



Amazon ELB - 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.

Alumno: Sebastian Antonio Saldaña Rodriguez

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
--protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=
HTTP,InstancePort=80 --availability-zones us-east-1d
```

CORREGIDO:

```
aws elb create-load-balancer --load-balancer-name nombre
--listeners
"Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HT
TP,InstancePort=80" --availability-zones us-east-1d
```

¿Cuál es el DNS_Name del balanceador de carga?

```
"LoadBalancerName": "sebastiansaldana",
"DNSName": "sebastiansaldana-1871654846.us-east-1.elb.amazonaws.com",
"CanonicalHostedZoneId": "sebastiansaldana-1871654846.us-east-1.elb.amazonaws.com"
```

2. 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?

```
[cloudshell-user@ip-10-4-59-167 ~]$ aws elb describe-load-balancers --load-balancer-name sebastiansaldana
{
  "LoadBalancerDescriptions": [
    {
      "LoadBalancerName": "sebastiansaldana",
      "DNSName": "sebastiansaldana-1871654846.us-east-1.elb.amazonaws.com",
      "CanonicalHostedZoneName": "sebastiansaldana-1871654846.us-east-1.elb.amazonaws.com",
      "CanonicalHostedZoneNameID": "Z35SXDOTRQ7X7K",
      "ListenerDescriptions": [
        {
          "Listener": {
            "Protocol": "HTTP",
            "LoadBalancerPort": 80,
            "InstanceProtocol": "HTTP",
            "InstancePort": 80
          },
          "PolicyNames": []
        }
      ],
      "Policies": {
        "AppCookieStickinessPolicies": [],
        "LBCookieStickinessPolicies": [],
        "OtherPolicies": []
      },
      "BackendServerDescriptions": [],
      "AvailabilityZones": [
        "us-east-1d"
      ],
      "Subnets": [
        "subnet-0cb4161aacf62353e"
      ],
      "VPCId": "vpc-0acae352ff95e0c1e",
      "Instances": [
        {

```



3. 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
```

```
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
```

```
[cloudshell-user@ip-10-4-59-167 ~]$ aws ec2 run-instances --image-id ami-d9a98cb0 --count 2 --instance-type
t1.micro --key-name sebastians-key --security-groups sebastian --user-data file:///apache-install --place
me nt AvailabilityZone=us-east-1d
{
  "Groups": [],
  "Instances": [
    {
      "AmiLaunchIndex": 0,
      "ImageId": "ami-d9a98cb0",
      "InstanceId": "i-08c6062d8587168ca",
      "InstanceType": "t1.micro",
      "KernelId": "aki-88aa75e1",
      "KeyName": "sebastians-key",
      "LaunchTime": "2023-06-14T22:51:13+00:00",
      "Monitoring": {
        "State": "disabled"
      },
      "Placement": {
        "AvailabilityZone": "us-east-1d",
        "GroupName": "",
        "Tenancy": "default"
      },
      "PrivateDnsName": "ip-172-31-94-150.ec2.internal",
      "PrivateIpAddress": "172.31.94.150",
      "ProductCodes": [],
      "PublicDnsName": "",
      "State": {
        "Code": 0,
        "Name": "pending"
      },
      "StateTransitionReason": "",
      "SubnetId": "subnet-0cb4161aacf62353e",
      "VpcId": "vpc-0acae352ff95e0c1e",
      "Architecture": "x86_64",
```



```
}
{
  "AmiLaunchIndex": 1,
  "ImageId": "ami-d9a98cb0",
  "InstanceId": "i-0eca58e5482addfec",
  "InstanceType": "t1.micro",
  "KernelId": "aki-88aa75e1",
  "KeyName": "sebastians-key",
  "LaunchTime": "2023-06-14T22:51:13+00:00",
  "Monitoring": {
    "State": "disabled"
  },
  "Placement": {
    "AvailabilityZone": "us-east-1d",
    "GroupName": "",
    "Tenancy": "default"
  },
  "PrivateDnsName": "ip-172-31-95-123.ec2.internal",
  "PrivateIpAddress": "172.31.95.123",
  "ProductCodes": [],
  "PublicDnsName": "",
  "State": {
    "Code": 0,
    "Name": "pending"
  },
  "StateTransitionReason": "",
  "SubnetId": "subnet-0cb4161aacf62353e",
  "VpcId": "vpc-0acae352ff95e0c1e",
  "Architecture": "x86_64",
  "BlockDeviceMappings": [],
  "ClientToken": "5f10168e-138f-49e8-87a6-fd951a91e9d2",
  "EbsOptimized": false,
  "Hypervisor": "xen",
  "NetworkInterfaces": [
    {

```

¿Qué parte de este comando indica que deseas dos instancias EC2? ¿Qué parte de este comando garantiza que tus instancias tendrán Apache instalado? ¿Cuál es el ID de instancia de la primera instancia? ¿Cuál es el ID de instancia de la segunda instancia?

