[ENGLISH]

Hello Cloud Gurus, and welcome to this lecture.

And this lecture is going to introduce DynamoDB TTL.

It's also known as DynamoDB Time to Live.

So what is DynamoDB TTL? Well, TTL, or Time to Live,

it's an attribute which defines an expiry time for your data.

So you define an expiry time,

and expired items are automatically marked for deletion.

And once an item has been marked for deletion,

that means it's going to be deleted within the next 48 hours.

And this is really good for removing any irrelevant or old data,

so data that's no longer useful to your application. So think session data,

event logs, any temporary data.

So any data that's no longer useful to your application after a certain period

of time.

And this is going to really help to reduce costs for storing data in DynamoDB

because it automatically removes data, which you no longer need.

So the data is no longer relevant.

So it just does that housekeeping for you and automatically deletes the data.

So let's take a look at how that actually works within DynamoDB.

And I've got an example Session Data table here.

So we're just storing the user ID, session ID, session creation time,

and the Time to Live, and then finally the session data itself.

And the first thing you're probably going to notice is that Time to Live is

expressed in a pretty strange format.

It's expressed in the epoch time format.

And if you haven't heard of that before, it's also known as Unix time,

and it's also known as POSIX time as well.

And epoch time or Unix time or POSIX time, that is a numerical value.

And it represents the number of seconds which have elapsed since 12:00 AM

on January the 1st, 1970. So even before I was born. In fact,

a long time before I was born.

And probably a long time before a lot of you were born as well.

So that is just the way that Unix systems see time.

And that's how they express time as a numerical value. Now, in this example,

I'm setting the expiration time on our item,

and I'm setting it for 2 hours after the session originally began.

And don't worry,

you can just use one of these really useful translation websites.

And I've got one to show you. I'm just going to head over there now.

So this is the website I normally use. It's just called EpochConverter,

but you I'm sure you can find one just by Googling or use your favorite one.

And you can just go in and set the day,

set the hour to what you want to,

and then just translate it into an epoch timestamp.

So that is the kind of thing that you can use if you are defining your

TTL.

And it's this epoch time number which actually allows DynamoDB to

easily compare your Time to Live to the current time right now.

And in this example,

the expiration of this item is set for 2 hours after the session

originally began. And so when the current time is greater than the TTL,

the item is going to be expired and it gets marked for deletion.

And then within the next 48 hours, it's actually going to be deleted.

So that's actually quite a long window between being marked for deletion and

actually getting deleted. So in order to cater for that,

you can actually filter out any expired items from your queries and scans.

But let's take a look at how this actually works within the DynamoDB console.

But first of all, we're going to need to create our own DynamoDB table.

And I prepared a couple of commands that we're going to run.

And before we do that,

you just need to make sure of course that your AWS CLI is configured

with credentials of a user who has the correct permissions to go

ahead and create DynamoDB tables.

So the first thing I'm going to do is I'm going to type aws iam get-user.

And this is just going to tell us which user we have used when we configured the

AWS CLI. So when we ran AWS configure,

this is the user credentials that I used originally.

So I'm just going to come back to my console and just double-check the

credentials of that particular user. So I'm going to go to Services,

and then IAM. I'm going to find my user.

So it was faye.

And I'm just going to check that that user has the correct permissions.

So my users actually got administrator access,

but if you don't have any DynamoDB access, you don't,

if you didn't have administrator access on that account,

then you can just go ahead and Add Permissions, go to Attach Existing Policies,

and then just search for DynamoDB. And you can add DynamoDB Full Access

to that user, and that will make sure that you have access to use DynamoDB.

So once you're happy with that, I'm going to come back to the console,

and I've actually prepared a couple of commands that we're going to run.

So we're going to run one command to firstly create our new table,

and then we're going to run another command to populate the table.

So I'll show you those commands before we run them.

And these are going to be included in a text file in the resources of the

course. So you'll be able to just copy and paste these.

So the first command is this one,

and this actually creates our Session Data table. So first of all,

it runs the create table command. It provides a table name of SessionData,

and then it defines our primary key, which it calls it the hash key.

