[ENGLISH]

Hello, Cloud Gurus, and welcome to this lecture,

which is going to cover CloudWatch actions.

We'll start off with what are CloudWatch actions,

and the most common actions that you're likely to use,

which are PutMetricData and PutMetricAlarm.

We'll go through an example together

and I'll show you how easy it is to use these 2 actions

with the AWS CLI.

And then finally my exam tips.

So what our CloudWatch actions?

Well the CloudWatch API

supports a long list of different actions.

And these actions allow you to publish,

monitor, and alert on a variety of metrics.

And these are particularly powerful

when creating custom metrics

for monitoring and alerting for your application.

And to review the latest supported CloudWatch Actions

you can view the API reference online,

and I've included a link in the resources section.

Now there are loads of different supported actions,

but the most commonly used ones

are PutMetricData and PutMetricAlarm.

So let's take a closer look at both of these

in more detail.

So PutMetricData is the one to use

if you want to publish metric data points to CloudWatch.

You can define the name of the metric,

the name of the namespace to publish to,

the value to publish

and a timestamp for the data point.

And the example on the left does just that.

So the metric name here is PageViewCount.

The namespace is MyService.

And just remember a namespace is just a container,

which allows you to group the metrics together.

The value to publish is 25.

And finally, we have a timestamp for the metric.

So after running this command,

this metric will become visible in CloudWatch.

So what about PutMetricAlarm?

Well, this creates an alarm

associated with a metric

to alert you if a threshold has been reached.

For example, let's say you want to be alerted

if the average PageViewCount

exceeds a threshold of 50

within a time period of 5 minutes or 300 seconds.

Well, that's exactly what this command on the left will do.

So the alarm name is PageViewMon,

the metric name is PageViewCount.

The namespace is MyService.

The statistic that we're using is the Average.

The period is 300 seconds or 5 minutes.

Our threshold is 50.

We're specifying a comparison operator

of GreaterThanOurThreshold.

And the alarm will be triggered

if the threshold is exceeded

for one evaluation period of 300 seconds.

So let's take a look at an example

of how you might use this in the real world.

So imagine you've got an application running on EC2

that is regularly throwing errors.

The application sends metrics into CloudWatch

every time an error occurs.

And we can create an alarm that will be triggered

if more than one critical error data point

is received by CloudWatch within a set time period,

for example, within 5 minutes.

And this is really, really easy to configure in CloudWatch.

And I'm going to show you exactly how easy it is.

Now I've got a simple script here,

which is going to use PutMetricData

to write a bunch of metrics to CloudWatch.

And I'm just using the AWS CLI to put the metrics,

but you could always code your application

to use an AWS SDK to call PutMetricData

and have your applications and metrics to CloudWatch.

And we're just simulating it here using the CLI.

Now I've included this script

in the resources for this lesson,

but you will need to update the timestamp

to reflect today's date

and use appropriate times for your metrics as well.

So I've set my timestamp for today's date

and I've set the times for a few minutes in the future.

So once you've got your script ready,

I'm just going to copy everything.

And then from the AWS Management Console,

I'm going to open up the Cloud Shell.

When your Cloud Shell is ready,

type vi publish\_metrics.sh

and hit Enter. Then hit Escape,

I for insert, then Ctl+v or Cmd+v to paste.

It's asking us to verify the text and hit Paste.

There's our script.

So now once again, hit the Escape key.

Then :wq,

then ! or bang

to save and hit Enter.

And that is our simple script.

And don't forget to add executable permission

to your script.

So we'll run chmod u+x

and the name of our script and hit Enter.

So now we are ready to run our script.

So type ./ and the name of your script and hit Enter.

And if it's all worked correctly,

you shouldn't receive any errors.

So now we can head to the CloudWatch console

and wait for our metrics to appear.

So head to services,

and you'll find CloudWatch.

It's under Management and Governance.

So select Management and Governance,

and I'm going to open the CloudWatch console in a new tab.

So from the CloudWatch console

head to Metrics, All Metrics,

and here is our namespace.

If we select My Service, we've got 1 metric.

So select our metric.

And there it is, CriticalError.

We can add that to our search

and it will appear on the graph above.

And if it's all worked, then within a few minutes,

you should start to see some metrics

appearing in your graph above.

So I'm just going to pause the video now

and come back in a few minutes

when my metrics have appeared.

And there we go.

After a few minutes,

you should start to see some data points appearing.

And I'm going to switch from the 3 hour view

to the 1 hour.

And there we go,

I've got a couple of data points appearing.

So now let's create an alarm

that is going to be triggered

if more than one critical error data point

is received within a 5 minute time period.

And here is the command that we're going to use.

And I've included this command

in the Resources section for the lecture.

So we're using PutMetricAlarm.

The alarm name is CriticalErrorMon.

The metric name is CriticalError.

