[ENGLISH]

Hello Cloud Gurus and welcome to this lesson

which is going to introduce Lambda environment variables

and the other parameters that are configurable

within Lambda.

We'll begin with what are environment variables,

give some example use cases,

we'll take a look at the other configurable parameters

within Lambda, and finish off with my exam tips.

So what are Lambda environment variables?

Well, they allow you to adjust your function behavior

without actually changing anything within your code.

So they allow you to configure your function

by passing different environment variables

to configure the function to behave differently,

for example within your development environment

than it does in production.

They consist of key value pairs

so a pair of strings, a key and a value.

For example, you could have a key

of environment and a value of development, or a key

of database name and the value of the name of the database.

So in this instance, mydevDB.

And environment variables are locked when the version

of your function is published.

So they're actually defined

before you go ahead and publish the version.

Example use cases include referencing an object

in an S3 bucket,

referencing a specific SNS Topic,

or referencing a DynamoDB table.

But what else can we configure within Lambda?

Well, in the Lambda console, there's a configuration tab

and it allows you to define the function configuration.

So for each function,

we can define the general configuration.

For instance, the amount of memory, ephemeral storage

or temporary storage, and also the function timeout,

so how long the function is gonna run for.

We can also define triggers

and that just means the service or resource

that's gonna be able to invoke your function,

permissions,

and the functions execution role determines the permissions

that the function's going to have.

We can also define a function URL,

which is an HTTP endpoint used to access your function

when using a web browser

and this is great for just simple functions

when you don't want to go to the trouble

of using API gateway.

We can also define tags,

so user defined key value pairs to organize our functions,

for instance, for cost reporting.

And we can allow a function to access resources

in a custom VPC by configuring the VPC parameters

within Lambda.

There's also the option to configure monitoring

using CloudWatch and X-Ray

which is used to help analyze performance issues.

There's concurrency options as well.

So reserved concurrency ensures

that a critical function can always run

and it's gonna restrict other concurrent requests

in favor of a critical function

whereas provision concurrency

lets your function scale consistently

without any fluctuations in latency.

And then finally, there are file system options as well.

So if your function needs to connect to a file system

like EFS so that it can read and write files,

there are options to configure that as well.

And your function must be connected

and associated with the same VPC

as the EFS file system if you want to do that.

But for the exam,

just be aware that Lambda is highly configurable.

We can use Lambda environment variables

to adjust the function's behavior

without actually changing the code itself.

For instance, define an environment, a database name,

or even an S3 object, et cetera.

And there are also loads of other configurable parameters

that allow you to adjust your function settings.

And these are located

in the configuration tab on the console.

The main ones to be aware of are firstly around capacity,

so configuring the amount of memory,

the amount of temporary ephemeral storage,

and concurrency,

and connectivity is another important one,

so the ability to connect to other AWS services

like CloudWatch and X-Ray for monitoring and to connect

to resources within your VPC like EFS file systems.

And that is all for this lesson.

Any questions, let me know.

Otherwise, please join me for the next one.

Thank you.

[SPANISH]

Hola Cloud Gurus y bienvenidos a esta lección.

que va a introducir variables de entorno Lambda

y los demás parámetros que son configurables

dentro de lambda.

Comenzaremos con lo que son las variables de entorno,

dar algunos ejemplos de casos de uso,

vamos a echar un vistazo a los otros parámetros configurables

dentro de Lambda, y termine con mis consejos para el examen.

Entonces, ¿qué son las variables de entorno de Lambda?

Bueno, te permiten ajustar el comportamiento de tu función.

sin cambiar nada dentro de su código.

Así te permiten configurar tu función

pasando diferentes variables de entorno

para configurar la función para que se comporte de manera diferente,

por ejemplo dentro de su entorno de desarrollo

que lo que hace en la producción.

Consisten en pares de valores clave

entonces un par de cadenas, una clave y un valor.

Por ejemplo, podría tener una clave

del medio ambiente y un valor del desarrollo, o una clave

del nombre de la base de datos y el valor del nombre de la base de datos.

Entonces, en este caso, mydevDB.

Y las variables de entorno están bloqueadas cuando la versión

de su función se publica.

Así que en realidad están definidos

antes de continuar y publicar la versión.

Ejemplos de casos de uso incluyen hacer referencia a un objeto

en un cubo S3,

haciendo referencia a un tema específico de SNS,

o haciendo referencia a una tabla de DynamoDB.

Pero, ¿qué más podemos configurar dentro de Lambda?

Bueno, en la consola Lambda, hay una pestaña de configuración

y te permite definir la configuración de funciones.

Entonces, para cada función,

podemos definir la configuración general.

Por ejemplo, la cantidad de memoria, almacenamiento efímero

o almacenamiento temporal, y también el tiempo de espera de la función,

Entonces, ¿cuánto tiempo se ejecutará la función?

También podemos definir disparadores

y eso solo significa el servicio o recurso

eso será capaz de invocar su función,

permisos,

y el rol de ejecución de funciones determina los permisos

que va a tener la función.

También podemos definir una URL de función,

que es un punto final HTTP utilizado para acceder a su función

al usar un navegador web

y esto es genial para funciones simples

cuando no quieres meterte en líos

de usar la puerta de enlace API.

También podemos definir etiquetas,

así que los pares de valores clave definidos por el usuario para organizar nuestras funciones,

por ejemplo, para informes de costos.

Y podemos permitir que una función acceda a los recursos.

en una VPC personalizada configurando los parámetros de VPC

dentro de lambda.

También existe la opción de configurar el monitoreo.

utilizando CloudWatch y X-Ray

que se utiliza para ayudar a analizar los problemas de rendimiento.

También hay opciones de concurrencia.

Así que la concurrencia reservada asegura

que una función crítica siempre puede ejecutarse

y va a restringir otras solicitudes concurrentes

a favor de una función crítica

Considerando que la concurrencia de provisiones

permite que su función escale consistentemente

sin fluctuaciones en la latencia.

Y finalmente, también hay opciones de sistema de archivos.

Entonces, si su función necesita conectarse a un sistema de archivos

como EFS para que pueda leer y escribir archivos,

hay opciones para configurar eso también.

Y tu función debe estar conectada.

y asociado con la misma VPC

como el sistema de archivos EFS si desea hacerlo.

Pero para el examen,

solo tenga en cuenta que Lambda es altamente configurable.

Podemos usar variables de entorno Lambda

para ajustar el comportamiento de la función

sin cambiar el código en sí.

Por ejemplo, defina un entorno, un nombre de base de datos,

o incluso un objeto S3, etcétera.

Y también hay un montón de otros parámetros configurables

que le permiten ajustar la configuración de su función.

Y estos se encuentran

en la pestaña de configuración de la consola.

Los principales a tener en cuenta son, en primer lugar, la capacidad,

así que configurando la cantidad de memoria,

la cantidad de almacenamiento efímero temporal,

y concurrencia,

y la conectividad es otra importante,

por lo que la capacidad de conectarse a otros servicios de AWS

como CloudWatch y X-Ray para monitorear y conectarse

a recursos dentro de su VPC como sistemas de archivos EFS.

Y eso es todo por esta lección.

Cualquier pregunta, hágamelo saber.

De lo contrario, por favor únase a mí para el próximo.

Gracias.