[ENGLISH]

Hello, Cloud Gurus and welcome to this lecture.

And this lecture is going to introduce CloudFormation.

Now CloudFormation,

it's a service that allows you to manage,

configure and provision your AWS infrastructure as code

and resources or the resources

that you're going to provision

are defined using a cloud formation template

and that is the code element.

And cloud formation,

it takes that template,

it interprets it

and makes the appropriate API calls

in order to create the resources

that you have defined in the template.

And the template supports

either YAML or JSON.

So you can use either YAML or JSON

whichever you're comfortable with

to create your cloud formation templates.

So what are some of the benefits of CloudFormation?

Well firstly it allows you to provision

your infrastructure in a really consistent way.

So it's the same every single time.

There's fewer mistakes

because there's less opportunity for human error.

It takes less time and effort than it would

if you were to configure things manually.

You can also Version Control

and peer review your templates

and it's free to use.

However, you will be charged

for any resources that you create.

It can be used to manage updates

to existing CloudFormation platforms

and it can also handle dependencies

between different resources

to ensure that resources are created in the correct order.

And it can also be used to really quickly

and easily roll back and delete the entire stack

that you've created within cloud formation as well.

So it's really easy to clear up environments

that were temporary or that just being used

to test something um you can really quickly have

CloudFormation roll back

and delete the whole lot for you as well.

So let's look at the CloudFormation process

in a little bit more detail.

Once again,

you can use either YAML or JSON

to describe the end state of the infrastructure

you are either provisioning or changing.

Once you've created the template,

you upload it to CloudFormation using S3.

So within the CloudFormation console,

you actually upload it

and it ends up in an S3 bucket

and then CloudFormation accesses

the template within the S3 bucket.

It reads the template

and then it makes the appropriate API calls

on your behalf to create the resources for you.

And the resulting resources are referred to as a "stack"

and it's called a CloudFormation Stack.

So let's have a look at

an example of CloudFormation template.

And most of these options or Parameters are optional.

It's only really the Resources section

that we'll see in the next slide that is mandatory.

So everything else

that you can see on this page is optional.

So firstly first of all,

we have the TemplateFormatVersion

and the only supported version is 2010-09-09.

And that just refers to

the version of the Template format.

So the acceptable or the supported Template format

for the CloudFornmation template.

Moving on,

we have the Description and of course this is optional.

So you can just add in a Description of your choice,

something meaningful to you.

So in this case,

it's uh just describing this as a

Template to create an EC2 instance.

Next is Metadata and that is simply data about data.

Once again it's completely optional.

So you can just put in a custom field.

and for this one we're just providing some data

about the specific instances that this

CloudFormation Template is going to provision.

So it just gives a Description of Web Server Instance.

Moving down to Parameters.

So these are input values that

you actually input to CloudFormation.

When you actually go ahead and

launch a stack using this Template.

So this is information that you need

to provide when you go ahead and launch this stack.

And in this case we're defining an Environment type

and the allowed values are either prod or test.

So we need to provide that when we actually

go ahead and launch a stack using this Template.

And then the next Parameter is Conditions

and we can use Conditions once again

it's completely user defined.

And we can actually use this

to test a Condition and take action

based on the outcome of testing that Condition.

So in this case,

we are evaluating the um the Parameter EnvType,

which was declared in the previous section

under the Parameters section.

So we're evaluating that

and we're saying if it equals to prod

then we should create prod resources only.

So you can use it to tell CloudFormation

to make decisions based on Parameters

that you have input at launch time.

Moving on to Mappings and we can use Mappings

to set our own user defined values.

So for example,

we could set the value of an AMI based on a Region,

so we can completely define this ourselves.

So for example for eu-west-1

that is the AMI we're going to use

and you could define a different AMI

for every different Region.

You could define a different AMI

for a different environment,

et cetera, et cetera.

It's all very flexible and completely user defined.

The next section called Transform.

This is quite important because this

allows you to actually include snippets of code,

which are outside the main Template.

Now you can either include,

for example,

code for a Lambda function.

You could include your own code that you want to reuse.

There's also a lot of code snippets which AWS provides.

And I'm just going to show you

the webpage where you can find these.

So I'm just going to search for CloudFormation Snippets.

And there's a whole load of different

examples scenarios that you can use to allow you

to declare various different CloudFormation Template parts.

So for example,

there's Templates Snippets for DynamoDB,

for EC2 Elastic Container Service,

RDS,

Route 53 et cetera, et cetera.

Load there's all sorts of different resources

that they have provided example code for.

