

TRABAJO DE GRADO
Opción Seminario-Diplomado.

Cloud Fats

Corporación Universitaria Remington.

Ingeniería
Tecnología desarrollo de software

Presentado:

Andres Julian Uribe Correa

Docente:
Juan Pablo Berrio Lopez
Seminario
2024

Tabla de Contenidos

Tabla de contenido

Resumen.....	3
Marco conceptual y contextual	4
AWS.....	4
Entrega 1	5
Instancias y recursos de aws que permite ejecutar instancias de aws.	5
Figuras y tablas	7
Entrega 2	21
Entrega final 3.....	57
Conclusiones	62
Referencias.....	64

Resumen

El siguiente trabajo es implementado con el fin de tener un entorno cloud de alta disponibilidad en aws por medio de instancias y contenedores, para esto se desarrollaron configuraciones de vpc en el cual se configura la estructura de conectividad, ec2 se implementan las instancias de los sistemas operativos, auto Scaling el cual permite tener el número deseado de instancias disponibles al mismo tiempo, Docker con el cual implementaremos los contenedores. entre otras herramientas las cuales dependerán una de la otras logrando así el objetivo de tener un alto rendimiento de disponibilidad.

Palabras clave

Las palabras claves del trabajo son:

Vpc.

Ec2.

Docker.

Contenedores.

Auto Scaling.

Marco conceptual y contextual

Este trabajo es de computaciones en la nube enfocado en aws los cuales son unos servicios de cómputo en la nube con el fin de lograr que estos servicios configurados logren estar la mayor parte del tiempo disponibles a los usuarios.

Para ello se configuran herramientas como vpc que permite configurar un espacio de enlace por medio de direcciones ip y grupos de conectividad, el putty es una aplicación que nos permite conectar por medio de un protocolo SSH a las instancias creamos por medio de ec2 la cual permite crear una instancia que va a tener un sistema operativo en el cual se configuran herramientas como apaches que son servicios que nos permiten tener una página web, nginx que permite tener un proxy reverso de conectividad sobre varios contenedores configurados por medio de la aplicación Docker y sus distintos puertos de red que permiten la navegación sin conflictos.

Tenemos el auto Scaling que permite tener instancias a necesidad de la implementación permitiendo tener de estar forma un sistema disponible para los usuarios en el momento que ellos desean conectar.

AWS

Entrega 1

Instancias y recursos de aws que permite ejecutar instancias de aws.

2. Instancias en Amazon Lightsail

- Diseñadas para ser fáciles de usar y económicas, ideales para usuarios que no requieren la flexibilidad completa de EC2.
- Incluyen una combinación de cómputo, almacenamiento y ancho de banda preconfigurado.

3. Instancias de Contenedores (Amazon ECS y AWS Fargate)

- **Amazon ECS (Elastic Container Service):** Permite ejecutar contenedores en instancias EC2 o mediante el servicio administrado AWS Fargate.
- **AWS Fargate:** Elimina la necesidad de administrar instancias directamente; ejecuta contenedores sin preocuparse por la infraestructura subyacente.

Entre las principales tenemos EC2:

1. Amazon EC2 (Elastic Compute Cloud)

En un servicio principal de aws el ec2 permite un entorno escalable.

Componentes Clave:

- AMIs (Amazon Machine Images): Plantillas preconfiguradas que definen el sistema operativo, software, y configuraciones de las instancias.
- Tipos de Instancia: Especifican los recursos de asignados a una instancia.

2. Almacenamiento

Las instancias y guardar sus aplicaciones y archivos.

- persisten incluso si la instancia se detiene.
- Instancia de Almacenamiento: Almacenamiento temporal vinculado a la instancia y que se pierde al detenerla.
- Amazon S3 almacina objetos

3. Redes

La conectividad de red es fundamental para que las instancias puedan comunicarse con otros recursos de manera eficiente. Dentro de AWS, se utilizan diversas herramientas y configuraciones para gestionar estas conexiones:

- Amazon VPC (Virtual Private Cloud): Permite crear redes virtuales aisladas donde se ejecutan las instancias de manera segura.
- Subredes: Son divisiones dentro de una VPC que organizan las instancias en diferentes zonas de disponibilidad.

4. Auto Scaling y Load Balancing

Para asegurar que las aplicaciones sean altamente disponibles y puedan manejar cambios en la demanda, AWS ofrece las siguientes herramientas:

- **Auto Scaling:** Ajusta automáticamente la cantidad de instancias en función de la carga de trabajo, garantizando eficiencia y optimización de recursos.
- **Elastic Load Balancer (ELB):** Distribuye el tráfico de red entre múltiples instancias, mejorando el rendimiento y la tolerancia a fallos.

