[ENGLISH]

Hello Cloud Gurus and welcome to this lesson

where we'll be configuring Cross Origin Resource Sharing

also known as CORS.

We'll begin by configuring two S3 buckets,

my index website and my CORS test bucket.

One is going to host our index.html

and this is where we're gonna be accessing the website from.

And our other bucket is going to host loadpage.html.

Next we're gonna update the index.html file

to reference loadpage.html which is stored

in the other bucket.

So I'll test what happens when we try

to access our index.html page,

and by default it's not going to be able

to access loadpage.html, because by default

this is not going to be allowed

because Cross Origin Resource Sharing

is not allowed by default.

So then we'll configure CORS

and by configuring CORS in my CORS test bucket

that is going to allow our index.html file

to access loadpage.html.

And after that we should be able to see the contents

of both pages when we log into our website.

So in this context CORS is basically a way

of allowing code that's in one S3 bucket

to access code that's in another S3 bucket.

And effectively we're allowing one resource

to access another resource,

and that's why it's called Cross Origin Resource Sharing.

So if you'd like to try it out for yourself

please join me in the AWS console.

From the console the first thing we'll do

is create our S3 bucket.

So search for S3.

Create bucket.

I'll call it my index bucket.

And add some random numbers on the end.

Scroll down, disable block public access,

and acknowledge.

Then scroll down to the bottom, and create bucket.

Now as this bucket is gonna host a website,

we need to update the bucket policy

to allow public access.

So select your bucket,

and select permissions.

Scroll down, and we need to edit our bucket policy.

So select edit.

First of all I'm gonna copy my bucket ARN,

and select policy generator.

The type of policy is a bucket policy.

Effect is allow.

Principle is going to be star.

AWS service is S3.

The action is going to be, get object.

So scroll right down until you find get object.

And there it is, so select that.

The Amazon resource name is the ARN

of our bucket that you just copied.

So paste that in there.

And then after that we also need to add

forward slash and star,

and that will enable this policy

to apply to everything in the bucket.

Then select add statement, generate policy,

and there's our bucket policy.

Copy the policy, come back to our bucket,

and paste in the policy.

Then scroll down.

We can ignore this error message,

and hit save.

Next we need to enable static website hosting

on our bucket.

So select properties.

You need to scroll right down to the bottom,

until you find static website hosting,

select edit, select enable.

Down here we'll specify the index document,

it's index.html, and the error document is error.html.

Then scroll down to the bottom and save.

Next we need to upload those HTML files.

And you will find all of the files that we're gonna use

in the resources for this lesson.

So here's our index.html, our error.html,

and these are the ones that we're gonna be adding.

So download all the files to your local machine,

and then in your S3 bucket select objects,

upload, add files,

and I'll select index.html

and error.html,

and upload.

After we've done that, close down that window,

head back to properties, scroll down to the bottom

until you find static website hosting,

copy the bucket website endpoint,

and I'm just gonna paste it in at this browser tab here.

And if it's all worked, this is what you should see.

So this is just displaying our index.html.

So far so good.

Next we're gonna create our second bucket.

So head back to the S3 console,

select buckets, create bucket.

I'm gonna call it my CORS test bucket,

and add some random numbers on the end.

Scroll down, deselect the block public access settings,

and acknowledge.

And create bucket.

Once again, we need to update the bucket policy

to allow public read access for this bucket.

So select your bucket,

select permissions,

scroll down to bucket policy,

select edit, copy your bucket ARN,

open up the policy generator,

select S3 bucket policy,

effect is allow,

principle is star,

service is S3,

actions is get object.

There it is.

Amazon resource name is the ARN of your bucket.

Remember to add forward slash and star,

add statement, generate policy,

copy the policy,

and paste it into your bucket policy.

Then scroll down,

we can ignore this message,

and save changes.

Next we'll enable static website hosting.

So in properties, scroll down,

to static website hosting.

Hit edit, enable.

The index document is going to be index.html,

and the error document is error.html,

and save changes.

Next we'll upload the relevant files,

so select objects, upload,

add files, and this time we're gonna add loadpage.html,

select open, and we can also add error.html as well.

Then scroll down, and upload.

After you've done that, we'll close down that view,

select properties,

and scroll down to the static website hosting section.

I'm gonna copy my bucket endpoint,

and paste in the URL, and hit enter.

Now by itself it's not gonna work

it's gonna tell us we've got the error,

so it's going straight to our error.html,

but I want you to type forward slash loadpage.html

and hit enter.

There we go.

