[ENGLISH]

Hello,

Cloud Gurus and welcome to this lecture where we're going to go into a little

bit more detail about CodeDeploy. So of course,

CodeDeploy is an automated deployment service.

It works with EC2 instances on premises systems and also Lambda

as well and because the deployments are automated,

it allows you to really quickly release new features,

avoid downtime during deployments,

and avoid the risks associated with manual processes

and with CodeDeploy,

there are two different deployment approaches that you're going to need to know

about and the first one is called an in place deployment and

this is where the application is stopped on each instance and a new release is

installed and this is also known as a rolling update and then

the second one that you're going to need to know about is called a blue-green

deployment and with blue-green deployments,

new instances are provisioned and the new release is installed on the brand

new instances and blue represents the active deployment whereas

green is the new release,

and it's really important to understand the differences between these two kinds

of deployment and to help you understand the difference,

we've got a couple of examples to go through.

So here is our application and it's running on a few EC2 instances

behind an elastic load balancer and we're currently running

version one of our application,

and we want to deploy a new version using an in place deployment

approach. So first of all,

CodeDeploy is going to stop the application on our first instance.

So the instance is going to be out of service during the deployment,

which means capacity within your environment is going to be reduced.

So in this example,

we've gone from having three application servers to only having two,

and you should configure your elastic load balancer to stop sending requests to

the instance while it is out of service and this is something that you can

configure within CodeDeploy as well.

So the next thing that happens is CodeDeploy installs the new version of your

application on the first instance, and this is known as a revision,

so they don't call it an application version within CodeDeploy.

They refer to it as a revision and once the new revision

has been deployed, then the instance comes back into service.

It gets reregistered with the load balancer and then CodeDeploy

continues to deploy to the next instance but what happens

if after you've deployed everything,

you change your mind and you want to roll back,

maybe you discover a problem with a new version,

which doesn't become apparent until you start running in production. Well,

unfortunately, with an in place deployment,

if you do change your mind and you need to roll back,

it's not going to be a quick fix and with an in-place deployment,

you're going to need to redeploy the previous version,

which of course can be time consuming because you're going to have to go through

the entire process all over again, just redeploying the old version.

So what about blue-green deployments?

And we've got a similar scenario here with three application servers and they're

all running version one, and they're sitting behind an elastic load balancer.

So blue represents the current version of our application and with a blue-green

deployment CodeDeploy provisions, new instances,

which are completely independent from the current production environment and in

the blue-green terminology,

these new instances represent the green environment.

So the revision is then deployed into the green environment.

The green instances are registered with the elastic load balancer and traffic is

routed away from the old environment and then finally,

when we're satisfied that everything is working as expected with our version two

environment, we can eventually terminate our old environment.

So what happens when we want to roll back from a blue-green deployment? Well,

with blue-green, it's really easy.

All you have to do is set the load balancer to direct the traffic back to our

original environment. So in our scenario,

we can simply reregister our blue environment with the load balancer

and de-register the green,

and you can easily switch between the old and new releases. Of course,

this is only possible if you haven't already terminated the old

environment.

So let's take a look at some of our exam tips for CodeDeploy deployments.

Well,

it's really important to know the difference between an in place deployment and

a blue-green deployment.

So remember with in place capacity is going to be reduced during the deployment.

Lambda is not supported, rolling back is fairly involved.

So it means that you're going to have to redeploy the original version.

And you might be wondering under which circumstances would I use an in place

deployment? Well, actually,

it's great when you're deploying for the very first time, because of course,

if it's the first time that you've deployed your application,

it's not possible to do a blue-green deployment.

So installing your application or deploying your application for the very first

time is a pretty good use case for using the in place deployment

approach. However, with a blue-green deployment,

there is of course no capacity reduction because we're deploying new

instances that are going to run alongside your existing environment

and the great thing with a blue-green deployment is that you can create the

instances and run the deployment ahead of time. Then on the day,

you'd like to cut over to the new version,

it's very easy to switch between the old and the new,

just a case of reconfiguring your load balancer,

but just keep in mind that you're going to be paying for two environments,

the old and the new until you're ready to terminate the old servers

and finally, if you do see a question in the exam,

which is asking you about the safest option for a production environment,

it's definitely the blue-green approach because you won't experience any

reduced capacity during the deployment process and it's very fast to roll

back.

If something goes wrong provided you haven't already terminated your original

environment. 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 en la que vamos a profundizar un poco

un poco más de detalle sobre CodeDeploy . Asi que por su puesto,

CodeDeploy es un servicio de implementación automatizado.

Funciona con instancias EC2 en sistemas locales y también con Lambda

también y porque las implementaciones están automatizadas,

le permite lanzar nuevas funciones muy rápidamente,

evitar el tiempo de inactividad durante las implementaciones,

y evitar los riesgos asociados a los procesos manuales

y con CodeDeploy ,

hay dos enfoques de implementación diferentes que necesitará saber

sobre y el primero se llama un despliegue en el lugar y

aquí es donde la aplicación se detiene en cada instancia y se lanza una nueva versión

instalado y esto también se conoce como una actualización continua y luego

el segundo que necesitará conocer se llama azul-verde

despliegue y con despliegues azul-verde,

se aprovisionan nuevas instancias y se instala la nueva versión en la marca

nuevas instancias y el azul representa la implementación activa mientras que

verde es el nuevo lanzamiento,

y es muy importante entender las diferencias entre estos dos tipos

de implementación y para ayudarlo a comprender la diferencia,

Tenemos un par de ejemplos para revisar.

