[ENGLISH]

Hello, Cloud Gurus,

and welcome to this lecture,

where we're going to take a closer look at the AppSpec file.

And this is a really important file used by CodeDeploy,

and it's basically a configuration file

which defines the parameters which are going to be used

during a CodeDeploy deployment.

And with deployments to EC2 and on-premises systems,

the AppSpec file can be written in YAML only.

However, with a Lambda-based deployment,

either YAML or JSON are supported.

And actually, the file structure depends on

whether you're deploying to Lambda or EC2.

There are some subtle differences.

But we're going to concentrate on an appspec.yml

for an EC2 basic deployment.

So let's take a look at the appspec.yml file structure.

And there are four significant sections of the file.

So first of all, we have version,

which is actually reserved for future use,

and currently, the only allowed value is 0.0.

So that's pretty simple.

The next one is OS,

and that relates to the operating system,

the type of operating system that you are deploying to.

So it's going to be either Linux or Windows.

Then we have files,

which relates to configuration files or packages,

which are going to be used during the deployment.

So we use this section to define the location

of any application files which need to be copied

and where they should be copied to during the deployment.

And then finally, we have hooks,

and hooks are often called lifecycle event hooks.

So what does that mean?

Well, it's basically, it's scripts which need to run

at set points during the deployment lifecycle.

And hooks actually have a very specific run order,

which we're going to learn a little bit more about later,

'cause they're actually really important for the exam.

And I've got some examples of different scripts

that might be used during an EC2 deployment.

So you might want to provide a script

which unzips your application files prior to the deployment,

you might want to run some functional tests

on a newly deployed application,

and you might also want to use scripts

for dealing with any registration

or deregistration of instances with your load balancer.

And for all of these scripts,

you're going to want to be in control

of exactly when CodeDeploy runs them.

So you're going to want to run them

at specific points in your deployment lifecycle,

and it's the lifecycle event hooks

which enable you to do that.

So let's take a look at an example appspec.yml file.

So first of all, we've got the version, which is 0.0.

We've then got the operating system,

and in this case, we're deploying to a Linux system.

We then have the file section,

which specifies any files that you want to copy

during the deployment.

So it specifies the source and the destination.

And then lastly, we've got our hooks section.

So this is the lifecycle hooks.

And this is where we specify the scripts

that we want to run during the deployment.

And when I mentioned to you that the hooks run

in a very specific order,

so this is defined using these parameters here.

And this is not an exhaustive list

of all the hooks that are available.

We actually have a lecture coming up

dedicated to the lifecycle hooks

and the order that they run in,

so we'll go through each of those in turn later on.

So moving on, where does this appspec.yml live?

Well, a typical setup is that you create a folder

or directory containing all the files needed

for the revision or the new version of your application,

so everything that you're going to need

during the deployment.

And in the root of the folder is the appspec.yml file,

and then everything else is organized into its own folder.

So we have our scripts folder,

which contains all the scripts that you want to run,

the config folder, which has any configuration files,

and then the source folder,

which contains any application source code.

But the most important thing to remember

is that the appspec.yml must be placed

in the root directory of your revision.

Otherwise, the deployment is going to fail.

And if we take another look

at our example appspec.yml file,

we can see that we need to reference the correct folders

in our files and hooks section.

So we're referencing the relative path

to our scripts and files

so that CodeDeploy knows exactly where to find everything.

So let's take a look at some of our exam tips

for the CodeDeploy AppSpec file.

Just remember, it's a configuration file,

and it's used to define the parameters

which are going to be used

by CodeDeploy during your deployment.

So for example, operating system version,

any files that are going to be used

or copied during the deployment,

and the lifecycle hooks,

which define scripts that you want to run

during the deployment and when to run them.

Your appspec.yml always lives

in the root of your revision directory,

and then everything else is organized into folders,

usually a folder containing scripts, config files,

and source code.

And then finally,

the lifecycle event hooks have a very specific run order,

and they're used to define when certain scripts will run

during the lifecycle of your deployment.

And we're going to take a look at lifecycle event hooks

in the next lecture.

So that is the end of this lecture.

And if you have any questions, please let me know.

If not, feel free to move on to 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 al archivo AppSpec .

Y este es un archivo muy importante utilizado por CodeDeploy ,

y es básicamente un archivo de configuración

que define los parámetros que se van a utilizar

durante una implementación de CodeDeploy .

Y con implementaciones en EC2 y sistemas locales,

el El archivo AppSpec solo se puede escribir en YAML.

Sin embargo, con una implementación basada en Lambda,

se admiten YAML o JSON.

Y en realidad, la estructura del archivo depende de

ya sea que esté implementando en Lambda o EC2.

Hay algunas diferencias sutiles.

Pero vamos a concentrarnos en un appspec.yml

para una implementación básica de EC2.

Así que echemos un vistazo a la estructura del archivo appspec.yml .

Y hay cuatro secciones significativas del archivo.

En primer lugar, tenemos la versión,

que en realidad está reservado para uso futuro,

y actualmente, el único valor permitido es 0.0.

