



PRÁCTICA 2: BASE DE DATOS, BALANCEO Y ESCALADO

RAÚL MATEUS SÁNCHEZ

COMPUTACIÓN EN LA NUBE
Escuela de Ingeniería Informática



Índice

Introducción	2
Objetivos	2
Desarrollo de la práctica	3
1. Despliega dos instancias en EC2 con un servidor web que muestre una página similar pero que se pueda reconocer que es un servidor distinto. E.g. [El servidor de Gabriel 1] [El servidor de Gabriel 2]. Estos servidores deben poder ser accedidos con un navegador desde fuera.....	3
2. Despliega un “load balancer” que distribuya las peticiones entre los dos servidores a partes iguales	8
3. Prepara un “template” de instancia para EC2 para generar servidores web. Con el “template” declarar un “Auto-Scaling Group”(ASG) que tenga como mínimo una instancia y como máximo 2. El ASG debe añadirse al “load balancer” previamente desplegado. Comprueba que el ASG mantiene al menos una instancia viva y que el “load balancer” le manda peticiones entrantes	13
Diagrama de arquitectura desplegada	20
Presupuesto y estimación de gasto de los recursos desplegados	21
Conclusiones	22

Introducción

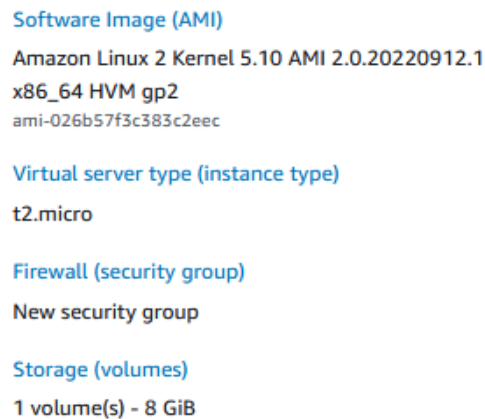
Dentro de Amazon Web Services, las posibilidades son infinitas. Por ello, deben ser exploradas y analizadas para saber su utilidad dependiendo del coste, contexto y dificultad de despliegue. Es por ello por lo que, en esta práctica, empleando los conocimientos adquiridos en las clases teóricas, se explorará más en profundidad EC2 y todas sus funcionalidades.

Objetivos

El objetivo de esta práctica es explorar y experimentar con las herramientas de balanceo de carga y escalado explicadas en la clase teórica.

Desarrollo de la práctica

1. Despliega dos instancias en EC2 con un servidor web que muestre una página similar pero que se pueda reconocer que es un servidor distinto. E.g. [El servidor de Gabriel 1] [El servidor de Gabriel 2]. Estos servidores deben poder ser accedidos con un navegador desde fuera



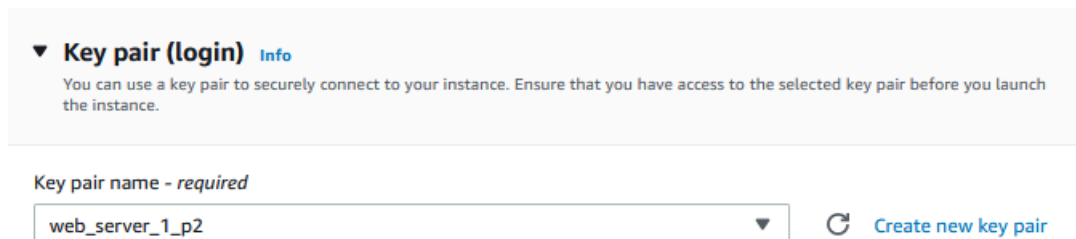
Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI 2.0.20220912.1
x86_64 HVM gp2
ami-026b57f3c383c2eec

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Figura 1: Resumen de la instancia 1



▼ **Key pair (login)** [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

web_server_1_p2 ▼

🔄 [Create new key pair](#)

Figura 2: Par de claves para la instancia 1

VPC - required [Info](#)

vpc-088feb7f494a9d6b (default) [Refresh](#)

172.31.0.0/16

Subnet [Info](#)

No preference [Refresh](#) [Create new subnet](#)

Auto-assign public IP [Info](#)

Enable

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group ☐ Select existing security group

Security group name - required

web_server

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-./()#,@[]+=&:~!\$*

Description - required [Info](#)

web_server|created 2022-10-20T09:51:50.886Z

Inbound security groups rules

▼ Security group rule 1 (TCP, 22, sg-0101e454a0c3e335d, SSH for admin access) [Remove](#)

Type Info	Protocol Info	Port range Info
ssh	TCP	22

Source type Info	Source Info	Description - optional Info
Custom	<input type="text" value="Add CIDR, prefix list or security group ID"/> sg-0101e454a0c3e335d X	SSH for admin access

▼ Security group rule 2 (TCP, 80, 0.0.0.0/0, HTTP from anywhere) [Remove](#)

Type Info	Protocol Info	Port range Info
HTTP	TCP	80

Source type Info	Source Info	Description - optional Info
Anywhere	<input type="text" value="Add CIDR, prefix list or security group ID"/> 0.0.0.0/0 X	HTTP from anywhere

Figura 3: Grupo de seguridad para la instancia 1

Para desplegar las instancias requeridas, se mantendrá la instancia SSH_Gate de la práctica anterior, la cual servirá de acceso mediante SSH a las instancias que se desplegarán a continuación. Para crear la instancia 1, se realiza el mismo proceso de anteriores prácticas. Para ello, a partir de la opción *Launch Instantes* de Amazon EC2, introducimos de nombre de instancia “web_server_1_p2”, y se selecciona Amazon Linux como imagen y t2.micro como tipo de instancia en su versión *Free Tier*.

