[ENGLISH]

Hello, Cloud Gurus, and welcome to this lesson,

which is going to cover "Version Control with Lambda."

We'll begin with what is $LATEST,

managing multiple versions, Lambda aliases,

we'll have a quick demo, and finish off with my exam tips.

Now, when you initially create a Lambda function,

there is only 1 version,

and they attach this label to it,

and it's called $LATEST.

And that is how you can reference

the latest code that you uploaded to your function.

But let's say you need to update your function code,

and so you upload a new version of the code to Lambda.

Well then that new version that you upload

is going to be the latest version.

And so the $LATEST label is going to be attached

to this new version,

but you can actually create multiple versions

of your function code

and then use different aliases

to reference the different versions.

And you can refer to the specific version

that you want to use using an alias.

For example, imagine a development environment

where you might want to maintain a few different versions

of the same function

as you develop and test and change your code.

And an alias is used to point to a specific version

of the function code,

so you can create different versions

of the same function called, for example,

development, test, production, etc.

And then an alias is just like a pointer

to the specific version of the function code

that you would like to use.

So let's have a look at an example.

So, we've got a developer on the left-hand side,

and they're busy updating their code

and adding new code into an S3 bucket.

And then within Lambda,

we've got our version 1 of the code,

and we've applied an alias called prod

to that version of the code.

Then we've also got a version 2,

which also happens to be the latest version of the code

and version 2 also has an alias of test,

and we can reference each of these aliases

using the Amazon Resource Name, or ARN.

And we use this ARN to identify

the version of the function that we want to invoke.

So, if we want to use the latest version of the code,

we can refer to it using the ARN which ends in latest,

or we could alternatively use the ARN which ends in test,

and that's the test alias.

But if we want to use our version 1 of the code,

which is our production version,

then we can use the ARN which ends in prod.

But what's going to happen if we add another version?

Let's say, for example,

we add a new version of our code to Lambda.

When we upload the new code to Lambda,

that is going to be our latest version.

So it will get the $LATEST label,

but we've still got our aliases

for the prod and test version,

so for version 1 and version 2.

But if your application is using an alias

to call the function,

then you might need to remember to update

the ARN that you're using within your application

if you want to use the new code.

Because if you are using the alias of test or prod,

then you're going to be pointing

to either version 1, if it's prod,

version 2, if it's test,

and your application will not be using

the latest version of your code.

So, if you want your application to use

the latest version you just uploaded,

then make sure that it's referring

to the $LATEST version,

but the best way to understand all of this

is if I show you what I mean in the console,

and you'll see how easy it is

to publish different versions of your code,

create aliases, and identify Amazon resource names.

So we're going to create a new function

using version 1 of my code.

We'll publish version 1

and create an alias called prod,

then we'll upload version 2 of my code.

We're going to publish version 2

and create an alias called test.

And then once we've done this, we'll

be able to review the ARNs and aliases that we created,

and you will find the 2 different versions of code

in the resources for this lesson.

So if you'd like to join me in the console,

I will show you how it all works.

So, from the AWS console,

let's head over to Lambda.

Select Lambda.

We create a function,

select Author From Scratch.

The function name is my-lambda-function.

The runtime is Node.js

and just select the latest version available.

Then scroll down to the bottom and Create Function.

When your function is ready,

scroll down to code source

and select Upload From Zip File.

Select Upload,

and we're going to upload version 1 of our code.

And these zip files

are in the resources section for this lesson

and just download them to your local machine

so that you can upload them to Lambda.

So select Version1 and Open, Save,

and there is our Lambda function.

So now let's test our function.

We'll configure a test event--

just name it te1,

scroll down, and Save.

And then to test it, just click Test again.

And if it's all worked correctly,

the response should include the status code of 200,

which means it's been successful,

and the body should include "Hello from Lambda!

"This is version1!\"

And if you scroll up to the top of the page,

you will find your function ARN, or Amazon resource name.

And when you see an ARN like this,

and it ends in your function name and nothing else,

then that is an unqualified function ARN.

And by default,

it relates to the latest version of your code,

or the $LATEST.

So now we are going to add a version and an alias.

So, from the top in the Actions section,

select Publish New Version.

And it says here that publishing a new version

saves a snapshot of the code

and the configuration of the $LATEST version.

We'll give it a description.

It's version 1 of my code., and Publish.

And there we go, we've got version 1 of my code.

