[ENGLISH]

Hello, Cloud Gurus and welcome to this lecture,

which is going to cover RDS and Elastic Beanstalk.

Now Elastic Beanstalk supports two ways of integrating an RDS database

with your Elastic Beanstalk environment,

and you can either deploy your RDS instance inside

your Elastic Beanstalk.

Environment, or outside of it.

So what does that actually mean? Well, with option 1 where you launch.

RDS within the Elastic Beanstalk.

Environment, you launch the.

RDS instance from within the Elastic Beanstalk console.

So that means it's created.

Within your Elastic Beanstalk environment.

However, the issue.

With this is that if you terminate the environment,

then the database will also be terminated.

So this is a good option.

For development and test deployments.

Because you deploy everything.

In one go,

and it's all very simple with Elastic Beanstalk doing everything for you in

terms of security groups, et cetera.

However, it's not.

So good for production because it lacks flexibility,

and if you terminate the environment,

then your database is going to be gone as well,

and there's no option to just terminate the application servers

and keep the RDS database. So what is the alternative?

Well with option 2, you can launch the.

RDS instance outside of Elastic.

Beanstalk, so you don't.

Use Elastic Beanstalk to create your RDS instance.

Instead you use the RDS console, or the AWS CLI,

and this allows you to tear down your application environment

without effecting the database instance.

So you can terminate your elastic Beanstalk environment and your RDS database

will continue running independently.

And this is the preferred.

Approach for production systems,

because it just gives you that flexibility to tear down your application servers

without affecting your.

Database. Now with this option, because we're.

Creating the RDS instance outside our elastic Beanstalk environment,

there are a couple of things we'll need to take care of to allow our application

server to communicate with the database. So firstly,

we'll need to add an additional security group to our environments auto scaling

group to allow our EC2 instances, to communicate.

With our RDS.

Database on the relevant port, and in my example,

it's a mySQL or MySQL database communicating on port

3306.

So I'll need to set up my security group to allow that network communication.

And then the second.

Thing that you'll need to do is you'll need to provide connection string

information to your application servers,

and you can do that using Elastic Beanstalk environment

properties,

and the connection string information is going to look something.

Like this. So it's the Amazon.

Resource name for the RDS database and a database

password. And that is the bare minimum.

So you basically need to supply all the information that is required for a

database client to connect to your RDS instance,

and I'll show you where we add this in the AWS console.

Here's my Elastic Beanstalk environment, and I'll just head to configuration.

And then the category will be software. So hit edit, and scroll.

Down to the bottom and you will find environment properties.

And this is where you add the database connection information.

And these are properties which are going to be passed into the application as

environment properties.

And this is where we add.

Our RDS connection information. So for the exam,

just remember, there are two different approaches.

Supported. You can either.

Deploy your RDS database inside Elastic Beanstalk,

or outside of your Elastic Beanstalk environment.

And creating.

It inside your Elastic Beanstalk environment is a really quick and easy way to

add your database and get started. However,

when you terminate the Elastic Beanstalk environment,

then the database will also be terminated and that may or may not suit

your requirements. So for that reason,

creating the RDS database inside your Elastic Beanstalk environment

is probably only suitable for development and test environments.

And then.

With the second option,

deploying your RDS database outside of your Elastic Beanstalk environment.

With this approach,

you can tear down the application stack without impacting your database at

all. And with this approach,

there are a couple of additional configuration steps required.

So you'll need to configure the security group to enable communication between

your application servers and your RDS instance.

And you will also.

Need to supply connection information as environment properties to your

Elastic Beanstalk environment.

And typically the connection information would be the Amazon resource name of

your audience instance, and a reference to the database.

Password.

And creating the RDS instance outside of your Elastic Beanstalk environment

is the preferred approach for production environments just because.

It gives you the flexibility to tear down your application servers without

losing your database.

So that is the end of this lecture. If you have any questions,

please let me know. Otherwise.

I'll see you in the next lecture. Thank you.

[SPANISH]

Hola, Cloud Gurus y bienvenidos a esta conferencia.

que cubrirá RDS y Elastic Beanstalk.

Ahora Elastic Beanstalk admite dos formas de integrar una base de datos RDS

con su entorno de Elastic Beanstalk,

y puede implementar su instancia de RDS dentro

tus habichuelas mágicas elásticas.

Entorno, o fuera de él.

Entonces, ¿qué significa eso realmente? Pues con la opción 1 donde te lanzas.

RDS dentro de Elastic Beanstalk.

Medio ambiente, se inicia el.

Instancia de RDS desde la consola de Elastic Beanstalk.

Entonces eso significa que está creado.

Dentro de su entorno de Elastic Beanstalk.

Sin embargo, el problema.

Con esto es que si terminas el entorno,

entonces la base de datos también será terminada.