- --count 2
 - --user-data file:///apache-install
 - i-08c6062d8587168ca
 - i-0eca58e5482addfec
4. 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
```

**aws elb register-instances-with-load-balancer --load-balancer-name
tu_nombre_de_usuario --instances instance1_id instancia2_id**

¿Cuál es la salida?



```
[cloudshell-user@ip-10-4-59-167 ~]$ aws elb register-instances-with-load-balancer --load-balancer-name sebastiansaldana --instances i-08c6062d8587168ca i-0eca58e5482addfec
{
  "Instances": [
    {
      "InstanceId": "i-0eca58e5482addfec"
    },
    {
      "InstanceId": "i-08c6062d8587168ca"
    }
  ]
}
[cloudshell-user@ip-10-4-59-167 ~]$ aws elb describe-instance-health --load-balancer-name sebastiansaldana
{
  "InstanceStates": [
    {
      "InstanceId": "i-08c6062d8587168ca",
      "State": "OutOfService",
      "ReasonCode": "Instance",
      "Description": "Instance has failed at least the UnhealthyThreshold number of health checks consecutively."
    },
    {
      "InstanceId": "i-0eca58e5482addfec",
      "State": "OutOfService",
      "ReasonCode": "Instance",
      "Description": "Instance has failed at least the UnhealthyThreshold number of health checks consecutively."
    }
  ]
}
```

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`

`aws elb describe-instance-health --load-balancer-name tu_nombre_de_usuario`

¿Cuál es la salida?

```
[cloudshell-user@ip-10-4-59-167 ~]$ aws elb describe-instance-health --load-balancer-name sebastiansaldana
{
  "InstanceStates": [
    {
      "InstanceId": "i-08c6062d8587168ca",
      "State": "InService",
      "ReasonCode": "N/A",
      "Description": "N/A"
    },
    {
      "InstanceId": "i-0eca58e5482addfec",
      "State": "InService",
      "ReasonCode": "N/A",
      "Description": "N/A"
    }
  ]
}
```

5. 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?



Esta página no funciona

La página **sebastiansaldana-1082915058.us-east-1.elb.amazonaws.com** no puede procesar esta solicitud ahora.

HTTP ERROR 503

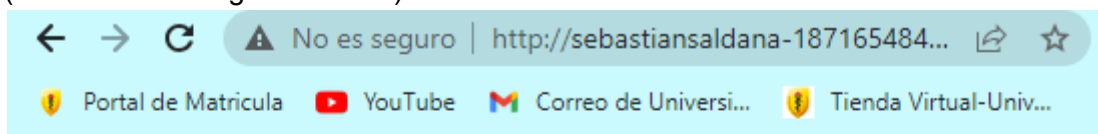
[Volver a cargar](#)



6. 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>
```

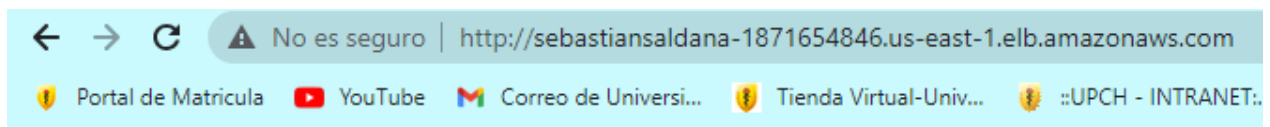
Para el segundo servidor, haz lo mismo excepto que use la instancia 2 y el servidor 2 para las líneas 2 y 3. En el navegador web, accede a tu balanceador de carga 4 veces (actualízelo/recárgalo 4 veces).



Funciona

La solicitud se envió a la instancia 1.

La solicitud fue atendida por el servidor web 1.



!Funciona!

La solicitud se envió a la instancia 2.

La solicitud fue atendida por el servidor web 2.

Esto genera 4 solicitudes a tu balanceador de carga.