And you will see in the function ARN,

it's added a 1 after our function name.

So now, let's create an alias.

Select Actions, Create Alias.

The alias is going to be called Prod.

We'll give it a description,

and we'll select the version,

and it's going to be version 1 of my code and hit Save.

So that is my alias created.

And you'll see that the alias

has also got its own function ARN,

with the alias appended to the function name.

So now what if we have a new version of our code

that we want to start testing with?

Well, let's head back to our function,

down to code source,

select Upload From Zip File,

Upload, and we're going to upload version2.zip,

hit Save, and there is our version 2.

We'll test again to make sure it works.

And if it's all worked,

you should get this response status code of 200,

which shows it's worked, and "Hello from Lambda!

"This is version 2!".

And then once again, if we check the ARN,

this is the latest version of our code,

and it's an unqualified function ARN,

which by default refers to the latest version of our code.

So now if we want to publish this

as a new version within Lambda,

we can select Actions, Publish New Version,

we'll give it a description, and hit Publish,

and there is our version 2,

and you will see version 2 reflected in the function ARN.

So now, let's create an alias and point it

to this new version.

So, under Actions,

Create Alias,

the name is going to be Test.

We'll add a description.

It will show you the versions that are available,

we'll select version 2.

Then down here, you can actually create a weighted alias,

and with a weighted alias,

Lambda allows you to direct traffic

to 1 or more versions of the function,

and you can shift the traffic between the 2 versions

based on weights, which are defined as a percentage.

So, you could send 90% of your traffic to 1 version

and 10% of your traffic to another version.

For example, if you wanted to test a new version,

with just a percentage of your production traffic.

But we're not going to do that,

so I'm just going to close that down and hit Save.

And that is our second alias created.

And as you can see, it's got its own ARN.

And now if you select your function name at the top,

select Aliases,

this is where you'll find the 2 aliases

that we just created,

and it will show the versions that they relate to.

Then if you select Versions,

you'll be able to see the versions

and any aliases that are associated with those versions.

And also, helpfully, it shows you

when the version was last modified as well.

So hopefully that's shown you how easy it is

to create versions and aliases in Lambda.

So, moving on to my exam tips,

just remember that $LATEST is always

the latest version of the code

that you've uploaded into Lambda.

Use Lambda versioning and aliases

to point your applications to a specific version

if you don't want to use the $LATEST.

Here's some example ARNs,

the first one showing my prod alias,

and you can also refer to the latest version

by adding the $LATEST label

at the end of your function name.

And if your application uses an alias instead of $LATEST,

then remember that it will not automatically use

the new code when you upload it.

So if you are using an alias, like prod or test,

and you upload new code

that you want your application to use,

then just remember to update your application

to point to the latest version of the code.

So, that is everything that I wanted to tell you

about Lambda versions.

If you have any questions, please let me know.

If not, feel free to move on to the next lesson, thank you.

[SPANISH]

Hola, Cloud Gurus, y bienvenidos a esta lección.

que cubrirá "Control de versiones con Lambda".

Comenzaremos con lo que es $ÚLTIMO,

administrar múltiples versiones, alias de Lambda,

tendremos una demostración rápida y terminaremos con mis consejos para el examen.

Ahora, cuando crea inicialmente una función Lambda,

solo hay 1 versión,

y le pegan esta etiqueta,

y se llama $ÚLTIMO.

Y así es como se puede hacer referencia

el último código que cargó en su función.

Pero digamos que necesita actualizar su código de función,

y así subes una nueva versión del código a Lambda.

pues entonces esa nueva version que subes

va a ser la última versión.

Y así se adjuntará la etiqueta $ÚLTIMO

a esta nueva versión,

pero en realidad puedes crear múltiples versiones

de su código de función

y luego usar diferentes alias

para hacer referencia a las diferentes versiones.

Y puede referirse a la versión específica

que desea utilizar utilizando un alias.

Por ejemplo, imagine un entorno de desarrollo

donde es posible que desee mantener algunas versiones diferentes

de la misma función

a medida que desarrolla, prueba y cambia su código.

Y un alias se usa para apuntar a una versión específica

del código de función,

para que puedas crear diferentes versiones

de la misma función llamada, por ejemplo,

desarrollo, prueba, producción, etc.

Y luego un alias es como un puntero

a la versión específica del código de función

que le gustaría usar.

