[ENGLISH]

Hello, Cloud Gurus,

and welcome to this lecture,

which is going to cover DynamoDB Streams.

So what are DynamoDB Streams?

Well, a DynamoDB Stream

is a time-ordered sequence, or stream,

and it records any modifications that are made to the items

in your DynamoDB table.

So that's any time there's an insert, update,

or delete operation,

it's going to record that in your DynamoDB stream.

And it records all of these actions in a log

and the logs are encrypted at rest

and stored for 24 hours only.

So after 24 hours,

it's going to start deleting those entries.

And typically, DynamoDB streams can be used for auditing

or archiving of transactions

or replaying transactions to a different table,

but they are mainly used to trigger events

based on a particular change within the DynamoDB table.

So they are really, really good

for serverless architectures.

So you can use a DynamoDB stream

to trigger a Lambda function, for example,

or to trigger your application to do something

based on the change in the DynamoDB table.

And the streams are accessed

using their own dedicated endpoint.

So there's one endpoint to access the DynamoDB table itself

and then there's a second separate endpoint

for the DynamoDB stream.

And by default,

the minimum amount of data that you can record

is the primary key of the item

that was being modified, updated or deleted,

but you can also store before and after images as well.

So you can capture the state of the item before the change

and the state of the item after the change as well.

Main use cases for DynamoDB streams are to audit

or archive transactions,

trigger an event based on a particular transaction

in your DynamoDB table,

or to replicate data across multiple DynamoDB tables.

So let's take a look

at a simple architecture for DynamoDB Streams.

We've got our application at the top

and it uses the DynamoDB API

to speak to its DynamoDB tables.

It's got one endpoint for the DynamoDB table

and then there is a completely different

DynamoDB Streams API

and a different endpoint for the DynamoDB Stream as well.

And just remember, that the event data

is only held for 24 hours before they start deleting it.

Now, the events are logged in near real-time.

So once again, that makes it really, really good

for serverless architectures

and it allows you to build applications

which respond to changes in the DynamoDB table.

Applications can take actions

based on the contents of your stream.

So based on what is actually going on

within your DynamoDB table.

So it makes a really good event source for Lambda

and Lambda can poll the DynamoDB Stream

and can then execute code

and launch functions based on the events

happening within your DynamoDB Stream.

So let's take a look at an example scenario

using DynamoDB Streams to trigger a workflow.

And in this example,

we have an invoicing and payment system,

which is recording invoice data into a DynamoDB table.

And each time a new invoice entry

is made into the DynamoDB table,

there is a Lambda function which reads the DynamoDB stream

to check for new transactions.

And then when it sees a new event,

it's sending a notification using SNS,

which in turn creates a message in an SNS queue.

And you can then have your payments application

polling that SQS queue, processing the message.

And let's say maybe it performs various tasks

like generating a payment request

or adding an item to a customer's bill, etc.

So you can see that this is a really powerful way

to generate triggers to Lambda,

to trigger your application

to take actions based on the changing contents

of your DynamoDB table.

So let's take a look at the exam tips for DynamoDB Streams.

And just remember, it's a time-ordered sequence

of item-level modifications in your DynamoDB tables.

So, every time an item gets changed in your DynamoDB table,

it's going to record that in the DynamoDB stream.

The data is only stored for 24 hours

and after that, it's going to start getting deleted.

And it's a really good event source for Lambda,

allowing you to create applications that can take actions

based on events that are happening in your DynamoDB table.

So that's it for this lecture.

If you have any questions, please let me know.

Otherwise, I will see you in the next lecture.

Thank you.

[SPANISH]

Hola, gurús de la nube,

y bienvenidos a esta conferencia,

que cubrirá DynamoDB Streams.

Entonces, ¿qué son los flujos de DynamoDB ?

Bueno, una secuencia de DynamoDB

es una secuencia ordenada en el tiempo, o flujo,

y registra las modificaciones que se realizan en los elementos

en su tabla de DynamoDB .

Eso es cada vez que hay una inserción, actualización,

o eliminar la operación,

va a registrar eso en su transmisión de DynamoDB .

Y registra todas estas acciones en un registro

y los registros están encriptados en reposo

y se almacena solo durante 24 horas.

Así que después de 24 horas,

va a empezar a borrar esas entradas.