The namespace is MyService

and the statistic we'll use is SampleCount.

The time period is 300 seconds

and our threshold is 1.

So if we get more than one critical error

within a 5 minute time period,

then the alarm will trigger.

So let's run that command now.

And then we can check our metrics and alarm in CloudWatch.

So come back to the CloudWatch console.

If we head to alarms,

select All Alarms.

Here's our alarm.

The state is insufficient data at the moment,

but hopefully in a few minutes

it should be triggered by our metrics.

So just be patient

and in a couple of minutes,

we should see the alarm getting triggered.

And there we go.

It has gone to a triggered state.

So it's gone to the In alarm state

because we have received more than one CriticalError

within 5 minutes.

So that is PutMetricData and PutMetricAlarm.

And when you're ready to delete everything,

you can just go ahead and delete your alarm.

Onto my exam tips,

and just remember that CloudWatch Actions

allow you to publish, monitor, and alert

on a variety of metrics.

PutMetricData allows you to publish

metric data points to CloudWatch.

So imagine you've got your application

running on EC2 and it's throwing a critical error.

Well, you can use PutMetricData

to send data points

about those critical errors into CloudWatch.

And then we can use PutMetricAlarm

to create an alarm associated with our metric

to alert if a threshold has been reached.

So that's it for this lesson.

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 las acciones de CloudWatch .

Comenzaremos con lo que son las acciones de CloudWatch ,

y las acciones más comunes que es probable que use,

que son PutMetricData y PutMetricAlarm .

Analizaremos un ejemplo juntos

y te mostraré lo fácil que es usar estas 2 acciones

con la CLI de AWS.

Y finalmente mis consejos para el examen.

Entonces, ¿cuáles son nuestras acciones de CloudWatch ?

Bueno, la API de CloudWatch

admite una larga lista de acciones diferentes.

Y estas acciones te permiten publicar,

monitorear y alertar sobre una variedad de métricas.

Y estos son particularmente poderosos

al crear métricas personalizadas

para monitorear y alertar para su aplicación.

Y para revisar las últimas acciones de CloudWatch compatibles

puede ver la referencia de la API en línea,

y he incluido un enlace en la sección de recursos.

Ahora hay un montón de diferentes acciones compatibles,

pero los mas usados

son PutMetricData y PutMetricAlarm .

Así que echemos un vistazo más de cerca a estos dos

con más detalle

Entonces PutMetricData es el que debe usar

si desea publicar puntos de datos de métricas en CloudWatch .

Puede definir el nombre de la métrica,

el nombre del espacio de nombres para publicar,

el valor de publicar

y una marca de tiempo para el punto de datos.

Y el ejemplo de la izquierda hace precisamente eso.

Entonces, el nombre de la métrica aquí es PageViewCount .

El espacio de nombres es MyService .

Y recuerde que un espacio de nombres es solo un contenedor,

que le permite agrupar las métricas.

El valor a publicar es 25.

Y finalmente, tenemos una marca de tiempo para la métrica.

Así que después de ejecutar este comando,

esta métrica se hará visible en CloudWatch .

Entonces, ¿qué pasa con PutMetricAlarm ?

Bueno, esto crea una alarma.

asociado con una métrica

para avisarle si se ha alcanzado un umbral.

Por ejemplo, supongamos que desea recibir una alerta

si el PageViewCount promedio

supera un umbral de 50

dentro de un período de tiempo de 5 minutos o 300 segundos.

Bueno, eso es exactamente lo que hará este comando a la izquierda.

Entonces el nombre de la alarma es PageViewMon ,

el nombre de la métrica es PageViewCount .

El espacio de nombres es MyService .

La estadística que estamos usando es el Promedio.

El período es de 300 segundos o 5 minutos.

Nuestro umbral es 50.

Estamos especificando un operador de comparación

de Mayor que nuestro umbral .

Y la alarma se disparará

si se supera el umbral

durante un período de evaluación de 300 segundos.

Así que echemos un vistazo a un ejemplo.

de cómo podrías usar esto en el mundo real.

Imagine que tiene una aplicación ejecutándose en EC2

que está arrojando errores regularmente.

La aplicación envía métricas a CloudWatch

cada vez que se produce un error.

Y podemos crear una alarma que se disparará

si hay más de un punto de datos de error crítico

es recibido por CloudWatch dentro de un período de tiempo establecido,

por ejemplo, en 5 minutos.

Y esto es muy, muy fácil de configurar en CloudWatch .

Y voy a mostrarte exactamente lo fácil que es.

Ahora tengo un script simple aquí,

que va a utilizar PutMetricData

para escribir un montón de métricas en CloudWatch .

Y solo estoy usando la CLI de AWS para poner las métricas,

pero siempre puedes codificar tu aplicación

usar un SDK de AWS para llamar a PutMetricData

y tener tus aplicaciones y métricas a CloudWatch .

