[ENGLISH]

Hello, Cloud Gurus,

and welcome to this lecture,

which is going to cover API Gateway caching and throttling.

And both of these features of API Gateway

are designed to improve performance and throughput

of your APIs.

So what is API Gateway caching?

Well, this is where API Gateway

caches your endpoint response.

And this reduces the number of calls made to your endpoint,

and it can also improve the latency

for requests to your API.

And when you enable caching,

API Gateway caches responses from your endpoint application

for a specified time-to-live period in seconds.

And the default is 300 seconds, or five minutes.

And API Gateway then responds to any new requests

by looking up the response from the cache

instead of making a new request to your application.

So API Gateway can help you improve

the performance of your APIs

and the latency your end users experience

by caching the output of API calls to avoid

calling your backend application every time.

So let's take a look at how this works.

So imagine you an application running in Lambda,

and you've configured an API Gateway endpoint

to make your application available to your users.

So here are our users.

They make a request to the application,

and the request hits the API Gateway endpoint.

API Gateway then sends the request

to your backend application, running in Lambda.

The Lambda function processes the request

and sends the response,

and the response is cached in API Gateway

for the specified time-to-live.

And by default, it's 300 seconds.

So the response is cached within API Gateway,

and then if somebody else comes along

and makes the same request,

instead of sending the request to be processed again,

it's going to return the response

that is held in the cache instead.

And this is usually much faster than processing

the entire request again.

Onto API Gateway throttling.

And the purpose of API Gateway throttling

is to prevent your API from being overwhelmed

by too many requests.

And by default, API Gateway limits

the steady-state request rate

to 10,000 requests per second per region.

And there is also a limit on concurrent requests as well.

So the maximum concurrent requests

is 5,000 requests across all APIs per region.

And you can request an increase on both of these limits.

However, if you exceed your limit

of 10,000 requests per second or 5,000 concurrent requests,

then you will receive an error message,

and it's the "429-Too Many Requests" error message.

So at that point, you will need to increase your limits.

So how does this work in practice?

Well, imagine you have an application,

and your application receives 10,000 requests per second,

and the requests are evenly spread

across each one second period.

So you're receiving approximately

10 requests per millisecond.

Well, that is within the limit,

so API Gateway processes all the requests

without dropping any and without any error messages.

But what if your application receives

10,000 requests in the very first millisecond?

Well, the limit of concurrent requests is 5,000.

Therefore, API Gateway is going to serve

5,000 of these requests immediately,

and then it will throttle the rest

within the one second period.

So it will throttle the remaining 5,000 requests

over the rest of the one second period.

And it will deal with all the requests within one second,

but no more than 5,000 concurrently.

But what about if you receive 5,000 requests

in the first millisecond

and then another 5,000 requests,

but these are evenly spread

throughout the remaining 999 milliseconds?

Well, this is still within the limit,

so it's still within the PERT region limit

of 10,000 requests per second in total

and 5,000 concurrent requests.

Therefore, API Gateway will process all requests,

so it will process all 10,000 requests

in the one second period without returning any errors.

So onto my exam tips, beginning with API Gateway caching.

And API Gateway caching is designed

to improve the performance of your APIs

by caching the output of API calls

to avoid calling your backend application every single time.

The responses are cached for the time-to-live period,

and the default is 300 seconds, or five minutes.

And this reduces the number of API calls

made to your application

because API Gateway will return the cached response

instead of making a new request to your application.

And the advantage of this is that we are improving

the performance of our application

and reducing the latency experienced by our end users.

And for API Gateway throttling,

remember there is a default limit

of 10,000 requests per second

and 5,000 concurrent requests.

And of course, you can ask to have these limits increased.

However, if you exceed the limit,

then API Gateway will return an error, and it's a 429 error,

usually accompanied by the message "Too Many Requests."

And API Gateway uses throttling to prevent your API

from being overwhelmed by too many requests.

So that's it for 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, gurús de la nube,

y bienvenidos a esta conferencia,

que cubrirá el almacenamiento en caché y la aceleración de API Gateway.

Y ambas características de API Gateway

están diseñados para mejorar el rendimiento y la producción

de sus API.

Entonces, ¿qué es el almacenamiento en caché de API Gateway?

Bueno, aquí es donde API Gateway

almacena en caché su respuesta de punto final.

Y esto reduce la cantidad de llamadas realizadas a su terminal,

y también puede mejorar la latencia

para solicitudes a su API.

Y cuando habilita el almacenamiento en caché,

API Gateway almacena en caché las respuestas de su aplicación de punto final

durante un período de tiempo de vida especificado en segundos.

Y el valor predeterminado es 300 segundos o cinco minutos.

Y API Gateway luego responde a cualquier solicitud nueva

buscando la respuesta del caché

en lugar de hacer una nueva solicitud a su aplicación.