Para configurar el grupo de seguridad de la máquina, estableceremos reglas para permitir solo el acceso SSH a la instancia SSH_Gate, y las solicitudes HTTP desde cualquier lado a la máquina, restringiendo cualquier otra conexión.

Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI... read more
ami-026b57f3c383c2eec
Virtual server type (instance type)
t2.micro
Firewall (security group)
web_server
Storage (volumes)
1 volume(s) - 8 GiB

Figura 4: Resumen de la instancia 2

▼ Network settings

Info

Edit

Network

Info

vpc-088febf7f494a9d6b

Subnet

Info

No preference (Default subnet in any availability zone)

Auto-assign public IP

Info

Enable

Firewall (security groups)

Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group

☒ Select existing security group

Common security groups

Info

Select security groups

web_server sg-0fc8ae4290e9f292a

×

VPC: vpc-088febf7f494a9d6b

↻ Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Figura 5: Grupo de seguridad para la instancia 2

Para la instancia 2, se genera del mismo modo que la instancia 1 pero de forma más simple. Esto es debido a que, aunque se genera una nueva imagen y un nuevo par de claves, el grupo de seguridad será el mismo que el de la instancia 1, pues al final ambas máquinas tienen la misma función, es decir, actuar como servidor web, y ambas serán accedidas desde la instancia SSH_gate.

```
[ec2-user@ip-172-31-27-246 .ssh]$ touch web_server_1_p2.pem
[ec2-user@ip-172-31-27-246 .ssh]$ touch web_server_2_p2.pem
[ec2-user@ip-172-31-27-246 .ssh]$ sudo vim web_server_1_p2.pem
[ec2-user@ip-172-31-27-246 .ssh]$ sudo vim web_server_2_p2.pem
[ec2-user@ip-172-31-27-246 .ssh]$ ls
authorized_keys  known_hosts  web_server_1_p2.pem  web_server_2_p2.pem
[ec2-user@ip-172-31-27-246 .ssh]$
```

Figura 6: Configuración para el acceso SSH

```
[ec2-user@ip-172-31-93-85 html]$ history
1  sudo yum update
2  yum -y install httpd
3  sudo yum -y install httpd
4  sudo systemctl enable httpd
5  sudo vim /var/www/http/index.html
6  cd /var
7  ls
8  cd www
9  ls
10 cd html/
11 ls
12 touch index.html
13 sudo touch index.html
14 sudo vim index.html
15 systemctl start httpd
16 sudo systemctl start httpd
```

Figura 7: Configuración del servicio Apache/HTTP

Una vez se han lanzado las dos instancias, podemos acceder a ellas a través de la instancia SSH_Gate para instalar el servicio Apache y configurar el servidor web en ambas. Para ello, se ejecuta el conjunto de órdenes apreciable en la **Figura 7**: Configuración del servicio Apache/HTTP, donde en ambas instancias logra instalar y arrancar el servicio.

Ya configurado, se necesita poder mostrar algo cuando accedamos a cada servidor además de poder diferenciarlos. Es por ello por lo que se modifica el index.html de cada servidor para que en uno de ellos muestre “El servidor de Raúl: 1”, y en el otro el “El servidor de Raúl 2”, como se logra ver accediendo a cada servidor desde sus direcciones IP públicas facilitadas por AWS.

Para poder acceder a ambos desde navegadores externos, se configuró, mencionado previamente, el grupo de seguridad al que pertenecen estas dos instancias para solo permitir el acceso SSH desde SSH_gate y peticiones HTTP por el puerto 80.