¿Cuántas solicitudes atendió el servidor web 1?

- 2

¿Cuántas solicitudes atendió el servidor web 2?

- 2

Parte 2: CloudWatch

7. 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

aws ec2 monitor-instances --instance-ids instance1_id instance2_id

¿Cuál es la salida?

```
[cloudshell-user@ip-10-4-70-241 ~]$ aws ec2 monitor-instances --instance-ids i-08c6062d8587168ca i-0eca58e5482addfec
{
  "InstanceMonitorings": [
    {
      "InstanceId": "i-08c6062d8587168ca",
      "Monitoring": {
        "State": "pending"
      }
    },
    {
      "InstanceId": "i-0eca58e5482addfec",
      "Monitoring": {
        "State": "pending"
      }
    }
  ]
}
```

Ahora examina las métricas disponibles con lo siguiente: aws cloudwatch

list-metrics --namespaces "AWS/EC2"

aws cloudwatch list-metrics --namespaces "AWS/EC2"

¿Viste la métrica CPUUtilization en el resultado?

```
{
  "Namespace": "AWS/EC2",
  "MetricName": "CPUUtilization",
  "Dimensions": [
    {
      "Name": "InstanceId",
      "Value": "i-05f918e5bb0fa74d1"
    }
  ]
}
```

8. 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

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?



```
[cloudshell-user@ip-10-4-70-241 ~]$ aws cloudwatch get-metric-statistics --metric-name CPUUtilization --start-time 2023-06-15T01:39:37Z --end-time 2023-06-15T02:19:37Z --period 3600 --namespace AWS/EC2 --statistics Maximum --dimensions Name=InstanceId,Value=i-0eca58e5482addfec
{
  "Label": "CPUUtilization",
  "Datapoints": [
    {
      "Timestamp": "2023-06-15T01:39:00+00:00",
      "Maximum": 0.499999999999995,
      "Unit": "Percent"
    }
  ]
}
```

9. Apache tiene un 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/

ab -n 50 -c 5 http://nombre_dns_de_tu_balanceador_carga



¿Qué significan -n 50 y -c 5?

- "-n 50": Esto indica el número total de solicitudes que se realizarán al servidor. En este caso, se están enviando 50 solicitudes.
- "-c 5": Esto establece el número de solicitudes concurrentes que se enviarán al servidor al mismo tiempo. En este caso, se están enviando 5 solicitudes simultáneamente.

¿Cuál es la salida?.

```
Server Software:      Apache/2.2.22
Server Hostname:      sebastiansaldana-1871654846.us-east-1.elb.amazonaws.com
Server Port:          80

Document Path:        /
Document Length:      151 bytes

Concurrency Level:     5
Time taken for tests:  0.023 seconds
Complete requests:     50
Failed requests:       25
  (Connect: 0, Receive: 0, Length: 25, Exceptions: 0)
Write errors:          0
Total transferred:     21225 bytes
HTML transferred:      7475 bytes
Requests per second:   2138.49 [#/sec] (mean)
Time per request:      2.338 [ms] (mean)
Time per request:      0.468 [ms] (mean, across all concurrent requests)
Transfer rate:         886.51 [Kbytes/sec] received
```

```
Connection Times (ms)
              min  mean[+/-sd] median  max
Connect:      0    1   0.3      1     2
Processing:    1    2   0.4      1     3
Waiting:       1    2   0.4      1     3
Total:         2    2   0.5      2     4
ERROR: The median and mean for the processing time are more than twice the standard
deviation apart. These results are NOT reliable.
ERROR: The median and mean for the waiting time are more than twice the standard
deviation apart. These results are NOT reliable.
```

```
Percentage of the requests served within a certain time (ms)
 50%    2
 66%    2
 75%    2
 80%    2
 90%    3
 95%    3
 98%    4
 99%    4
100%    4 (longest request)
```

10. 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 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
```

```
[cloudshell-user@ip-10-4-70-241 ~]$ aws cloudwatch get-metric-statistics --metric-name Latency --start-time 2023-06-15T01:39:37Z --end-time 2023-06-15T02:19:37Z --period 3600 --namespace AWS/ELB --statistics Maximum --dimensions Name=LoadBalancerName,Value=sebastiansaldana
{
  "Label": "Latency",
  "Datapoints": [
    {
      "Timestamp": "2023-06-15T01:39:00+00:00",
      "Maximum": 0.002036571502685547,
      "Unit": "Seconds"
    }
  ]
}
```