Entonces API Gateway puede ayudarlo a mejorar

el rendimiento de sus API

y la latencia que experimentan sus usuarios finales

almacenando en caché la salida de las llamadas a la API para evitar

llamando a su aplicación backend cada vez.

Así que echemos un vistazo a cómo funciona esto.

Así que imagina una aplicación ejecutándose en Lambda,

y ha configurado un punto final de API Gateway

para que su aplicación esté disponible para sus usuarios.

Así que aquí están nuestros usuarios.

Hacen una solicitud a la aplicación,

y la solicitud llega al extremo de API Gateway.

API Gateway luego envía la solicitud

a su aplicación de back-end, ejecutándose en Lambda.

La función Lambda procesa la solicitud

y envía la respuesta,

y la respuesta se almacena en caché en API Gateway

durante el tiempo de vida especificado.

Y por defecto, son 300 segundos.

Entonces, la respuesta se almacena en caché dentro de API Gateway,

y luego si viene alguien más

y hace la misma petición,

en lugar de enviar la solicitud para ser procesada nuevamente,

va a devolver la respuesta

que se mantiene en el caché en su lugar.

Y esto suele ser mucho más rápido que el procesamiento

toda la solicitud de nuevo.

En la limitación de API Gateway.

Y el propósito de la aceleración de API Gateway

es evitar que su API se vea abrumada

por demasiadas solicitudes.

Y de forma predeterminada, los límites de API Gateway

la tasa de solicitud de estado estable

a 10.000 solicitudes por segundo por región.

Y también hay un límite en las solicitudes simultáneas.

Así que el máximo de solicitudes simultáneas

es de 5000 solicitudes en todas las API por región.

Y puede solicitar un aumento en ambos límites.

Sin embargo, si supera su límite

de 10.000 solicitudes por segundo o 5.000 solicitudes simultáneas,

entonces recibirá un mensaje de error,

y es el mensaje de error "429-Too Many Requests".

Entonces, en ese punto, deberá aumentar sus límites.

Entonces, ¿cómo funciona esto en la práctica?

Bueno, imagina que tienes una aplicación,

y tu aplicación recibe 10.000 solicitudes por segundo,

y las solicitudes se distribuyen uniformemente

en cada período de un segundo.

Así que estás recibiendo aproximadamente

10 solicitudes por milisegundo.

Bueno, eso está dentro del límite,

para que API Gateway procese todas las solicitudes

sin soltar ninguno y sin ningún mensaje de error.

Pero, ¿y si su aplicación recibe

¿10.000 solicitudes en el primer milisegundo?

Bueno, el límite de solicitudes simultáneas es de 5.000.

Por lo tanto, API Gateway va a servir

5.000 de estas solicitudes inmediatamente,

y luego acelerará el resto

dentro del período de un segundo.

Por lo tanto, limitará las 5000 solicitudes restantes.

durante el resto del período de un segundo.

Y se ocupará de todas las solicitudes en un segundo,

pero no más de 5.000 al mismo tiempo.

Pero, ¿qué pasa si recibes 5000 solicitudes?

en el primer milisegundo

y luego otras 5.000 solicitudes,

pero estos están distribuidos uniformemente

durante los 999 milisegundos restantes?

Bueno, esto todavía está dentro del límite,

por lo que todavía está dentro del límite de la región PERT

de 10.000 solicitudes por segundo en total

y 5.000 solicitudes simultáneas.

Por lo tanto, API Gateway procesará todas las solicitudes,

por lo que procesará las 10,000 solicitudes

en el período de un segundo sin devolver ningún error.

Entonces, en mis consejos para el examen, comenzando con el almacenamiento en caché de API Gateway.

Y el almacenamiento en caché de API Gateway está diseñado

para mejorar el rendimiento de sus API

almacenando en caché la salida de las llamadas a la API

para evitar llamar a su aplicación backend cada vez.

Las respuestas se almacenan en caché durante el período de tiempo de vida,

y el valor predeterminado es 300 segundos o cinco minutos.

Y esto reduce la cantidad de llamadas a la API.

hecho a su aplicación

porque API Gateway devolverá la respuesta almacenada en caché

en lugar de hacer una nueva solicitud a su aplicación.

Y la ventaja de esto es que estamos mejorando

el rendimiento de nuestra aplicación

y reduciendo la latencia experimentada por nuestros usuarios finales.

Y para la limitación de API Gateway,

recuerda que hay un límite predeterminado

de 10.000 solicitudes por segundo

y 5.000 solicitudes simultáneas.

Y, por supuesto, puede solicitar que se aumenten estos límites.

Sin embargo, si supera el límite,

entonces API Gateway devolverá un error, y es un error 429,

suele ir acompañada del mensaje "Demasiadas solicitudes".

Y API Gateway utiliza la limitación para evitar que su API

de estar abrumado por demasiadas solicitudes.

Así que eso es todo por esta conferencia.

Si tiene alguna pregunta, por favor hágamelo saber.

De lo contrario, siéntase libre de pasar a la siguiente lección.

Gracias.