



## 175- [JAWS] - Lab - Uso de Auto Scaling en AWS (Linux)

### Datos Generales:

**Nombre:** Tomás Alfredo Villaseca Constantinescu

**País:** Chile

**Fecha:** 21/10/2023

**Contacto:** [tomas.villaseca.c@gmail.com](mailto:tomas.villaseca.c@gmail.com)

Después de completar este laboratorio, usted será capaz de hacer lo siguiente:

- Crear una instancia EC2 mediante un comando de la CLI de AWS.
- Crear una nueva AMI mediante la CLI de AWS.
- Crear una plantilla de lanzamiento de Amazon EC2.
- Crear una configuración de lanzamiento de Amazon EC2 Auto Scaling.
- Configurar políticas de escalado y crear un grupo de Auto Scaling para aumentar y reducir el número de servidores en función de una carga variable.

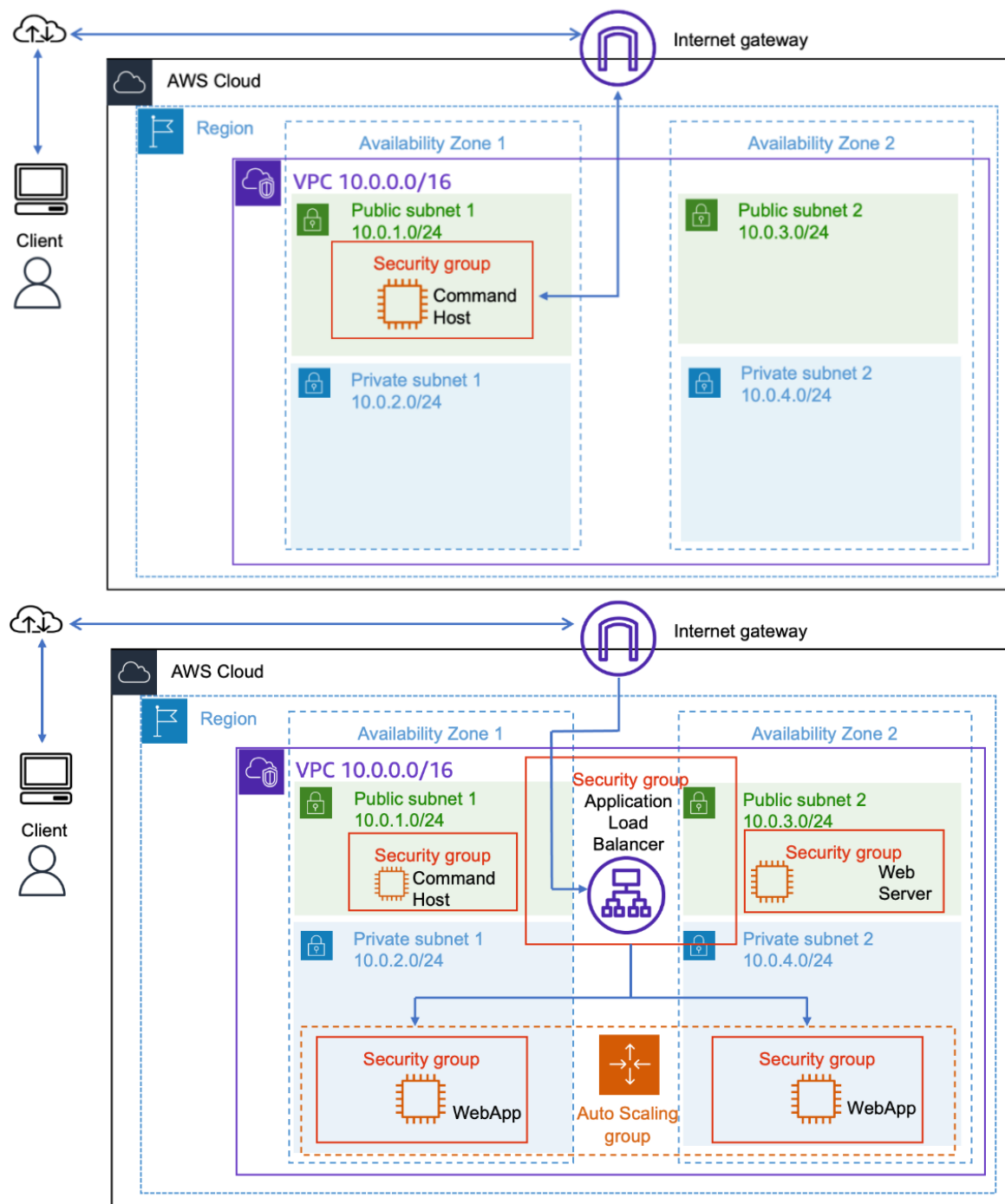
## Resumen Laboratorio:

