[ENGLISH]

Hello Cloud Gurus, and welcome to this lecture

where we're going to discuss CodeDeploy lifecycle event

Hooks and as you know, the lifecycle event hooks define

scripts that we want to run and the order that we want to

run them in during our deployment.

And for the exam they really focus on understanding

lifecycle event hooks for an in-place deployment.

So here's our example again,

and in the hooks section of the file

we've got a few different scripts running at

different stages during the deployment,

and of course, the lifecycle event hooks,

they're run in a specific order and

that is know as the run order.

So if you see anything in the exam referring to

the run order of lifecycle event hooks,

then this is what they're talking about.

And we only have a few included here in this example,

but there are actually lots of lifecycle event hooks

that you can include and you just include the ones

that you need for your own specific deployment.

So whatever makes sense for your own application.

So these are just a few examples,

but I'm going to show you an exhaustive list

of all the lifecycle event hooks

which apply to an in-place deployment.

And it's actually quite a lot to take in and remember,

so when I think about lifecycle event hooks,

I try and think of them as three distinct phases,

and that's just a way that I try and remember it all.

So hopefully that will make sense to you as well.

And first of all, in my phase one,

this is all to do with de-registering

our instances from a Load Balancer.

We then have phase two which is the

real nuts and bolts of the application deployment,

so all the activities that need to happen

in order to deploy the application itself.

And then my third phase is re-registering

our instances with the Load Balancer,

so once everything has been installed

and it's all up and running again,

then we re-register the instances with the Load Balancer.

So let's go in and look at these

in a little bit more detail.

So the first lifecycle event hook

is called BeforeBlockTraffic,

and this covers all the tasks that you want to run

on instances before they are de-registered

from the Load Balancer.

So if you have any scripts that you want to run

before your instances are de-registered

from the Load Balancer,

then this is the lifecycle event hook

that you're going to use.

Then next one is called BlockTraffic,

and this is where we actually de-register

the instances from a Load Balancer,

so any scripts relating to that,

you can include in BlockTraffic.

After BlockTraffic is any tasks that you want

to run on instances after they've

been de-registered from a Load Balancer,

and remember, this is all quite flexible,

again you just use the lifecycle event hooks

that make sense for your own application.

So now the next few stages are all about

the nuts and bolts of actually deploying

the application to your instances.

So the next lifecycle event hook that

we're going to look at is ApplicationStop,

and this is all about gracefully stopping the application,

so what scripts would you need to run in order to

make sure your application shuts down gracefully?

And of course with an in-place deployment

you may well be deploying a new version

of an application which is already installed in production,

so it makes sense that you're going to need to

shut down the application before you do anything else.

So the next step is DownloadBundle,

and this is where the CodeDeploy agent

copies the application revision files

or the new version of your application.

So it copies all the files to a temporary location.

The next event is BeforeInstall,

so this is any pre-installation scripts

that you want to run, for example,

you might want to back up the current version

of your application, you might want to back up

some configuration files, et cetera,

you might want to decrypt files from

the DownloadBundle that need decrypting.

The next step is to actually install the application itself.

And this often involves copying the

application files to their final location.

The next step is AfterInstall,

and this is any post installation scripts

that you might want to run, so for example,

you might want to update a configuration file,

or you might want to change file permissions as well.

So any scripts that you would want to run

after the installation has been done.

We then have ApplicationStart,

so at this point we start up any services

which was stopped during ApplicationStop,

and you might have a script or set of commands

that you want to run for Application Start,

and we then have ValidateService,

and this covers off any test that we want to run

to validate that the application is working as expected.

So those are all the lifecycle event hooks

which we're later actually installing your application,

and then the last bit is going to be all about

re-registering with the LoadBalancer.

So we have BeforeAllowTraffic,

which is any tasks that you want to run on your instances

before they get registered with the Load Balancer.

AllowTraffic, so this is when we actually go ahead

and register our instances with the Load Balancer,

and then AfterAllowTraffic,

which is any tasks that you want to run on instances

after they've been registered with the Load Balancer.

So that is all of the lifecycle event hooks

for an in-place deployment.