So for example,

if we take a look at the DynamoDB one,

its got um an example piece of code

for creating a DynamoDB Table.

Um the various different Properties and Attributes um

it's the GlobalSecondaryIndexes,

ProvisionedThroughput et cetera.

All the different things you need

to define for a DynamoDB table.

And this example here is in JSON

but they also show you it in YAML as well.

So you can actually create reusable pieces of code

which you can store in S3 and CloudFormation

will allow you to actually reference that

within your CloudFormation Template.

So once again it just allows you to reuse code

and maintain consistency throughout your code base.

Then moving on to the Resources section.

So this section of the file is the most important.

And this is where you define the AWS Resources

that you want CloudFormation to deploy for you.

So in this case we're gonna be deploying an EC2Instance.

We're defining the InstanceType as a t2.micro

and we're also specifying which AMI we want to use as well.

And of course,

this Resources section is

the only mandatory section of the CloudFormation Template.

And finally the last section of this example Template

is the Outputs section.

And again this is completely user defined.

And in this particular Template

it gets you to output the Instance ID

of the EC2Instance that we are provisioning.

And the Outputs you can see them

displayed on the console and they can also be

input into another CloudFormation stack as well.

And you don't need to know the Template off by heart.

It's unlikely that they're gonna ask you

really detailed questions about it in the exam

but it's important to understand.

Okay, what is the anatomy of a CloudFormation Template?

What all the different sections are used for.

And remember that the Resources section

is the most important section

and it's the only mandatory section of the Template.

And also remember that the Transform Section,

that is what is used to reference

additional code stored in S3.

It allows for code reuse.

For example,

you can specify Lambda code

and also Templates snippets

or reusable pieces of CloudFormation code.

So let's take a look at some of our

exam tips for CloudFormation.

So CloudFormation allows you to manage,

configure and provision AWS Infrastructure as code.

And remember we use either YAML or JSON

for the CloudFormation Template file.

And remember the main sections

of the CloudFormation Template.

So we have Parameters,

which allows you to input custom values.

And in the example we used an Environment type.

So in that example CloudFormation,

we'll expect you to state which Environment,

the Resources are going to be for.

For example production or test,

Conditions are used to make decisions

based on a set of parameters.

For example,

it will allow you to provision

a Resources based on Environment

if you've used an Environment Parameter.

The Resources Section remember that's the mandatory section

and defines all the AWS Resources to create

and that could be an S3 bucket,

EC2Instances and RDS Database.

Dynamo, DB database.

Pretty much anything that you can think of within AWS.

You can create as a Resource using CloudFormation.

Mappings allow you to create custom mappings.

For example,

mapping a different AMI to a different Region.

So depending on which Region you are working in

you can tell CloudFormation to use

a specific AMI for that Region and finally,

Transform so this allows you

to reference code located in S3.

For example,

Lambda code or reusable snippets of CloudFormation code

and both of those located in S3.

However the best way to learn CloudFormation

is definitely to start using it

and get your hands dirty with it.

And that is exactly what we're going

to be doing in the next lecture.

So the next lecture is going to be a Lab.

So if you have time,

please join me in the next lecture.

Thank you.

[SPANISH]

Hola, Cloud Gurus y bienvenidos a esta conferencia.

Y esta conferencia va a presentar CloudFormation .

Ahora CloudFormation ,

es un servicio que te permite gestionar,

configure y aprovisione su infraestructura de AWS como código

y recursos o los recursos

que vas a provisionar

se definen utilizando una plantilla de formación de nubes

y ese es el elemento de código.

y formación de nubes,

se necesita esa plantilla,

lo interpreta

y hace las llamadas API apropiadas

para crear los recursos

que ha definido en la plantilla.

Y la plantilla soporta

ya sea YAML o JSON.

Entonces puedes usar YAML o JSON

lo que te sientas cómodo

para crear sus plantillas de formación de nubes.

Entonces, ¿cuáles son algunos de los beneficios de CloudFormation ?

Bueno, en primer lugar, le permite aprovisionar

su infraestructura de una manera realmente consistente.

Así que es lo mismo cada vez.

hay menos errores

porque hay menos oportunidad para el error humano.

Se necesita menos tiempo y esfuerzo de lo que sería

si tuviera que configurar las cosas manualmente.

También puedes Control de versiones

y revise sus plantillas por pares

y es de uso gratuito.

Sin embargo, se le cobrará

para cualquier recurso que cree.

Se puede utilizar para gestionar las actualizaciones.

a las plataformas CloudFormation existentes

y también puede manejar dependencias

entre diferentes recursos

para garantizar que los recursos se crean en el orden correcto.