Eso es bastante simple.

El siguiente es el sistema operativo,

y que se relaciona con el sistema operativo,

el tipo de sistema operativo que está implementando.

Así que va a ser Linux o Windows.

Entonces tenemos archivos,

que se relaciona con archivos de configuración o paquetes,

que se van a utilizar durante el despliegue.

Así que usamos esta sección para definir la ubicación

de cualquier archivo de aplicación que necesite ser copiado

y dónde deben copiarse durante la implementación.

Y finalmente, tenemos ganchos,

y los ganchos a menudo se denominan ganchos de eventos del ciclo de vida.

¿Entonces que significa eso?

Bueno, básicamente son scripts que deben ejecutarse

en puntos establecidos durante el ciclo de vida de la implementación.

Y los ganchos en realidad tienen un orden de ejecución muy específico,

del que vamos a aprender un poco más más adelante,

porque en realidad son muy importantes para el examen.

Y tengo algunos ejemplos de diferentes guiones

que podría usarse durante una implementación de EC2.

Por lo tanto, es posible que desee proporcionar un script

que descomprime los archivos de su aplicación antes de la implementación,

desee ejecutar algunas pruebas funcionales

en una aplicación recién implementada,

y es posible que también desee utilizar secuencias de comandos

para tramitar cualquier registro

o cancelar el registro de instancias con su balanceador de carga.

Y para todos estos guiones,

vas a querer tener el control

de cuándo CodeDeploy los ejecuta exactamente .

Así que vas a querer ejecutarlos

en puntos específicos de su ciclo de vida de implementación,

y son los ganchos de eventos del ciclo de vida

que le permiten hacer eso.

Así que echemos un vistazo a un archivo appspec.yml de ejemplo .

En primer lugar, tenemos la versión, que es 0.0.

Entonces tenemos el sistema operativo,

y en este caso, estamos implementando en un sistema Linux.

Luego tenemos la sección de archivos,

que especifica los archivos que desea copiar

durante el despliegue.

Por lo tanto, especifica la fuente y el destino.

Y por último, tenemos nuestra sección de ganchos.

Así que estos son los ganchos del ciclo de vida.

Y aquí es donde especificamos los scripts.

que queremos ejecutar durante la implementación.

Y cuando te mencioné que corren los anzuelos

en un orden muy específico,

así que esto se define usando estos parámetros aquí.

Y esta no es una lista exhaustiva

de todos los anzuelos que hay disponibles.

De hecho, tenemos una conferencia próximamente.

dedicado a los ganchos del ciclo de vida

y el orden en que corren,

así que revisaremos cada uno de ellos más adelante.

Entonces, ¿dónde vive este appspec.yml ?

Bueno, una configuración típica es que creas una carpeta

o directorio que contiene todos los archivos necesarios

para la revisión o la nueva versión de su aplicación,

así que todo lo que vas a necesitar

durante el despliegue.

Y en la raíz de la carpeta está el archivo appspec.yml ,

y luego todo lo demás se organiza en su propia carpeta.

Así que tenemos nuestra carpeta de scripts,

que contiene todos los scripts que desea ejecutar,

el carpeta de configuración , que tiene archivos de configuración,

y luego la carpeta de origen,

que contiene cualquier código fuente de la aplicación.

Pero lo más importante para recordar

es que se debe colocar el appspec.yml

en el directorio raíz de su revisión.

De lo contrario, la implementación fallará.

Y si echamos otro vistazo

en nuestro archivo appspec.yml de ejemplo,

podemos ver que necesitamos hacer referencia a las carpetas correctas

en nuestra sección de archivos y ganchos.

Así que estamos haciendo referencia a la ruta relativa

a nuestros scripts y archivos

para que CodeDeploy sepa exactamente dónde encontrar todo.

Así que echemos un vistazo a algunos de nuestros consejos para el examen.

para CodeDeploy \_ archivo AppSpec .

Solo recuerda, es un archivo de configuración,

y se utiliza para definir los parámetros

que se van a utilizar

por CodeDeploy durante su implementación.

Por ejemplo, la versión del sistema operativo,

cualquier archivo que se vaya a utilizar

o copiado durante la implementación,

y los ganchos del ciclo de vida,

que definen los scripts que desea ejecutar

durante el despliegue y cuándo ejecutarlos.

Tu appspec.yml siempre vive

en la raíz de su directorio de revisión,

y luego todo lo demás se organiza en carpetas,

generalmente una carpeta que contiene scripts, archivos de configuración ,

y código fuente.

Y finalmente,

los ganchos de eventos del ciclo de vida tienen un orden de ejecución muy específico,

y se usan para definir cuándo se ejecutarán ciertos scripts

durante el ciclo de vida de su implementación.

Y vamos a echar un vistazo a los ganchos de eventos del ciclo de vida

en la próxima conferencia.

Así que ese es el final de esta conferencia.

Y si tiene alguna pregunta, por favor hágamelo saber.

Si no, siéntase libre de pasar a la siguiente lección.

Gracias.