[ENGLISH]

Hello Cloud Gurus and welcome to this lesson

which is going to cover Lambda deployment packaging options.

We'll start off with the Lambda deployment package.

What to do if you have deployment packages that are greater

than 50 megabytes.

We'll discuss Lambda layers

and finish with my exam tips.

Now, the Lambda deployment package is used

to deploy your code to Lambda,

but you may not be aware of it

if you've only created a function using the AWS console.

Because when you create a function using the console

and you are typing the code directly into the console,

a zip file containing the code that you provide

is automatically created by Lambda in the background.

So, you may just not be aware of it.

The .zip file is your deployment package

and it contains the application code

and optionally, any dependencies that your code needs

in order to run.

Now, you can also create the deployment package yourself

and then upload it directly from your local machine

but you can only do that with deployment packages

that are up to 50 megabytes in size.

Now, if you end up

with a deployment package that's greater than 50 megabytes,

then you will need to upload your .zip file to S3

in the region where you are creating your function,

and then specify the S3 object

when you create your function.

Because direct upload from your local machine

is not supported for deployment packages

that are over 50 megabytes in size

and that is just a limitation within Lambda.

Now, there is an alternative to including everything

in the deployment package and that is Lambda layers.

And a Lambda layer is really a distribution mechanism

and it's a .zip file archive that's referenced

by your function, and it contains function dependencies

like libraries or custom run times, et cetera.

And here's an example.

Say for instance, you wanted to include a specific SDK.

Well, you can do that using a layer like this one

instead of including it in your deployment package.

Once created, a layer can be used by multiple functions

that have the same dependencies, so multiple functions

can reference the same layer.

And using a Lambda layer to provide function dependencies

like libraries or custom run times or SDKs

is a best practice because it reduces the size

of your deployment package, enabling your function

to initialize faster.

So when the function starts up for the first time,

a smaller deployment package is going to allow it

to initialize faster.

So for the exam, just remember

that the deployment package is a .zip file

that contains your function code

and optionally, any dependencies.

And if you're creating your function in the console

and you are adding your code directly into the console,

then the console will create the deployment package for you.

However, for large deployment packages

over 50 megabytes in size,

you will need to create the deployment package

and upload it to S3.

And finally, using a Lambda layer is actually best practice

because it reduces the size of the deployment package

allowing your functions to initialize faster

and another good reason to use it is that multiple functions

can reference the same layer.

So if you do see any questions in the exam relating

to how to provide code dependencies, specifically,

if you've got larger libraries that you want to include,

or if you've got multiple functions that need

to reference the same dependencies,

then the approach to go for is to use a Lambda layer.

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.

que cubrirá las opciones de empaquetado de implementación de Lambda.

Comenzaremos con el paquete de implementación de Lambda.

Qué hacer si tiene paquetes de implementación que son mayores

de 50 megas.

Hablaremos de las capas de Lambda

y terminar con mis consejos de examen.

Ahora, se utiliza el paquete de implementación de Lambda

para implementar su código en Lambda,

pero puede que no te des cuenta

si solo ha creado una función utilizando la consola de AWS.

Porque cuando creas una función usando la consola

y está escribiendo el código directamente en la consola,

un archivo zip que contiene el código que proporciona

es creado automáticamente por Lambda en segundo plano.

Por lo tanto, es posible que simplemente no seas consciente de ello.

El archivo .zip es su paquete de implementación

y contiene el código de la aplicación

y opcionalmente, cualquier dependencia que su código necesite

para correr

Ahora, también puede crear el paquete de implementación usted mismo

y luego cárguelo directamente desde su máquina local

pero solo puede hacer eso con paquetes de implementación

que tienen un tamaño de hasta 50 megabytes.

Ahora, si terminas

con un paquete de implementación de más de 50 megabytes,

entonces deberá cargar su archivo .zip a S3

en la región donde está creando su función,

y luego especifique el objeto S3

cuando creas tu función.

Porque carga directa desde su máquina local

no es compatible con los paquetes de implementación

que tienen más de 50 megabytes de tamaño

y eso es solo una limitación dentro de Lambda.

Ahora, hay una alternativa para incluir todo

en el paquete de implementación y son capas Lambda.

Y una capa Lambda es realmente un mecanismo de distribución

y es un archivo comprimido .zip al que se hace referencia

por su función, y contiene dependencias de función

como bibliotecas o tiempos de ejecución personalizados, etcétera.

Y aquí hay un ejemplo.

Digamos, por ejemplo, que desea incluir un SDK específico.

Bueno, puedes hacerlo usando una capa como esta.

en lugar de incluirlo en su paquete de implementación.

Una vez creada, una capa puede ser utilizada por múltiples funciones

que tienen las mismas dependencias, por lo que múltiples funciones

puede hacer referencia a la misma capa.

Y usar una capa Lambda para proporcionar dependencias de funciones

como bibliotecas o tiempos de ejecución personalizados o SDK

es una buena práctica porque reduce el tamaño

de su paquete de implementación, habilitando su función

para inicializar más rápido.

Entonces, cuando la función se inicia por primera vez,

un paquete de implementación más pequeño lo permitirá

para inicializar más rápido.

Entonces, para el examen, solo recuerda

que el paquete de implementación es un archivo .zip

que contiene su código de función

y opcionalmente, cualquier dependencia.

Y si está creando su función en la consola

y está agregando su código directamente en la consola,

luego, la consola creará el paquete de implementación por usted.

Sin embargo, para paquetes de implementación grandes

más de 50 megabytes de tamaño,

deberá crear el paquete de implementación

y subirlo a S3.

Y finalmente, usar una capa Lambda es en realidad la mejor práctica

porque reduce el tamaño del paquete de implementación

permitiendo que sus funciones se inicialicen más rápido

y otra buena razón para usarlo es que múltiples funciones

puede hacer referencia a la misma capa.

Entonces, si ve alguna pregunta en el examen relacionada con

a cómo proporcionar dependencias de código, específicamente,

si tiene bibliotecas más grandes que desea incluir,

o si tiene múltiples funciones que necesitan

para hacer referencia a las mismas dependencias,

entonces el enfoque a seguir es usar una capa Lambda.

Eso es todo por esta lección.

Cualquier pregunta, hágamelo saber.

De lo contrario, te veré en la próxima.

Gracias.