And this is the message you should see,

"The page has loaded successfully Cloud Gurus."

So now after we've got that working,

I want you to copy this website URL.

So copy the full URL including loadpage.html,

and we are gonna update our index.html file.

So here is my file.

And my index.html actually references loadpage.html.

But as that's in a different bucket at the moment,

it can't actually access it.

So we need to tell the code where to find this page.

So we're gonna replace loadpage.html

with our website URL.

And there it is.

So once you've finished, this is what it should look like.

I'm gonna save my file.

And then upload this new version to S3.

Come back to S3.

At the top select buckets.

Come to my index bucket.

First of all, we're gonna delete this one.

And we're gonna upload the new version.

Hit close, and select properties.

Scroll down until you find static website hosting,

and select your bucket website endpoint.

Now, even though we added the web URL

for loadpage.html, it cannot access it.

And this is because by default AWS does not allow

a file in one S3 bucket to access a file

that's located in another bucket.

And if you are a Chrome user like me,

you can actually see the CORS related error messages

if you select view, developer,

developer tools, then make sure that console is selected.

And here is our error message.

So we can see that the HTTP request

to our loadpage.html, from this origin

has been blocked by the CORS policy.

So no access control allow origin header

is present on the requested resource.

And it has failed to load that part of the page.

So this is why we need to set up CORS.

So to configure CORS I'm gonna close down

my developer tools, head back to the S3 console,

we're gonna grab the website URL

of our bucket website endpoint for our index bucket,

so just make sure you've copied that.

Then scroll back up to buckets,

select my CORS test bucket,

permissions, scroll down,

until you get to Cross Origin Resource Sharing

and select edit.

Then paste in the website URL

because we're gonna need it in a moment.

And I've included a sample policy

in the resources for this lesson.

Here it is, it's called Sample CORS Configuration.

This is the file.

And I'm just gonna copy everything,

then back in my CORS test bucket

I'll paste the sample policy in there.

And the only thing we need to do

is replace this star, with the URL of our bucket.

So I'll copy the URL, and paste it in there,

and I'm just gonna get rid of that top line.

And when you've finished, this is how it should look.

So it's allowing the get method

for only this particular origin

for my index bucket.

Once you've done that, save changes,

and we should be good to go.

Now head back to buckets, select my index bucket,

properties, scroll down to the bottom,

and select our bucket website endpoint.

And there we go.

And if it's all worked, this is what you should see.

So you should see the message from both pages.

This one coming from index.html

and this one is coming from loadpage.html

located in the other bucket.

And if you do have problems, then just make sure,

that you've configured your CORS policy correctly.

So this is what it should look like.

So if you are experiencing any issues,

just make sure that you've configured

this Cross Origin Resource Sharing policy correctly.

So for the exam, just remember

that Cross Origin Resource Sharing or CORS,

can be used to allow resources in one S3 bucket

to access resources that are located

in another S3 bucket.

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 configuraremos el uso compartido de recursos de origen cruzado

también conocido como CORS.

Comenzaremos configurando dos cubos S3,

mi sitio web de índice y mi cubo de prueba CORS.

Uno va a alojar nuestro index.html

y aquí es desde donde accederemos al sitio web.

Y nuestro otro depósito alojará loadpage.html.

A continuación, actualizaremos el archivo index.html.

para hacer referencia a loadpage.html que se almacena

en el otro balde.

Así que voy a probar lo que sucede cuando lo intentamos

para acceder a nuestra página index.html,

y por defecto no va a poder

para acceder a loadpage.html, porque por defecto

esto no se va a permitir

porque el intercambio de recursos de origen cruzado

está permitido por defecto.

Entonces configuraremos CORS

y configurando CORS en mi cubo de prueba CORS

eso permitirá que nuestro archivo index.html

para acceder a loadpage.html.

Y después de eso deberíamos poder ver los contenidos.

de ambas páginas cuando iniciamos sesión en nuestro sitio web.

Entonces, en este contexto, CORS es básicamente una forma

de permitir el código que está en un cubo S3

para acceder al código que está en otro depósito S3.

Y efectivamente estamos permitiendo que un recurso

para acceder a otro recurso,

y es por eso que se llama intercambio de recursos de origen cruzado.

Así que si quieres probarlo por ti mismo

Únase a mí en la consola de AWS.

Desde la consola lo primero que haremos

es crear nuestro depósito S3.

Así que busca S3.

Crear balde.

Lo llamaré mi cubo de índice.

Y agregue algunos números aleatorios al final.