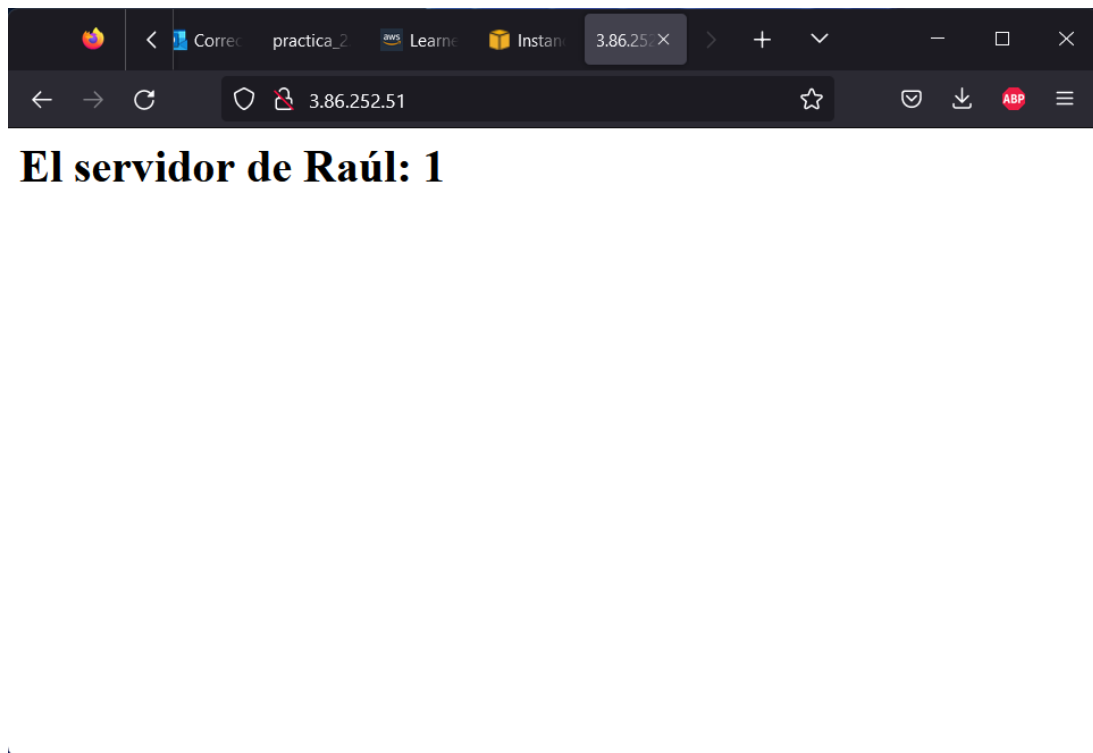


Figura 8: Servidor 1 en activo



Figura 9: Servidor 2 en activo

2. Despliega un “load balancer” que distribuya las peticiones entre los dos servidores a partes iguales

En esta parte de la práctica, se desplegará un *load balancer* que gestionará la carga de las instancias asociadas a él para repartir las peticiones a partes iguales. Para crearlo, se accede al apartado de “*load balancers*” dentro de EC2 para crearlo.

En la configuración básica, se le asigna un nombre identificativo como es “web_server_load_balancer”, así como se identifica el esquema del *load balancer* como Internet-facing, ya que enrutará las peticiones de los clientes por internet a las instancias objetivo. Por último, se seleccionan las IP de tipo IPv4, recomendadas para este tipo de balanceo de carga.

Create Application Load Balancer [Info](#)

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

► How Application Load Balancers work

Basic configuration

Load balancer name
Name must be unique within your AWS account and cannot be changed after the load balancer is created.

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme [Info](#)
Scheme cannot be changed after the load balancer is created.
☒ **Internet-facing**
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#) [↗](#)
☐ **Internal**
An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type [Info](#)
Select the type of IP addresses that your subnets use.
☒ **IPv4**
Recommended for internal load balancers.
☐ **Dualstack**
Includes IPv4 and IPv6 addresses.

Figura 10: Configuración básica del Load Balancer

Network mapping [Info](#)
The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC [Info](#)
Select the virtual private cloud (VPC) for your targets. Only VPCs with an internet gateway are enabled for selection. The selected VPC cannot be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

-
vpc-088febf7f494a9d6b
IPv4: 172.31.0.0/16

Mappings [Info](#)
Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

☐ us-east-1a

☒ us-east-1b
Subnet
subnet-00db343d4ea75c3f8
IPv4 settings
Assigned by AWS

☒ us-east-1c
Subnet
subnet-067218daac8437c2d
IPv4 settings
Assigned by AWS

Figura 11: Mapeo de red

Como el load balancer enruta el tráfico a las instancias en las subredes seleccionadas, debemos elegir las acorde con las instancias ya creadas. Por ello, en este caso, se eligen las subredes visibles en la **Figura 11:** Mapeo de red

Por otro lado, se debe asignar un grupo de seguridad al *load balancer*, por lo cual, aplicaremos el mismo que a las dos instancias creadas ya que el fin sigue siendo que solo se acepten peticiones HTTP desde el exterior.

Security groups [Info](#)
A security group is a set of firewall rules that control the traffic to your load balancer.

Security groups

Select up to 5 security groups

Create new security group

web_server sg-0fc8ae4290e9f292a X
VPC: vpc-088febf7f494a9d6b

Figura 12: Grupo de seguridad del load balancer

Summary

Review and confirm your configurations. [Estimate cost](#)

Basic configuration [Edit](#)

web_server_load_balancer

- Internet-facing
- IPv4

Security groups [Edit](#)

- web_server
[sg-0fc8ae4290e9f292a](#)

Network mapping [Edit](#)

VPC [vpc-088feb7f7f494a9d6b](#)

- us-east-1b
[subnet-00db343d4ea75c3f8](#)
- us-east-1c
[subnet-067218daac8437c2d](#)

Listeners and routing [Edit](#)

- HTTP:80 defaults to
Target group not defined

Add-on services [Edit](#)

None

Tags [Edit](#)

None

Attributes

Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.

Figura 13: Resumen del load balancer

EC2 > Target groups

Target groups (1/1) [Info](#)

Refresh

Actions

Create target group

< 1 >

⚙

<input checked="" type="checkbox"/>	Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
<input checked="" type="checkbox"/>	WebServerTargetGroup	arn:aws:elasticloadbalancing...	80	HTTP	Instance	None associated	vpc-088feb7f7f494a9d6b

Target group: WebServerTargetGroup

Details

Targets

Monitoring

Health checks

Attributes

Tags

Registered targets (2)

Refresh

Deregister

Register targets

< 1 >

⚙

<input type="checkbox"/>	Instance ID	Name	Port	Zone	Health status	Health status details
<input type="checkbox"/>	i-0fbb74fc74ab23044	web_server_1_p2	80	us-east-1b	unused	Target group is not configured to receive traffic from the load balancer
<input type="checkbox"/>	i-068f44fd97591e897	web_server_2_p2	80	us-east-1c	unused	Target group is not configured to receive traffic from the load balancer

Figura 14: Target group del load balancer

Como era de esperar, de nada sirve un load balancer si no asignamos un grupo de instancias las cuales debe gestionar su tráfico. En EC2, esto se llama *Target Group*, el cual se debe configurar para incluir en él las instancias que necesitemos y asociarlo al *load balancer*.

10

▼ Listener HTTP:80

Remove

Protocol

HTTP ▼

Port

80

1-65535

Default action

Info

Forward to

WebServerTargetGroup

HTTP ▼

Target type: Instance, IPv4

↺

Create target group

Listener tags - optional

Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add listener tag

You can add up to 50 more tags.