Y también se puede utilizar para muy rápidamente

y revertir y eliminar fácilmente toda la pila

que ha creado dentro de la formación de nubes también.

Así que es muy fácil limpiar los entornos.

que fueron temporales o que acaban de usarse

para probar algo que puedes tener muy rápidamente

Reversión de CloudFormation

y elimine todo el lote para usted también.

Así que echemos un vistazo al proceso de CloudFormation

con un poco más de detalle.

Una vez más,

puedes usar YAML o JSON

para describir el estado final de la infraestructura

usted está aprovisionando o cambiando.

Una vez que haya creado la plantilla,

lo subes a CloudFormation usando S3.

Entonces, dentro de la consola de CloudFormation ,

en realidad lo subes

y termina en un cubo S3

y luego accede a CloudFormation

la plantilla dentro del cubo S3.

Lee la plantilla

y luego hace las llamadas API apropiadas

en su nombre para crear los recursos para usted.

Y los recursos resultantes se denominan "pila"

y se llama CloudFormation Stack.

Así que echemos un vistazo a

un ejemplo de plantilla de CloudFormation .

Y la mayoría de estas opciones o Parámetros son opcionales.

Es solo la sección de Recursos

que veremos en la próxima diapositiva que es obligatorio.

Así que todo lo demás

que puedes ver en esta página es opcional.

Así que en primer lugar en primer lugar,

tenemos la TemplateFormatVersion

y la única versión compatible es 2010-09-09.

Y eso solo se refiere a

la versión del formato Plantilla.

Entonces, el formato de plantilla aceptable o compatible

para la plantilla CloudFormmation .

Hacia adelante,

tenemos la Descripción y por supuesto esto es opcional.

Así que puedes simplemente agregar una Descripción de tu elección,

algo significativo para ti.

Entonces, en este caso,

es uh simplemente describir esto como un

Plantilla para crear una instancia EC2.

El siguiente es Metadatos y eso es simplemente datos sobre datos.

Una vez más, es completamente opcional.

Así que puedes simplemente poner un campo personalizado.

y para este solo estamos proporcionando algunos datos

sobre los casos específicos en que esto

de CloudFormation se va a aprovisionar.

Entonces solo da una descripción de la instancia del servidor web.

Bajando a Parámetros.

Así que estos son valores de entrada que

usted realmente ingresa a CloudFormation .

Cuando realmente sigues adelante y

inicie una pila usando esta Plantilla.

Así que esta es la información que necesita

para proporcionar cuando siga adelante y lance esta pila.

Y en este caso estamos definiendo un tipo de entorno.

y los valores permitidos son prod o test.

Así que tenemos que proporcionar eso cuando en realidad

continúe y lance una pila usando esta Plantilla.

Y luego el siguiente parámetro es Condiciones

y podemos usar Condiciones una vez más

es completamente definido por el usuario.

Y en realidad podemos usar esto

para probar una Condición y tomar acción

basado en el resultado de probar esa Condición.

Entonces, en este caso,

estamos evaluando el um el parámetro EnvType ,

lo declarado en el apartado anterior

en la sección Parámetros.

Así que estamos evaluando eso

y estamos diciendo si es igual a prod

entonces deberíamos crear recursos de producción solamente.

Entonces puede usarlo para decirle a CloudFormation

para tomar decisiones basadas en Parámetros

que haya ingresado en el momento del lanzamiento.

Pasando a Mappings y podemos usar Mappings

para establecer nuestros propios valores definidos por el usuario.

Así por ejemplo,

podríamos establecer el valor de una AMI en función de una Región,

por lo que podemos definir completamente esto nosotros mismos.

Entonces, por ejemplo, para eu-west-1

esa es la AMI que vamos a usar

y podría definir una AMI diferente

para cada región diferente.

Podría definir una AMI diferente

por un ambiente diferente,

etcétera , etcétera.

Todo es muy flexible y completamente definido por el usuario.

La siguiente sección llamada Transformar.

Esto es bastante importante porque este

permite incluir fragmentos de código,

que están fuera de la plantilla principal.

Ahora puede incluir,

por ejemplo,

código para una función Lambda.

Puede incluir su propio código que desee reutilizar.

También hay muchos fragmentos de código que proporciona AWS.

Y solo voy a mostrarte

la página web donde puedes encontrarlos.

Así que solo voy a buscar fragmentos de CloudFormation .

Y hay un montón de diferentes

ejemplos de escenarios que puede usar para permitirle

para declarar varias partes diferentes de la plantilla de CloudFormation .

