[ENGLISH]

Hello, Cloud Gurus, and welcome to this lecture,

which will cover some advanced Elastic Beanstalk topics

that you will need to know for the exam.

And we'll begin with customizing

your Elastic Beanstalk environment

and how we do that for Pre-Amazon Linux 2.

So Amazon Linux 1 environments

and Amazon Linux 2 as well.

And then finally, we'll cover my exam tips.

Now you can customize your Elastic Beanstalk environment

in a number of different ways.

And the configuration is going to be different

depending on whether you are running Amazon Linux 1

or Amazon Linux 2.

So for Pre-Amazon Linux 2 environments.

So for Amazon Linux 1,

we can use a configuration file

to define packages to install,

create Linux users and groups,

run shell commands, enable services

and configure load balancers as well.

And these are just a few examples.

These configuration files can be written

in either YAML or JSON format

and the files must have a .config extension

and be located inside a folder named .ebextensions

in the top-level directory

of your application source code bundle.

And here is an example.

This example configures an application health check URL

that can be used by a load balancer

which will make an HTTP request to the specified path

to check if the instances are healthy.

And once again, this is the way to customize

Elastic Beanstalk for Amazon Linux 1 only.

And once you've created your file,

you just save it into the folder named .ebextensions.

Now you might be thinking

that Amazon Linux 2 has been available for quite some time,

and I agree,

however, you may still see questions

relating to .ebextensions mentioned in the exam.

So it's definitely worth knowing what it's used for

because it is still supported for any Amazon Linux 1 systems

that you may have.

So what about Amazon Linux 2?

Well, according to the Elastic Beanstalk Developer Guide,

on Amazon Linux 2 platforms, they are recommending

that you use the Buildfile, Procfile, and platform hooks

to configure and run custom code

on your EC2 instances during provisioning.

But what does that actually mean?

Well, we can use a Buildfile

for commands that run for short time periods

and then exit upon task completion.

For example, running a simple shell script.

Create your Buildfile in the root directory

of your application source.

And the format looks like this.

So it's key-value pairs consisting of a process name

and a path to the script that you want to run.

And this script should be included

in your source code bundle.

Next, we have the Procfile,

and this is for long-running application processes.

For example, commands that you use to start

and run your application.

You create the file named Procfile

in your root directory of your application source.

And the format is the same as the Buildfile.

So it's key-value pairs consisting of a process name

and the command that you want to run.

And Elastic Beanstalk expects any processes

that are defined in the Procfile to run continuously.

And it's going to monitor

and restart any processes, which terminate.

And then the third way to customize Amazon Linux 2

is by using platform hooks.

And this is for custom scripts or executable files

that you would like Elastic Beanstalk

to run at a chosen stage of the EC2 provisioning process.

The scripts are stored in dedicated directories

in your application source code.

And we use the following directory structure.

So there's .platform/hooks/prebuild

and this is for files that you want Elastic Beanstalk

to run before it builds, sets up,

and configures the application and web server.

Then there's .platform/hooks/predeploy,

and this is for files that you want to run

after Elastic Beanstalk sets up

and configures the application and web server,

but before it deploys them to their final runtime location.

And then there is .platform/hooks/postdeploy,

and this is for files that run after Elastic Beanstalk

has completely deployed the application,

and this is the final deployment workflow step.

And all of these are optional,

but they are available to use

should you require them for your application.

So for the exam, just remember that you can customize

your Elastic Beanstalk environment

in a number of different ways.

For Amazon Linux 1, you can use the .ebextensions folder,

and this should be located in the top-level directory

of your application source code bundle.

Any files that you save there

should have a .config extension.

For example, myhealthcheckurl.config.

And once again, that is only for Amazon Linux 1,

but it may still come up in the exam.

For Amazon Linux 2, create a Buildfile

in the root directory of your application source

for commands that exit upon completion like shells scripts.

Create a Procfile for long running processes.

For example, custom commands to start up your application.

And use platform hooks for custom scripts or executables

that run at various stages

when your EC2 instances are being provisioned,

and use a folder structure like this

to organize and store your scripts.

And these are the ones to use for Amazon Linux 2.

So that's it for 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 algunos temas avanzados de Elastic Beanstalk

que necesitará saber para el examen.

Y comenzaremos con la personalización.

su entorno de Elastic Beanstalk

y cómo lo hacemos para Pre-Amazon Linux 2.

Así que los entornos de Amazon Linux 1

y Amazon Linux 2 también.

Y finalmente, cubriremos mis consejos para el examen.

Ahora puedes personalizar tu entorno de Elastic Beanstalk

en un número de diferentes maneras.

Y la configuración va a ser diferente.

dependiendo de si está ejecutando Amazon Linux 1

o Amazon Linux 2.

Entonces, para entornos Pre-Amazon Linux 2.

Así que para Amazon Linux 1,

podemos usar un archivo de configuración

para definir paquetes a instalar,

crear usuarios y grupos de Linux,

ejecutar comandos de shell, habilitar servicios

y configurar balanceadores de carga también.

Y estos son solo algunos ejemplos.