Y, por lo general, los flujos de DynamoDB se pueden usar para auditar

o archivo de transacciones

o reproducir transacciones en una tabla diferente,

pero se utilizan principalmente para desencadenar eventos.

basado en un cambio particular dentro de la tabla de DynamoDB .

Así que son muy, muy buenos.

para arquitecturas sin servidor .

Entonces puede usar una transmisión de DynamoDB

para activar una función Lambda, por ejemplo,

o para activar su aplicación para hacer algo

basado en el cambio en la tabla de DynamoDB .

Y se accede a los streams

usando su propio punto final dedicado.

Entonces, hay un punto final para acceder a la tabla de DynamoDB en sí

y luego hay un segundo punto final separado

para la transmisión de DynamoDB .

Y por defecto,

la cantidad mínima de datos que puede registrar

es la clave principal del elemento

que estaba siendo modificado, actualizado o eliminado,

pero también puede almacenar imágenes de antes y después.

Para que pueda capturar el estado del artículo antes del cambio

y el estado del artículo después del cambio también.

Los principales casos de uso de flujos de DynamoDB son para auditar

o archivar transacciones,

desencadenar un evento basado en una transacción en particular

en su tabla de DynamoDB ,

o para replicar datos en varias tablas de DynamoDB .

Así que echemos un vistazo

en una arquitectura simple para DynamoDB Streams.

Tenemos nuestra aplicación en la parte superior

y utiliza la API de DynamoDB

para hablar con sus tablas de DynamoDB .

Tiene un punto final para la tabla de DynamoDB

y luego hay una completamente diferente

API de secuencias de DynamoDB

y también un punto final diferente para DynamoDB Stream.

Y recuerde, que los datos del evento

se retiene durante 24 horas antes de que comiencen a eliminarlo.

Ahora, los eventos se registran casi en tiempo real.

Entonces, una vez más, eso lo hace muy, muy bueno.

para arquitecturas sin servidor

y te permite construir aplicaciones

que responden a cambios en la tabla de DynamoDB .

Las aplicaciones pueden realizar acciones

basado en el contenido de su transmisión.

Así que en base a lo que realmente está pasando

dentro de su tabla de DynamoDB .

Por lo tanto, es una muy buena fuente de eventos para Lambda.

y Lambda puede sondear el flujo de DynamoDB

y luego puede ejecutar el código

y lanzar funciones basadas en los eventos

sucediendo dentro de su flujo de DynamoDB .

Así que echemos un vistazo a un escenario de ejemplo

usando DynamoDB Streams para activar un flujo de trabajo.

Y en este ejemplo,

contamos con un sistema de facturación y pago,

que está registrando los datos de la factura en una tabla de DynamoDB .

Y cada vez que una nueva entrada de factura

se convierte en la tabla de DynamoDB ,

hay una función Lambda que lee el flujo de DynamoDB

para comprobar si hay nuevas transacciones.

Y luego, cuando ve un nuevo evento,

está enviando una notificación usando SNS,

que a su vez crea un mensaje en una cola SNS.

Y luego puede tener su aplicación de pagos

sondeando esa cola SQS, procesando el mensaje.

Y digamos que tal vez realiza varias tareas

como generar una solicitud de pago

o agregar un artículo a la factura de un cliente, etc.

Entonces puedes ver que esta es una manera realmente poderosa

para generar disparadores a Lambda,

para activar su aplicación

para tomar acciones basadas en los contenidos cambiantes

de su tabla de DynamoDB .

Así que echemos un vistazo a los consejos de examen para DynamoDB Streams.

Y recuerda, es una secuencia ordenada por tiempo

de modificaciones a nivel de elemento en sus tablas de DynamoDB .

Entonces, cada vez que se cambia un elemento en su tabla de DynamoDB ,

va a registrar eso en la transmisión de DynamoDB .

Los datos solo se almacenan durante 24 horas.

y después de eso, comenzará a eliminarse.

Y es una muy buena fuente de eventos para Lambda,

permite crear aplicaciones que pueden tomar acciones

en función de los eventos que están ocurriendo en su tabla de DynamoDB .

Así que eso es todo por esta conferencia.

Si tiene alguna pregunta, por favor hágamelo saber.

De lo contrario, te veré en la próxima conferencia.

Gracias.