```
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
```

```
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
```

```
[cloudshell-user@ip-10-4-70-241 ~]$ aws cloudwatch get-metric-statistics --metric-name RequestCount --start-time 2023-06-15T01:39:37Z --end-time 2023-06-15T02:19:37Z --period 3600 --namespace AWS/ELB --statistics Sum --dimensions Name=LoadBalancerName,Value=sebastiansaldana
{
  "Label": "RequestCount",
  "Datapoints": [
    {
      "Timestamp": "2023-06-15T01:39:00+00:00",
      "Sum": 52.0,
      "Unit": "Count"
    }
  ]
}
```



Parte 3: Limpieza

11. Necesitamos cancelar el registro de las instancias de ELB. Haz lo siguiente. `aws elb deregister-instances-from-load-balancer`

`--load-balancer-name tu_nombre_de_usuario`

`--instances instance1_id instance2_id`

`aws elb deregister-instances-from-load-balancer --load-balancer-name tu_nombre_de_usuario --instances instance1_id instance2_id`

¿Cuál es la salida?

```
[cloudshell-user@ip-10-4-70-241 ~]$ aws elb deregister-instances-from-load-balancer --load-balancer-name sebastiansaldana --instances i-08c6062d8587168ca i-0eca58e5482addfec
{
  "Instances": []
}
```

12. A continuación, eliminamos la instancia de ELB de la siguiente manera. `aws elb delete-load-balancer --load-balancer-name tu_nombre_de_usuario`

`aws elb delete-load-balancer --load-balancer-name tu_nombre_de_usuario`

```
[cloudshell-user@ip-10-4-70-241 ~]$ aws elb delete-load-balancer --load-balancer-name sebastiansaldana
[cloudshell-user@ip-10-4-70-241 ~]$
```

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

¿Qué comandos usaste?

- Finalizar las instancias del servidor web:
 - `aws ec2 terminate-instances --instance-ids i-08c6062d8587168ca i-0eca58e5482addfec`

¿Cuál es la salida?



```
"TerminatingInstances": [  
  {  
    "CurrentState": {  
      "Code": 32,  
      "Name": "shutting-down"  
    },  
    "InstanceId": "i-0eca58e5482addfec",  
    "PreviousState": {  
      "Code": 16,  
      "Name": "running"  
    }  
  },  
  {  
    "CurrentState": {  
      "Code": 32,  
      "Name": "shutting-down"  
    },  
    "InstanceId": "i-08c6062d8587168ca",  
    "PreviousState": {  
      "Code": 16,  
      "Name": "running"  
    }  
  }  
]
```



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
```

```
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?

```
[cloudshell-user@ip-10-6-170-27 ~]$ aws autoscaling describe-launch-configurations --launch-conf
figuration-names sebastiansal-lc
{
  "LaunchConfigurations": [
    {
      "LaunchConfigurationName": "sebastiansal-lc",
      "LaunchConfigurationARN": "arn:aws:autoscaling:us-east-1:903131493500:launchConfigur
ation:33814c53-4094-48cd-b322-14cb94f6aac3:launchConfigurationName/sebastiansal-lc",
      "ImageId": "ami-d9a98cb0",
      "KeyName": "sebastians-key",
      "SecurityGroups": [
        "sebastian"
      ],
      "ClassicLinkVPCSecurityGroups": [],
      "UserData": "IyEvYmluL2Jhc2gKIyBQcmV2ZW50IGFwdC1nZXQgZnJvbSBicmluZ2luZyB1cCBhbnkgaw5
0ZXJhY3RpdmUgcXVlcmllcwpleHBvcnQgREVCSUF0X0ZST05URU5EPW5vaW50ZXJhY3RpdmUKIyBVcGdyYWRLIHBhY2thZ2V
zCmFwdC1nZXQgdXBkYXRlCiMgSW5zdGFsbCBBCGFjaGUKYXB0LWdlldCBpbmN0YWxsIGFwYWN0ZTIgLXkK",
      "InstanceType": "t1.micro",
      "KernelId": "",
      "RamdiskId": "",
      "BlockDeviceMappings": [],
      "InstanceMonitoring": {
        "Enabled": true
      },
      "CreatedTime": "2023-06-16T01:49:34.471000+00:00"
```

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