En este laboratorio, utilizará la AWS CLI para crear una instancia EC2 para alojar un servidor web y crear una AMI a partir de dicha instancia.

Utilizará esa AMI como base para lanzar un sistema que se escala automáticamente bajo una carga variable utilizando Amazon EC2 Auto Scaling.

También creará un Elastic Load Balancer para distribuir la carga entre las instancias EC2 creadas en varias zonas de disponibilidad mediante la configuración de escalado automático.

Arquitectura Inicial y final:



Tarea 1: Crear una nueva AMI para  
Amazon EC2 Auto Scaling



En esta tarea, lanzará una nueva instancia EC2 y luego creará una nueva AMI basada en esa instancia en ejecución.

## 1.1 – Conectarse a la instancia Command Host

En esta tarea, se utiliza EC2 Instance Connect para conectarse a la instancia Command Host que se creó cuando se aprovisionó el laboratorio.

Esta instancia se utiliza para ejecutar comandos de AWS CLI.

**Paso 1:** AWS Management Console → Search → EC2 → Instances → Command Host

Instances (1/1) <a href="#">Info</a>				
<input type="text" value="Find instance by attribute or tag (case-sensitive)"/>				
<input checked="" type="checkbox"/>	Name ▾	Instance ID	Instance state ▾	Instance type ▾
<input checked="" type="checkbox"/>	Command Host	i-085f4a2ab68fceab2	<span>✓</span> Running  	t3.medium

## Paso 2: Command Host → Connect → EC2 Instance Connect

# Connect to instance [Info](#)

Connect to your instance i-085f4a2ab68fceab2 (Command Host) using any of these options


EC2 Instance Connect

Session Manager

SSH client

EC2 serial console

Instance ID


 i-085f4a2ab68fceab2 (Command Host)

Connection Type

☒ Connect using EC2 Instance Connect  
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.


☐ Connect using EC2 Instance Connect Endpoint  
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address

 35.91.94.82

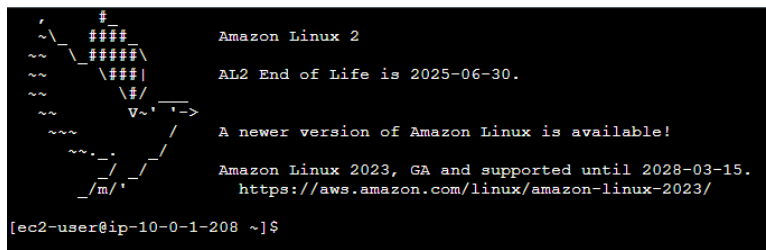
User name

Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ec2-user.

 **Note:** In most cases, the default user name, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Cancel

Connect



## 1.2 – Configurar AWS CLI en Command Host

La AWS CLI está preconfigurada en la instancia Command Host.

**Paso 1:** Para confirmar que la región en la que se está ejecutando la instancia del host de comandos es la misma que la del laboratorio (us-west-2), ejecute el siguiente comando:

```
curl http://169.254.169.254/latest/dynamic/instance-identity/document | grep region
```

```
[ec2-user@ip-10-0-1-208 ~]$ curl http://169.254.169.254/latest/dynamic/instance-identity/document | grep region
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %             Dload  Upload  Total   Spent    Left   Speed
100  475    100  475    0    0   277k      0  --:--:--  --:--:--  --:--:--  463k
"region" : "us-west-2",
[ec2-user@ip-10-0-1-208 ~]$
```

**Paso 2:** Para actualizar AWS CLI con las credenciales correctas utilizar el comando **aws configure**

```
aws configure
```

