**AmazonELB- Auto Scaling**

**Indicaciones**

1. Las respuestas deben ser explicadas, solo colocar resultados sin ninguna referencia no puntúa en las preguntas de la evaluación.
2. Realiza una copia de este documento y coloca todas tus respuestas y sube a tu repositorio personal de github en formato markdown. Presenta capturas de pantalla del procedimiento y las explicaciones necesarias. No puntúa si solo se hace la presentación de imágenes.
3. De preferencia adiciona un video adicional explicando los pasos realizados. Utiliza el sandbox de AWS usado en la práctica anterior.
4. Sube a la plataforma de Blackboard el enlace de github donde están todas tus respuestas. No olvides colocar tu nombre y apellido antes de subir el enlace de tus respuestas a la plataforma
5. Cualquier evidencia de copia elimina el examen se informará de la situación a la coordinación.

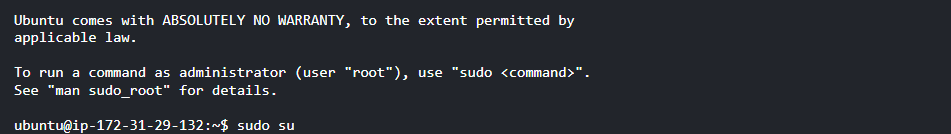
**Primero, haremos la instalación correcta del Apache.**

Estando dentro del Ubuntu con el comando: ssh -i [devenv-key.pem ubuntu@ip\_public\_address](mailto:devenv-key.pem%20ubuntu@34.202.165.14%20)  haremos uso del comando *sudo su* el cual nos mandará al root.

En este caso es: root@ip-172-31-29-132:/home/ubuntu#

Texto

Descripción generada automáticamente



Ingresamos a: ( root@ip-172-31-29-132:/home/ubuntu#) vi /etc/apt/sources.list

Depende de lo que salga en tu pc

Estando dentro escribimos los comandos:

#Insertar

deb http://old-releases.ubuntu.com/ubuntu precise main restricted universe multiverse

deb http://old-releases.ubuntu.com/ubuntu precise-updates main restricted universe multiverse

deb http://old-releases.ubuntu.com/ubuntu precise-security main restricted universe multiverse

#insertar

Guardamos con *wq!* Como se muestra en la imagen de abajo.

**Texto

Descripción generada automáticamente**

Colocamos el comando: root@ip-172-31-26-123:/home/ubuntu# apt-get clean

Para limpiar el caché.

Colocamos el comando: root@ip-172-31-26-123:/home/ubuntu# apt-get update

Texto

Descripción generada automáticamente

Comenzamos la instalación del apache utilizando:

root@ip-172-31-26-123:/home/ubuntu# apt-get install apache2

Preguntamos también el estado del apache.





Reiniciamos el apache2 y volvemos a preguntar su estado

Texto

Descripción generada automáticamente

Insertamos en el navegador la ip pública, en este caso es 3.90.103.236**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente**

¡Se levantó el servidor!

# Amazon ELB

Aquí, usamos Amazon Elastic Load Balancing (ELB) y Amazon Cloud Watch a través de la CLI de AWS para equilibrar la carga de un servidor web.

## Parte 1: ELB

1. Inicia sesión en el sandbox del curso AWS . Ve al directorio donde guarda el archivo de script de instalación de apache del laboratorio de la práctica calificada 3. Para crear un balanceador de carga, haz lo siguiente.

*aws elb create-load-balancer --load-balancer-name tu\_nombre de usuario --oyentes*

*"Protocolo=HTTP,LoadBalancerPort=80,InstanceProtocol=*

*HTTP,InstancePort=80" --availability-zones us-east-1d*

**Corregimos el código:**

*aws elb create-load-balancer --load-balancer-name tu\_nombre de usuario --listeners*

*"Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=*

*HTTP,InstancePort=80" --availability-zones us-east-1a*

¿Cuál es el DNS\_Name del balanceador de carga?

Texto

Descripción generada automáticamente con confianza baja

Como se puede ver en la imagen, el nombre de DNS es:

valery-1036477025.us-east-1.elb.amazonaws.com

1. El comando describe-load-balancers describe el estado y las propiedades de tu(s) balanceador(es) de carga. Presenta este comando.

*aws elb describe-load-balancers*

*--load-balancer-name tu\_nombre\_de\_usuario* ¿Cuál es la salida?