Figura 15: Escucha por el puerto 80

Por último, se especifica que cualquier petición que llegue por el puerto 80, debe ser tratada por el Load Balancer, haciendo uso del Target Group creado para saber a qué instancia debe enrutar la petición. Una vez creado, podremos acceder al panel de control del mismo y acceder a su nombre en el DNS.

Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monitoring
WebServerLoadBalancer	WebServerLoadBalancer-189...	Provisioning	vpc-088feb7f1494a9d6b	us-east-1b, us-east-1c	application	October 20, 2022 at 12:05:04 ...	

Load balancer: WebServerLoadBalancer

Description

Listeners

Monitoring

Integrated services

Tags

Basic Configuration

Name

WebServerLoadBalancer

ARN

am:aws:elasticloadbalancing:us-east-1:340954026236:loadbalancer/app/WebServerLoadBalancer/aab7886ac44b63fb

DNS name

WebServerLoadBalancer-1896943446.us-east-1.elb.amazonaws.com (A Record)

State

Provisioning

Type

application

Scheme

internet-facing

IP address type

ipv4

Edit IP address type

VPC

vpc-088feb7f1494a9d6b

Availability Zones

subnet-00db343d4ea75c3f8 - us-east-1b

IPv4 address: Assigned by AWS

subnet-067218daac8437c2d - us-east-1c

IPv4 address: Assigned by AWS

Edit subnets

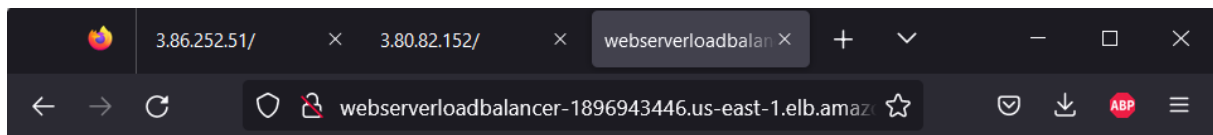
Hosted zone

Z35SXDOTRQ7X7K

Creation time

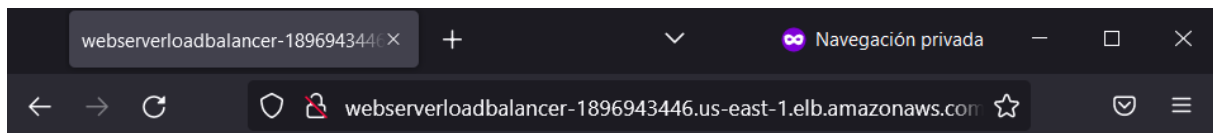
October 20, 2022 at 12:05:04 PM UTC+1

Figura 16: Load Balancer creado



El servidor de Raúl: 2

Figura 17: Prueba de funcionamiento del load balancer 1



El servidor de Raúl: 1

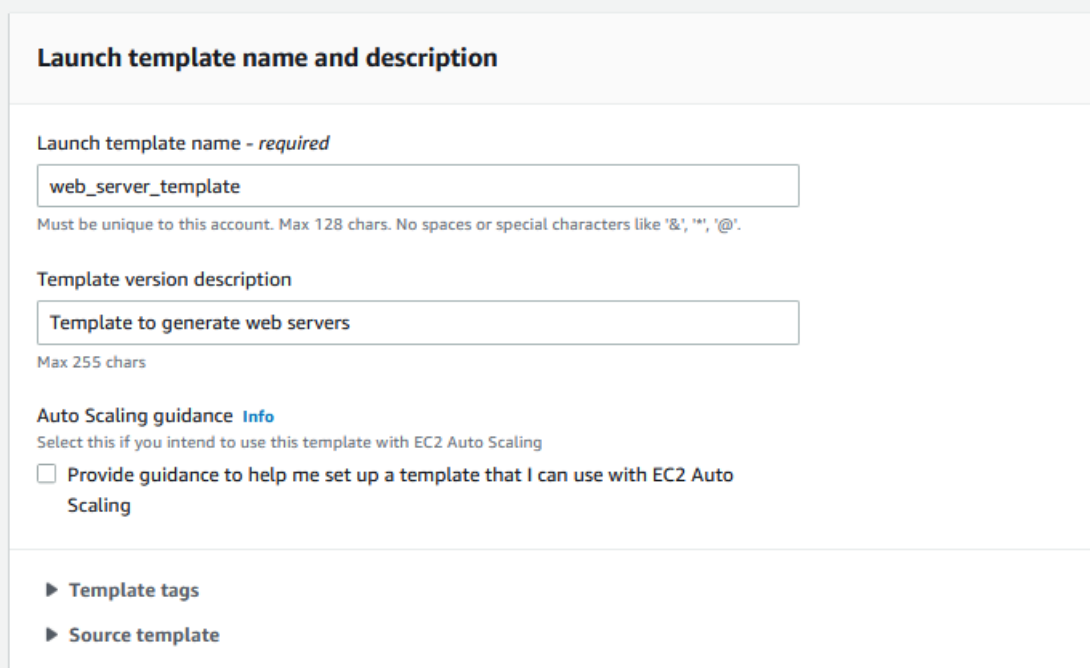
Figura 18: Prueba de funcionamiento del load balancer 2

Para probar el funcionamiento del load balancer una vez este esté desplegado, no hay más que llevar su *DNS name* al navegador y actualizar la página algunas veces, para ver como unas veces nos mostrará el servidor 1 y otras el servidor 2, lo que nos dice que el balanceo de carga está funcionando correctamente.

3. Prepara un “template” de instancia para EC2 para generar servidores web. Con el “template” declarar un “Auto-Scaling Group”(ASG) que tenga como mínimo una instancia y como máximo 2. El ASG debe añadirse al “load balancer” previamente desplegado. Comprueba que el ASG mantiene al menos una instancia viva y que el “load balancer” le manda peticiones entrantes

Para generar servidores web sin tener que estar constantemente creando y configurando instancias, podemos generar un “*template*” que lo haga por nosotros. Para ello, desde el apartado *Launch template*, podemos comenzar a desarrollar nuestra plantilla.