Así que echemos un vistazo a un ejemplo.

Entonces, tenemos un desarrollador en el lado izquierdo,

y están ocupados actualizando su código

y agregando código nuevo en un depósito S3.

Y luego dentro de Lambda,

tenemos nuestra versión 1 del código,

y le hemos aplicado un alias llamado prod

a esa versión del código.

Luego también tenemos una versión 2,

que también resulta ser la última versión del código

y la versión 2 también tiene un alias de prueba,

y podemos hacer referencia a cada uno de estos alias

utilizando el Nombre de recurso de Amazon, o ARN.

Y usamos este ARN para identificar

la versión de la función que queremos invocar.

Entonces, si queremos usar la última versión del código,

podemos referirnos a él usando el ARN que termina en más reciente,

o alternativamente podríamos usar el ARN que termina en prueba,

y ese es el alias de prueba.

Pero si queremos usar nuestra versión 1 del código,

cuál es nuestra versión de producción,

entonces podemos usar el ARN que termina en prod.

Pero, ¿qué va a pasar si agregamos otra versión?

Digamos, por ejemplo,

agregamos una nueva versión de nuestro código a Lambda.

Cuando subimos el nuevo código a Lambda,

esa va a ser nuestra última versión.

Entonces obtendrá la etiqueta $ÚLTIMO,

pero todavía tenemos nuestros alias

para la versión de producción y prueba,

así que para la versión 1 y la versión 2.

Pero si su aplicación está usando un alias

para llamar a la función,

entonces es posible que deba recordar actualizar

el ARN que está utilizando dentro de su aplicación

si desea utilizar el nuevo código.

Porque si está utilizando el alias de prueba o prod,

entonces vas a estar señalando

a la versión 1, si es prod,

versión 2, si es de prueba,

y su aplicación no estará usando

la última versión de su código.

Entonces, si desea que su aplicación use

la última versión que acabas de subir,

luego asegúrate de que se esté refiriendo

a la versión $ÚLTIMA,

pero la mejor manera de entender todo esto

es si te muestro lo que quiero decir en la consola,

y verás que fácil es

para publicar diferentes versiones de su código,

crear alias e identificar nombres de recursos de Amazon.

Así que vamos a crear una nueva función.

usando la versión 1 de mi código.

Publicaremos la versión 1

y crear un alias llamado prod,

luego subiremos la versión 2 de mi código.

Vamos a publicar la versión 2

y crea un alias llamado test.

Y luego, una vez que hayamos hecho esto, vamos a

poder revisar los ARN y los alias que creamos,

y encontrarás las 2 versiones diferentes de código

en los recursos para esta lección.

Entonces, si desea unirse a mí en la consola,

Te mostraré cómo funciona todo.

Entonces, desde la consola de AWS,

Vayamos a Lambda.

Seleccione lambda.

Creamos una función,

seleccione Autor desde cero.

El nombre de la función es my-lambda-function.

El tiempo de ejecución es Node.js

y simplemente seleccione la última versión disponible.

Luego desplácese hacia abajo hasta la parte inferior y Crear función.

Cuando su función esté lista,

desplácese hacia abajo hasta el código fuente

y seleccione Cargar desde archivo zip.

Seleccione Cargar,

y vamos a subir la versión 1 de nuestro código.

Y estos archivos zip

están en la sección de recursos para esta lección

y simplemente descárguelos a su máquina local

para que pueda cargarlos en Lambda.

Así que seleccione Version1 y Abrir, Guardar,

y ahí está nuestra función Lambda.

Así que ahora probemos nuestra función.

Configuraremos un evento de prueba--

solo nómbralo te1,

desplácese hacia abajo y Guardar.

Y luego, para probarlo, simplemente haga clic en Probar nuevamente.

Y si todo ha funcionado correctamente,

la respuesta debe incluir el código de estado de 200,

lo que significa que ha tenido éxito,

y el cuerpo debe incluir "¡Hola de Lambda!

"¡Esta es la versión 1!\"

Y si se desplaza hacia arriba hasta la parte superior de la página,

encontrará el ARN de su función o el nombre del recurso de Amazon.

Y cuando ves un ARN como este,

y termina en el nombre de tu función y nada más,

entonces ese es un ARN de función no calificado.

Y por defecto,

se relaciona con la última versión de su código,

o el $ ÚLTIMO.

Así que ahora vamos a agregar una versión y un alias.