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

1. Creamos dos instancias EC2, cada una ejecutando un servidor web Apache. Emite lo siguiente.

*aws ec2 run-instances --image-id ami-d9a98cb0 --count 2*

*--instance-type t1.micro --key-name tu\_nombre\_de\_usuario-key*

*--security-groups tu\_nombre\_de\_usuario*

*--user-data file://./apache-install --placement AvailabilityZone=us-east-1d*

**Corregido:**

aws ec2 run-instances --image-id ami-d9a98cb0 --count 2

--instance-type t1.micro --key-name tu\_nombre\_de\_usuario-key

--security-groups tu\_nombre\_de\_usuario

--user-data file:/home/ubuntu/./apache --placement AvailabilityZone=us-east-1a

**Resultado:**

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

¿Qué parte de este comando indica que deseas dos instancias EC2?

En la parte de *--count 2* menciona que se desea las dos instancias.

¿Qué parte de este comando garantiza que tus instancias tendrán Apache instalado?

En la parte de *--user-data* [*file:/home/ubuntu/./apache*](file://./apachel)se garantiza la instalación del apache. Solo que, gracias a la pc3, sabemos que el file es /home/ubuntu/./apache-install

¿Cuál es el ID de instancia de la primera instancia?

i-0f5e786efc1fdc70e

¿Cuál es el ID de instancia de la segunda instancia?

i-0c419d2cd70667a9f

1. Para usar ELB, tenemos que registrar las instancias EC2. Haz lo siguiente, donde instance1\_id e instance2\_id son los obtenidos del comando en el paso 3.

*aws elb register-instances-with-load-balancer*

*--load-balancer-name tu\_nombre\_de\_usuario*

*--instances instance1\_id instancia2\_id*

¿Cuál es la salida?

Se actualizó la lista de instancias, que con anterioridad estaba vacío, ahora aparecen las dos distancias que nosotros registramos.

Texto

Descripción generada automáticamente

Ahora vea el estado de la instancia de los servidores cuya carga se equilibra.

*aws elb describe-instance-health*

*--load-balancer-name tu\_nombre\_de\_usuario* ¿Cuál es la salida?

Se observa que las dos instancias se encuentran fuera de servicio, por ende, cuando se ponga en el navegador el DNS no saldrá nada.

Texto

Descripción generada automáticamente

1. Abre el navegador del sandox. Recupera la dirección IP de tu balanceador de carga del paso 1, ingresa la URL http://nombre\_dns\_de\_tu\_balanceador\_carga/ en tu navegador web.

¿Qué apareció en el navegador?

Como se explicó en la anterior pregunta y como se ve en la imagen de abajo, aparece que la página no funciona debido a que no están en funcionamiento ambas instancias.

Interfaz de usuario gráfica, Texto, Sitio web

Descripción generada automáticamente

1. Abre dos ventanas de terminal adicionales y ssh en ambos servidores web. En cada uno, cd al directorio DocumentRoot (probablemente /usr/local/apache/htdocs) y modifique la página de inicio predeterminada, index.html, de la siguiente manera.

*<html><body><h1>¡Funciona!</h1>*

*<p>La solicitud se envió a la instancia 1.</p>*

*<p>La solicitud fue atendida por el servidor web 1.</p>*

*</body></html>*

Para el segundo servidor, haz lo mismo excepto que use la instancia 2 y el servidor 2 para las líneas 2 y 3. En el navegador web, accede a tu balanceador de carga 4 veces (actualícelo/recárgalo 4 veces). Esto genera 4 solicitudes a tu balanceador de carga.

**Para hacer esto necesitamos:**

-Para la primera instancia:

Texto

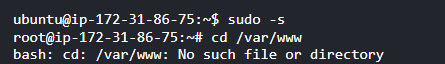
Descripción generada automáticamente

Con el ip público:

