[ENGLISH]

Hello, Cloud Gurus, and welcome to this lesson,

which is going to be a demo of the "AWS CLI."

So in this lesson, we're going to get our hands dirty

using the AWS Command Line Interface.

We'll start off by launching an EC2 instance,

and this is where we're going to use the AWS CLI.

Next, we'll create an Identity and Access Management user,

and we'll give the user permissions

to access and create S3 resources.

And then finally, we'll configure the AWS CLI

on our EC2 instance using the IAM user credentials.

And then we'll use the AWS CLI

to create an S3 bucket and upload a file.

So if you'd like to get your hands dirty with the AWS CLI,

then please join me in the console now.

So from the AWS console, I'm going to be doing everything

from the Northern Virginia Region.

And first of all, we are going to launch an EC2 instance.

So search for EC2 in the search box

and select Launch Instance.

We'll give it a name

and make sure that the Amazon Linux AMI is selected.

Under instance type, we can stick with the t2.micro.

We'll create a new key pair

so that we can log into the EC2 instance.

And I'm just going to call it mykp and Create Key Pair,

and it will download the key pair to your local machine.

Under network settings,

make sure that you've got auto-assigned public IP

set to enable.

By default, the launch wizard

is going to create a security group,

which allows SSH from anywhere.

So now you can just go ahead,

accept the rest of the defaults,

and Launch Instance.

So while that is launching, you can go ahead

and select your instance ID,

and we can view the progress of our instance here.

And at the moment, it's still in a pending state,

so just be patient,

and once it's passed its status checks,

we should be good to continue.

We can refresh the screen to refresh our status.

Refresh again.

And once you've passed the 2 status checks,

you are good to go.

So now, select your instance ID,

find your public IP address--it's here at the top,

copy that to your clipboard,

and then head to a terminal window on your local machine.

And we're going to log into our EC2 instance.

So here I am on a terminal window on my local machine,

and I'm in the Downloads directory.

And before I log into my EC2 instance,

I need to change the permissions on my SSH key.

So type chmod 400 and the name of your key pair.

And then we are ready to SSH onto our EC2 instance.

So type ssh ec2-user@

and then paste the IP address,

then -I, and the name of your key pair,

and hit Enter.

Answer yes to this question.

And there we are.

We are logged into our EC2 instance.

Now the great thing about the Amazon Linux 2 AMI

is that it comes with the AWS CLI pre-installed.

So when we do it this way,

we don't need to install anything.

We can just go ahead and start running AWS commands

straight away on our instance.

So we can run aws s3 ls

and hit Enter.

And this is a command that lists all of the S3 buckets.

So type that command and hit Enter.

However, we're getting an error message at this stage

because we haven't yet configured our AWS CLI.

So we're getting this error message,

which tells us it's unable to locate any credentials.

So let's go ahead and configure the AWS CLI.

So type aws configure,

and hit Enter.

And now it's asking us for an access key.

And this is where our Identity

and Access Management user comes in.

So we need to create an IAM user

so that we get an access key and secret access key

that we can use to configure the CLI.

So let's head back to the console and do that.

So back in the console,

search for IAM,

then head to Users,

Add Users,

and I'm going to add a user called faye,

but you can add a user with your own name.

I'll give my user programmatic access.

Hit Next.

And now we need to set up some permissions for our user.

Now using Identity and Access Management groups

is the best practice way to manage your user permissions

by job function.

So let's go ahead and create a group.

So select Create Group,

and I'm going to call my group developers.

And then down here, we can select a ready-made policy

to give some permissions to the users

who will be members of this group.

Now we could select AdministratorAccess,

but this is going to provide full access to all AWS

services. And do we really want our developers

to have complete access to every AWS service?

Probably not.

And this user for this lab is just going to be using S3.

So let's search for S3 instead.

And I'm going to select S3FullAccess.

So this is going to give access

to all the buckets within the account

and the user will be able to create and delete S3 buckets.

Now, if you had a user that only needed read-only access,

then you could use this policy instead.

But I want this user to have full access to all S3 buckets.

So to be able to create, delete, and upload objects as well.

So once you've selected S3FullAccess,

you can go ahead and create the group.

So our user is a member of this group.

Here's the attached policy.

We can go ahead and hit Next.

Here's where you can add an optional tag,

but I'm just going to hit Next and Create User.