En primer lugar, como configuración básica, se le dará un nombre y una descripción, tal y como se ve en la **Figura 19**: Configuración básica del template.



The screenshot shows the 'Launch template name and description' form in the AWS console. It includes a text input for the launch template name, a description input, and a checkbox for 'Provide guidance to help me set up a template that I can use with EC2 Auto Scaling'. Below the form are expandable sections for 'Template tags' and 'Source template'.

Launch template name and description

Launch template name - *required*

web_server_template

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description

Template to generate web servers

Max 255 chars

Auto Scaling guidance [Info](#)

Select this if you intend to use this template with EC2 Auto Scaling

☐ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

► Template tags

► Source template

Figura 19: Configuración básica del template

En consecuencia, crear un template es prácticamente idéntico que crear una instancia, por lo que deberemos asignar una imagen, en este caso, Amazon Linux, así como el tipo de instancia (t2.micro en su versión *free tier*), un par de claves y un grupo de seguridad, el cual emplearemos el mismo que hemos empleado tanto para instancias anteriores como para el Load balancer, pues es justo lo que necesitamos para permitir peticiones por el puerto 80 al servidor web.

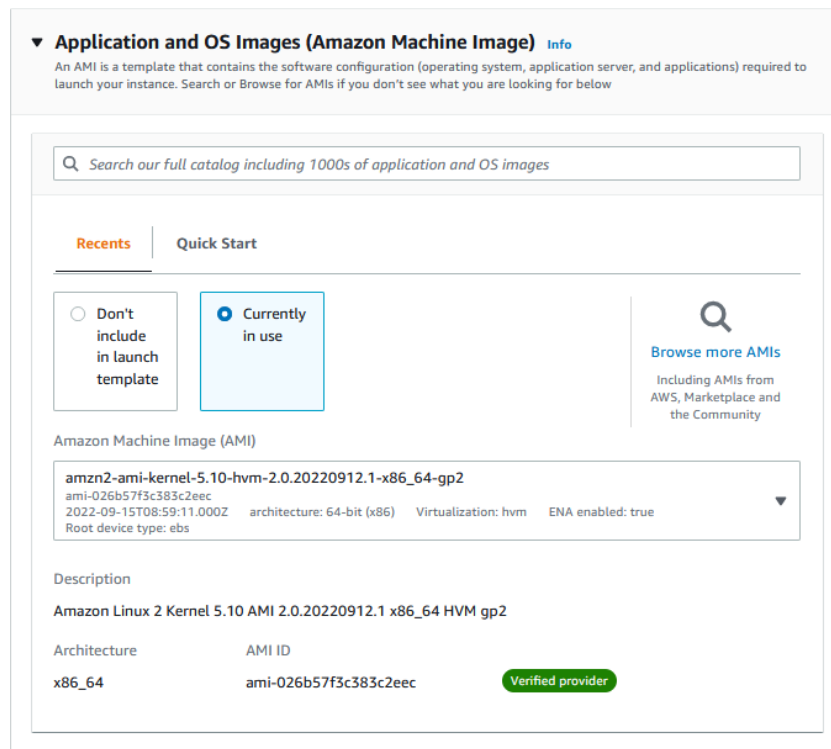


Figura 20: Amazon Linux como imagen de las instancias creadas por el template

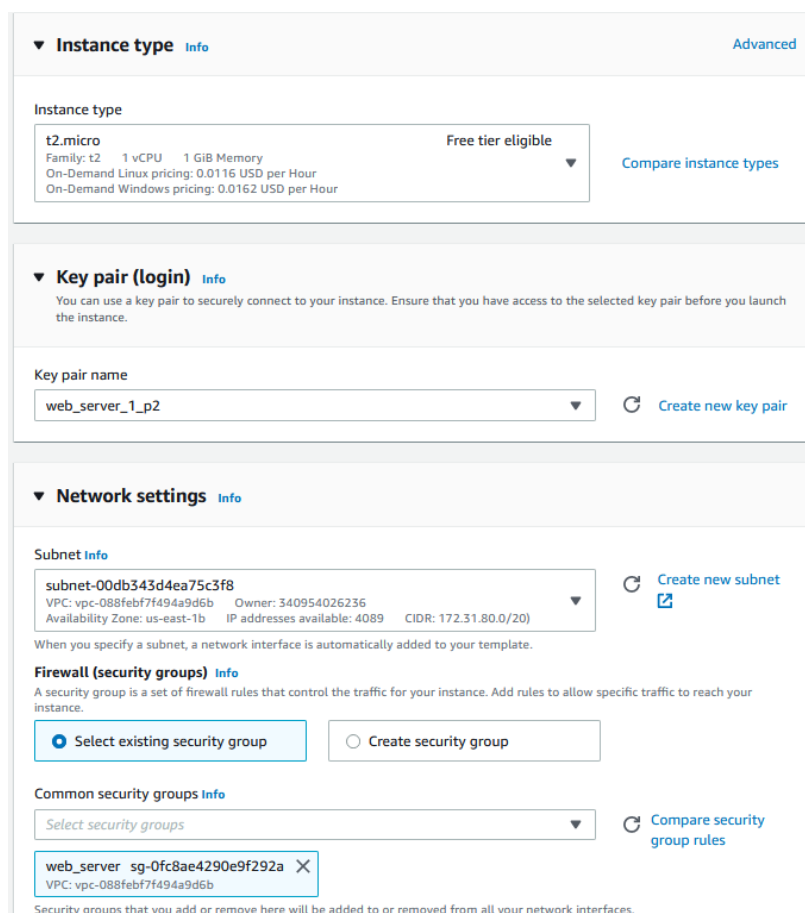
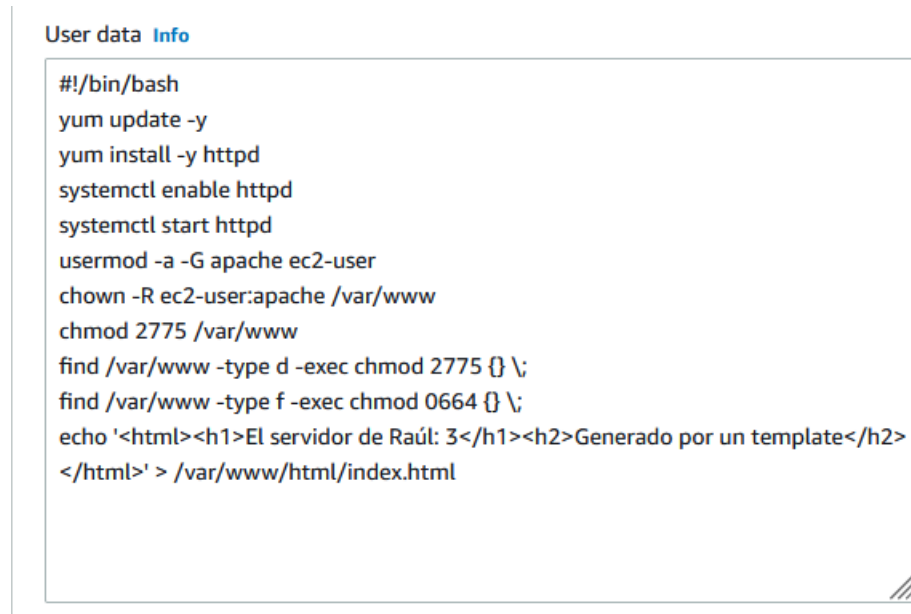