And that's just an alternative name that they use for the primary key or the

partition key. So we're going to define our hash key as being the user ID,

and the attribute type is going to be of type number. And then finally,

the last thing we'd stipulate in this command is the provisioned throughput read

capacity units of 5 and write capacity units of 5.

So you just copy that command, and don't worry,

you don't need to understand all of the different options.

You don't need to understand the DynamoDB commands,

the CLI commands for the exam. They're not going to test you on that.

It's just the quickest way to create the table for the purposes of this lecture.

So come back to the terminal.

I'm going to paste that into my terminal and hit Enter.

And hopefully if that's been successful,

you should see this kind of output summarizing what we've done in DynamoDB.

And then the next command we're going to run is going to perform a

BatchWriteItem,

and this is going to write a number of items into our new table.

And it takes as input this, this request items file,

which is going to be a file called items.json.

And you're going to need to download this from the resources section of the

course. So if I show you items.json,

it's simply a json file which is going to create these items in our

Session Data table.

So the first thing it does is it stipulates the name of the table,

which is SessionData, and then it defines a number of Put requests,

and the items that we're actually going to put into that table.

So we've got our user ID. It's going to be a number.

This number here is the actual user ID.

Then we've got the session creation time, expiration time,

and a session ID as well. So it just creates 3 different items,

which it's going to put into the table. So if we come back to our command.

I'm just going to copy and paste that.

Okay, and if that's successful,

we can come back to our console and we can just check whether that table got

created successfully. So come to Services,

DynamoDB,

and make sure that you are in the correct region where you have run aws

configure. So on my system,

I'm in us-east-1.

So I just need to make sure that I am looking at the us-east-1 region,

which I am. And then if we go to Tables, and there is our Session Data table.

So if you click on the table, click on Items,

and that should be the items that we have just populated.

So if that has run successfully for you,

you should have 3 items in your table with a user ID, creation time,

expiry time, and a session ID.

And I've just set this expiry time to midnight tonight.

And today is the 6th of December.

So I have set the expiration time for these items to be the

6th of December at midnight. So you might want to update this,

set it to midnight tonight for whatever day it is when you are watching this

lecture. So if you want to do that once again,

just go back to the EpochConverter,

and you can create an epoch timestamp using whatever date you want to.

So Let's say February the 2nd, 2019.

You can just configure the date you want and then grab the epoch timestamp.

And you can set it to whatever hour, minute, and second you want as well,

and set whatever the local time is for wherever you are located.

So feel free to go in and update that expiration time if you want to follow

along on this lecture. So once you're happy with all of that,

we can actually go ahead and stop playing around with the Time to Live for these

items. So if you head to Actions,

click on Manage TTL.

And the first thing it's going to ask you to do is define which attribute you

are basing the Time to Live on.

And we are going to be using this expiration time as our TTL

attribute.

So you need to just type in the name of your attribute,

and DynamoDB actually allows you to preview TTL before you actually go

ahead and apply it. So we can, first of all,

we can run a preview based on previewing items expiring by 3:30

on today's date. So I'm just going to hit Run Preview,

and that's what we should expect to see.

We shouldn't expect it to see any items expiring by 3:30 today

because all of these items are set to expire at midnight tonight.

So let's just change that to December the 7th

and midnight, and we can run the preview again,

and there we are.

It should catch all 3 of those items because they are due you to expire

at midnight.

So if you want to go ahead and configure Time to Live for all of these items,

you can hit Continue. And then if we go back to Overview,

you should see under the Table Details the Time to Live attribute.

It's configured as the attribute that we defined earlier,

so the expiration time. So that is DynamoDB TTL.

The main thing to remember is that you can use it to define expiration times for

your data.

It will mark your data for expiration as soon as the current time is greater

than the TTL. The item will then be expired, marked for deletion,

and then deleted within 48 hours.

It's great for removing irrelevant or old data,

for example, session data, event logs,

and temporary data that your application no longer requires.

