[ENGLISH]

Hello, Cloud Gurus,

and welcome to part 1 of the CodePipeline demo.

So let's just run

through exactly what we're going to be doing.

First of all, we're going to use CloudFormation

to create an EC2 instance.

We're going to upload version 1 of our application

into an S3 bucket,

and then we're going to deploy our application

to our EC2 instance using CodeDeploy.

Then, in the next stage, we're going to build a CodePipeline

and manually trigger CodeDeploy

to update our application to version 2.

And then, finally,

we are going to configure an automated pipeline,

which is going to be triggered when we add a new version

of our code to the S3 bucket.

So that's automatically going to trigger a CloudWatch event,

which is going to trigger CodePipeline to redeploy the code

to our EC2 instance using CodeDeploy.

And there are a few prerequisites

that we need to take care of before we begin.

So first of all,

there are some resources that you will need to download

to your local machine,

which include application code

and a text file containing the commands

that we're going to run.

And you will find these resources

in the Resources section of the lecture.

Secondly, we need to configure an EC2 instance

and we'll use CloudFormation to automatically create

our EC2 instance, install the CodeDeploy agent,

and configure the identity and access management permissions

that are required to run CodeDeploy and CodePipeline.

And don't worry if you haven't used CloudFormation

very much before.

We are going to cover it in a lot more detail later on.

And the main thing that I want you to appreciate

about CloudFormation at this stage is

that it's a really easy way to automate the provisioning

of your AWS resources.

So without further ado, let's get started.

Now, the first thing I want to show you is

that there are a load of resources provided

for this lecture.

So firstly, we have a text file,

which contains all of the CloudFormation commands.

So these are the commands that we're going to use

to create our EC2 instance.

And there is also this code\_revisions ZIP file.

And when you unzip it,

it's going to create 3 different directories,

and that's going to be for 3 different versions

of the web application that we're going to install.

So if I go into the first directory,

we've got our appspec.yml,

which you should be familiar with already

if you've already done our CodeDeploy lab.

We have an index.html,

which is the webpage for our web application.

We have a scripts directory,

which contains 3 different scripts.

So there's an install\_dependencies script.

So CodeDeploy is going to go in and install httpd.

Then, we have a script to start httpd

and to stop it as well.

And these are all contained in this mywebapp.zip.

And that folder structure is reflected

in each of these 3 different directories.

So when you download the ZIP file

and it creates these directories for you,

so you don't need to unzip this mywebapp.zip,

because that's what we're going to be uploading

into CodeDeploy.

So just keep that folder structure exactly

as you see it on my screen.

So now, from the AWS Management Console,

we're going to create an EC2 key pair,

and this is going to enable us to log into our EC2 instance

in case we need to troubleshoot anything.

So head to Services,

and you'll find EC2 under the Compute section.

So select EC2.

From the EC2 Dashboard,

select Key Pairs and Create Key Pair.

Make sure that you're in the Northern Virginia region.

Give your key pair a simple name.

I'm going to call it nvkp, and Create Key Pair.

The next thing we need to do is create a couple

of S3 buckets and we need one

to store our CloudFormation template

and one to store our code in.

So head to Services, scroll down to Storage, and select S3.

Click on Create Bucket.

Make sure you're in the Northern Virginia region.

Give your bucket a name, and Create Bucket.

We'll also create our second bucket

and this is going to hold the application code.

For this bucket, we're going to enable versioning,

because we're going to be working with different versions

of our application code,

and eventually, we want CodePipeline

to recognize when we add a new version of our code

to this S3 bucket

and automatically launch the pipeline

when we add a new version.

And Create Bucket.

So that is our 2 buckets created.

And now we're ready to upload our CloudFormation template

to our bucket.

So select your CloudFormation template bucket,

Upload, Add Files,

and you need to select your CloudFormation template

and Upload.

Once that's uploaded, just close that view.

Let's click on our template and open it.

And here is our CloudFormation template.

And it basically just creates an EC2 instance

with the correct instance profile for CodeDeploy

and it creates a service role for CodeDeploy as well.

So it's basically setting up all the permissions

that we need.

So this is just automating everything for us.

Now, in order to run our CloudFormation command,

we need to grab the object URL

of our CloudFormation template and you'll find it down here.

So just copy that.

Here's our CloudFormation command

and we need to replace this URL

with the correct URL from our template.

So just paste that in here.

And this is what it should look like.

You also need to make sure that you've got the correct

parameter value for your key pair.

And mine was called nvkp.

These parameters are just allowing SSH from any IP address

and it's also configuring some tags as well.

So once you've done that, copy that entire command,

come back to the AWS Console,

and we can run this command using the CloudShell.

So select CloudShell at the top of your screen.

And the CloudShell is just a browser-based shell.

It gives you command line access to your AWS resources.

So we can just run the CloudFormation command from here.

And it's a really quick way of running AWS commands

without having to configure the CLI on your local machine.