Entonces, desde la parte superior de la sección Acciones,

seleccione Publicar nueva versión.

Y aquí dice que publicar una nueva versión

guarda una instantánea del código

y la configuración de la versión $LATEST.

Le daremos una descripción.

Es la versión 1 de mi código y Publicar.

Y ahí vamos, tenemos la versión 1 de mi código.

Y verás en la función ARN,

se agrega un 1 después de nuestro nombre de función.

Así que ahora, vamos a crear un alias.

Seleccione Acciones, Crear alias.

El alias se llamará Prod.

Le daremos una descripción,

y seleccionaremos la versión,

y será la versión 1 de mi código y presione Guardar.

Así que ese es mi alias creado.

Y verás que el alias

también tiene su propia función ARN,

con el alias añadido al nombre de la función.

Entonces, ¿qué pasa si tenemos una nueva versión de nuestro código?

con el que queremos empezar a probar?

Bueno, volvamos a nuestra función,

hasta el código fuente,

seleccione Cargar desde archivo zip,

Subir, y vamos a subir version2.zip,

presione Guardar, y ahí está nuestra versión 2.

Probaremos de nuevo para asegurarnos de que funciona.

Y si todo ha funcionado,

debería obtener este código de estado de respuesta de 200,

que muestra que funcionó, y "¡Hola de Lambda!

"¡Esta es la versión 2!".

Y luego, una vez más, si verificamos el ARN,

esta es la última versión de nuestro código,

y es un ARN de función no calificado,

que por defecto se refiere a la última versión de nuestro código.

Entonces ahora si queremos publicar esto

como una nueva versión dentro de Lambda,

podemos seleccionar Acciones, Publicar nueva versión,

le daremos una descripción y le daremos a Publicar,

y ahí está nuestra versión 2,

y verás la versión 2 reflejada en el ARN de la función.

Así que ahora, vamos a crear un alias y apuntarlo

a esta nueva versión.

Entonces, en Acciones,

crear alias,

el nombre va a ser Test.

Añadiremos una descripción.

Le mostrará las versiones que están disponibles,

seleccionaremos la versión 2.

Luego, aquí abajo, puedes crear un alias ponderado,

y con un alias ponderado,

Lambda le permite dirigir el tráfico

a 1 o más versiones de la función,

y puedes cambiar el tráfico entre las 2 versiones

basado en pesos, que se definen como un porcentaje.

Entonces, podrías enviar el 90% de tu tráfico a 1 versión

y el 10% de su tráfico a otra versión.

Por ejemplo, si desea probar una nueva versión,

con solo un porcentaje de su tráfico de producción.

Pero no vamos a hacer eso,

así que voy a cerrar eso y presionar Guardar.

Y ese es nuestro segundo alias creado.

Y como puedes ver, tiene su propio ARN.

Y ahora, si selecciona el nombre de su función en la parte superior,

seleccione Alias,

aquí es donde encontrarás los 2 alias

que acabamos de crear,

y mostrará las versiones con las que se relacionan.

Luego, si selecciona Versiones,

podrás ver las versiones

y cualquier alias asociado con esas versiones.

Y también, útilmente, te muestra

cuando la versión fue modificada por última vez también.

Así que espero que eso te haya mostrado lo fácil que es

para crear versiones y alias en Lambda.

Entonces, pasando a mis consejos para el examen,

solo recuerda que $ÚLTIMO es siempre

la última versión del código

que ha cargado en Lambda.

Usar versiones y alias de Lambda

para apuntar sus aplicaciones a una versión específica

si no desea utilizar el $ÚLTIMO.

Aquí hay algunos ARN de ejemplo,

el primero que muestra mi alias prod,

y también puede consultar la última versión

agregando la etiqueta $ÚLTIMO

al final del nombre de su función.

Y si su aplicación usa un alias en lugar de $ÚLTIMO,

entonces recuerda que no usará automáticamente

el nuevo código cuando lo subas.

Entonces, si usa un alias, como prod o test,

y subes código nuevo

que desea que utilice su aplicación,

entonces solo recuerda actualizar tu aplicación

para apuntar a la última versión del código.

Entonces, eso es todo lo que quería decirte.

sobre las versiones Lambda.

Si tiene alguna pregunta, por favor hágamelo saber.

Si no es así, siéntase libre de pasar a la siguiente lección, gracias.