And this is great because it saves you money and reduces the cost of your

DynamoDB table by automatically removing data,

which is no longer relevant or useful to your application.

So that is everything you need to know about TTL. If you have any questions,

please let me know. If not, feel free to move on to the next lecture. Thank you.

[SPANISH]

Hola Cloud Gurus, y bienvenidos a esta conferencia.

Y esta conferencia presentará DynamoDB TTL.

También se conoce como DynamoDB Time to Live.

Entonces, ¿qué es DynamoDB TTL? Bueno, TTL, o Time to Live,

es un atributo que define un tiempo de caducidad para sus datos.

Así que defines un tiempo de caducidad,

y los elementos caducados se marcan automáticamente para su eliminación.

Y una vez que se ha marcado un elemento para su eliminación,

eso significa que se eliminará dentro de las próximas 48 horas.

Y esto es realmente bueno para eliminar datos irrelevantes o antiguos,

por lo que los datos que ya no son útiles para su aplicación. Así que piensa en los datos de la sesión,

de eventos , cualquier dato temporal.

Entonces, cualquier dato que ya no sea útil para su aplicación después de un cierto período

del tiempo

Y esto realmente ayudará a reducir los costos de almacenamiento de datos en DynamoDB .

porque elimina automáticamente los datos que ya no necesita.

Así que los datos ya no son relevantes.

Así que solo hace esa limpieza por usted y elimina automáticamente los datos.

Así que echemos un vistazo a cómo funciona realmente dentro de DynamoDB .

Y aquí tengo un ejemplo de tabla de datos de sesión.

Entonces, solo estamos almacenando la ID de usuario, la ID de la sesión, la hora de creación de la sesión,

y el tiempo de vida, y finalmente los datos de la sesión en sí.

Y lo primero que probablemente notará es que Time to Live es

expresado en un formato bastante extraño.

Se expresa en el formato de tiempo de época.

Y si no has oído hablar de eso antes, también se conoce como tiempo de Unix ,

y también se conoce como tiempo POSIX.

Y tiempo de época o tiempo Unix o tiempo POSIX, ese es un valor numérico.

Y representa el número de segundos que han transcurrido desde las 00:00

el 1 de enero de 1970. Incluso antes de que yo naciera. De hecho,

mucho antes de que yo naciera.

Y probablemente mucho antes de que nacieran muchos de ustedes también.

Así es como los sistemas Unix ven el tiempo.

Y así es como expresan el tiempo como un valor numérico. Ahora, en este ejemplo,

Estoy configurando el tiempo de caducidad de nuestro artículo,

y lo estoy configurando para 2 horas después de que comenzó la sesión originalmente.

Y no te preocupes,

puede usar uno de estos sitios web de traducción realmente útiles.

Y tengo uno para mostrarte. Sólo voy a ir allí ahora.

Así que este es el sitio web que uso normalmente. Simplemente se llama EpochConverter ,

pero estoy seguro de que puedes encontrar uno simplemente buscando en Google o usando tu favorito.

Y puedes entrar y establecer el día,

ajusta la hora a lo que quieras,

y luego simplemente traducirlo a una marca de tiempo de época.

Ese es el tipo de cosas que puede usar si está definiendo su

TTL.

Y es este número de tiempo de época lo que realmente permite a DynamoDB

fácilmente su tiempo de vida con la hora actual en este momento.

Y en este ejemplo,

la caducidad de este artículo está establecida para 2 horas después de la sesión

originalmente . Y así, cuando la hora actual es mayor que el TTL,

el elemento caducará y se marcará para su eliminación.

Y luego, dentro de las próximas 48 horas, en realidad se eliminará.

Así que en realidad hay una ventana bastante larga entre que se marca para su eliminación y

en realidad siendo eliminado. Entonces, para atender eso,

puede filtrar cualquier elemento caducado de sus consultas y escaneos.

Pero echemos un vistazo a cómo funciona esto realmente dentro de la consola de DynamoDB .

Pero antes que nada, necesitaremos crear nuestra propia tabla de DynamoDB .

Y preparé un par de comandos que vamos a ejecutar.