- AWS Access Key ID → Apretar Enter
- AWS Secret Access Key → Apretar Enter
- Default region name = us-west-2
- Default output format = json

```
[ec2-user@ip-10-0-1-208 ~]$ aws configure
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [us-west-2]: us-west-2
Default output format [None]: json
[ec2-user@ip-10-0-1-208 ~]$
```

**Paso 3:** Ejecutar el siguiente comando para acceder al directorio donde se encuentran el script:

```
cd /home/ec2-user/
```

```
[ec2-user@ip-10-0-1-208 ~]$ cd /home/ec2-user/
[ec2-user@ip-10-0-1-208 ~]$ pwd
/home/ec2-user
[ec2-user@ip-10-0-1-208 ~]$ ls
UserData.txt
[ec2-user@ip-10-0-1-208 ~]$
```

## 1.3 – Crear una nueva instancia EC2

En esta tarea, se utilizará AWS CLI para crear una nueva instancia que aloje un servidor web.

**Paso 1:** Inspeccionar el script UserData.txt

```
more UserData.txt
```

```
[ec2-user@ip-10-0-1-208 ~]$ more UserData.txt
#!/bin/bash
yum update -y --security
amazon-linux-extras install epel -y
yum -y install httpd php stress
systemctl enable httpd.service
systemctl start httpd
cd /var/www/html
wget http://aws-tc-largeobjects.s3.amazonaws.com/CUR-TF-100-TULABS-1/10-lab-autoscaling-linux/s3/ec2-stress.zip
unzip ec2-stress.zip

echo 'UserData has been successfully executed. ' >> /home/ec2-user/result
find -wholename /root/.history -wholename /home/*/.history -exec rm -f {} \;
find / -name 'authorized_keys' -exec rm -f {} \;
rm -rf /var/lib/cloud/data/scripts/*
[ec2-user@ip-10-0-1-208 ~]$
```

Este script realiza una serie de tareas de inicialización, incluyendo la actualización de todo el software instalado en la caja y la instalación de una pequeña aplicación web PHP que puede utilizar para simular una alta carga de CPU en la instancia.

**Paso 2:** Ejecutar el siguiente comando reemplazando los valores correspondientes por los entregados en los detalles del laboratorio para lanzar una nueva instancia EC2.

- AMIID = ami-0025f0db847eb6254
- HTTPACCESS = sg-07975c12874a8573a
- KEYNAME = vockey
- SUBNETID = subnet-0bd28d44407aae936

```
aws ec2 run-instances --key-name KEYNAME --instance-type t3.micro --image-id AMIID --user-data file:///home/ec2-user/UserData.txt --security-group-ids HTTPACCESS --subnet-id SUBNETID --associate-public-ip-address --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=WebServer}]' --output text --query 'Instances[*].InstanceId'
```

```
[ec2-user@ip-10-0-1-208 ~]$ aws ec2 run-instances
73a --subnet-id subnet-0bd28d44407aae936 --associ
i-0e9482d229a814b32
[ec2-user@ip-10-0-1-208 ~]$
```

La salida de este comando le proporciona un InstanceId.

- Instance ID = i-0e9482d229a814b32

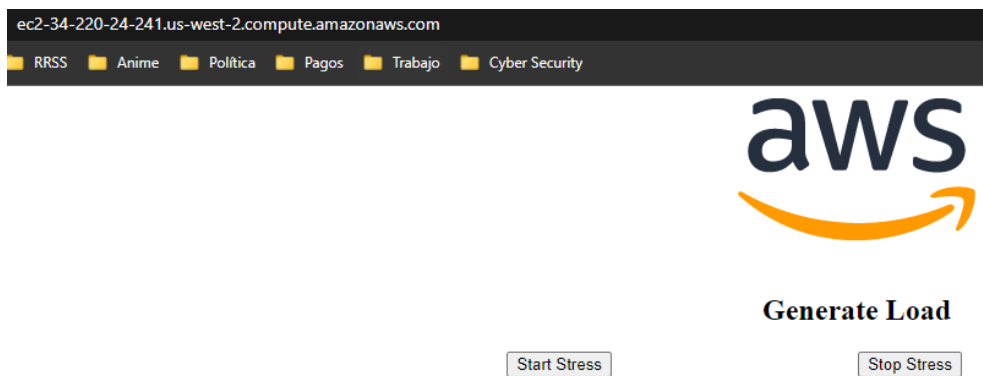
**Paso 2:** Para obtener el Public DNS name ejecutar el siguiente comando:

```
aws ec2 describe-instances --instance-id NEW-INSTANCE-ID --query  
'Reservations[0].Instances[0].NetworkInterfaces[0].Association.PublicDnsName'
```

```
[ec2-user@ip-10-0-1-208 ~]$ aws ec2 describe-instances --i  
"ec2-34-220-24-241.us-west-2.compute.amazonaws.com"  
[ec2-user@ip-10-0-1-208 ~]$
```