Now in terms of our exam tips,

for the exam you'll need to understand

the logical flow at a high level.

Just remember, if you see run order

in relation to lifecycle event hooks,

then this is what they're talking about,

they're talking about the specific order

that everything runs in.

And for an in-place deployment,

in my mind it's broadly three phases,

so you've got a de-registering phase,

so you're de-registering with the Load Balancer,

then the installation itself and

everything involved in that,

and then we've got the re-registering with a Load Balancer.

And finally, for the exam, I'd say the most important thing

is to try and get straight in your mind

the order that all of these things happen.

And if you think about it, it's all pretty logical,

and it's only the same steps that you would do

if you were installing a new version

of an application manually.

So if you do see a question in the exam

and you're not sure about how to answer it,

just have a think about the steps that you might take

if you were performing a manual application update.

What would you do, you know,

what would the sequence of steps look like?

So that is the end of this lecture.

If you have any questions, please let me know.

If not, feel free to move on to the next lecture, thank you.

[SPANISH]

Hola Cloud Gurus, y bienvenidos a esta conferencia.

donde vamos a discutir el evento del ciclo de vida de CodeDeploy

Los ganchos y, como saben, los ganchos de eventos del ciclo de vida definen

scripts que queremos ejecutar y el orden que queremos

ejecútelos durante nuestra implementación.

Y para el examen realmente se enfocan en entender

de ciclo de vida para una implementación en el lugar.

Así que aquí está nuestro ejemplo de nuevo,

y en la sección de ganchos del archivo

tenemos algunos scripts diferentes ejecutándose en

diferentes etapas durante el despliegue,

y , por supuesto, los ganchos de eventos del ciclo de vida,

se ejecutan en un orden específico y

eso se conoce como el orden de ejecución.

Entonces, si ve algo en el examen que se refiera a

el orden de ejecución de los ganchos de eventos del ciclo de vida,

entonces esto es de lo que están hablando.

Y solo tenemos algunos incluidos aquí en este ejemplo,

pero en realidad hay muchos ganchos de eventos de ciclo de vida

que puedes incluir y solo incluyes los

que necesita para su propia implementación específica.

Entonces, lo que tenga sentido para su propia aplicación.

Así que estos son solo algunos ejemplos,

pero les voy a mostrar una lista exhaustiva

de todos los enlaces de eventos del ciclo de vida

que se aplican a una implementación en el lugar.

Y en realidad es mucho para asimilar y recordar,

así que cuando pienso en los ganchos de eventos del ciclo de vida,

Trato de pensar en ellos como tres fases distintas,

y esa es solo una forma en que trato de recordarlo todo.

Así que espero que eso también tenga sentido para ti.

Y antes que nada, en mi fase uno,

esto tiene que ver con la cancelación del registro

nuestras instancias desde un Load Balancer.

Entonces tenemos la fase dos que es la

tuercas y tornillos reales de la implementación de la aplicación,

así que todas las actividades que deben suceder

para desplegar la propia aplicación.

Y luego mi tercera fase es volver a registrarme.

nuestras instancias con el Load Balancer,

así que una vez que todo ha sido instalado

y todo está funcionando de nuevo,

luego volvemos a registrar las instancias con el Load Balancer.

Así que entremos y miremos estos

con un poco más de detalle.

Entonces, el primer gancho de evento del ciclo de vida

se llama BeforeBlockTraffic ,

y esto cubre todas las tareas que desea ejecutar

en las instancias antes de que se den de baja

del equilibrador de carga.

Entonces, si tiene algún script que desee ejecutar

antes de que se anule el registro de sus instancias

del equilibrador de carga,

entonces este es el gancho del evento del ciclo de vida

que vas a usar.

Luego, el siguiente se llama BlockTraffic ,

y aquí es donde realmente cancelamos el registro

las instancias de un Load Balancer,

así que cualquier guión relacionado con eso,

puede incluir en BlockTraffic .

Después de BlockTraffic hay cualquier tarea que desee

para ejecutarse en instancias después de que hayan

se ha dado de baja de un Load Balancer,

y recuerda, todo esto es bastante flexible,

de nuevo, solo usa los ganchos de eventos del ciclo de vida

que tengan sentido para su propia aplicación.