```
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?

```
[cloudshell-user@ip-10-6-170-27 ~]$ aws elb create-load-balancer --load-balancer-name sebastiansaldana-elb --listeners "Protocol=HTTP, LoadBalancerPort=80, InstanceProtocol=HTTP, InstancePort=80" --availability-zones us-east-1c  
{  
  "DNSName": "sebastiansaldana-elb-545995836.us-east-1.elb.amazonaws.com"  
}  
[cloudshell-user@ip-10-6-170-27 ~]$
```

2. 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
```

```
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?

```
[cloudshell-user@ip-10-6-170-27 ~]$ aws autoscaling create-auto-scaling-group --auto-scaling-group-name sebastiansaldana-asg --launch-configuration-name sebastiansaldana-lc --min-size 1 --max-size 3 --load-balancer-names sebastiansaldana-elb --availability-zones us-east-1c  
[cloudshell-user@ip-10-6-170-27 ~]$
```

3. 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
```

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name  
tu_nombre_de_usuario-asg
```

¿Cuál es la salida?



```
[cloudshell-user@ip-10-6-170-27 ~]$ aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names sebastiansalda-asg
{
  "AutoScalingGroups": [
    {
      "AutoScalingGroupName": "sebastiansalda-asg",
      "AutoScalingGroupARN": "arn:aws:autoscaling:us-east-1:903131493500:autoScalingGroup:4b316920-8b59-4580-a9ab-41e54bdf9e11:autoScalingGroupName/sebastiansalda-asg",
      "LaunchConfigurationName": "sebastiansal-lc",
      "MinSize": 1,
      "MaxSize": 3,
      "DesiredCapacity": 1,
      "DefaultCooldown": 300,
      "AvailabilityZones": [
        "us-east-1c"
      ],
      "LoadBalancerNames": [
        "sebastiansaldana-elb"
      ],
      "TargetGroupARNs": [],
      "HealthCheckType": "EC2",
      "HealthCheckGracePeriod": 0,
      "Instances": [
        {

```

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?

i-03193c8a55d8579ee

4. Ahora creamos una política de escalado hacia arriba seguida de una alarma de CloudWatch para determinar, en caso de que la política 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 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
```

```
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?

```
[cloudshell-user@ip-10-6-170-27 ~]$ aws autoscaling put-scaling-policy --auto-scaling-group-name se
bastiansalda-asg --policy-name sebastiansalda-scaleup --scaling-adjustment 1 --adjustment-type ChangeInCap
acity --cooldown 120
{
  "PolicyARN": "arn:aws:autoscaling:us-east-1:903131493500:scalingPolicy:ada4846d-c1d5-4202-979f-
46289a66c12f:autoScalingGroupName/sebastiansalda-asg:policyName/sebastiansalda-scaleup",
  "Alarms": []
}
[cloudshell-user@ip-10-6-170-27 ~]$
```



```
[cloudshell-user@ip-10-6-170-27 ~]$ aws cloudwatch put-metric-alarm --alarm-name sebastiansal-highcpualarm --metric-name CPUUtilization --namespace AWS/EC2 --statistic Average --period 120 --threshold 70 --comparison-operator GreaterThanThreshold --dimensions "Name=AutoScalingGroupName,Value=sebastiansal-asg" --evaluation-periods 1 --alarm-actions arn:aws:autoscaling:us-east-1:903131493500:scalingPolicy:ada4846d-c1d5-4202-979f-46289a66c12f:autoScalingGroupName/sebastiansalda-asg:policyName/sebassa-scalaup
[cloudshell-user@ip-10-6-170-27 ~]$
```

¿Cuál es el valor de PolicyARN? (El valor es una cadena larga sin comillas).

arn:aws:autoscaling:us-east-1:903131493500:scalingPolicy:ada4846d-c1d5-4202-979f-46289a66c12f:autoScalingGroupName/sebastiansalda-asg:policyName/sebassa-scalaup

Ejecuta el siguiente comando para ver una descripción de tu alarma.

```
aws cloudwatch describe-alarms
--alarm-names
tu_nombre_de_usuario-highcpualarm
```

```
aws cloudwatch describe-alarms
--alarm-names
tu_nombre_de_usuario-highcpualarm
```