Public DNS name = ec2-34-220-24-241.us-west-2.compute.amazonaws.com

**Paso 4:** Copiar Public DNS name en una nueva pestaña de navegador web.



## 1.4 – Crear una nueva instancia EC2

En esta tarea, creará una nueva AMI basada en la instancia que se acaba de crear.

**Paso 1:** Ejecutar el siguiente comando para crear una nueva AMI:

- Reemplazar NEW-INSTANCE-ID por Instance ID obtenido anteriormente.

```
aws ec2 create-image --name WebServerAMI --instance-id NEW-INSTANCE-ID
```

Por defecto, el comando `aws ec2 create-image` reinicia la instancia actual antes de crear la AMI para asegurar la integridad de la imagen en el sistema de archivos.

```
[ec2-user@ip-10-0-1-208 ~]$ aws ec2 create-image --name WebServerAMI --instance-id i-0e9482d229a814b32  
{  
  "ImageId": "ami-021504398952a42d3"  
}  
[ec2-user@ip-10-0-1-208 ~]$
```

- AMI ID = ami-021504398952a42d3



## Tarea 2: Crear un entorno de Auto Scaling

En esta tarea, se creará un equilibrador de carga que agrupa un grupo de instancias EC2 bajo una única dirección DNS.

Utilizará el escalado automático para crear un grupo escalable dinámicamente de instancias EC2 basado en la AMI que creó en la tarea anterior.

Por último, creará un conjunto de alarmas que reduzcan o aumenten el número de instancias de su grupo de equilibradores de carga siempre que el rendimiento de la CPU de cualquier máquina del grupo supere o quede por debajo de un conjunto de umbrales especificados.

### 2.1 – Crear un Application Load Balancer

En esta tarea, se creará un equilibrador de carga que puede equilibrar el tráfico entre varias instancias EC2 y zonas de disponibilidad.

**Paso 1:** EC2 → Load Balancing → Load Balancers → Create Load Balancer

▼ Load Balancing

Load Balancers

Target Groups



Actions ▼

Create load balancer



**Paso 2:** Create Load Balancer → Load Balancer Types

- Application Load Balancer → Create

#### Load balancer types

Application Load Balancer [Info](#)

Network Load Balancer [Info](#)

Gateway Load Balancer [Info](#)

**Paso 3:** Create Load Balancer → Basic Configuration

- Load Balancer Name = WebServerELB

#### Basic configuration

Load balancer name

Name must be unique within your AWS account and can't be changed after the load balancer is created.

WebServerELB

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

### Paso 4: Create Load Balancer → Network Mapping

- VPC → Lab VPC
- Mappings → Seleccionar ambas AZs disponibles.
- AZ 1 → Public Subnet 1
- AZ 2 → Public Subnet 2

## 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 or you can [create a new VPC](#). Only VPCs with an internet gateway are enabled for selection. The selected VPC can't be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

Lab VPC  
vpc-0a9273ae058a83816  
IPv4: 10.0.0.0/16



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-west-2a (usw2-az1)**

Subnet

subnet-0b6a42a78f593a07cPublic Subnet 1 ▼

IPv4 address

Assigned by AWS

☒ **us-west-2b (usw2-az2)**

Subnet

subnet-0bd28d4407aae936Public Subnet 2 ▼

IPv4 address

Assigned by AWS

### Paso 5: Create Load Balancer → Security Groups

- Desplegar lista → HTTPAccess


## Security groups [Info](#)

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

## Security groups

Select up to 5 security groups 



HTTPAccess   
sg-07975c12874a8573a VPC: vpc-0a9273ae058a83816



**Paso 6:** Create Load Balancer → Listeners and routing

- Create target group → Link → Se abre una nueva pestaña del navegador web

▼ Listener HTTP:80

Protocol

HTTP ▼

Port

80

1-65535

Default action

Forward to

Info

Select a target group

Create target group [↗](#)

**Paso 7:** Create target group

- Target type → Instances

Choose a target type

☒ Instances

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#) [↗](#) to manage and scale your EC2 capacity.