Así que aquí está nuestra aplicación y se está ejecutando en algunas instancias de EC2

detrás de un balanceador de carga elástico y actualmente estamos ejecutando

versión uno de nuestra aplicación,

y queremos implementar una nueva versión usando una implementación en el lugar

acercamiento \_ Entonces, en primer lugar,

CodeDeploy detendrá la aplicación en nuestra primera instancia.

Entonces, la instancia estará fuera de servicio durante la implementación,

lo que significa que la capacidad dentro de su entorno se reducirá.

Así que en este ejemplo,

hemos pasado de tener tres servidores de aplicaciones a tener solo dos,

y debe configurar su balanceador de carga elástico para dejar de enviar solicitudes a

la instancia mientras está fuera de servicio y esto es algo que puede

configurar dentro de CodeDeploy también.

Entonces, lo siguiente que sucede es que CodeDeploy instala la nueva versión de su

solicitud en primera instancia, y esto se conoce como revisión,

por lo que no lo llaman una versión de la aplicación dentro de CodeDeploy .

Se refieren a ella como una revisión y una vez que la nueva revisión

se ha implementado, luego la instancia vuelve a estar en servicio.

Se vuelve a registrar con el balanceador de carga y luego CodeDeploy

continúa desplegándose a la siguiente instancia, pero lo que sucede

si después de haber implementado todo,

cambias de opinión y quieres retroceder,

tal vez descubras un problema con una nueva versión,

que no se vuelve aparente hasta que comienza a ejecutarse en producción. Bien,

desafortunadamente , con un despliegue en el lugar,

si cambia de opinión y necesita retroceder,

no va a ser una solución rápida y con una implementación en el lugar,

tendrá que volver a implementar la versión anterior,

lo que, por supuesto, puede llevar mucho tiempo porque tendrá que pasar por

el proceso de nuevo, simplemente volviendo a implementar la versión anterior.

Entonces, ¿qué pasa con las implementaciones azul-verde?

Y tenemos un escenario similar aquí con tres servidores de aplicaciones y están

todos ejecutan la versión uno, y están sentados detrás de un balanceador de carga elástico.

Entonces el azul representa la versión actual de nuestra aplicación y con un azul verdoso

despliegue Disposiciones de CodeDeploy , nuevas instancias,

que son completamente independientes del entorno de producción actual y en

la terminología azul-verde,

estas nuevas instancias representan el entorno verde.

Entonces, la revisión se implementa en el entorno verde.

Las instancias verdes se registran con el balanceador de carga elástico y el tráfico se

desviado del antiguo entorno y finalmente,

cuando estamos satisfechos de que todo funciona como se esperaba con nuestra versión dos

environment , eventualmente podemos terminar nuestro antiguo entorno.

Entonces, ¿qué sucede cuando queremos revertir una implementación azul-verde? Bien,

con azul-verde, es muy fácil.

Todo lo que tiene que hacer es configurar el balanceador de carga para dirigir el tráfico de regreso a nuestro

original . Así que en nuestro escenario,

podemos volver a registrar nuestro entorno azul con el balanceador de carga

y dar de baja el verde,

y puede cambiar fácilmente entre las versiones antiguas y nuevas. Por supuesto,

esto solo es posible si aún no ha terminado el antiguo

medio ambiente

Así que echemos un vistazo a algunos de nuestros consejos de examen para las implementaciones de CodeDeploy .

Bien,

es realmente importante saber la diferencia entre una implementación en el lugar y

un despliegue azul-verde.

Así que recuerde que la capacidad establecida se reducirá durante la implementación.

Lambda no es compatible, la reversión es bastante complicada.

Entonces significa que tendrá que volver a implementar la versión original.

Y es posible que se pregunte en qué circunstancias usaría un

despliegue ? Bueno en realidad,

es genial cuando se implementa por primera vez porque, por supuesto,

si es la primera vez que implementa su aplicación,

es posible hacer una implementación azul-verde.

Entonces, instalar su aplicación o implementar su aplicación por primera vez

el tiempo es un caso de uso bastante bueno para usar la implementación en el lugar

acercamiento \_ Sin embargo, con un despliegue azul-verde,

hay reducción de capacidad porque estamos implementando nuevos

instancias que se ejecutarán junto con su entorno existente

y lo mejor de una implementación azul-verde es que puede crear el

instancias y ejecutar la implementación antes de tiempo. Entonces en el día,

le gustaría pasar a la nueva versión,

es muy fácil cambiar entre lo antiguo y lo nuevo,

solo un caso de reconfigurar su balanceador de carga,

pero tenga en cuenta que pagará por dos entornos,

el antiguo y el nuevo hasta que esté listo para terminar los servidores antiguos

y finalmente, si ve una pregunta en el examen,

que le pregunta sobre la opción más segura para un entorno de producción,

es el enfoque azul-verde porque no experimentará ningún

reducida durante el proceso de implementación y es muy rápido para rodar

atrás \_

Si algo sale mal, siempre que no haya cancelado su contrato original

medio ambiente Así que ese es el final de esta conferencia.

Si tiene alguna pregunta, hágamelo saber si no,

Siéntete libre de pasar a la siguiente lección. Gracias.