Texto

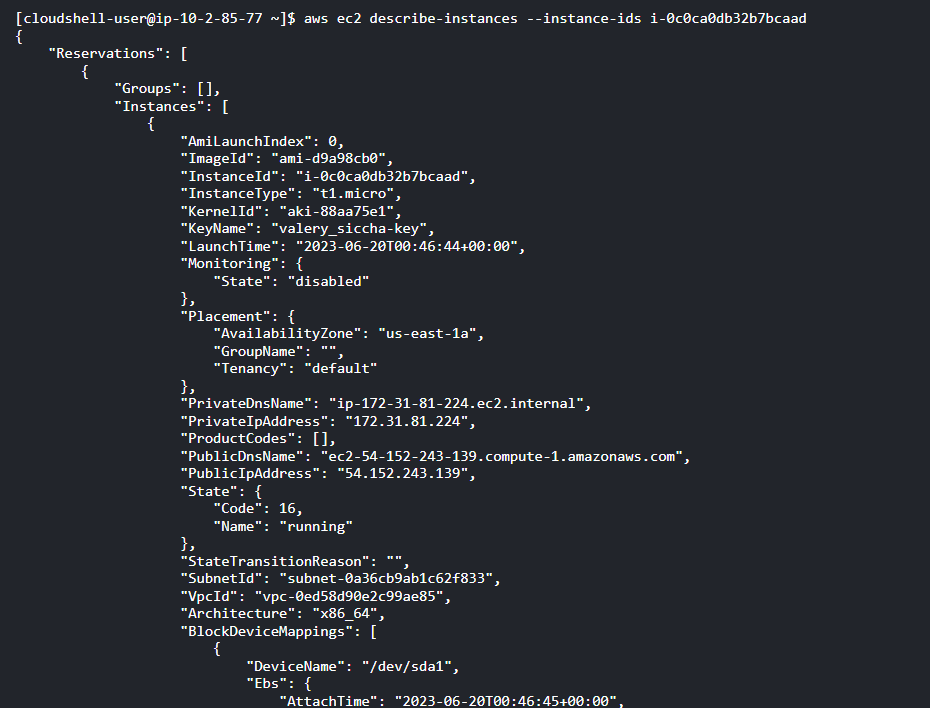
Descripción generada automáticamente

A este punto reinicié la práctica por completo (que me acuerde) y continuó este error:



NO RECONOCE EL DIRECTORIO

-Para la segunda instancia:



Ocurrirá lo mismo que con la primera instancia.

NOTA:

Lo más seguro es que por haber reiniciado la primera vez al ver que no salía, sin querer habré eliminado alguna instancia que afectaba el trabajo anterior/este, de ahí a pesar de haberlo reiniciado un montón de veces en estos días que estuve intentando la práctica, además de ir borrando las instancias que creaba junto con los grupos de seguridad, seguía el mismo error desde el punto número 4 aproximadamente, lo cual me parece extraño porque tenía instalado ya el apache.

Aún así intenté avanzar un poco más de la práctica, aunque me seguía saliendo errores o no me daba datos que deberían de darme. ):

¿Cuántas solicitudes atendió el servidor web 1?

¿Cuántas solicitudes atendió el servidor web 2?

## Parte 2: CloudWatch

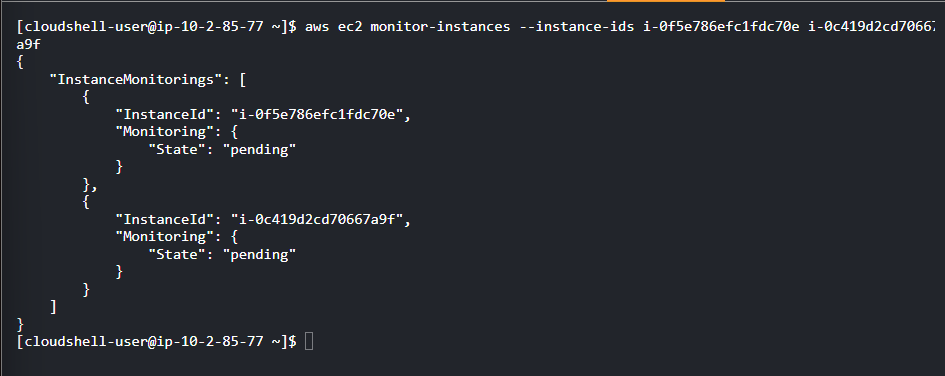
1. CloudWatch se utiliza para monitorear instancias. En este caso, queremos monitorear los dos servidores web. Inicia CloudWatch de la siguiente manera.

*aws ec2 monitor-instances*

*--instance-ids instance1\_id instance2\_id*

¿Cuál es la salida?