Desplácese hacia abajo, deshabilite bloquear el acceso público,

y reconocer

Luego, desplácese hacia abajo hasta la parte inferior y cree un depósito.

Ahora que este depósito alojará un sitio web,

necesitamos actualizar la política de depósito

para permitir el acceso público.

Así que seleccione su cubo,

y seleccione los permisos.

Desplácese hacia abajo y necesitamos editar nuestra política de depósito.

Así que selecciona editar.

En primer lugar, voy a copiar el ARN de mi depósito,

y seleccione el generador de políticas.

El tipo de política es una política de depósito.

El efecto está permitido.

Principio va a ser estrella.

El servicio de AWS es S3.

La acción va a ser, obtener objeto.

Así que desplácese hacia abajo hasta que encuentre el objeto.

Y ahí está, así que seleccione eso.

El nombre del recurso de Amazon es el ARN

de nuestro balde que acabas de copiar.

Así que pega eso ahí.

Y luego, después de eso, también debemos agregar

oblicua y estrella,

y eso habilitará esta política

para aplicar a todo en el balde.

Luego seleccione agregar declaración, generar política,

y ahí está nuestra política de depósito.

Copie la política, vuelva a nuestro cubo,

y pegar en la política.

Luego, desplácese hacia abajo.

Podemos ignorar este mensaje de error,

y pulsa guardar.

A continuación, debemos habilitar el alojamiento de sitios web estáticos

en nuestro balde.

Así que selecciona propiedades.

Tienes que desplazarte hacia abajo hasta el final,

hasta que encuentre alojamiento de sitio web estático,

seleccione editar, seleccione habilitar.

Aquí abajo especificaremos el documento índice,

es index.html y el documento de error es error.html.

Luego, desplácese hacia abajo hasta el final y guarde.

A continuación, debemos cargar esos archivos HTML.

Y encontrarás todos los archivos que vamos a usar

en los recursos para esta lección.

Así que aquí está nuestro index.html, nuestro error.html,

y estos son los que vamos a agregar.

Así que descargue todos los archivos a su máquina local,

y luego en su cubo S3 seleccione objetos,

cargar , agregar archivos,

y seleccionaré index.html

y error.html,

y subir

Después de que hayamos hecho eso, cierre esa ventana,

regrese a las propiedades, desplácese hacia abajo hasta la parte inferior

hasta que encuentre alojamiento de sitio web estático,

copie el punto final del sitio web del depósito,

y voy a pegarlo aquí en esta pestaña del navegador.

Y si todo funcionó, esto es lo que deberías ver.

Así que esto solo muestra nuestro index.html.

Hasta ahora, todo bien.

A continuación, vamos a crear nuestro segundo cubo.

Así que regresa a la consola S3,

seleccionar cubos, crear cubo.

Voy a llamarlo mi cubo de prueba CORS,

y agregue algunos números aleatorios al final.

Desplácese hacia abajo, anule la selección de la configuración de bloqueo de acceso público,

y reconocer

Y crea un balde.

Una vez más, necesitamos actualizar la política de depósitos.

para permitir el acceso público de lectura para este depósito.

Así que seleccione su cubo,

seleccionar permisos,

desplácese hacia abajo hasta la política del depósito,

seleccione editar, copie el ARN de su depósito,

abre el generador de políticas,

seleccione la política de depósito de S3,

el efecto es permitir,

el principio es estrella,

el servicio es S3,

acciones es obtener objeto.

Ahí está.

El nombre del recurso de Amazon es el ARN de su depósito.

Recuerde agregar barra diagonal y estrella,

agregar declaración, generar política,

copiar la política,

y péguelo en su política de depósito.

Luego desplácese hacia abajo,

podemos ignorar este mensaje,

y guardar los cambios.

A continuación, habilitaremos el alojamiento de sitios web estáticos.

Entonces, en propiedades, desplácese hacia abajo,

al alojamiento de sitios web estáticos.

Presiona editar, habilitar.

El documento de índice va a ser index.html,

y el documento de error es error.html,

y guardar los cambios.

A continuación, subiremos los archivos relevantes,

así que seleccione objetos, cargue,

agregar archivos, y esta vez vamos a agregar loadpage.html,

seleccione abrir, y también podemos agregar error.html también.

Luego, desplácese hacia abajo y cargue.

Después de que haya hecho eso, cerraremos esa vista,

seleccionar propiedades,

y desplácese hacia abajo hasta la sección de alojamiento de sitios web estáticos.

Voy a copiar el punto final de mi depósito,

y pegue la URL, y presione enter.