Once your Cloud Shell is up and running,

you can just hit Ctrl+V.

It's asking you to verify.

So just verify the command, and paste, and hit Enter.

And that is going to launch our CloudFormation template.

Now, it does take a few minutes for CloudFormation

to complete.

And there's a command here which allows you

to verify the status.

So I'm just going to copy that,

paste that into my CloudShell,

and we can see the CloudFormation job is still in progress.

So I'm just going to leave that for a few minutes

and it normally takes about 5 minutes to complete.

So now's a good time to have a cup of tea.

And within about 5 minutes,

we should be good to continue.

After a few minutes, we can run that command again,

just using the up arrow, and hit Enter.

And if it's all completed, you should see Create Complete.

So once everything is completed,

we can go in and log into our EC2 instance

and just make sure

that the CodeDeploy agent was successfully installed.

So I'm going to head to Services, EC2,

come to the EC2 Dashboard.

Here's our instance.

Select your instance on the left and hit Connect.

And we'll use EC2 Instance Connect

to connect to our instance.

So just hit Connect,

and then type sudo service codedeploy-agent status,

and hit Enter.

And there we go.

The CodeDeploy agent has been successfully installed

on our EC2 instance.

So that's it for part one of this demo.

And if you're ready to continue to part two,

I will see you in the next lecture.

Thank you.

[SPANISH]

Hola, gurús de la nube,

y bienvenido a la parte 1 de la demostración de CodePipeline .

Así que solo corramos

a través de exactamente lo que vamos a estar haciendo.

En primer lugar, vamos a utilizar CloudFormation

para crear una instancia EC2.

Vamos a subir la versión 1 de nuestra aplicación

en un cubo S3,

y luego vamos a implementar nuestra aplicación

a nuestra instancia EC2 usando CodeDeploy .

Luego, en la siguiente etapa, vamos a construir un CodePipeline

y activar manualmente CodeDeploy

para actualizar nuestra aplicación a la versión 2.

Y entonces, finalmente,

vamos a configurar un pipeline automatizado,

que se activará cuando agreguemos una nueva versión

de nuestro código al depósito S3.

Eso activará automáticamente un evento de CloudWatch ,

que activará CodePipeline para volver a implementar el código

a nuestra instancia EC2 usando CodeDeploy .

Y hay algunos requisitos previos

que debemos cuidar antes de comenzar.

Entonces, en primer lugar,

hay algunos recursos que necesitarás descargar

a su máquina local,

que incluyen código de aplicación

y un archivo de texto que contiene los comandos

que vamos a correr.

Y encontrarás estos recursos

en la sección Recursos de la conferencia.

En segundo lugar, necesitamos configurar una instancia EC2

y usaremos CloudFormation para crear automáticamente

nuestra instancia EC2, instale el agente CodeDeploy ,

y configurar los permisos de administración de identidad y acceso

que son necesarios para ejecutar CodeDeploy y CodePipeline .

Y no se preocupe si no ha usado CloudFormation

mucho antes.

Vamos a cubrirlo con mucho más detalle más adelante.

Y lo principal que quiero que aprecies

acerca de CloudFormation en esta etapa es

que es una manera muy fácil de automatizar el aprovisionamiento

de sus recursos de AWS.

Así que sin más preámbulos, comencemos.

Ahora, lo primero que quiero mostrarles es

que hay un montón de recursos proporcionados

para esta conferencia.

En primer lugar, tenemos un archivo de texto,

que contiene todos los comandos de CloudFormation .

Entonces estos son los comandos que vamos a usar

para crear nuestra instancia EC2.

Y también está este archivo ZIP code\_revisions .

Y cuando lo abres,

va a crear 3 directorios diferentes,

y eso va a ser para 3 versiones diferentes

de la aplicación web que vamos a instalar.

Entonces, si entro en el primer directorio,

tenemos nuestro appspec.yml ,

el que ya deberías estar familiarizado

si ya ha realizado nuestro laboratorio de CodeDeploy .

Tenemos un index.html,

que es la página web de nuestra aplicación web.

Tenemos un directorio de scripts,

que contiene 3 guiones diferentes.

Entonces hay un script install\_dependencies .

Por lo tanto , CodeDeploy ingresará e instalará httpd .

Luego, tenemos un script para iniciar httpd

y para detenerlo también.

Y todo esto está contenido en este mywebapp.zip.

Y esa estructura de carpetas se refleja

en cada uno de estos 3 directorios diferentes.

Así que cuando descargas el archivo ZIP

y crea estos directorios para usted,

por lo que no necesita descomprimir este mywebapp.zip,

porque eso es lo que vamos a estar subiendo

en CodeDeploy .

Así que mantén esa estructura de carpetas exactamente

como lo ves en mi pantalla.

Así que ahora, desde la consola de administración de AWS,

vamos a crear un par de claves EC2,

y esto nos permitirá iniciar sesión en nuestra instancia EC2

en caso de que necesitemos solucionar cualquier problema.