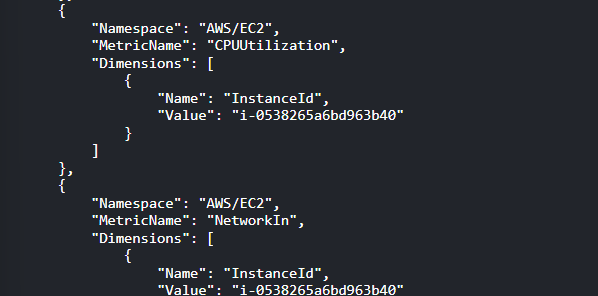
Se muestra que el estado de monitoreo de ambas instancias está en pendiente.



Ahora examina las métricas disponibles con lo siguiente:

*aws cloudwatch list-metrics --namespaces "AWS/EC2"* ¿Viste la métrica CPUUtilization en el resultado?

Después de bajar un poco entre toda la cantidad extensa de resultados que nos dio, se encuentra la métrica de CPUUtilization, el cuál nos muestra el valor del Id de Instancia.



1. Ahora configuramos una métrica para recopilar la utilización de la CPU. Obtener la hora actual con date -u. Esta será tu hora de inicio. Tu hora de finalización debe ser 30 minutos más tarde. Haz lo siguiente.

*aws cloudwatch get-metric-statistics*

*--metric-name CPUUtilization*

*--start-time your\_start\_time --end-time your\_end\_time --period 3600 --namespace AWS/EC2*

*--statistics Maximum*

*– dimensions Name=InstanceId,Value=instance2\_id* ¿Cuál es la salida?

No da ningún valor puesto que la instancia 2 al igual que la 1 está fuera de servicio, por lo tanto, no hay monitoreo en ambas tampoco se estará utilizando el CPU.

Texto

Descripción generada automáticamente

1. Apache tiene una herramienta benchmark llamada ab. Si desea ver más información sobre ab, consulte http://httpd.apache.org/docs/2.0/programs/ab.html. Para ejecutar ab, emita el siguiente comando en tu sistema de trabajo. ab -n 50 -c 5 http://nombre\_dns\_de\_tu\_balanceador\_carga/

¿Qué significan -n 50 y -c 5?

¿Cuál es la salida?

1. Ahora queremos examinar la métrica de latencia del ELB. Utiliza el siguiente comando con las mismas horas de inicio y finalización que especificó en el paso 8.

*aws cloudwatch get-metric-statistics --metric-name Latency*

*--start-time your\_start\_time --end-time your\_end\_time --period 3600 --namespace*

*AWS/ELB*

*--statistics Maximum --dimensions*

*Name=LoadBalancerName,Value=tu\_nombre\_de\_usuario*

*aws cloudwatch get-metric-statistics --metric-name*

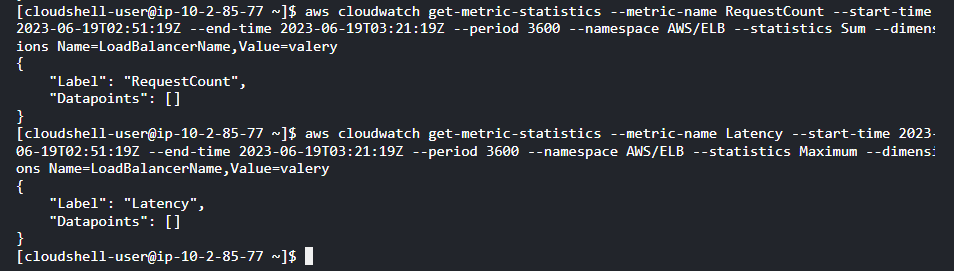
*RequestCount --start-time your\_start\_time --end-time your\_end\_time --period 3600*

*--namespace AWS/ELB*

*--statistics Sum --dimensions*

*Name=LoadBalancerName,Value=tu\_nombre\_de\_usuario* ¿Qué resultados recibió de ambos comandos?

Al tener error previamente con ambas instancias y también problemas con el apache, no habrá contador de solicitudes recibidas ni de latencia.



## Parte 3: Limpieza

1. Necesitamos cancelar el registro de las instancias de ELB. Haz lo siguiente.

*aws elb desregister-instances-from-load-balancer*

*--load-balancer-name tu\_nombre\_de\_usuario*

*--instances instance1\_id instance2\_id* ¿Cuál es la salida?

Ya no aparecen en la lista de Instancias que anteriormente estaba rellenado con los ids de ambas instancias.