Así que esta es una buena opción.

Para implementaciones de desarrollo y prueba.

Porque despliegas todo.

En una ida,

y todo es muy simple con Elastic Beanstalk haciendo todo por ti en

términos de grupos de seguridad, etcétera.

Sin embargo, no lo es.

Tan bueno para la producción porque carece de flexibilidad,

y si terminas el entorno,

entonces su base de datos también desaparecerá,

y no hay opción para simplemente terminar los servidores de aplicaciones

y mantener la base de datos RDS. entonces cual es la alternativa?

Bueno, con la opción 2, puede iniciar el.

Instancia RDS fuera de Elastic.

Beanstalk, para que no lo hagas.

Utilice Elastic Beanstalk para crear su instancia de RDS.

En su lugar, utiliza la consola RDS o la CLI de AWS,

y esto le permite derribar su entorno de aplicación

sin afectar la instancia de la base de datos.

Para que pueda terminar su entorno elástico Beanstalk y su base de datos RDS

seguirá funcionando de forma independiente.

Y este es el preferido.

Enfoque para los sistemas de producción,

porque simplemente le da esa flexibilidad para derribar sus servidores de aplicaciones

sin afectar su .

Base de datos. Ahora con esta opción, porque estamos.

Creando la instancia RDS fuera de nuestro entorno elástico Beanstalk,

hay un par de cosas de las que debemos ocuparnos para permitir que nuestra aplicación

servidor para comunicarse con la base de datos. Así que en primer lugar,

tendremos que agregar un grupo de seguridad adicional al escalado automático de nuestros entornos

group para permitir que nuestras instancias EC2 se comuniquen.

Con nuestro RDS.

Base de datos en el puerto relevante, y en mi ejemplo,

es una base de datos mySQL o MySQL que se comunica en el puerto

3306.

Así que tendré que configurar mi grupo de seguridad para permitir esa comunicación de red.

Y luego el segundo.

Lo que deberá hacer es proporcionar una cadena de conexión

información a sus servidores de aplicaciones,

y puede hacerlo usando el entorno de Elastic Beanstalk

propiedades ,

y la información de la cadena de conexión se verá algo.

Como esto. Así que es el Amazonas.

Nombre de recurso para la base de datos RDS y una base de datos

contraseña \_ Y eso es lo mínimo.

Así que básicamente necesita proporcionar toda la información que se requiere para un

de base de datos para conectarse a su instancia de RDS,

y le mostraré dónde agregamos esto en la consola de AWS.

Este es mi entorno de Elastic Beanstalk y me dirigiré a la configuración.

Y luego la categoría será software. Así que presiona editar y desplázate.

Baja hasta el fondo y encontrarás las propiedades del entorno.

Y aquí es donde agrega la información de conexión de la base de datos.

Y estas son propiedades que se van a pasar a la aplicación como

del entorno .

Y aquí es donde añadimos.

Nuestra información de conexión RDS. Así que para el examen,

solo recuerda, hay dos enfoques diferentes.

Soportado. Tu también puedes.

Implemente su base de datos RDS dentro de Elastic Beanstalk,

o fuera de su entorno de Elastic Beanstalk.

Y creando.

Dentro de su entorno de Elastic Beanstalk es una manera realmente rápida y fácil de

agregue su base de datos y comience. Sin embargo,

cuando termina el entorno de Elastic Beanstalk,

entonces la base de datos también se cerrará y eso puede o no ser adecuado

sus requisitos Así que por esa razón,

crear la base de datos RDS dentro de su entorno de Elastic Beanstalk

sea adecuado para entornos de desarrollo y prueba.

Y luego.

Con la segunda opción,

implementar su base de datos RDS fuera de su entorno de Elastic Beanstalk.

Con este enfoque,

puede derribar la pila de aplicaciones sin afectar su base de datos en

todos \_ Y con este enfoque,

se requieren un par de pasos de configuración adicionales.

Por lo tanto, deberá configurar el grupo de seguridad para habilitar la comunicación entre

sus servidores de aplicaciones y su instancia RDS.

Y tú también lo harás.

Necesita proporcionar información de conexión como propiedades del entorno a su

Entorno de Beanstalk elástico.

Y, por lo general, la información de conexión sería el nombre del recurso de Amazon de

su instancia de audiencia y una referencia a la base de datos.

Contraseña.

Y crear la instancia de RDS fuera de su entorno de Elastic Beanstalk

es el enfoque preferido para los entornos de producción porque sí.

Le brinda la flexibilidad de desmantelar sus servidores de aplicaciones sin

perder su base de datos.

Así que ese es el final de esta conferencia. Si tienes alguna pregunta,

por favor hágamelo saber De lo contrario.

Te veré en la próxima conferencia. Gracias.