So there we go, our user's been created,

and we've got this access key and secret access key

that we'll be using to configure our AWS CLI.

And this is what's going to give us programmatic access to

AWS based on the permissions attached to this user.

So first of all, copy the access key,

and come back to the terminal window.

Paste the access key in here

and hit Enter.

Now it's asking for the secret access key.

So back in the AWS console,

you can copy your secret access key

and paste it in your terminal window and hit Enter.

The default region name is going to be us-east-1,

and remember to type a Region and not an Availability Zone.

So it must be us-east-1,

not us-east-1a, or b,

and hit Enter.

And the default output format can be JSON

or it can be text, but I like JSON,

so I'm just going to type JSON and hit Enter.

And that is our AWS Command Line Interface configured.

And if you run AWS configure list, and hit Enter,

it will show you your configuration,

and it just shows the last 4 characters

of the access key and secret access key.

Here's the default Region,

and there's also a configuration file as well.

So we can take a look at the configuration file,

and it just shows our output format and our Region.

So now I'll just clear the screen.

And now, if we run AWS S3 ls

and hit Enter,

we should no longer get that error message.

And if you had any S3 buckets in your account,

then you would see them in the output of this command.

So now let's go ahead and create an S3 bucket.

So type aws s3 mb, for make bucket,

s3:// and the name of your bucket.

And I'm just going to call my bucket acloudguru

and a load of random numbers,

and hopefully that's going to give me a unique bucket name.

And hit Enter.

So that has created our bucket.

And now, if we type aws s3 ls and hit Enter,

it's displaying the name of our bucket.

So now let's create a text file and upload it to our bucket.

So type echo "Hello Cloud Gurus"

And then we're going to redirect this text to a new file.

So type &gt;, which is a shift

and a full stop on my keyboard

and the name of the file that we are creating,

which is hello.txt.

And hit Enter.

If you type ls, you'll see your text file.

And now we're going to upload it to S3.

So type aws s3

cp, for copy,

the name of our text file, hello.txt,

then s3://

and paste in the name of your bucket.

And hit Enter.

So that has uploaded our file to the S3 bucket.

And now, if we type aws s3 ls and the name of our bucket

and hit Enter, it will list out the contents of that bucket.

And there is our file.

Let's head back to the console.

Here's our user with the access key and secret access key,

which is going to close down that screen.

And I'm going to select my user over here.

Head over to Security Credentials

and find your access key.

So here is our access key,

but you'll notice that it's not displaying

the secret access key

because you only get to see the secret access key

when you first create it.

And you do get an opportunity when you first create it

to download those credentials

and store them in a safe place should you need to.

However, if you lose your access key and secret access key

or they become corrupted, then you will need to come

to the user's security credentials,

and you will need to delete them,

and you can actually make them inactive,

or you can just go ahead and delete them.

So let's go ahead and do that right now.

And I'm going to deactivate my access key.

So select Deactivate,

and then I'm going to delete it as well.

And when we delete the access key,

it means that any AWS API call

made using this key is going to fail.

So I'm going to select Delete.

And if you ever wanted to create new ones for this user,

you can just go ahead and create a new one,

and it will create a new access key and secret access key.

And then that user will need to start again with the AWS CLI

and configure it all over again.

But we are not going to do that.

What I want to show you is that the AWS CLI

will have stopped working now, or it should have,

so come back to your terminal window.

And if we try to run aws s3 ls again,

you should see an error message similar to this.

So it's telling us that an error has occurred.

We've got an invalid access key ID

when calling the list buckets operation,

and the AWS access key ID that you provided

no longer exists.

So in order to fix this, we'd have to go in,

create new access keys, and run aws configure again

using the new access key and secret access key credentials.

So now let's head back to the console,

and I'm going to search for S3 in the search box.

Here's the bucket that we created using the CLI.

Here's our text file.

And if we select Open,

we can view the contents of the text file.

And we've done all of this

just using the AWS Command Line Interface.

And then the last thing I wanted to show you

is the documentation for the CLI.

So if you open a new browser tab

and search for AWS CLI v2,

you'll find the documentation for the AWS CLI.

And the syntax is always aws,

followed by the service that you're trying to access,

and then followed by some command or list of commands.

And if we scroll down to the bottom of this page,

you'll see all the available services

that you can access using the AWS CLI.