Así que ahora las próximas etapas tienen que ver con

los aspectos prácticos de la implementación real

la aplicación a sus instancias.

Entonces, el próximo evento del ciclo de vida engancha ese

vamos a ver es ApplicationStop ,

y esto se trata de detener con gracia la aplicación,

Entonces, ¿ qué scripts necesitaría ejecutar para

Asegúrese de que su aplicación se cierra correctamente?

Y, por supuesto, con una implementación en el lugar

es posible que esté implementando una nueva versión

de una aplicación que ya está instalada en producción,

por lo que tiene sentido que usted va a necesitar

cierre la aplicación antes de hacer cualquier otra cosa.

Así que el siguiente paso es DownloadBundle ,

y aquí es donde el agente de CodeDeploy

copia los archivos de revisión de la aplicación

o la nueva versión de su aplicación.

Entonces copia todos los archivos a una ubicación temporal.

El siguiente evento es BeforeInstall ,

así que este es cualquier script de preinstalación

que desea ejecutar, por ejemplo,

desee hacer una copia de seguridad de la versión actual

de su aplicación, es posible que desee hacer una copia de seguridad

algunos archivos de configuración, etcétera,

es posible que desee descifrar archivos de

el DownloadBundle que necesita descifrado.

El siguiente paso es instalar la aplicación en sí.

Y esto a menudo implica copiar el

de la aplicación a su ubicación final.

El siguiente paso es AfterInstall ,

y este es cualquier script posterior a la instalación

que es posible que desee ejecutar, por ejemplo,

es posible que desee actualizar un archivo de configuración,

o es posible que también desee cambiar los permisos de los archivos.

Entonces, cualquier script que desee ejecutar

después de que se haya realizado la instalación.

Entonces tenemos ApplicationStart ,

así que en este punto ponemos en marcha cualquier servicio

que se detuvo durante ApplicationStop ,

y es posible que tenga un script o un conjunto de comandos

que desea ejecutar para Inicio de la aplicación,

y luego tenemos ValidateService ,

y esto cubre cualquier prueba que queramos ejecutar

para validar que la aplicación funciona como se esperaba.

Esos son todos los ganchos de eventos del ciclo de vida.

que más tarde instalaremos su aplicación,

y luego lo último será todo sobre

volver a registrarse con LoadBalancer .

Entonces tenemos BeforeAllowTraffic ,

que es cualquier tarea que desee ejecutar en sus instancias

antes de que se registren con Load Balancer.

AllowTraffic , entonces aquí es cuando realmente seguimos adelante

y registrar nuestras instancias con el Load Balancer,

y luego AfterAllowTraffic ,

que es cualquier tarea que desee ejecutar en instancias

después de que se hayan registrado con Load Balancer.

Esos son todos los ganchos de eventos del ciclo de vida.

para una implementación en el lugar.

Ahora, en términos de nuestros consejos para el examen,

para el examen necesitarás entender

el flujo lógico a un alto nivel.

Solo recuerda, si ves orden de ejecución

en relación con los ganchos de eventos del ciclo de vida,

entonces esto es de lo que están hablando,

están hablando del orden específico

que todo corre.

Y para una implementación en el lugar,

en mi mente son básicamente tres fases,

así que tienes una fase de cancelación de registro,

por lo que está cancelando el registro con Load Balancer,

luego la instalación en sí y

todo lo que implica eso,

y luego tenemos el nuevo registro con un Load Balancer.

Y por último, para el examen, diría lo más importante

es tratar de aclarar tu mente

el orden en que suceden todas estas cosas.

Y si lo piensas bien, todo es bastante lógico,

y son solo los mismos pasos que harías

si estuvieras instalando una nueva versión

de una aplicación manualmente.

Así que si ves una pregunta en el examen

y no estás seguro de cómo responderla,

solo piensa en los pasos que podrías tomar

si estuviera realizando una actualización manual de la aplicación.

¿Qué harías, ya sabes,

Cómo sería la secuencia de pasos?

Así que ese es el final de esta conferencia.

Si tiene alguna pregunta, por favor hágamelo saber.

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