*Texto

Descripción generada automáticamente*

1. A continuación, eliminamos la instancia de ELB de la siguiente manera.

*aws elb delete-load-balancer --load-balancer-name tu\_nombre\_de\_usuario*

Finalmente, finaliza las instancias del servidor web de tus instancias y tu instancia EC2.



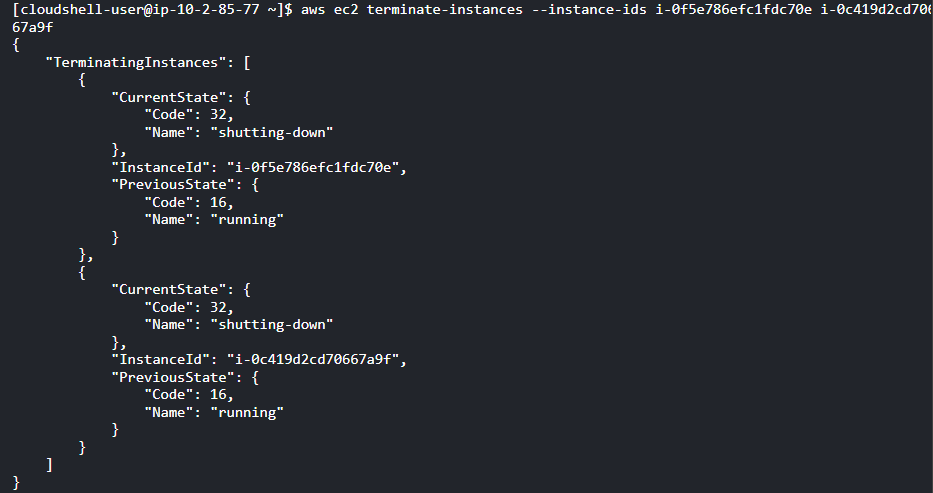
¿Qué comandos usaste?

Para terminar con ambas instancias aplicamos el comando de:

aws ec2 terminate-instances --instance-ids id\_instancia\_1 id\_instancia\_2

¿Cuál es la salida?

Se están finalizando ambas instancias.



# Auto Scaling

Usamos AWS CLI para configurar sus instancias EC2 para el escalado automático.

## Parte 1: escalar hacia arriba

1. Inicia sesión en el sandbox virtual. Cambia al directorio donde guarda el archivo de script de instalación de apache. Inicie una instancia de la siguiente manera.

*aws autoscaling create-launch-configuration*

*--launch-configuration-name tu\_nombre\_de\_usuario-lc*

*--image-id ami-d9a98cb0 --instance-type t1.micro --key-name tu\_nombre\_usuario-key --security-groups tu\_nombre\_usuario --user-data file://./apache-install*

¿Cuál es la salida? A continuación, crea un equilibrador de carga.

*aws elb create-load-balancer --load-balancer-name tu\_nombre\_de\_usuario-elb --listeners*

*"Protocol=HTTP, LoadBalancerPort=80,*

*InstanceProtocol=HTTP,InstancePort=80"*

*– availability-zones us-east-1c* ¿Cuál es la salida?

1. Ahora creamos un grupo de escalado automático. Haz lo siguiente.

*aws autoscaling create-auto-scaling-group*

*--auto-scaling-group-name tu\_nombre\_de\_usuario-asg*

*--launch-configuration-name tu\_nombre\_de\_usuario-lc*

*--min-size 1 --max-size 3 --load-balancer-names tu\_nombre\_de\_usuario-elb --availability-zones us-east-1c*

¿Cuál es la salida?

1. Para describir el grupo de escalado automático que acabas de crear, emite el siguiente comando.

*aws autoscaling describe-auto-scaling-groups*

*--auto-scaling-group-name tu\_nombre\_de\_usuario-asg*

¿Cuál es la salida? Deberías ver que se crea una nueva instancia EC2. Si no lo ves, espera 2 minutos y vuelve a ejecutar el comando ¿Cuál es el ID de la instancia?

1. Ahora creamos una política de escalado hacía arriba seguida de una alarma de CloudWatch para determinar, en caso de que lapolítica sea cierta, que AWS necesita ampliar nuestros recursos. Escribe los dos comandos que siguen. AnotA el valor de PolicyARN del primer comando.