And if we scroll right down

almost to the bottom,

this is where you'll find s3, so select that,

and it will show you all the different options available.

So here's the cp command that we used earlier,

mb, which is used to make the bucket,

ls, which is used to list the contents of a bucket

or list the buckets that you have in your account,

rb just removes the bucket.

And you don't need to remember

all of these commands for the exam--

it's just good to understand how to read the documentation.

And then down here, we've got a few examples as well,

and all of the available commands are here.

So you can click on the available commands

to learn a little bit more about each of these commands.

So that's just a little bit about the documentation,

and it's just good to understand how to read it

when you need to find a new command.

And if you're working in your own AWS account,

just remember to go in and delete your EC2 instance

once you've finished.

And I've also got some exam tips for you as well.

So first of all, be mindful of the principle

of least privilege,

and that means always give your users

the minimum amount of access

that is required to do their job.

So we gave our user access to S3 but not to everything else.

It's best practice to use Identity and Access Management

groups. So create IAM groups, assign your users to groups,

and group permissions are assigned

using IAM policy documents.

And if you remember,

we assigned the S3FullAccess policy document

to our developers group.

And any user we add to that group

will automatically inherit the permissions of the group.

So by adding our user to the developers group,

we inherited the S3 full access permissions.

When it comes to the secret access key,

you are only going to see this once

when you first create it.

So if you lose it, you can delete the access key

and secret access key like we did

and then just go in and regenerate them.

But you will need to run aws configure again

to provide the new credentials.

Remember not to share your key pairs,

and each developer should have their own access key ID

and secret access key.

Just like passwords,

they should not be shared within the team.

And the AWS CLI is available for Linux, Windows,

and even MacOS,

and you can also use it on EC2 instances as well.

And the great thing about the Amazon Linux 2 AMI

is that it comes with the CLI already installed.

So that is it for this lesson.

If you have any questions, please let me know.

Otherwise, I will see you in the next lesson. Thank you.

[SPANISH]

Hola, Cloud Gurus, y bienvenidos a esta lección.

que será una demostración de "AWS CLI".

Entonces, en esta lección, vamos a ensuciarnos las manos.

mediante la interfaz de línea de comandos de AWS.

Comenzaremos lanzando una instancia EC2,

y aquí es donde vamos a utilizar la CLI de AWS.

A continuación, crearemos un usuario de Gestión de acceso e identidad,

y le daremos permisos al usuario

para acceder y crear recursos de S3.

Y finalmente, configuraremos la CLI de AWS

en nuestra instancia EC2 utilizando las credenciales de usuario de IAM.

Y luego usaremos la CLI de AWS

para crear un depósito S3 y cargar un archivo.

Entonces, si desea ensuciarse las manos con la CLI de AWS,

entonces por favor únase a mí en la consola ahora.

Entonces, desde la consola de AWS, haré todo

de la Región del Norte de Virginia.

Y antes que nada, vamos a lanzar una instancia EC2.

Entonces busque EC2 en el cuadro de búsqueda

y seleccione Iniciar instancia.

Le daremos un nombre

y asegúrese de que la AMI de Amazon Linux esté seleccionada.

Bajo el tipo de instancia, podemos quedarnos con el t2.micro.

Crearemos un nuevo par de claves

para que podamos iniciar sesión en la instancia EC2.

Y lo llamaré mykp y Crear par de claves,

y descargará el par de claves a su máquina local.

En la configuración de red,

asegúrese de tener una IP pública asignada automáticamente

configurado para habilitar.

Por defecto, el asistente de lanzamiento

va a crear un grupo de seguridad,

que permite SSH desde cualquier lugar.

Así que ahora puedes seguir adelante,

aceptar el resto de los valores predeterminados,

y Instancia de lanzamiento.

Entonces, mientras se inicia, puede continuar

y seleccione su ID de instancia,

y podemos ver el progreso de nuestra instancia aquí.

Y por el momento, sigue en estado pendiente,

así que ten paciencia,

y una vez que ha pasado sus controles de estado,

debemos ser buenos para continuar.

Podemos refrescar la pantalla para refrescar nuestro estado.

Actualizar de nuevo.

Y una vez que haya pasado las 2 verificaciones de estado,

eres bueno para ir

Así que ahora, seleccione su ID de instancia,

encuentre su dirección IP pública; está aquí en la parte superior,

copiar eso a su portapapeles,

y luego diríjase a una ventana de terminal en su máquina local.