Y antes de hacer eso,

necesita asegurarse, por supuesto, de que su AWS CLI esté configurada

con credenciales de un usuario que tiene los permisos correctos para ir

adelante y cree tablas de DynamoDB .

Así que lo primero que voy a hacer es escribir aws iam get-usuario.

Y esto solo nos dirá qué usuario hemos usado cuando configuramos el

CLI de AWS. Así que cuando ejecutamos AWS configure,

estas son las credenciales de usuario que usé originalmente.

Así que volveré a mi consola y volveré a verificar el

credenciales de ese usuario en particular. Así que voy a ir a Servicios,

y luego YO SOY. Voy a buscar mi usuario.

Así que fue Faye .

Y solo voy a verificar que ese usuario tenga los permisos correctos.

Así que mis usuarios realmente obtuvieron acceso de administrador,

pero si no tiene acceso a DynamoDB , no lo tiene,

si no tenía acceso de administrador en esa cuenta,

luego puede continuar y Agregar permisos, ir a Adjuntar políticas existentes,

y luego simplemente busque DynamoDB . Y puede agregar DynamoDB Full Access

a ese usuario, y eso asegurará que tenga acceso para usar DynamoDB .

Entonces, una vez que esté satisfecho con eso, volveré a la consola,

y de hecho he preparado un par de comandos que vamos a ejecutar.

Vamos a ejecutar un comando para crear primero nuestra nueva tabla,

y luego vamos a ejecutar otro comando para llenar la tabla.

Así que les mostraré esos comandos antes de que los ejecutemos.

Y estos van a ser incluidos en un archivo de texto en los recursos de la

curso \_ Por lo tanto, podrá simplemente copiar y pegar estos.

Así que el primer comando es este,

y esto en realidad crea nuestra tabla de datos de sesión. Entonces, en primer lugar,

ejecuta el comando crear tabla. Proporciona un nombre de tabla de SessionData ,

y luego define nuestra clave principal, a la que llama clave hash.

Y ese es solo un nombre alternativo que usan para la clave principal o la

clave de partición Así que vamos a definir nuestra clave hash como el ID de usuario,

y el tipo de atributo va a ser de tipo número. Y finalmente,

lo último que estipularíamos en este comando es la lectura de rendimiento aprovisionado

capacidad de 5 y unidades de capacidad de escritura de 5.

Así que solo copia ese comando y no te preocupes,

no es necesario que comprenda todas las diferentes opciones.

No necesita comprender los comandos de DynamoDB ,

los comandos CLI para el examen. No te van a poner a prueba en eso.

Es simplemente la forma más rápida de crear la tabla para los propósitos de esta lección.

Así que vuelve a la terminal.

Voy a pegar eso en mi terminal y presionar Enter.

Y con suerte, si eso ha tenido éxito,

debería ver este tipo de salida que resume lo que hemos hecho en DynamoDB .

Y luego el siguiente comando que vamos a ejecutar va a realizar una

Elemento de escritura por lotes ,

y esto va a escribir una serie de elementos en nuestra nueva tabla.

Y toma como entrada este, este archivo de elementos de solicitud,

que va a ser un archivo llamado items.json .

Y tendrá que descargarlo de la sección de recursos de la

curso \_ Entonces, si te muestro items.json ,

es simplemente un archivo json que va a crear estos elementos en nuestro

Tabla de datos de sesión.

Así que lo primero que hace es estipular el nombre de la tabla,

que es SessionData , y luego define una cantidad de solicitudes Put,

y los artículos que vamos a poner en esa mesa.

Así que tenemos nuestra identificación de usuario. Va a ser un número.

Este número aquí es la identificación de usuario real.

Luego tenemos el tiempo de creación de la sesión, el tiempo de vencimiento,

y una ID de sesión también. Entonces solo crea 3 elementos diferentes,

que va a poner en la mesa. Así que si volvemos a nuestro comando.

Solo voy a copiar y pegar eso.

