[ENGLISH]

Hello, Cloud Gurus, and welcome to this lesson

where we'll be getting our hands dirty using the CDK.

And we'll begin by using the CDK,

or Cloud Development Kit,

to initialize a sample application

written in TypeScript.

Next, we'll compile the app using npm run build,

we'll synthesize a CloudFormation template

using the cdk synth command,

and finally, we'll deploy our stack.

And if we're using the CDK for the very first time,

we'll need to run cdk bootstrap

before we can run cdk deploy.

And all of the commands that we're going to use

are in the text file in the resources for this lesson.

So if you're ready to get started using CDK,

please join me in the AWS Console.

Now we're going to run all

of the commands using the CloudShell.

So open up the CloudShell,

and the first thing we need to do

is install the latest version of the CDK.

Next, we'll update to the latest version of TypeScript.

I'll clear my screen,

and now we are ready to initialize our sample app.

First of all, we need to make a new directory.

Change directory to your new directory.

Then we'll use this command to initialize our sample app,

cdk init sample-app --language=typescript.

After everything's completed, we can check

that the files have been successfully created by typing ls.

And you should have some files created in your directory.

I'll clear my screen.

And next, we're going to manually compile the app.

So type npm run build.

After that's completed, we can run cdk ls,

and the name of our application directory

is the name of our stack.

Next, we're going to synthesize our CloudFormation template.

So type cdk synth,

and here is the CloudFormation template that it's created.

And it actually stores the template in the cdk.out folder.

So if you type ls, it's this cdk.out folder

where the template is going to be stored.

So if you type ls cdk.out,

here is our template written in JSON.

So now we are ready to deploy our stack.

So I'm going to clear my screen.

And if we haven't used CDK before,

then before we deploy it,

we will need to run the bootstrap command.

And this is just to provision the required resources

for CDK before we can deploy our CDK apps

into an AWS account and region.

And it actually uses an S3 bucket to store any files,

and it creates Identity and Access Management roles

with permission to perform deployments.

So first of all, run cdk bootstrap,

and we can see it creating our Identity

and Access Management roles and policies

and an S3 bucket as well.

Then after that's completed,

we can deploy everything using cdk deploy.

And remember, it's running

CloudFormation in the background,

so it does just take a few minutes to complete everything.

It's asking us now, do we wish to deploy the changes?

And the answer is, yes.

And now it's deploying everything using CloudFormation.

Then, once everything's completed,

we can check the CloudFormation console

to review our stack.

I'm going to close down the CloudShell,

search for CloudFormation.

Here's my stack.

So select that and select Resources,

and within your stack you should have your

SNS topic and your SQS queue.

So there we go.

We have provisioned these resources using code

written in TypeScript and using just a few CDK commands.

So for the exam, just remember the high-level process.

So we used cdk init inside an empty directory

to create our new CDK project.

Next, we ran npm build to compile our app.

Next, we used cdk synth

to create our CloudFormation template,

and we used cdk deploy

to deploy the stack using CloudFormation.

So that is it for this lesson.

Any questions, let me know.

Otherwise, I'll see you in the next one.

Thank you.

[SPANISH]

Hola, Cloud Gurus, y bienvenidos a esta lección.

donde nos ensuciaremos las manos usando el CDK.

Y comenzaremos usando el CDK,

o kit de desarrollo en la nube,

para inicializar una aplicación de muestra

escrito en TypeScript .

A continuación, compilaremos la aplicación usando npm run build,

sintetizaremos una plantilla de CloudFormation

usando el comando de sintetizador cdk ,

y finalmente, implementaremos nuestra pila.

Y si estamos usando el CDK por primera vez,

necesitaremos ejecutar cdk bootstrap

antes de que podamos ejecutar cdk deployment.

Y todos los comandos que vamos a usar

están en el archivo de texto en los recursos para esta lección.

Entonces, si está listo para comenzar a usar CDK,

Únase a mí en la consola de AWS.

Ahora vamos a ejecutar todo

de los comandos usando CloudShell .

Así que abre CloudShell ,

y lo primero que tenemos que hacer

es instalar la última versión del CDK.

A continuación, actualizaremos a la última versión de TypeScript .

Borraré mi pantalla,

y ahora estamos listos para inicializar nuestra aplicación de muestra.

En primer lugar, necesitamos crear un nuevo directorio.

Cambie de directorio a su nuevo directorio.

Luego usaremos este comando para inicializar nuestra aplicación de muestra,

cdk init muestra-aplicación --idioma=mecanografiado.

Después de que todo esté completo, podemos verificar

que los archivos se han creado correctamente escribiendo ls .

Y debería tener algunos archivos creados en su directorio.

Borraré mi pantalla.

Y luego, vamos a compilar manualmente la aplicación.

Entonces escriba npm run build.

Una vez que se haya completado, podemos ejecutar cdk ls ,

y el nombre de nuestro directorio de aplicaciones

es el nombre de nuestra pila.

A continuación, sintetizaremos nuestra plantilla de CloudFormation .

Así que escribe cdk synth,

y aquí está la plantilla de CloudFormation que se creó.

Y en realidad almacena la plantilla en la carpeta cdk.out .

Entonces, si escribe ls , es esta carpeta cdk.out

donde se va a almacenar la plantilla.

Así que si escribes ls cdk.fuera ,

aquí está nuestra plantilla escrita en JSON.

Así que ahora estamos listos para implementar nuestra pila.

Así que voy a limpiar mi pantalla.

Y si no hemos usado CDK antes,

luego, antes de implementarlo,

necesitaremos ejecutar el comando bootstrap.

Y esto es solo para aprovisionar los recursos necesarios.

para CDK antes de que podamos implementar nuestras aplicaciones de CDK

en una cuenta y región de AWS.

Y en realidad usa un depósito S3 para almacenar cualquier archivo,

y crea roles de gestión de acceso e identidad

con permiso para realizar implementaciones.

Entonces, antes que nada, ejecute cdk bootstrap,

y podemos verlo creando nuestra Identidad

y roles y políticas de administración de acceso

y un cubo S3 también.

Luego, después de que se haya completado,

podemos implementar todo usando cdk deployment.

Y recuerda, está corriendo

CloudFormation en el fondo,

por lo que solo toma unos minutos completar todo.

Nos está preguntando ahora, ¿queremos implementar los cambios?

Y la respuesta es sí.

Y ahora está implementando todo usando CloudFormation .

Entonces, una vez que todo esté terminado,

podemos comprobar la consola de CloudFormation

para revisar nuestra pila.

Voy a cerrar CloudShell ,

busque CloudFormation . \_

Aquí está mi pila.

Así que seleccione eso y seleccione Recursos,

y dentro de tu pila deberías tener tu

Tema de SNS y su cola de SQS.

Así que ahí vamos.

Hemos aprovisionado estos recursos usando código

escrito en TypeScript y usando solo unos pocos comandos CDK.

Entonces, para el examen, solo recuerda el proceso de alto nivel.

Así que usamos cdk init dentro de un directorio vacío

para crear nuestro nuevo proyecto CDK.

Luego, ejecutamos npm build para compilar nuestra aplicación.

A continuación, usamos el sintetizador cdk

para crear nuestra plantilla de CloudFormation ,

y usamos cdk deployment

para implementar la pila usando CloudFormation .

Eso es todo por esta lección.

Cualquier pregunta, hágamelo saber.

De lo contrario, te veré en la próxima.

Gracias.