[ENGLISH]

Hello cloud gurus,

and welcome to this lecture,

where we going to take a closer look at CodeCommit.

So as you know, CodeCommit is a central code repository.

So it's a place to store everything you need

to build your application.

So think source code, binaries, images,

and any libraries needed as well.

And it manages updates from multiple sources.

So allows multiple developers to work

on the code at the same time.

It enables collaboration and tracks

and manages changes to your code base

and maintains a version history.

But how do we actually use CodeCommit?

Well, if you've used a source control

or version control system, like git before,

this will all be pretty familiar to you.

But let's take a look at how we might use

CodeCommit in the real world.

So here is our repository with all our files in.

So we've got our source code, our libraries and binaries,

and any images needed for our application.

And here's our developer, Sally.

And Sally starts off by creating a new branch,

which is a copy of the master branch.

And she clones a local copy of the dev branch

to her local machine.

And this is where she's going to work on her code updates.

So she makes the changes and tests them locally.

So she's updated the source code,

and she's also changed the image

from Ryan holding his deep racer

to an image of one of our

other awesome instructors, Tim Berry,

and then she goes ahead and merges her changes

to the development branch.

And then finally, if there's no merge conflicts,

then Sally's code can be merged with the master branch.

But what happens if another developer in Sally's team

is also working on the same files?

So here is Sam, and Sam is also working

on updates to the source code.

And what do you think happens

if Sam merges his own change before Sally?

Well, if that happens,

then when Sally tries to merge her own changes,

there's likely to be a merge conflict.

And conflicts generally happen if two people

are working on the same file,

and they change the same line of code.

And it's then the developers' responsibility

to work out which is going to be the correct version.

Is it going to be Sally's version?

Or Sam's or is it going to be something else?

And then once that's resolved,

once they've come to an agreement,

Sally will update her code to incorporate Sam's changes,

or vice versa.

And then once that's done,

Sally's merge will be good to go.

So that is just one example of how

we can use CodeCommit to manage our source code

and enable multiple developers

to work on the same project.

So let's take a look at some

of our exam tips for CodeCommit.

First of all, just remember,

it's a centralized code repository.

So it's a place to store source code, binaries,

libraries, images, HTML files, et cetera,

and it's based on git.

Code commit enables collaboration,

and manages updates from multiple users,

and it's a version control system.

So it tracks and manages code changes,

and maintains a version history of your files.

So that is CodeCommit at a high level.

And in the next lecture,

we going to be getting our hands dirty with CodeCommit.

So if you have any questions,

please let me know.

If not, please join me in the next lecture, thank you.

[SPANISH]

Hola gurús de la nube,

y bienvenidos a esta conferencia,

donde vamos a echar un vistazo más de cerca a CodeCommit .

Como saben, CodeCommit es un repositorio de código central.

Así que es un lugar para guardar todo lo que necesitas

para construir su aplicación.

Así que piensa en código fuente, binarios, imágenes,

y cualquier biblioteca necesaria también.

Y gestiona las actualizaciones de múltiples fuentes.

Por lo tanto, permite que varios desarrolladores trabajen

en el código al mismo tiempo.

Permite la colaboración y rastrea

y administra los cambios en su base de código

y mantiene un historial de versiones.

Pero, ¿cómo usamos realmente CodeCommit ?

Bueno, si ha usado un control de fuente

o sistema de control de versiones, como git antes,

esto te resultará bastante familiar.

Pero echemos un vistazo a cómo podríamos usar

CodeCommit en el mundo real.

Así que aquí está nuestro repositorio con todos nuestros archivos.

Así que tenemos nuestro código fuente, nuestras bibliotecas y binarios,

y cualquier imagen necesaria para nuestra aplicación.

Y aquí está nuestra desarrolladora, Sally.

Y Sally comienza creando una nueva sucursal,

que es una copia de la rama maestra.

Y ella clona una copia local de la rama de desarrollo

a su máquina local.

Y aquí es donde va a trabajar en sus actualizaciones de código.

Entonces ella hace los cambios y los prueba localmente.

Ella actualizó el código fuente,

y también ha cambiado la imagen

de Ryan sosteniendo su deep racer

a una imagen de uno de nuestros

otros increíbles instructores, Tim Berry,

y luego continúa y fusiona sus cambios

a la rama de desarrollo.

Y finalmente, si no hay conflictos de fusión,

entonces el código de Sally se puede fusionar con la rama principal.

Pero, ¿qué sucede si otro desarrollador del equipo de Sally

está trabajando en los mismos archivos?

Así que aquí está Sam, y Sam también está trabajando.

en las actualizaciones del código fuente.

y que crees que pasa

si Sam fusiona su propio cambio antes que Sally?

Bueno, si eso sucede,

luego , cuando Sally intenta fusionar sus propios cambios,

es probable que haya un conflicto de fusión.

Y los conflictos generalmente ocurren si dos personas

están trabajando en el mismo archivo,

y cambian la misma línea de código.

Y es entonces responsabilidad de los desarrolladores

para averiguar cuál va a ser la versión correcta.

¿Será la versión de Sally?

¿O de Sam o va a ser otra cosa?

Y luego, una vez que se resuelve,

una vez que han llegado a un acuerdo,

Sally actualizará su código para incorporar los cambios de Sam,

o viceversa.

Y luego, una vez hecho esto,

La fusión de Sally estará lista.

Así que ese es solo un ejemplo de cómo

podemos usar CodeCommit para administrar nuestro código fuente

y habilitar múltiples desarrolladores

para trabajar en el mismo proyecto.

Así que echemos un vistazo a algunos

de nuestros consejos de examen para CodeCommit .

En primer lugar, solo recuerda,

es un repositorio de código centralizado.

Es un lugar para almacenar código fuente, binarios,

bibliotecas , imágenes, archivos HTML, etcétera,

y está basado en git .

La confirmación de código permite la colaboración,

y administra las actualizaciones de múltiples usuarios,

y es un sistema de control de versiones.

Por lo tanto, rastrea y administra los cambios de código,

y mantiene un historial de versiones de sus archivos.

Eso es CodeCommit a un alto nivel.

Y en la próxima conferencia,

nos vamos a ensuciar las manos con CodeCommit .

Así que si tienes alguna pregunta,

por favor hágamelo saber

Si no, por favor únase a mí en la próxima conferencia, gracias.