*aws autoscaling put-scaling-policy*

*--auto-scaling-group-name tu\_nombre\_de\_usuario-asg*

*--policy-name tu\_nombre\_de\_usuario-scaleup*

*--scaling-adjustment 1*

*--adjustment-type ChangeInCapacity --cooldown 120 aws cloudwatch put-metric-alarm --alarm-name tu\_nombre\_de\_usuario-highcpualarm*

*--metric-name CPUUtilization --namespace AWS/EC2*

*--statistic Average --period 120 --threshold 70*

*--comparison-operator GreaterThanThreshold --dimensions "Name=AutoScalingGroupName,Value= tu\_nombre\_de\_usuario-asg" --evaluation-periods 1*

*--alarm-actions value\_of\_PolicyARN*

¿Cuál es la salida de ambos comandos? ¿Cuál es el valor de PolicyARN? (El valor es una cadena larga sin comillas). Ejecuta el siguiente comando para ver una descripción de tu alarma.

*aws cloudwatch describe-alarms --alarm-names tu\_nombre\_de\_usuario-highcpualarm*

¿Cuál es la salida?

5. Inicia sesión en la instancia EC2 desde el paso 1 mediante ssh. Cambia al root. Cargaremos y ejecutaremos una herramienta stress para aumentar la utilización de procesamiento del servidor. Emite los siguientes comandos de Linux.

*apt-get install stress stress--cpu 1*

Inicia un nuevo terminal. Repite el siguiente comando cada 2 minutos hasta que veas la segunda instancia EC2 y luego una tercera instancia EC2. En este punto, emite las siguientes instrucciones en la ventana de tu terminal inicial.

*aws autoscaling describe-auto-scaling-groups*

*--auto-scaling-group-name tu\_nombre\_de\_usuario -asg*

¿Cuál es la salida?

## Parte 2: reducir la escala

6. Ahora exploraremos cómo AWS puede controlar el escalado hacia abajo mediante la eliminación de algunas de las máquinas virtuales creadas. Ejecuta los siguientes dos comandos, nuevamente tomando nota del PolicyARN creado a partir del primer comando para usar en el segundo.

*aws autoscaling put-scaling-policy*

*--auto-scaling-group-name tu\_nombre\_de\_usuario-asg*

*--policy-name tu\_nombre\_de\_usuario-scaledown*

*--scaling-adjustment -1 --adjustment-type*

*ChangeInCapacity --cooldown 120 aws cloudwatch put-metric-alarm --alarm-name tu\_nombre\_de\_usuario-lowcpualarm*

*--metric-name CPUUtilization --namespace AWS/EC2*

*--statistic Average --period 120 --threshold 30*

*--comparison-operator LessThanThreshold --dimensions*

*"Name=AutoScalingGroupName,Value=tu\_nombre\_de\_usuario-asg"*

*--evaluation-periods 1 --alarm-actions value\_of\_PolicyARN*

¿Cuáles son las salidas?

¿Cuál es el valor de PolicyARN?

7. Cambia al terminal de la instancia EC2. Escribe ctrl+c para detener el comando stress. Vuelva a la ventana del terminal y repite el siguiente comando cada 2 minutos hasta que vea solo un EC2 en su grupo de escalado automático.

*aws autoscaling describe-auto-scaling-groups*

*--auto-scaling-group-name tu\_nombre\_de\_usuario-asg* ¿Cuál es la salida?

## Parte 3: Limpieza

8. Elimina el grupo de escalado automático mediante el siguiente comando. Si el grupo tiene instancias o actividades de escalado en curso, debes especificar la opción para forzar la eliminación para que se realice correctamente. Si el grupo tiene políticas, al eliminar el grupo se eliminan las políticas, las acciones de alarma subyacentes y cualquier alarma que ya no tenga una acción asociada. Después de eliminar tu grupo de escala, elimina tus alarmas como se muestra a continuación.

*aws autoscaling delete-auto-scaling-group*

*--auto-scaling-group-name tu\_nombre\_de\_usuario-asg*

*--force-delete aws cloudwatch delete-alarms --alarm-name tu\_nombre\_de\_usuario-lowcpualarm*

¿Cuál es la salida?

*aws cloudwatch delete-alarms --alarm-name*

*tu\_nombre\_de\_usuario-highcpualarm*

¿Cuál es la salida?

Elimina tu configuración de lanzamiento de la siguiente manera.

*aws autoescaling delete-launch-configuration*

*--launch-configuration-name tu\_nombre\_de\_usuario-lc*

¿Cuál es la salida?

Finalmente, elimina tu ELB.

¿Qué comando ejecutaste?