Estos archivos de configuración se pueden escribir

en formato YAML o JSON

y los archivos deben tener una extensión . extensión de configuración

y estar ubicado dentro de una carpeta llamada . ebextensiones

en el directorio de nivel superior

del paquete de código fuente de su aplicación.

Y aquí hay un ejemplo.

Este ejemplo configura una URL de verificación de estado de la aplicación

que puede ser utilizado por un balanceador de carga

que hará una solicitud HTTP a la ruta especificada

para comprobar si las instancias están en buen estado.

Y una vez más, esta es la forma de personalizar

Elastic Beanstalk solo para Amazon Linux 1.

Y una vez que haya creado su archivo,

guárdelo en la carpeta llamada . ebextensiones .

Ahora podrías estar pensando

que Amazon Linux 2 ha estado disponible durante bastante tiempo,

y estoy de acuerdo,

sin embargo , aún puede ver preguntas

relacionado con ebextensions mencionadas en el examen.

Así que definitivamente vale la pena saber para qué se usa.

porque todavía es compatible con cualquier sistema Amazon Linux 1

que puedas tener.

Entonces, ¿qué pasa con Amazon Linux 2?

Bueno, de acuerdo con la Guía para desarrolladores de Elastic Beanstalk,

en plataformas Amazon Linux 2, están recomendando

que utiliza los ganchos Buildfile , Procfile y plataforma

para configurar y ejecutar código personalizado

en sus instancias EC2 durante el aprovisionamiento.

Pero, ¿qué significa eso realmente?

Bueno, podemos usar un Buildfile

para comandos que se ejecutan por períodos cortos de tiempo

y luego salir al completar la tarea.

Por ejemplo, ejecutar un script de shell simple.

Cree su archivo de compilación en el directorio raíz

de la fuente de su aplicación.

Y el formato se ve así.

Entonces son pares clave-valor que consisten en un nombre de proceso

y una ruta al script que desea ejecutar.

Y este script debe ser incluido

en su paquete de código fuente.

A continuación, tenemos el Procfile ,

y esto es para procesos de aplicación de larga duración.

Por ejemplo, los comandos que utiliza para iniciar

y ejecute su aplicación.

Creas el archivo llamado Procfile

en su directorio raíz de la fuente de su aplicación.

Y el formato es el mismo que el Buildfile .

Entonces son pares clave-valor que consisten en un nombre de proceso

y el comando que desea ejecutar.

Y Elastic Beanstalk espera cualquier proceso

que se definen en el Procfile para ejecutarse continuamente.

Y va a monitorear

y reinicie cualquier proceso que finalice.

Y luego la tercera forma de personalizar Amazon Linux 2

es mediante el uso de ganchos de plataforma.

Y esto es para scripts personalizados o archivos ejecutables.

que te gustaría Elastic Beanstalk

para ejecutarse en una etapa elegida del proceso de aprovisionamiento de EC2.

Los scripts se almacenan en directorios dedicados.

en el código fuente de su aplicación.

Y usamos la siguiente estructura de directorios.

Entonces hay .platform/hooks/prebuild

y esto es para los archivos que desea Elastic Beanstalk

para ejecutarse antes de construir, configurar,

y configura la aplicación y el servidor web.

Luego está .platform/hooks/ predeploy ,

y esto es para los archivos que desea ejecutar

después de que se configura Elastic Beanstalk

y configura la aplicación y el servidor web,

pero antes de implementarlos en su ubicación de tiempo de ejecución final.

Y luego está .platform/hooks/ postdeploy ,

y esto es para archivos que se ejecutan después de Elastic Beanstalk

ha desplegado completamente la aplicación,

y este es el paso final del flujo de trabajo de implementación.

Y todos estos son opcionales,

pero están disponibles para usar

en caso de que los requiera para su aplicación.

Entonces, para el examen, recuerde que puede personalizar

su entorno de Elastic Beanstalk

en un número de diferentes maneras.

Para Amazon Linux 1, puede usar . carpeta de extensiones eb ,

y esto debe estar ubicado en el directorio de nivel superior

del paquete de código fuente de su aplicación.

Cualquier archivo que guardes allí

debería tener un . extensión de configuración .

Por ejemplo, myhealthcheckurl.config .

Y una vez más, eso es solo para Amazon Linux 1,

pero aún puede aparecer en el examen.

Para Amazon Linux 2, cree un archivo de compilación

en el directorio raíz de la fuente de su aplicación

para comandos que salen al completarse como scripts de shells.

Cree un Procfile para procesos de ejecución prolongada.

Por ejemplo, comandos personalizados para iniciar su aplicación.

Y use ganchos de plataforma para scripts personalizados o ejecutables

que se ejecutan en varias etapas

cuando se aprovisionan sus instancias EC2,

y usa una estructura de carpetas como esta

para organizar y almacenar sus scripts.

Y estos son los que se usan para Amazon Linux 2.

Así que eso es todo por esta conferencia.

Si tiene alguna pregunta, por favor hágamelo saber.

De lo contrario, te veré en la próxima lección.

Gracias.