Figura 21: Tipo de instancia, par de claves y grupo de seguridad del template

En lo que si cambia la creación de este template de las instancias que hemos generado anteriormente, es que para no estar instalando y configurando siempre el servidor web, en el apartado User data justo al final en opciones avanzadas, introducimos el pequeño script apreciable en la **Figura 22**: Script para instalar y configurar el servidor web.

En él, se realizan todas las actualizaciones pertinentes para a continuación instalar e iniciar el servicio Apache, así como conceder los permisos necesarios y configurar la página principal del servidor para diferenciarlo de las demás instancias.



```
User data Info
#!/bin/bash
yum update -y
yum install -y httpd
systemctl enable httpd
systemctl start httpd
usermod -a -G apache ec2-user
chown -R ec2-user:apache /var/www
chmod 2775 /var/www
find /var/www -type d -exec chmod 2775 {} \;
find /var/www -type f -exec chmod 0664 {} \;
echo '<html><h1>El servidor de Raúl: 3</h1><h2>Generado por un template</h2></html>' > /var/www/html/index.html
```

Figura 22: Script para instalar y configurar el servidor web

Una vez creado el *template*, podemos desplegar un *Auto-Scaling Group*, el cual deberá tener siempre al menos una instancia viva generada por el *template* y que el *load balancer* le reenvíe peticiones entrantes. Para crearlo, como en cualquier cosa, nos solicita un nombre y el template que queremos emplear para generar instancias.

Step 1

Choose launch template or configuration

Step 2

Choose instance launch options

Step 3 (optional)

Configure advanced options

Step 4 (optional)

Configure group size and scaling policies

Step 5 (optional)

Add notifications

Step 6 (optional)

Add tags

Step 7

Review

Choose launch template or configuration [Info](#)

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

Name

Auto Scaling group name

Enter a name to identify the group.

ASG_Web_Server

Must be unique to this account in the current Region and no more than 255 characters.

Launch template [Info](#)

[Switch to launch configuration](#)

Launch template

Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

web_server_template

[Create a launch template](#)

Version

Default (1)

[Create a launch template version](#)

Figura 23: Creación del ASG

Network [Info](#)

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC

Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-088febf7f494a9d6b

172.31.0.0/16 Default

[Create a VPC](#)

Availability Zones and subnets

Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets

us-east-1b | subnet-00db343d4ea75c3f8

172.31.80.0/20 Default

us-east-1c | subnet-067218daac8437c2d

172.31.16.0/20 Default

[Create a subnet](#)

Instance type requirements [Info](#)

[Override launch template](#)

You can keep the same instance attributes or instance type from your launch template, or you can choose to override the launch template by specifying different instance attributes or manually adding instance types.

Launch template	Version	Description
web_server_template	Default	Template to generate web servers
lt-0fc7d277e5d53e527		
Instance type		
t2.micro		

Figura 24: Configuración de Red

16

Para la configuración de red, debemos elegir las zonas disponibles y subredes en las cuales el ASG puede generar nuevas instancias. Tratando por anticipado de evitar problemas con el *load balancer*, seleccionamos las mismas subredes que este.

Configure advanced options [Info](#)

Choose a load balancer to distribute incoming traffic for your application across instances to make it more reliable and easily scalable. You can also set options that give you more control over health check replacements and monitoring.

Load balancing - optional [Info](#)

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

☐ **No load balancer**
Traffic to your Auto Scaling group will not be fronted by a load balancer.

☒ **Attach to an existing load balancer**
Choose from your existing load balancers.

☐ **Attach to a new load balancer**
Quickly create a basic load balancer to attach to your Auto Scaling group.

