[ENGLISH]

Hello Cloud Gurus and welcome to this lecture

which is going to introduce ElastiCache.

So what is ElastiCache?

Well, as you probably know, memory is faster than disk.

And wouldn't it be great if we could speed up

database queries by storing frequently accessed data

in an in-memory cache?

Well, ElastiCache enables you to do exactly that.

So ElastiCache is an in-memory cache.

Unlike many data stores in AWS,

it is a key value data store.

An ElastiCache makes it really easy to deploy,

operate, and scale an in-memory cache in the cloud.

And it's designed to improve database performance.

So it allows you to retrieve information

from fast in memory caches,

instead of slower disc based storage.

And it's great for read heavy database workloads.

So it allows you to catch the results

of I/O intensive database queries,

and it's also really useful for storing session data

for distributed applications.

But it might make a little bit more sense

if we consider an example.

So imagine you have a busy database,

serving thousands of queries a day,

and many of these queries are requesting the same data.

Well, in this scenario, you can use an ElastiCache cluster

to cache frequently accessed data.

So frequently accessed data will be stored

here in ElastiCache to enable faster access.

And then when your application needs to access the data,

it can query ElastiCache instead of the database.

And by doing this, we can reduce the load on the database

and speed up data access for our application.

And there are two types of ElastiCache available.

Firstly, we have Memcached,

and this is great for basic object caching.

It scales horizontally,

but there is no persistence for the data

and there is no Multi-AZ or failover option either.

However, is a good choice if you just want basic caching

and you want your caching model to be as simple as possible.

And then the second option that we have is Redis.

And this is a more sophisticated solution

with enterprise features like persistence,

replication, Multi-AZ and failover.

and it also supports sorting and ranking data

for example, for gaming leaderboards.

And it also supports complex data types

like lists and hashes.

But don't get too hung up on complex data types,

the main thing to remember is we've got two options,

you've got Memcache D for the basic object caching,

and Redis, is just the more sophisticated solution

with enterprise features.

So let's take a look at a typical scenario

that you might see in the exam.

And you might be given a scenario

where a particular database is under a lot of stress

and you'll be asked to find a solution.

So you might be asked at which service

should you use to alleviate this.

And you will need to know when to use ElastiCache.

So ElastiCache is a really good choice

if your database is particularly read-heavy

and the data is not prone to frequent changing.

Because if the data is changing too frequently,

then ElastiCache will struggle

to have the latest data available

and it will not be a very effective solution.

And you should also be aware of when ElastiCache can't help.

And it's not gonna help if your database

is struggling with heavy write loads.

So caching will not help alleviate heavy write loads

so in that case you may need to scale up

your database instead.

And if you're running online, analytical processing

or OLAP queries, and your database is feeling stress

because you're performing online, analytical processing,

then ElastiCache is not going to help with that

because you really need a data warehouse for that

rather than a database.

And you should think about using Redshift instead.

So onto my exam tips,

ElastiCache is an in-memory cache designed to improve

read performance for read-heavy databases.

And we've got two different flavors,

so we've got Memcache D and Redis.

And Memcache D is an in-memory key value data store.

And it's the one to use if object caching

is your primary goal.

And you want to keep things as simple as possible.

You don't need persistence or Multi-AZ features

and you don't need to support advanced data types

or sorting your data.

And then we have Redis as well,

and it's also an in-memory key value data store.

And this is the one to use

if you're performing data sorting, and ranking,

the kind of thing that you would need

for gaming leaderboards.

If you've got advanced data types such as lists and hashes,

and you need your data to be persistent,

and you need it to be Multi-AZ.

So that's it for this lecture.

If you have any questions, please let me know,

otherwise I will see you in the next lecture.

Thank you.

[SPANISH]

Hola Cloud Gurus y bienvenidos a esta conferencia.

que presentará ElastiCache .

Entonces, ¿qué es ElastiCache ?

Bueno, como probablemente sepa, la memoria es más rápida que el disco.

Y no sería genial si pudiéramos acelerar

Consultas de bases de datos mediante el almacenamiento de datos de acceso frecuente.

en un caché en memoria?

Bueno, ElastiCache le permite hacer exactamente eso.

Entonces, ElastiCache es un caché en memoria.

A diferencia de muchos almacenes de datos en AWS,

es un almacén de datos de valor clave.

Un ElastiCache hace que sea realmente fácil de implementar,

operar y escalar un caché en memoria en la nube.

Y está diseñado para mejorar el rendimiento de la base de datos.