Ahora por sí solo no va a funcionar

es nos dirá que tenemos el error,

por lo que va directamente a nuestro error.html,

pero quiero que escribas barra diagonal loadpage.html

y pulsa enter.

Aquí vamos.

Y este es el mensaje que deberías ver,

"La página se ha cargado con éxito Cloud Gurus".

Así que ahora, después de que tengamos eso funcionando,

Quiero que copie la URL de este sitio web.

Así que copie la URL completa, incluido loadpage.html,

y vamos a actualizar nuestro archivo index.html.

Así que aquí está mi archivo.

Y mi index.html en realidad hace referencia a loadpage.html.

Pero como eso está en un cubo diferente en este momento,

no puede acceder a él.

Así que necesitamos decirle al código dónde encontrar esta página.

Así que vamos a reemplazar loadpage.html

con la URL de nuestro sitio web.

Y ahí está.

Entonces, una vez que haya terminado, así es como debería verse.

Voy a guardar mi archivo.

Y luego cargue esta nueva versión en S3.

Vuelve a S3.

En la parte superior seleccione cubos.

Ven a mi cubo de índice.

En primer lugar, vamos a eliminar este.

Y vamos a subir la nueva versión.

Presiona cerrar y selecciona propiedades.

Desplácese hacia abajo hasta que encuentre alojamiento de sitios web estáticos,

y seleccione el punto final del sitio web de su depósito.

Ahora, aunque agregamos la URL web

para loadpage.html, no puede acceder a él.

Y esto se debe a que por defecto AWS no permite

un archivo en un depósito S3 para acceder a un archivo

que está ubicado en otro cubo.

Y si eres usuario de Chrome como yo,

puede ver los mensajes de error relacionados con CORS

si selecciona ver, desarrollador,

de desarrollador , luego asegúrese de que la consola esté seleccionada.

Y aquí está nuestro mensaje de error.

Entonces podemos ver que la solicitud HTTP

a nuestro loadpage.html, desde este origen

ha sido bloqueado por la política CORS.

Entonces ningún control de acceso permite el encabezado de origen

está presente en el recurso solicitado.

Y no ha podido cargar esa parte de la página.

Entonces, esta es la razón por la que necesitamos configurar CORS.

Así que para configurar CORS voy a cerrar

mis herramientas de desarrollador, regrese a la consola S3,

eran voy a tomar la URL del sitio web

del punto final del sitio web de nuestro depósito para nuestro depósito de índice,

así que asegúrate de haber copiado eso.

Luego desplácese hacia atrás hasta cubos,

seleccione mi cubo de prueba CORS,

permisos , desplácese hacia abajo,

hasta llegar a Intercambio de recursos de origen cruzado

y seleccione editar.

Luego pegue la URL del sitio web

porque lo vamos a necesitar en un momento.

Y he incluido una política de muestra

en los recursos para esta lección.

Aquí está, se llama Ejemplo de configuración de CORS.

Este es el archivo.

Y voy a copiar todo,

luego de vuelta en mi cubo de prueba CORS

Pegaré la política de muestra allí.

Y lo único que tenemos que hacer

es reemplazar esta estrella, con la URL de nuestro cubo.

Así que copiaré la URL y la pegaré allí,

y me voy a deshacer de esa línea superior.

Y cuando hayas terminado, así es como debería verse.

Entonces está permitiendo el método get

por este origen particular

para mi cubo de índice.

Una vez que haya hecho eso, guarde los cambios,

y deberíamos estar listos para irnos.

Ahora regrese a los cubos, seleccione mi cubo de índice,

propiedades , desplácese hacia abajo hasta la parte inferior,

y seleccione el punto de conexión del sitio web de nuestro depósito.

Y ahí vamos.

Y si todo funcionó, esto es lo que deberías ver.

Entonces deberías ver el mensaje de ambas páginas.

Este viene de index.html

y este viene de loadpage.html

ubicado en el otro balde.

Y si tienes problemas, entonces solo asegúrate,

que ha configurado su política CORS correctamente.

Así es como debería verse.

Entonces, si tiene algún problema,

solo asegúrese de haber configurado

esta política de uso compartido de recursos entre orígenes.

Entonces, para el examen, solo recuerda

que Intercambio de recursos de origen cruzado o CORS,

se puede usar para permitir recursos en un depósito S3

para acceder a los recursos que se encuentran

en otro cubo S3.

Eso es todo por esta lección.

Cualquier pregunta, hágamelo saber.

De lo contrario, te veré en la próxima.

Gracias.