Y solo lo estamos simulando aquí usando la CLI.

Ahora he incluido este script

en los recursos para esta lección,

pero tendrá que actualizar la marca de tiempo

para reflejar la fecha de hoy

y use tiempos apropiados para sus métricas también.

Así que configuré mi marca de tiempo para la fecha de hoy

y he fijado los tiempos para unos minutos en el futuro.

Así que una vez que tengas tu guión listo,

Voy a copiar todo.

Y luego desde la consola de administración de AWS,

Voy a abrir el Cloud Shell.

Cuando su Cloud Shell esté listo,

escriba vipublish\_metrics.sh

y presione Entrar. Luego presione Escape,

I para insertar, luego Ctl+v o Cmd+v para pegar.

Nos pide que verifiquemos el texto y presionemos Pegar.

Ahí está nuestro guión.

Así que ahora, una vez más, presione la tecla Escape.

Entonces: wq ,

entonces ! o explosión

para guardar y presione Entrar.

Y ese es nuestro guión simple.

Y no te olvides de agregar permiso ejecutable

a tu guión.

Así que ejecutaremos chmod u+x

y el nombre de nuestro script y presiona Enter.

Así que ahora estamos listos para ejecutar nuestro script.

Así que escribe ./ y el nombre de tu script y presiona Enter.

Y si todo ha funcionado correctamente,

deberías recibir ningún error.

Así que ahora podemos dirigirnos a la consola de CloudWatch

y esperar a que aparezcan nuestras métricas.

Así que dirígete a servicios,

y encontrará CloudWatch .

Está bajo Gestión y Gobernanza.

Así que seleccione Gestión y Gobernanza,

y voy a abrir la consola de CloudWatch en una pestaña nueva.

Así que desde la consola de CloudWatch

diríjase a Métricas, Todas las métricas,

y aquí está nuestro espacio de nombres.

Si seleccionamos Mi servicio, tenemos 1 métrica.

Así que selecciona nuestra métrica.

Y ahí está, CriticalError .

Podemos agregar eso a nuestra búsqueda.

y aparecerá en el gráfico de arriba.

Y si todo funcionó, en unos minutos,

deberías empezar a ver algunas métricas

que aparece en su gráfico anterior.

Así que solo voy a pausar el video ahora.

y vuelve en unos minutos

cuando han aparecido mis métricas.

Y ahí vamos.

Despues de unos minutos,

debería comenzar a ver aparecer algunos puntos de datos.

Y voy a cambiar de la vista de 3 horas

a la 1 hora.

Y ahí vamos,

Aparecen un par de puntos de datos.

Así que ahora vamos a crear una alarma.

que se va a desencadenar

si hay más de un punto de datos de error crítico

se recibe dentro de un período de tiempo de 5 minutos.

Y aquí está el comando que vamos a usar.

Y he incluido este comando

en la sección Recursos de la conferencia.

Así que estamos usando PutMetricAlarm .

El nombre de la alarma es CriticalErrorMon .

El nombre de la métrica es CriticalError .

El espacio de nombres es MyService

y la estadística que usaremos es SampleCount .

El tiempo es de 300 segundos

y nuestro umbral es 1.

Entonces, si obtenemos más de un error crítico

dentro de un período de tiempo de 5 minutos,

entonces la alarma se disparará.

Así que ejecutemos ese comando ahora.

Y luego podemos consultar nuestras métricas y alarmas en CloudWatch .

Así que vuelva a la consola de CloudWatch .

Si nos dirigimos a las alarmas,

seleccione Todas las alarmas.

Aquí está nuestra alarma.

El estado es datos insuficientes en este momento,

pero espero que en unos minutos

debe ser activado por nuestras métricas.

Así que ten paciencia

y en un par de minutos,

Deberíamos ver cómo se dispara la alarma.

Y ahí vamos.

Ha pasado a un estado activado.

Entonces ha pasado al estado En alarma

porque hemos recibido más de un CriticalError

dentro de 5 minutos.

Eso es PutMetricData y PutMetricAlarm .

Y cuando esté listo para borrar todo,

puede continuar y eliminar su alarma.

En mis consejos de examen,

y recuerde que las acciones de CloudWatch

permite publicar, monitorear y alertar

en una variedad de métricas.

PutMetricData le permite publicar

métricos apuntan a CloudWatch .

Así que imagina que tienes tu aplicación

ejecutándose en EC2 y arroja un error crítico.

Bueno, puedes usar PutMetricData

para enviar puntos de datos

sobre esos errores críticos en CloudWatch .

Y luego podemos usar PutMetricAlarm

para crear una alarma asociada a nuestra métrica

para alertar si se ha alcanzado un umbral.

Eso es todo por esta lección.

Si tiene alguna pregunta, por favor hágamelo saber.

De lo contrario, te veré en la próxima conferencia, gracias.