- Target group name = webserver-app

Target group name

webserver-app

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

- Health checks → Health check path → /index.php

Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

HTTP ▼

Health check path

Use the default path of "/" to ping the root, or specify a custom path if preferred.

/index-php

Up to 1024 characters allowed.

- Register targets → Create target groups

Cancel

Previous

Create target group

<input checked="" type="checkbox"/>	Name	ARN	Port	Protocol	Target type
<input checked="" type="checkbox"/>	webserver-app	arn:aws:elasticloadbalanci...	80	HTTP	Instance

**Paso 8:** Volver a la pestaña de navegador de Load Balancers.

- Refrescar → Desplegar lista → webserver-app

Protocol	Port	Default action	<a href="#">Info</a>
HTTP ▼	: 80 1-65535	Forward to	webserver-app Target type: Instance, IPv4
HTTP ▼			
<a href="#">Create target group</a>			

**Paso 9:** Create Load Balancer → Create

- View Load Balancer
- Copiar DNS name = WebServerELB-907497419.us-west-2.elb.amazonaws.com

Cancel	Create load balancer
--------	----------------------

<input checked="" type="checkbox"/>	Name ▼	DNS name ▼
<input checked="" type="checkbox"/>	<a href="#">WebServerELB</a>	WebServerELB-90749741...

DNS name [Info](#)

WebServerELB-907497419.us-west-2.elb.amazonaws.com (A Record)

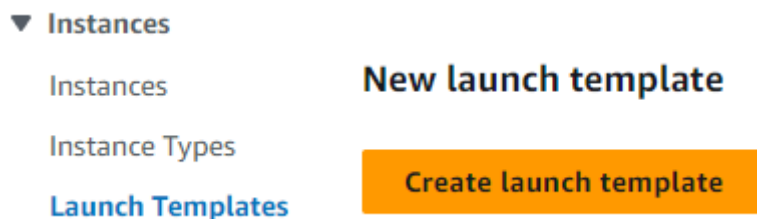
## 2.2 – Crear un Launch Template

En esta tarea, se crea una plantilla de lanzamiento (launch template) para el Auto Scaling Group.

Una plantilla de lanzamiento es una plantilla que un Auto Scaling Group utiliza para lanzar instancias EC2.

Al crear una plantilla de lanzamiento, se especifica información para las instancias, como la AMI, el tipo de instancia, el par de claves, el grupo de seguridad y los discos.

**Paso 1:** EC2 → Instances → Launch Templates → Create Launch Template



**Paso 2:** Create Launch Template → Name and Description

- Launch Template Name = web-app-launch-template
- Template version description = a web server for the load test app.
- Auto Scaling Guidance → Seleccionar casilla asociada.

### Launch template name and description

Launch template name - *required*

web-app-launch-template

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

Template version description

a web server for the load test app

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

### Paso 3: Create Launch Template → Application and OS images

- My AMIs → WebServerAMI

Recents

**My AMIs**

Quick Start

☒ Owned by me

☐ Shared with me

Amazon Machine Image (AMI)

WebServerAMI

ami-021504398952a42d3

2023-10-22T01:09:20.000Z   Virtualization: hvm   ENA enabled: true   Root device type: ebs

### Paso 4: Create Launch Template → Instance type

- Instance type → t3.micro

#### Instance type

**t3.micro**

Family: t3   2 vCPU   1 GiB Memory   Current generation: true

On-Demand SUSE base pricing: 0.0104 USD per Hour

On-Demand Windows base pricing: 0.0196 USD per Hour

On-Demand RHEL base pricing: 0.0704 USD per Hour

On-Demand Linux base pricing: 0.0104 USD per Hour

### Paso 5: Create Launch Template → Key pair (login)

- Desplegar lista → Don't include in launch template

▼ **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

Don't include in launch template

↕ Create new key pair

**Paso 6:** Create Launch Template → Network Settings

- Security Groups → HTTPAccess

**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.


☒ Select existing security group

☐ Create security group

Security groups [Info](#)