Y vamos a iniciar sesión en nuestra instancia EC2.

Así que aquí estoy en una ventana de terminal en mi máquina local,

y estoy en el directorio de Descargas.

Y antes de iniciar sesión en mi instancia EC2,

Necesito cambiar los permisos en mi clave SSH.

Entonces escriba chmod 400 y el nombre de su par de claves.

Y luego estamos listos para SSH en nuestra instancia EC2.

Así que escriba ssh ec2-user@

y luego pegue la dirección IP,

luego -I, y el nombre de su par de claves,

y presione Entrar.

Responda sí a esta pregunta.

Y ahí estamos.

Hemos iniciado sesión en nuestra instancia EC2.

Ahora, lo mejor de la AMI de Amazon Linux 2

es que viene con la CLI de AWS preinstalada.

Así que cuando lo hacemos de esta manera,

necesitamos instalar nada.

Podemos seguir adelante y comenzar a ejecutar los comandos de AWS

inmediatamente en nuestra instancia.

Entonces podemos ejecutar aws s3 ls

y presione Entrar.

Y este es un comando que enumera todos los cubos S3.

Así que escribe ese comando y presiona Enter.

Sin embargo, recibimos un mensaje de error en esta etapa.

porque aún no hemos configurado nuestra AWS CLI.

Así que estamos recibiendo este mensaje de error,

lo que nos dice que no puede localizar ninguna credencial.

Así que sigamos adelante y configuremos la CLI de AWS.

Así que escriba aws configure,

y presione Entrar.

Y ahora nos pide una clave de acceso.

Y aquí es donde nuestra Identidad

y entra el usuario de Gestión de acceso.

Entonces necesitamos crear un usuario de IAM

para que obtengamos una clave de acceso y una clave de acceso secreta

que podemos usar para configurar la CLI.

Así que regresemos a la consola y hagamos eso.

Así que de vuelta en la consola,

buscar IAM,

luego dirígete a Usuarios,

Agregar usuarios,

y voy a agregar un usuario llamado faye ,

pero puede agregar un usuario con su propio nombre.

Le daré a mi usuario acceso programático.

Presiona Siguiente.

Y ahora necesitamos configurar algunos permisos para nuestro usuario.

Ahora usando grupos de administración de acceso e identidad

es la mejor forma de administrar sus permisos de usuario

por función laboral.

Así que sigamos adelante y creemos un grupo.

Así que seleccione Crear grupo,

y voy a llamar a mi grupo de desarrolladores.

Y luego aquí abajo, podemos seleccionar una política preparada

para dar algunos permisos a los usuarios

quienes serán miembros de este grupo.

Ahora podríamos seleccionar AdministratorAccess ,

pero esto proporcionará acceso completo a todos los AWS

servicios \_ ¿Y realmente queremos que nuestros desarrolladores

tener acceso completo a todos los servicios de AWS?

Probablemente no.

Y este usuario de este laboratorio solo usará S3.

Así que busquemos S3 en su lugar.

Y voy a seleccionar S3FullAccess.

Así que esto va a dar acceso

a todos los cubos dentro de la cuenta

y el usuario podrá crear y eliminar depósitos S3.

Ahora, si tenía un usuario que solo necesitaba acceso de solo lectura,

entonces podría usar esta política en su lugar.

Pero quiero que este usuario tenga acceso completo a todos los depósitos de S3.

Entonces, para poder crear, eliminar y cargar objetos también.

Una vez que haya seleccionado S3FullAccess,

puede continuar y crear el grupo.

Así que nuestro usuario es miembro de este grupo.

Aquí está la política adjunta.

Podemos continuar y pulsar Siguiente .

Aquí es donde puede agregar una etiqueta opcional,

pero voy a pulsar Siguiente y Crear usuario.

Ahí vamos, nuestro usuario ha sido creado,

y tenemos esta clave de acceso y clave de acceso secreta

que usaremos para configurar nuestra CLI de AWS.

Y esto es lo que nos dará acceso programático a

AWS en función de los permisos asociados a este usuario.

Entonces, antes que nada, copie la clave de acceso,

y vuelve a la ventana de la terminal.

Pegue la clave de acceso aquí

y presione Entrar.

Ahora está pidiendo la clave de acceso secreta.