Por lo que le permite recuperar información

de rápido en cachés de memoria,

en lugar de un almacenamiento basado en disco más lento.

Y es genial para leer cargas de trabajo pesadas de bases de datos.

Así que te permite captar los resultados.

de consultas de base de datos intensivas de E/S,

y también es muy útil para almacenar datos de sesión

para aplicaciones distribuidas.

Pero podría tener un poco más de sentido

si consideramos un ejemplo.

Así que imagina que tienes una base de datos ocupada,

atendiendo miles de consultas al día,

y muchas de estas consultas solicitan los mismos datos.

Bueno, en este escenario, puede usar un clúster de ElastiCache

para almacenar en caché los datos a los que se accede con frecuencia.

Por lo tanto, los datos de acceso frecuente se almacenarán

aquí en ElastiCache para permitir un acceso más rápido.

Y luego, cuando su aplicación necesite acceder a los datos,

puede consultar ElastiCache en lugar de la base de datos.

Y al hacer esto, podemos reducir la carga en la base de datos.

y acelerar el acceso a los datos de nuestra aplicación.

Y hay dos tipos de ElastiCache disponibles.

En primer lugar, tenemos Memcached ,

y esto es excelente para el almacenamiento en caché de objetos básicos.

Se escala horizontalmente,

pero no hay persistencia para los datos

y tampoco hay opción Multi-AZ o failover.

Sin embargo, es una buena opción si solo desea un almacenamiento en caché básico

y desea que su modelo de almacenamiento en caché sea lo más simple posible.

Y luego la segunda opción que tenemos es Redis .

Y esta es una solución más sofisticada.

con características empresariales como persistencia,

replicación , Multi-AZ y conmutación por error.

y también es compatible con la clasificación y clasificación de datos

por ejemplo, para tablas de clasificación de juegos.

Y también admite tipos de datos complejos.

como listas y hashes.

Pero no se obsesione demasiado con los tipos de datos complejos,

lo más importante que hay que recordar es que tenemos dos opciones,

tienes Memcache D para el almacenamiento en caché básico de objetos,

y Redis , es solo la solución más sofisticada

con características empresariales.

Así que echemos un vistazo a un escenario típico

que podrías ver en el examen.

Y se le puede dar un escenario

donde una base de datos en particular está bajo mucho estrés

y se le pedirá que encuentre una solución.

Por lo tanto, es posible que le pregunten en qué servicio

debe usar para aliviar esto.

Y necesitará saber cuándo usar ElastiCache .

Entonces ElastiCache es una muy buena opción

si su base de datos es particularmente de lectura pesada

y los datos no son propensos a cambios frecuentes.

Porque si los datos cambian con demasiada frecuencia,

entonces ElastiCache tendrá problemas

tener los últimos datos disponibles

y no será una solución muy eficaz.

Y también debe saber cuándo ElastiCache no puede ayudar.

Y no va a ayudar si su base de datos

está luchando con grandes cargas de escritura.

Por lo tanto, el almacenamiento en caché no ayudará a aliviar las cargas de escritura pesadas.

entonces, en ese caso, es posible que deba ampliar

su base de datos en su lugar.

Y si está ejecutando en línea, el procesamiento analítico

o consultas OLAP, y su base de datos está bajo presión

porque está realizando un procesamiento analítico en línea,

entonces ElastiCache no va a ayudar con eso

porque realmente necesitas un almacén de datos para eso

en lugar de una base de datos.

Y debería pensar en usar Redshift en su lugar.

Entonces, en mis consejos para el examen,

ElastiCache es un caché en memoria diseñado para mejorar

de lectura para bases de datos con muchas lecturas.

Y tenemos dos sabores diferentes,

así que tenemos Memcache D y Redis .

Y Memcache D es un almacén de datos de valor clave en memoria.

Y es el que se debe usar si el almacenamiento en caché de objetos

es su objetivo principal.

Y desea mantener las cosas lo más simples posible.

No necesita características de persistencia o Multi-AZ

y no necesita admitir tipos de datos avanzados

o ordenar sus datos.

Y luego tenemos Redis también,

y también es un almacén de datos de valor clave en memoria.

Y este es el que hay que usar

si está ordenando y clasificando datos,

el tipo de cosas que necesitarías

para tablas de clasificación de juegos.

Si tiene tipos de datos avanzados como listas y hashes,

y necesita que sus datos sean persistentes,

y necesita que sea Multi-AZ.

Así que eso es todo por esta conferencia.

Si tiene alguna pregunta, por favor hágamelo saber,

de lo contrario , te veré en la próxima conferencia.

Gracias.