Así que dirígete a Servicios,

y encontrará EC2 en la sección Compute.

Así que seleccione EC2.

Desde el panel EC2,

seleccione Pares de claves y Crear par de claves.

Asegúrate de estar en la región del norte de Virginia.

Dale a tu par de claves un nombre simple.

Voy a llamarlo nvkp y Crear par de claves.

Lo siguiente que tenemos que hacer es crear un par

de cubos S3 y necesitamos uno

para almacenar nuestra plantilla de CloudFormation

y otro para almacenar nuestro código.

Diríjase a Servicios, desplácese hacia abajo hasta Almacenamiento y seleccione S3.

Haga clic en Crear cubo.

Asegúrate de estar en la región del norte de Virginia.

Dale un nombre a tu depósito y crea un depósito.

También crearemos nuestro segundo cubo.

y esto va a contener el código de la aplicación.

Para este grupo, vamos a habilitar el control de versiones,

porque vamos a estar trabajando con diferentes versiones

de nuestro código de aplicación,

y eventualmente, queremos CodePipeline

para reconocer cuando agregamos una nueva versión de nuestro código

a este cubo S3

y lanzar automáticamente la canalización

cuando añadimos una nueva versión.

Y Crear Cubo.

Así que esos son nuestros 2 cubos creados.

Y ahora estamos listos para cargar nuestra plantilla de CloudFormation

a nuestro balde.

Así que seleccione su grupo de plantillas de CloudFormation ,

Cargar, agregar archivos,

y debe seleccionar su plantilla de CloudFormation

y Subir.

Una vez que se haya cargado, simplemente cierre esa vista.

Hagamos clic en nuestra plantilla y abrámosla.

Y aquí está nuestra plantilla de CloudFormation .

Y básicamente solo crea una instancia EC2

con el perfil de instancia correcto para CodeDeploy

y también crea un rol de servicio para CodeDeploy .

Así que básicamente está configurando todos los permisos.

que necesitamos

Así que esto es simplemente automatizar todo para nosotros.

Ahora, para ejecutar nuestro comando CloudFormation ,

necesitamos tomar la URL del objeto

de nuestra plantilla CloudFormation y la encontrará aquí.

Así que copia eso.

Aquí está nuestro comando CloudFormation

y necesitamos reemplazar esta URL

con la URL correcta de nuestra plantilla.

Así que solo pega eso aquí.

Y así es como debería verse.

También debe asegurarse de que tiene la información correcta

del parámetro para su par de claves.

Y el mío se llamaba nvkp .

Estos parámetros solo permiten SSH desde cualquier dirección IP

y también está configurando algunas etiquetas.

Entonces, una vez que haya hecho eso, copie todo el comando,

volver a la consola de AWS,

y podemos ejecutar este comando usando CloudShell .

Así que seleccione CloudShell en la parte superior de su pantalla.

Y CloudShell es solo un shell basado en navegador.

Le da acceso de línea de comandos a sus recursos de AWS.

Así que podemos ejecutar el comando CloudFormation desde aquí.

Y es una forma realmente rápida de ejecutar comandos de AWS

sin tener que configurar la CLI en su máquina local.

Una vez que su Cloud Shell esté funcionando,

puede pulsar Ctrl+V .

Te está pidiendo que verifiques.

Así que simplemente verifique el comando, péguelo y presione Enter.

Y eso va a lanzar nuestra plantilla de CloudFormation .

CloudFormation tarda unos minutos

para completar

Y hay un comando aquí que te permite

para verificar el estado.

Así que solo voy a copiar eso,

pega eso en mi CloudShell ,

y podemos ver que el trabajo de CloudFormation todavía está en progreso.

Así que voy a dejar eso por unos minutos.

y normalmente tarda unos 5 minutos en completarse.

Así que ahora es un buen momento para tomar una taza de té.

Y en unos 5 minutos,

debemos ser buenos para continuar.

Después de unos minutos, podemos ejecutar ese comando nuevamente,

simplemente usando la flecha hacia arriba y presione Entrar.

Y si todo está completo, debería ver Create Complete.

Así que una vez que todo esté terminado,

podemos entrar e iniciar sesión en nuestra instancia EC2

y solo asegúrate

que el agente de CodeDeploy se instaló correctamente.

Voy a dirigirme a Servicios, EC2,

acceda al panel de EC2.

Aquí está nuestra instancia.

Seleccione su instancia a la izquierda y presione Conectar.

Y usaremos EC2 Instance Connect

para conectarnos a nuestra instancia.

Así que solo presiona Conectar,

y luego escriba sudo service codedeploy -agent status,

y presione Entrar.

Y ahí vamos.

El agente de CodeDeploy se instaló correctamente

en nuestra instancia EC2.

Así que eso es todo para la primera parte de esta demostración.

Y si estás listo para continuar con la segunda parte,

Te veré en la próxima conferencia.

Gracias.