¿Cuál es la salida?



```
[cloudshell-user@ip-10-6-170-27 ~]$ aws cloudwatch describe-alarms --alarm-names sebassal-highcpualarm
{
  "MetricAlarms": [
    {
      "AlarmName": "sebassal-highcpualarm",
      "AlarmArn": "arn:aws:cloudwatch:us-east-1:903131493500:alarm:sebassal-highcpualarm",
      "AlarmConfigurationUpdatedTimestamp": "2023-06-16T02:16:55.561000+00:00",
      "ActionsEnabled": true,
      "OKActions": [],
      "AlarmActions": [
        "arn:aws:autoscaling:us-east-1:903131493500:scalingPolicy:ada4846d-c1d5-4202-979f-46289a66c12f:autoScalingGroupName/sebastiansalda-asg:policyName/sebassal-scaleup"
      ],
      "InsufficientDataActions": [],
      "StateValue": "INSUFFICIENT_DATA",
      "StateReason": "Unchecked: Initial alarm creation",
      "StateUpdatedTimestamp": "2023-06-16T02:16:55.561000+00:00",
      "MetricName": "CPUUtilization",
      "Namespace": "AWS/EC2",
      "Statistic": "Average",
    }
  ]
}
```

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
```

```
root@ip-172-31-7-133:/home/ubuntu# stress --cpu 1
stress: info: [3530] dispatching hogs: 1 cpu, 0 io, 0 vm, 0 hdd
```

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
```

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name
sebastiansalda-asg
```

¿Cuál es la salida?



```
[cloudshell-user@ip-10-6-170-27 ~]$ aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name sebastiansalda-asg
{
  "AutoScalingGroups": [
    {
      "AutoScalingGroupName": "sebastiansalda-asg",
      "AutoScalingGroupARN": "arn:aws:autoscaling:us-east-1:903131493500:autoScalingGroup:4b316920-8b59-4580-a9ab-41e54bdf9e11:autoScalingGroupName/sebastiansalda-asg",
      "LaunchConfigurationName": "sebastiansal-lc",
      "MinSize": 1,
      "MaxSize": 3,
      "DesiredCapacity": 1,
      "DefaultCooldown": 300,
      "AvailabilityZones": [
        "us-east-1c"
      ],
      "LoadBalancerNames": [
        "sebastiansaldana-elb"
      ],
      "TargetGroupARNs": [],
      "HealthCheckType": "EC2",
      "HealthCheckGracePeriod": 0,
      "Instances": [
        {
          "InstanceId": "i-03193c8a55d8579ee",
          "InstanceType": "t2.micro",
          "InstanceProfile": "AWSAutoScalingProfile",
          "SubnetId": "subnet-0a1b2c3d",
          "EBSOptimized": false,
          "PlacementGroup": null,
          "AvailabilityZone": "us-east-1c",
          "LaunchConfigurationName": "sebastiansal-lc",
          "AutoScalingGroup": "sebastiansalda-asg",
          "HealthStatus": "Healthy",
          "ProtectedFromScaleIn": false
        }
      ]
    }
  ]
}
```

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 autoscaling put-scaling-policy --auto-scaling-group-name sebastiansal-asg
--policy-name sebastianrodri-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
```

```
aws cloudwatch put-metric-alarm --alarm-name
sebasrodri-lowcpualarm --metric-name CPUUtilization
--namespace AWS/EC2 --statistic Average --period 120
--threshold 30 --comparison-operator LessThanThreshold
--dimensions
"Name=AutoScalingGroupName,Value=sebastiansalda-asg"
--evaluation-periods 1 --alarm-actions value_of_PolicyARN
```

¿Cuáles son las salidas?

```
[cloudshell-user@ip-10-6-170-27 ~]$ aws autoscaling put-scaling-policy --auto-scaling-group-name sebastiansalda-asg --policy-name sebastianrodri-scaledown --scaling-adjustment -1 --adjustment-type ChangeInCapacity --cooldown 120
{
  "PolicyARN": "arn:aws:autoscaling:us-east-1:903131493500:scalingPolicy:8907534d-eae7-476e-9204-8c237708b1fa:autoScalingGroupName/sebastiansalda-asg:policyName/sebastianrodri-scaledown",
  "Alarms": []
}
[cloudshell-user@ip-10-6-170-27 ~]$
```



```
[cloudshell-user@ip-10-6-170-27 ~]$ aws cloudwatch put-metric-alarm --alarm-name
sebasrodri-lowcpualarm --metric-name CPUUtilization --namespace AWS/EC2 --statist
ic Average --period 120 --threshold 30 --comparison-operator LessThanThreshold --
dimensions "Name=AutoScalingGroupName,Value=sebastiansalda-asg" --evaluation-peri
ods 1 --alarm-actions arn:aws:autoscaling:us-east-1:903131493500:scalingPolicy:89
07534d-eae7-476e-9204-8c237708b1fa:autoScalingGroupName/sebastiansalda-asg:policy
Name/sebastianrodri-scaledown
[cloudshell-user@ip-10-6-170-27 ~]$
```

¿Cuál es el valor de PolicyARN?