Select security groups ▼

HTTPAccess sg-07975c12874a8573a ✕  
VPC: vpc-0a9273ae058a83816

 [Compare security group rules](#)

**Paso 7:** Create Launch Template → Create

- View Launch Template

Cancel

Create launch template

**Launch Templates (1/1)** [Info](#)

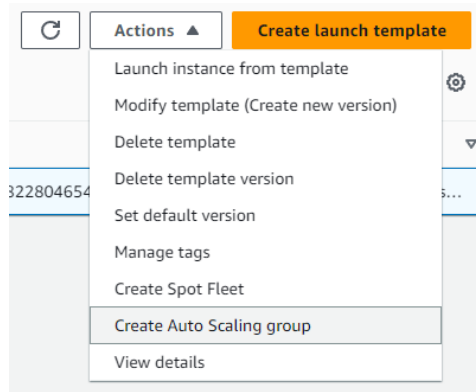
 Search

	Launch Template ID ▼	Launch Template Name
<input checked="" type="radio"/>	lt-01d4d9719cc778ced	web-app-launch-template

## 2.3 – Crear un Auto Scaling Group

En esta tarea, utilizará la launch template para crear un Auto Scaling Group.

**Paso 1:** Launch Templates → web-app-launch-template → Actions → Create Auto Scaling Group



**Paso 2:** Create Auto Scaling Group → Name

- Auto Scaling Group Name = Web App Auto Scaling Group

Auto Scaling group name

Enter a name to identify the group.

Web Auto Scaling Group

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

**Paso 3:** Create Auto Scaling Group → Network

- VPC → Lab VPC
- Availability Zones and subnets → Private Subnet 1 / Private Subnet 2

### 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-0a9273ae058a83816 (Lab VPC)  
10.0.0.0/16

[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-west-2a | subnet-0b39f90647acaf251 (Private Subnet 1)  
10.0.2.0/24

us-west-2b | subnet-0e537cc5521c43cf4 (Private Subnet 2)  
10.0.4.0/24

#### Paso 4: Create Auto Scaling Group → Configure advanced options

- Load balancing → Attach to an existing load balancer

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.

- Existing load balancer target groups → webserver-app | HTTP

☒ 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 ▼

webserver-app | HTTP X  
Application Load Balancer: WebServerELB

- Health checks → Additional health check types → Seleccionar casilla ELB

### Health checks

Health checks increase availability by replacing unhealthy instances. When you use multiple health checks, all are evaluated, and if one fails, instance replacement occurs.

#### EC2 health checks

 Always enabled

#### Additional health check types - optional

[Info](#)

- ☒ Turn on Elastic Load Balancing health checks **Recommended**

Elastic Load Balancing monitors whether instances are available to handle requests. When it reports an unhealthy instance, Auto Scaling can replace it on its next periodic check.



## Paso 5: Create Auto Scaling Group → Configure group size and scaling policies

- Group size → Desired capacity = 2, Minimum capacity = 2, Maximum capacity = 4

### 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

Minimum capacity

Maximum capacity

- Scaling policies → Target tracking scaling policies
- Scaling policies → Metric type → Average CPU utilization
- Scaling policies → Target value = 50

### 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

Scaling policy name

Metric type [Info](#)

Monitored metric that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.



Target value

**Paso 6:** Create Auto Scaling Group → Add tags → Add tag

- Key = Name
- Value = WebApp

**Tags (1)**

Key

Value - optional

Tag new instances

Name

WebApp

☒

Add tag

49 remaining

**Paso 7:** Create Auto Scaling Group → Create

Cancel

Previous

Create Auto Scaling group

Auto Scaling groups (1/1) <a href="#">Info</a>			
<div><div>Q</div><div>Search your Auto Scaling groups</div></div>			
<input checked="" type="checkbox"/>	Name	Launch template/configuration <a href="#">↗</a>	Instances
<input checked="" type="checkbox"/>	Web Auto Scaling Group	web-app-launch-template   Version Defa	2

## Tarea 3: Verificar la configuración de Auto Scaling

En esta tarea, verificará que tanto la configuración de Auto Scaling como el equilibrador de carga funcionan accediendo a un script preinstalado en uno de sus servidores que consumirá ciclos de CPU, lo que invoca la alarma de escalado.