Attach to an existing load balancer
Select the load balancers that you want to attach to your Auto Scaling group.

☒ **Choose from your load balancer target groups**
This option allows you to attach Application, Network, or Gateway Load Balancers.

☐ **Choose from Classic Load Balancers**

Existing load balancer target groups
Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups ▼

WebServerTargetGroup | HTTP ✕
Application Load Balancer: WebServerLoadBalancer

Figura 25: Selección del load balancer

En la propia configuración del ASG permite asociarlo con un *load balancer* ya existente, por lo que en este caso se asocia ya con el que creamos anteriormente y se selecciona el *Target Group* creado para que, cualquier instancia creada por el ASG, se incluya en este y el load balancer pueda redirigirle peticiones entrantes.

Configure group size and scaling policies [Info](#)

Set the desired, minimum, and maximum capacity of your Auto Scaling group. You can optionally add a scaling policy to dynamically scale the number of instances in the group.

Group size - *optional* [Info](#)

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity

1

Minimum capacity

1

Maximum capacity

2

Scaling policies - *optional*

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. [Info](#)

☐ Target tracking scaling policy
Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

☒ None

Figura 26: Tamaño del ASG

Para que al menos siempre exista una instancia viva, en la capacidad deseada se debe seleccionar 1, y se establece el mínimo y máximo de instancias en 1 y 2 respectivamente, para cumplir con las especificaciones de la práctica. Con esto se finalizaría la configuración del ASG y solo se deberá esperar a que esté en funcionamiento para ver si el *load balancer* envía peticiones a las instancias creadas por el ASG.

EC2 > Auto Scaling groups

Auto Scaling groups (1) [Info](#)

<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max
<input type="checkbox"/>	ASG_Web_Server	web_server_template Version Default	1	-	1	1	2

Buttons: Refresh, Edit, Delete, Create an Auto S...

Figura 27: ASG creado

Instances (1/4) Info

Find instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
web_server_1...	i-0fbb74fc74ab23044	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-34-238-53-252.co...	34.238.53.252	-
-	i-0933607ee74f00083	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-18-208-168-99.co...	18.208.168.99	-
SSH_gate	i-02b70a13521c15a9f	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-3-91-248-40.comp...	3.91.248.40	-
web_server_2...	i-068f44fd97591e897	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-34-203-35-115.co...	34.203.35.115	-

Instance: i-0933607ee74f00083

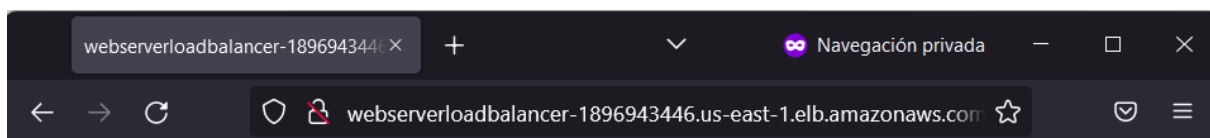
Details Security Networking Storage Status checks Monitoring Tags

▼ Instance summary Info

Instance ID i-0933607ee74f00083	Public IPv4 address 18.208.168.99 open address	Private IPv4 addresses 172.31.85.187
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-18-208-168-99.compute-1.amazonaws.com open address
Hostname type IP name: ip-172-31-85-187.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-85-187.ec2.internal	Elastic IP addresses -
Answer private resource DNS name -	Instance type t2.micro	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 18.208.168.99 [Public IP]	VPC ID vpc-088feb7f7494a9d6b	Auto Scaling Group name ASG Web Server
IAM Role -	Subnet ID	

Una vez inicializado, se crea automáticamente una instancia. Es por ello, que al acceder al DNS name del load balancer desde un navegador, si enviamos varias peticiones debería en alguna respondernos esta nueva instancia, como se aprecia en la **Figura 28**: ASG integrado correctamente en el Load Balancer.

De esta manera, finaliza el desarrollo de la práctica.