De vuelta en la consola de AWS,

puede copiar su clave de acceso secreta

y péguelo en la ventana de su terminal y presione Entrar.

El nombre de región predeterminado será us-east-1,

y recuerde escribir una Región y no una Zona de disponibilidad.

Así que debe ser us-east-1,

no us-east-1a, o b,

y presione Entrar.

Y el formato de salida predeterminado puede ser JSON

o puede ser texto, pero me gusta JSON,

así que voy a escribir JSON y presionar Enter.

Y esa es nuestra interfaz de línea de comandos de AWS configurada.

Y si ejecuta la lista de configuración de AWS y pulsa Intro,

te mostrará su configuración,

y solo muestra los últimos 4 caracteres

de la clave de acceso y la clave secreta de acceso.

Esta es la región predeterminada,

y también hay un archivo de configuración.

Así que podemos echar un vistazo al archivo de configuración,

y solo muestra nuestro formato de salida y nuestra Región.

Así que ahora limpiaré la pantalla.

Y ahora, si ejecutamos AWS S3 ls

y pulsa Intro,

deberíamos recibir ese mensaje de error.

Y si tenía depósitos de S3 en su cuenta,

entonces los vería en la salida de este comando.

Así que ahora sigamos adelante y creemos un depósito S3.

Así que escriba aws s3 mb , para hacer cubo,

s3:// y el nombre de su depósito.

Y solo voy a llamar a mi balde acloudguru

y un montón de números aleatorios,

y con suerte eso me dará un nombre de depósito único.

Y presiona Entrar.

Eso ha creado nuestro cubo.

Y ahora, si escribimos aws s3 ls y presionamos Enter ,

está mostrando el nombre de nuestro cubo.

Así que ahora vamos a crear un archivo de texto y subirlo a nuestro cubo.

Así que escribe echo "Hola Cloud Gurus"

Y luego vamos a redirigir este texto a un nuevo archivo.

Así que escriba & gt ;, que es un cambio

y un punto en mi teclado

y el nombre del archivo que estamos creando,

que es hola.txt.

Y presiona Entrar.

Si escribe ls , verá su archivo de texto.

Y ahora vamos a subirlo a S3.

Así que escriba aws s3

cp , para copiar,

el nombre de nuestro archivo de texto, hola.txt,

entonces s3://

y pega el nombre de tu depósito.

Y presiona Entrar.

Eso ha cargado nuestro archivo en el depósito S3.

Y ahora si tecleamos aws s3 ls y el nombre de nuestro bucket

y presione Entrar, mostrará una lista del contenido de ese cubo.

Y ahí está nuestro archivo.

Volvamos a la consola.

Aquí está nuestro usuario con la clave de acceso y la clave de acceso secreta,

que va a cerrar esa pantalla.

Y voy a seleccionar mi usuario aquí.

Dirígete a Credenciales de seguridad

y encuentre su clave de acceso.

Así que aquí está nuestra clave de acceso,

pero notará que no se muestra

la clave de acceso secreta

porque solo puedes ver la clave de acceso secreta

cuando lo creas por primera vez .

Y obtienes una oportunidad cuando la creas por primera vez.

para descargar esas credenciales

y guárdelos en un lugar seguro si es necesario.

Sin embargo, si pierde su clave de acceso y su clave de acceso secreta

o se corrompen, entonces tendrás que venir

a las credenciales de seguridad del usuario,

y tendrás que eliminarlos,

y en realidad puedes hacerlos inactivos,

o simplemente puede continuar y eliminarlos.

Así que sigamos adelante y hagámoslo ahora mismo.

Y voy a desactivar mi clave de acceso.

Así que seleccione Desactivar,

y luego voy a borrarlo también.

Y cuando borramos la clave de acceso,

significa que cualquier llamada a la API de AWS

hecho usando esta clave va a fallar.

Así que voy a seleccionar Eliminar.

Y si alguna vez quisiste crear otros nuevos para este usuario,

puedes seguir adelante y crear uno nuevo,

y creará una nueva clave de acceso y una clave de acceso secreta.

Y luego ese usuario deberá comenzar de nuevo con la AWS CLI

y configurarlo todo de nuevo.

Pero no vamos a hacer eso.

Lo que quiero mostrarles es que la CLI de AWS

habrá dejado de funcionar ahora, o debería haberlo hecho,

así que vuelve a la ventana de tu terminal.