### Paso 1: EC2 → Instances

- Se puede ver 2 nuevas instancias WebApp creadas por Auto Scaling Group.

Instances (2/4) <a href="#">Info</a>						
<input type="text" value="Find instance by attribute or tag (case-sensitive)"/>						
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	
<input type="checkbox"/>	Command Host	i-085f4a2ab68fceab2	Running	t3.medium	2/2 checks passed	
<input checked="" type="checkbox"/>	WebApp	i-01eab0cbfa87601ce	Running	t3.micro	2/2 checks passed	
<input checked="" type="checkbox"/>	WebApp	i-0a7373e3d03b516b2	Running	t3.micro	2/2 checks passed	
<input type="checkbox"/>	WebServer	i-0e9482d229a814b32	Running	t3.micro	2/2 checks passed	

### Paso 2: EC2 → Load balancing → Target Groups → webserver-app

- Targets → Puede ver el status de ambas instancias → Healthy Status.

Target groups (1/1) <a href="#">Info</a>						
<input type="text" value="Filter target groups"/>						
<input checked="" type="checkbox"/>	Name	ARN	Port	Protocol	Target type	
<input checked="" type="checkbox"/>	webserver-app	arn:aws:elasticloadbalanci...	80	HTTP	Instance	

Registered targets (2/3)					Health status
<input type="text" value="Filter targets"/>					
<input checked="" type="checkbox"/>	Instance ID	Name	Port	Zone	
<input checked="" type="checkbox"/>	i-0e9a1e73ad867708a	WebApp	80	us-west-2b	healthy
<input checked="" type="checkbox"/>	i-01eab0cbfa87601ce	WebApp	80	us-west-2a	healthy

## Tarea 4: Probar la configuración de Auto Scaling

**Paso 1:** Abrir una nueva pestaña de navegador y copiar DNS name del Load Balancer.



### Generate Load

Start Stress

Stop Stress

**Paso 2:** En la página web seleccionar “Start Stress”.

## Generating load

**Paso 3:** EC2 → Auto Scaling → Auto Scaling Group → Web App Auto Scaling Group → Activity

- Tras unos minutos se puede ver que su Auto Scaling Group añade una nueva instancia.
- Lo anterior ocurre porque Amazon CloudWatch detectó que la utilización media de la CPU de su grupo de Auto Scaling superó el 50 por ciento y su política de escalado se ha invocado en respuesta.

### Auto Scaling group: Web Auto Scaling Group

#### ▼ Auto Scaling

#### Auto Scaling Groups

✓ Successful

Launching a new EC2  
instance: i-  
0e9a1e73ad867708a

**Paso 4:** Comprobar que las nuevas instancias EC2 se están lanzando en el panel de instancias de EC2.

Instances (3/5) <a href="#">Info</a>						
<input type="text" value="Find instance by attribute or tag (case-sensitive)"/>						
<input type="checkbox"/>	Name ▲	Instance ID	Instance state ▼	Instance type ▼	Status check	
<input type="checkbox"/>	Command Host	i-085f4a2ab68fceab2	✓ Running	t3.medium	✓ 2/2 checks passed	
<input checked="" type="checkbox"/>	WebApp	i-01eab0cbfa87601ce	✓ Running	t3.micro	✓ 2/2 checks passed	
<input checked="" type="checkbox"/>	WebApp	i-0e9a1e73ad867708a	✓ Running	t3.micro	⌚ Initializing	
<input checked="" type="checkbox"/>	WebApp	i-0a7373e3d03b516b2	✓ Running	t3.micro	✓ 2/2 checks passed	
<input type="checkbox"/>	WebServer	i-0e9482d229a814b32	✓ Running	t3.micro	✓ 2/2 checks passed	

Laboratorio Completado