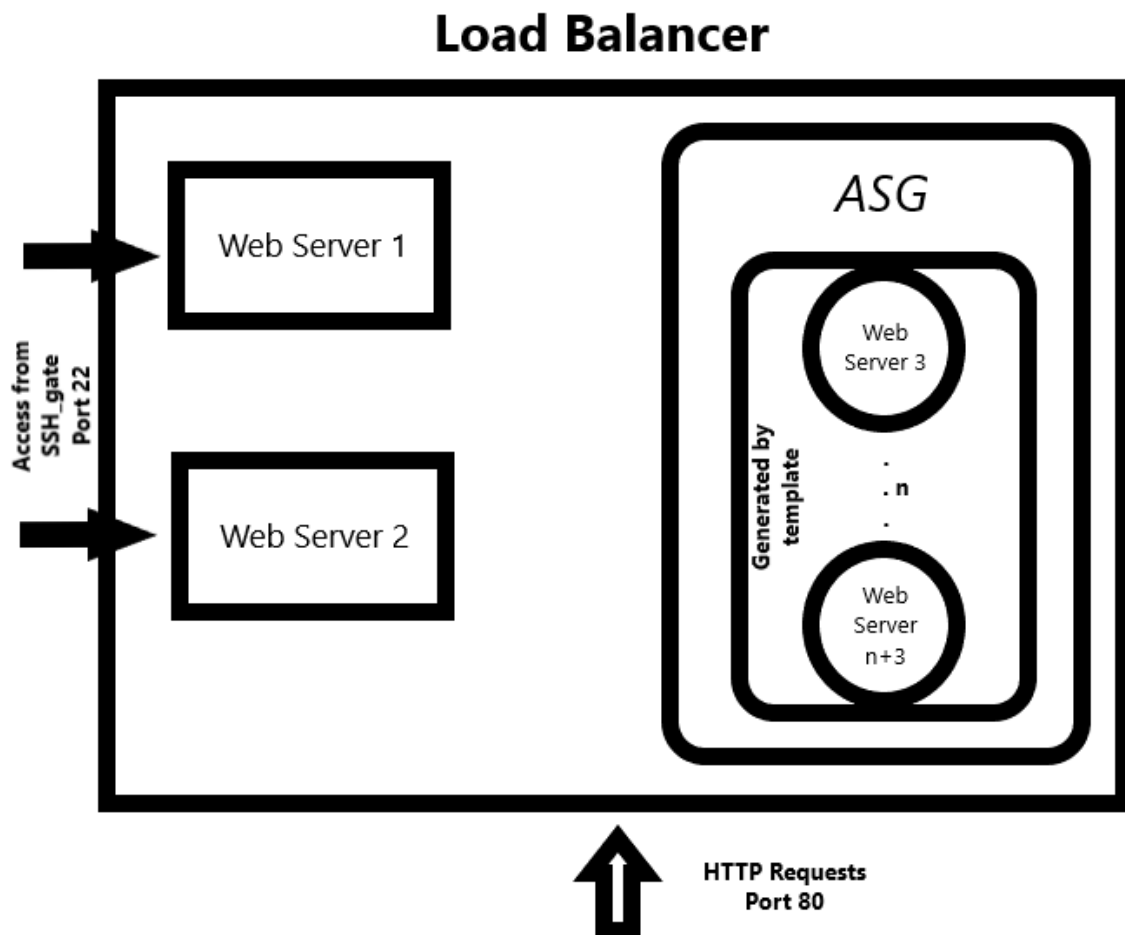


El servidor de Raúl: 3

Generado por un template

Figura 28: ASG integrado correctamente en el Load Balancer

Diagrama de arquitectura desplegada



La arquitectura desplegada corresponde a un *load balancer* que distribuye las peticiones HTTP a partes iguales entre las instancias que forman parte del Target Group asociado al *Load Balancer*.

Por una parte, tenemos las dos instancias del apartado 1, las cuales son accedidas mediante la instancia *SSH_gate*, creada en la anterior práctica. Por otra parte, el ASG creado se asocia con el load balancer para que las instancias que cree, ya configuradas como servidores web, se incluyan en el target group y puedan recibir peticiones entrantes por el puerto 80.

Presupuesto y estimación de gasto de los recursos desplegados

Entendiendo el presupuesto como el dinero que tenemos para montar y desarrollar esta práctica, se puede decir que este es del 100\$. Ahora bien, siendo realistas, durante el transcurso de la asignatura se realizarán 8 prácticas, donde la práctica 1 se estimó su gasto en aproximadamente 1\$, por lo que tendríamos 99\$ a repartir entre 7 prácticas restantes, siendo 14,14\$ de presupuesto para cada práctica.

Para la estimación de gasto, hay que tener en cuenta sobre todo el precio de un *Elastic Load Balancer*, en su versión *Application Load Balancer*. Por hora, nos encontramos con un precio de 0.0225\$, mientras que por cada instancia en ejecución el precio será de 0.0116\$ cada una por hora. Además, el uso de volúmenes SSD de uso general (gp2) cuesta 0,10\$ por GB-mes¹.

Es por todo ello que, teniendo en cuenta que el desarrollo normal de la práctica debería durar unas dos semanas aproximadamente, aunque puede ser menos o más, y que el load balancer siempre está activo dando el servicio al exterior, la estimación de gasto en esta práctica quedaría tal que:

$$(0.0225 + 3 * (0.0116)) * 336 \text{ horas} + 0.10 * 14 \text{ días} = 20.65\$$$

Nuestra estimación de gasto máximo será de 20.65\$, en un caso en el que tanto el *load balancer* como las instancias estén las 24 horas del día, durante dos semanas, siempre en ejecución. Estimando que, para completar la práctica en dos semanas, cada día estén activas como mucho 1 hora, la estimación de gastos quedaría tal que:

$$\begin{aligned} 0.0225 * 336 \text{ horas} + (3 * 0.0116) * 14 \text{ horas} + 0.10 * 14 \text{ días} = \\ = 7.56 + 0.4872 + 1.4 = 9.4472\$ \end{aligned}$$

Ya en el caso más realista, la práctica se ha completado en menos de una semana, por lo que, aproximando justamente 7 días para su desarrollo, la estimación más precisa sería de **4,72\$**, justo la mitad.

En casos reales donde el *load balancer* debiera estar siempre dando servicio, mensualmente saldría a $0.0225 * 744 = 16,74\$$, sin contar cuantas instancias están asignadas a su *target group* ni el precio de los volúmenes de esas instancias, pero es evidente que saldría mucho más cara que cualquiera de las estimaciones desarrolladas.

¹ <https://aws.amazon.com/es/premiumsupport/knowledge-center/ebs-volume-charges/>

Conclusiones

Una vez concluida la práctica, se ha adquirido el conocimiento necesario para desplegar un balanceador de carga con vistas a dar servicios públicos, en este caso, un servidor web que puede recibir peticiones en cualquier instancia del *load balancer*.

Otro de los puntos clave aprendidos es el uso del “template” y como puede ser de gran utilidad y eficacia a la hora de generar nuevas instancias a partir de un modelo, ahorrando al encargado el tener que configurar el servicio a prestar y realizar un mantenimiento.

En definitiva, se han visto herramientas enfocadas en dar servicios al exterior, combinando los fundamentos antiguos con las nuevas tecnologías y creando infraestructuras nunca vistas y más fáciles de crear que nunca.