Así por ejemplo,

hay fragmentos de plantillas para DynamoDB ,

para el servicio de contenedor elástico EC2,

RDS,

Ruta 53, etcétera, etcétera.

Cargue hay todo tipo de recursos diferentes

el que han proporcionado un código de ejemplo.

Así por ejemplo,

si echamos un vistazo al de DynamoDB ,

tiene un ejemplo de código

para crear una tabla de DynamoDB .

Um las diversas propiedades y atributos diferentes um

son los GlobalSecondaryIndexes ,

Rendimiento aprovisionado , etcétera.

Todas las cosas diferentes que necesitas

definir para una tabla de DynamoDB .

Y este ejemplo aquí está en JSON

pero también te lo muestran en YAML.

Entonces puedes crear piezas de código reutilizables

que puede almacenar en S3 y CloudFormation

permitirá hacer referencia a eso

dentro de su plantilla de CloudFormation .

Entonces, una vez más, solo te permite reutilizar el código.

y mantenga la consistencia en toda su base de código.

Luego pasar a la sección de Recursos.

Así que esta sección del archivo es la más importante.

Y aquí es donde define los recursos de AWS

que desea que CloudFormation implemente por usted.

Entonces, en este caso, implementaremos una instancia EC2.

Estamos definiendo InstanceType como t2.micro

y también estamos especificando qué AMI queremos usar también.

Y por supuesto,

esta sección de Recursos es

la única sección obligatoria de la plantilla de CloudFormation .

Y finalmente la última sección de este ejemplo Plantilla

es la sección Salidas.

Y nuevamente, esto está completamente definido por el usuario.

Y en esta plantilla en particular

lleva a generar la ID de la instancia

de la EC2Instance que estamos aprovisionando.

Y las Salidas las puedes ver

se muestran en la consola y también se pueden

entrada en otra pila de CloudFormation también.

Y no es necesario que te sepas la Plantilla de memoria.

Es poco probable que te pregunten

muy detalladas al respecto en el examen.

pero es importante entender.

Bien, ¿cuál es la anatomía de una plantilla de CloudFormation ?

Para qué se utilizan todas las diferentes secciones.

Y recuerda que la sección de Recursos

es la sección más importante

y es la única sección obligatoria de la plantilla.

Y también recuerda que la Sección Transformar,

eso es lo que se usa para hacer referencia

adicional almacenado en S3.

Permite la reutilización de código.

Por ejemplo,

puede especificar el código Lambda

y también fragmentos de plantillas

o piezas reutilizables de código de CloudFormation .

Así que echemos un vistazo a algunos de nuestros

consejos para el examen de CloudFormation .

Entonces, CloudFormation le permite administrar,

configurar y aprovisionar la infraestructura de AWS como código.

Y recuerda que usamos YAML o JSON

para el archivo de plantilla de CloudFormation .

Y recuerda las secciones principales

de la plantilla de CloudFormation .

Así que tenemos Parámetros,

que le permite ingresar valores personalizados.

Y en el ejemplo usamos un tipo de entorno.

Entonces, en ese ejemplo , CloudFormation ,

esperamos que indique qué entorno,

los Recursos van a ser para.

Por ejemplo, producción o prueba,

Las condiciones se utilizan para tomar decisiones.

en base a un conjunto de parámetros.

Por ejemplo,

le permitirá aprovisionar

a Recursos basados en el Medio Ambiente

si ha utilizado un parámetro de entorno.

La Sección de Recursos recuerda que es la sección obligatoria

y define todos los recursos de AWS para crear

y eso podría ser un cubo S3,

EC2Instances y base de datos RDS.

Dynamo, base de datos de base de datos.

Prácticamente cualquier cosa que se le ocurra dentro de AWS.

Puede crear como recurso utilizando CloudFormation .

Las asignaciones le permiten crear asignaciones personalizadas.

Por ejemplo,

mapear una AMI diferente a una región diferente.

Entonces, dependiendo de la región en la que esté trabajando

puede decirle a CloudFormation que use

un AMI específico para esa Región y finalmente,

Transforme para que esto le permita

al código de referencia ubicado en S3.

Por ejemplo,

Código lambda o fragmentos reutilizables del código de CloudFormation

y ambos ubicados en S3.

Sin embargo, la mejor manera de aprender CloudFormation

es definitivamente para empezar a usarlo

y ensúciate las manos con eso.

Y eso es exactamente lo que vamos

hacer en la próxima lección.

Entonces, la próxima conferencia será un laboratorio.

Así que si tienes tiempo,

por favor únase a mí en la próxima conferencia.

Gracias.