Bien, y si eso tiene éxito,

podemos volver a nuestra consola y podemos simplemente verificar si esa tabla obtuvo

creado con éxito. Así que ven a Servicios,

dinamoDB ,

y asegúrese de estar en la región correcta donde ha ejecutado aws

configurar \_ Así que en mi sistema,

Estoy en us-east-1.

Así que solo necesito asegurarme de que estoy viendo la región us -east-1,

que soy Y luego, si vamos a Tablas, y allí está nuestra tabla de Datos de sesión.

Entonces, si hace clic en la tabla, haga clic en Elementos,

y esos deberían ser los elementos que acabamos de llenar.

Entonces, si eso ha funcionado con éxito para usted,

debe tener 3 elementos en su tabla con una identificación de usuario, hora de creación,

de caducidad y un ID de sesión.

Y acabo de establecer este tiempo de caducidad para la medianoche de esta noche.

Y hoy es 6 de diciembre.

Así que he establecido el tiempo de caducidad de estos artículos para que sea el

6 de diciembre a la medianoche. Así que es posible que desee actualizar esto,

ajústelo a la medianoche de esta noche para cualquier día que sea cuando esté viendo esto

conferencia \_ Así que si quieres hacer eso una vez más,

simplemente regrese a EpochConverter ,

y puede crear una marca de tiempo de época usando la fecha que desee.

Entonces , digamos el 2 de febrero de 2019.

Simplemente puede configurar la fecha que desee y luego tomar la marca de tiempo de la época.

Y también puede configurarlo en cualquier hora, minuto y segundo que desee,

y establezca la hora local que sea para el lugar donde se encuentre.

Así que siéntase libre de ingresar y actualizar ese tiempo de vencimiento si desea seguir

a lo largo de esta conferencia. Así que una vez que estés satisfecho con todo eso,

podemos seguir adelante y dejar de jugar con Time to Live para estos

artículos \_ Entonces, si se dirige a Acciones,

haga clic en Administrar TTL.

Y lo primero que te va a pedir que hagas es definir qué atributo

están basando el tiempo de vida en.

Y vamos a utilizar este tiempo de caducidad como nuestro TTL

atributo \_

Entonces, solo necesita escribir el nombre de su atributo,

y DynamoDB en realidad le permite obtener una vista previa de TTL antes de ir

adelante y aplicarlo. Entonces podemos, en primer lugar,

podemos ejecutar una vista previa basada en la vista previa de los elementos que caducan a las 3:30

en la fecha de hoy. Voy a presionar Ejecutar vista previa,

y eso es lo que deberíamos esperar ver.

No deberíamos esperar que vea ningún elemento que caduque a las 3:30 de hoy.

porque todos estos artículos expirarán a la medianoche de esta noche.

Así que cambiemos eso al 7 de diciembre.

y medianoche, y podemos volver a ejecutar la vista previa,

y ahí estamos

Debería capturar los 3 de esos artículos porque están a punto de caducar.

a medianoche

Entonces, si desea continuar y configurar Time to Live para todos estos elementos,

puede pulsar Continuar. Y luego, si volvemos a la descripción general,

debería ver debajo de Detalles de la tabla el atributo Tiempo de vida.

Está configurado como el atributo que definimos anteriormente,

por lo que el tiempo de caducidad. Eso es DynamoDB TTL.

Lo más importante que debe recordar es que puede usarlo para definir tiempos de vencimiento para

tus datos

Marcará sus datos para la caducidad tan pronto como la hora actual sea mayor

que el TTL. El elemento caducará, se marcará para su eliminación,

y luego eliminado dentro de las 48 horas.

Es excelente para eliminar datos irrelevantes o antiguos,

por ejemplo, datos de sesión, registros de eventos,

y datos temporales que su aplicación ya no requiere.

Y esto es genial porque le ahorra dinero y reduce el costo de su

tabla de DynamoDB mediante la eliminación automática de datos,

que ya no es relevante o útil para su aplicación.

Eso es todo lo que necesitas saber sobre TTL. Si tienes alguna pregunta,

por favor hágamelo saber Si no, siéntase libre de pasar a la siguiente lección. Gracias.