Y si intentamos ejecutar aws s3 ls nuevamente,

Deberías ver un mensaje de error similar a este.

Entonces nos está diciendo que ha ocurrido un error.

Tenemos un ID de clave de acceso no válido.

al llamar a la operación de cubos de lista,

y el ID de la clave de acceso de AWS que proporcionó

ya no existe

Entonces, para arreglar esto, tendríamos que entrar,

cree nuevas claves de acceso y ejecute aws configure nuevamente

utilizando la nueva clave de acceso y las credenciales de la clave de acceso secreta.

Así que ahora volvamos a la consola,

y voy a buscar S3 en el cuadro de búsqueda.

Aquí está el cubo que creamos usando la CLI.

Aquí está nuestro archivo de texto.

Y si seleccionamos Abrir,

podemos ver el contenido del archivo de texto.

Y hemos hecho todo esto

simplemente usando la interfaz de línea de comandos de AWS.

Y luego lo último que quería mostrarte

es la documentación para la CLI.

Así que si abres una nueva pestaña del navegador

y busque AWS CLI v2,

encontrará la documentación para la CLI de AWS.

Y la sintaxis siempre es aws ,

seguido del servicio al que intenta acceder,

y luego seguido de algún comando o lista de comandos.

Y si nos desplazamos hacia abajo hasta el final de esta página,

verás todos los servicios disponibles

a los que puede acceder mediante la CLI de AWS.

Y si nos desplazamos hacia abajo

casi hasta el fondo,

aquí es donde encontrarás s3, así que selecciónalo,

y le mostrará todas las diferentes opciones disponibles.

Así que aquí está el comando cp que usamos antes,

mb , que se usa para hacer el balde,

ls , que se usa para enumerar el contenido de un cubo

o enumere los cubos que tiene en su cuenta,

rb simplemente elimina el cubo.

Y no necesitas recordar

todos estos comandos para el examen--

es bueno entender cómo leer la documentación.

Y aquí abajo, también tenemos algunos ejemplos,

y todos los comandos disponibles están aquí.

Para que pueda hacer clic en los comandos disponibles

para aprender un poco más sobre cada uno de estos comandos.

Eso es solo un poco sobre la documentación,

y es bueno entender cómo leerlo

cuando necesite encontrar un nuevo comando.

Y si está trabajando en su propia cuenta de AWS,

solo recuerde ingresar y eliminar su instancia EC2

una vez que haya terminado.

Y también tengo algunos consejos de examen para ti.

Así que antes que nada, tenga en cuenta el principio

de menor privilegio,

y eso significa dar siempre a sus usuarios

la cantidad mínima de acceso

que se requiere para hacer su trabajo.

Así que le dimos a nuestro usuario acceso a S3 pero no a todo lo demás.

Es una buena práctica usar Gestión de acceso e identidad

grupos \_ Así que cree grupos de IAM, asigne sus usuarios a grupos,

y se asignan permisos de grupo

mediante documentos de política de IAM.

Y si recuerdas,

asignamos el documento de política S3FullAccess

a nuestro grupo de desarrolladores.

Y cualquier usuario que agreguemos a ese grupo

heredará automáticamente los permisos del grupo.

Entonces, al agregar nuestro usuario al grupo de desarrolladores,

heredamos los permisos de acceso completo de S3.

Cuando se trata de la clave de acceso secreta,

vas a ver esto una vez

cuando lo creas por primera vez.

Entonces, si la pierde, puede eliminar la clave de acceso

y clave de acceso secreta como hicimos nosotros

y luego entrar y regenerarlos.

Pero deberá ejecutar aws configure nuevamente

para proporcionar las nuevas credenciales.

Recuerde no compartir sus pares de claves,

y cada desarrollador debe tener su propia ID de clave de acceso

y clave de acceso secreta.

Al igual que las contraseñas,

compartirse dentro del equipo.

Y la AWS CLI está disponible para Linux, Windows,

e incluso macOS ,

y también puede usarlo en instancias EC2.

Y lo bueno de la AMI de Amazon Linux 2

es que viene con el CLI ya instalado.

Eso es todo por esta lección.

Si tiene alguna pregunta, por favor hágamelo saber.

De lo contrario, te veré en la próxima lección. Gracias.