- arn:aws:autoscaling:us-east-1:903131493500:scalingPolicy:8907534d-eae7-476e-9204-8c237708b1fa:autoScalingGroupName/sebastiansalda-asg:policyName/sebastianrodri-scaledown

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
```

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name
sebastiansalda-asg
```

¿Cuál es la salida?

Efectivamente solo me aparece una instancia.

```
cloudshell-user@ip-10-6-170-27 ~]$ aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name sebastiansalda-asg
{
  "AutoScalingGroups": [
    {
      "AutoScalingGroupName": "sebastiansalda-asg",
      "AutoScalingGroupARN": "arn:aws:autoscaling:us-east-1:903131493500:autoScalingGroup:4b316920-8b59-4580-a9ab-41e54bdf9e11:autoScalingGroupName/sebastiansalda-asg",
      "LaunchConfigurationName": "sebastiansal-lc",
      "MinSize": 1,
      "MaxSize": 3,
      "DesiredCapacity": 1,
      "DefaultCooldown": 300,
      "AvailabilityZones": [
        "us-east-1c"
      ],
      "LoadBalancerNames": [
        "sebastiansaldana-elb"
      ],
      "TargetGroupARNs": [],
      "HealthCheckType": "EC2",
      "HealthCheckGracePeriod": 0,
      "Instances": [
        {
          "InstanceId": "i-03193c8a55d8579ee",
          "InstanceType": "t1.micro",
          "LaunchConfigurationName": "sebastiansal-lc",
          "CreationTime": "2015-06-15T15:35:00.000Z",
          "State": "Running",
          "StateReason": "Successful completion of state transition"
        }
      ]
    }
  ]
}
```

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 autoscaling delete-auto-scaling-group  
--auto-scaling-group-name  
sebastiansalda-asg --force-delete
```

```
aws cloudwatch delete-alarms  
--alarm-name  
tu_nombre_de_usuario-lowcpualarm
```

```
aws cloudwatch delete-alarms  
--alarm-names  
sebasrodri-lowcpualarm
```

¿Cuál es la salida?

```
[cloudshell-user@ip-10-6-170-27 ~]$ aws autoscaling delete-auto-scaling-group --a  
uto-scaling-group-name sebastiansalda-asg --force-delete  
[cloudshell-user@ip-10-6-170-27 ~]$ █
```

```
[cloudshell-user@ip-10-6-170-27 ~]$ aws cloudwatch delete-alarms --alarm-names se  
basrodri-lowcpualarm  
[cloudshell-user@ip-10-6-170-27 ~]$ █
```

aws cloudwatch delete-alarms --alarm-name



tu_nombre_de_usuario-highcpualarm

aws cloudwatch delete-alarms --alarm-names sebasal-highcpualarm

¿Cuál es la salida?

```
[cloudshell-user@ip-10-6-170-27 ~]$ aws cloudwatch delete-alarms --alarm-names sebasal-highcpualarm
[cloudshell-user@ip-10-6-170-27 ~]$
```

Elimina tu configuración de lanzamiento de la siguiente manera.

aws autoecaling delete-launch-configuration

--launch-configuration-name tu_nombre_de_usuario-lc

aws autoscaling delete-launch-configuration

--launch-configuration-name sebastiansal-lc

¿Cuál es la salida?

```
[cloudshell-user@ip-10-6-170-27 ~]$ aws autoscaling delete-launch-configuration --launch-configuration-name sebastiansal-lc
[cloudshell-user@ip-10-6-170-27 ~]$
```

Finalmente, elimina tu ELB.

¿Qué comando ejecutaste?.

- **aws elb delete-load-balancer --load-balancer-name sebastiansaldana-elb**

```
[cloudshell-user@ip-10-6-170-27 ~]$ aws elb delete-load-balancer --load-balancer-name sebastiansaldana-elb
[cloudshell-user@ip-10-6-170-27 ~]$
```