2. Implementando un servidor web en Amazon Linux:

Figuras y tablas

Figura 1. Creación y configuración de vpc

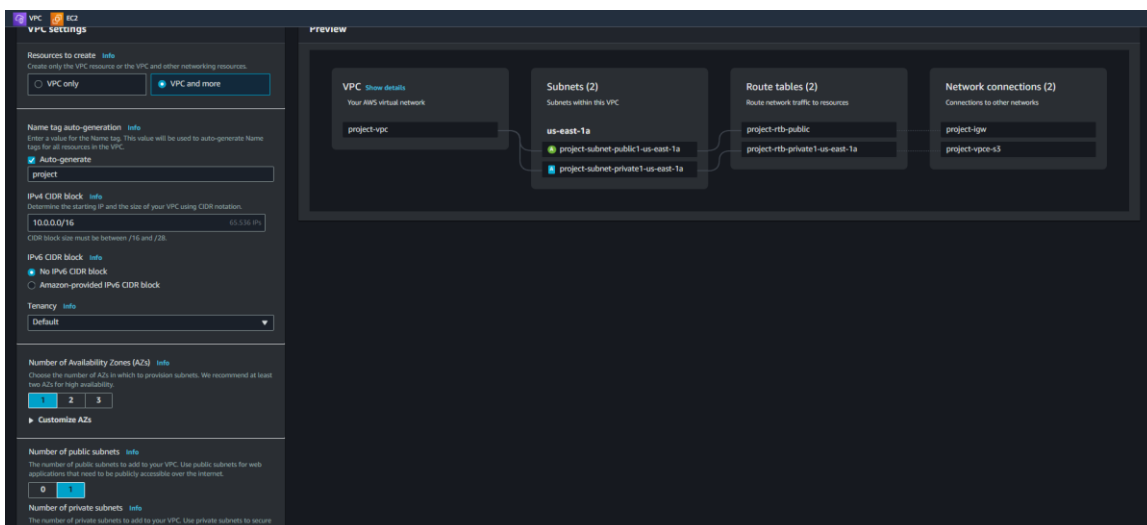


Figura 2. Creamos el certificado

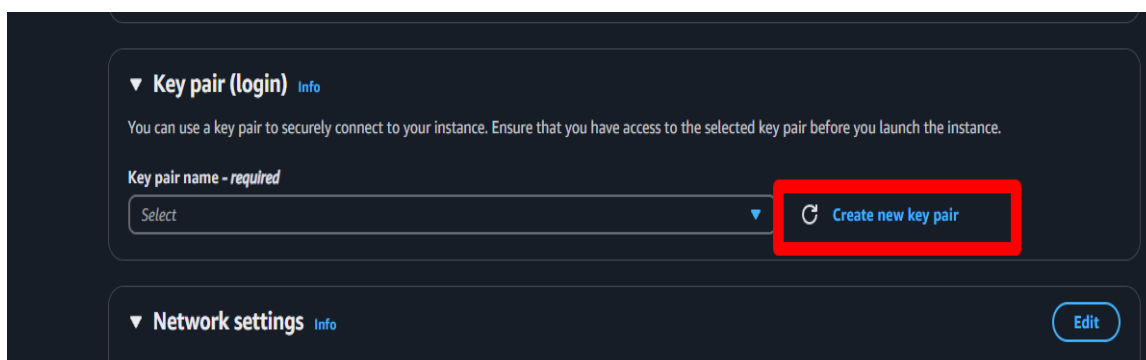
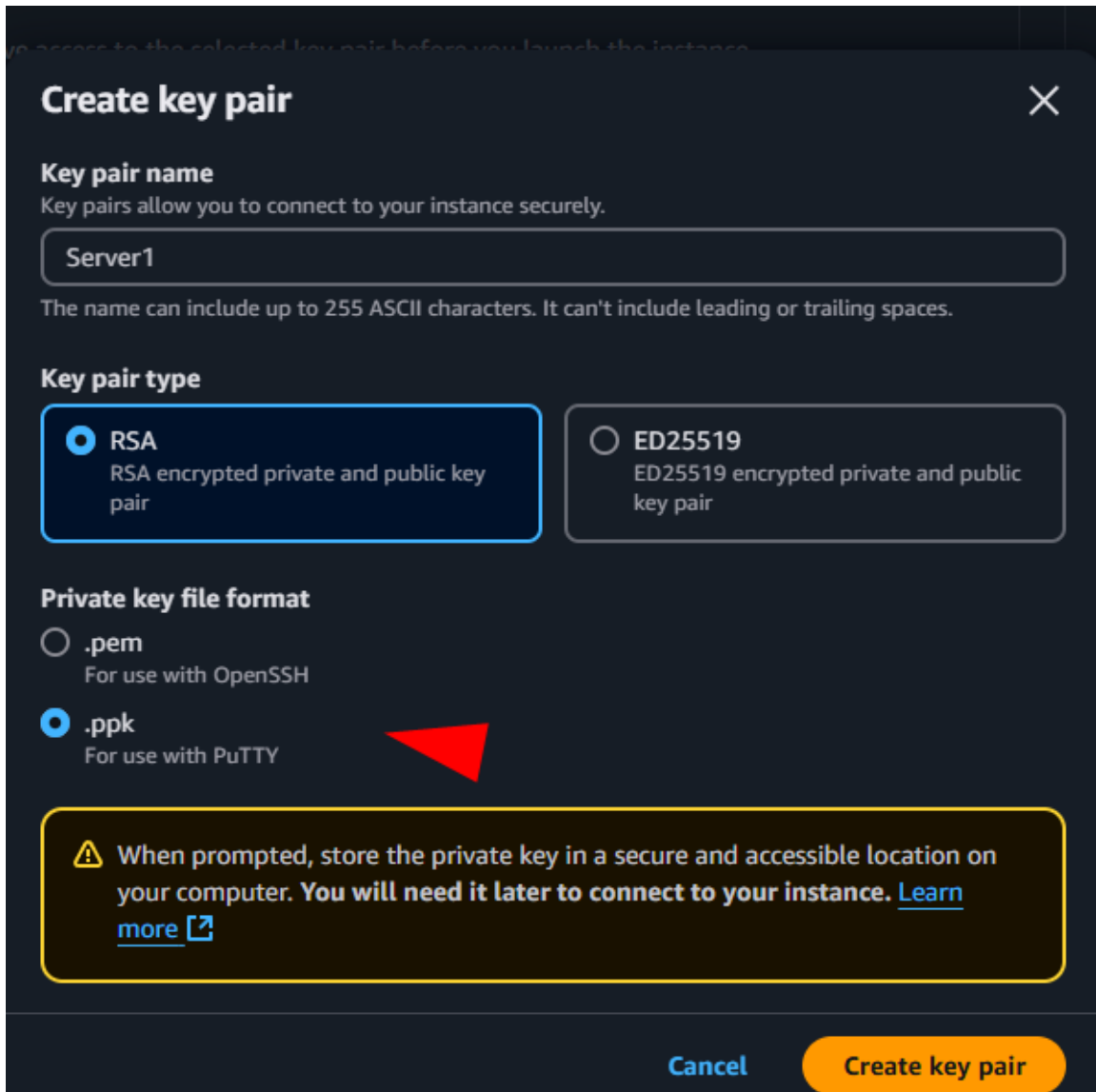


Figura 3. Seleccionamos para conectar por medio de putty



you access to the selected key pair before you launch the instance.

Create key pair

Key pair name
Key pairs allow you to connect to your instance securely.

Server1

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

RSA
RSA encrypted private and public key pair

ED25519
ED25519 encrypted private and public key pair

Private key file format

.pem
For use with OpenSSH

.ppk
For use with PuTTY

Warning: When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Cancel Create key pair

Le damos crear y guardamos el certificado

Figura 4. Configuración de red

▼ Network settings [Info](#)

VPC - required [Info](#)

vpc-0cd34ab243a3c8408 (proyecto-vpc)
10.0.0.0/16 ↕ ↻

Subnet [Info](#)

subnet-0c739260fb4d880dd proyecto-subnet-public1-us-east-1a
VPC: vpc-0cd34ab243a3c8408 Owner: 354918403906
Availability Zone: us-east-1a Zone type: Availability Zone
IP addresses available: 4091 CIDR: 10.0.0.0/20 ↕ ↻ [Create new subnet](#)

Auto-assign public IP [Info](#)

Enable ↕

Additional charges apply when outside of free tier allowance

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

eg_server1

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](#)

launch-wizard-1 created 2024-11-23T23:26:13.754Z

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0) Remove

Type Info	Protocol Info	Port range Info
ssh ↕	TCP	22
Source type Info	Source Info	Description - optional Info
Anywhere ↕	<input type="text" value="0.0.0.0/0"/> ×	e.g. SSH for admin desktop

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. ×

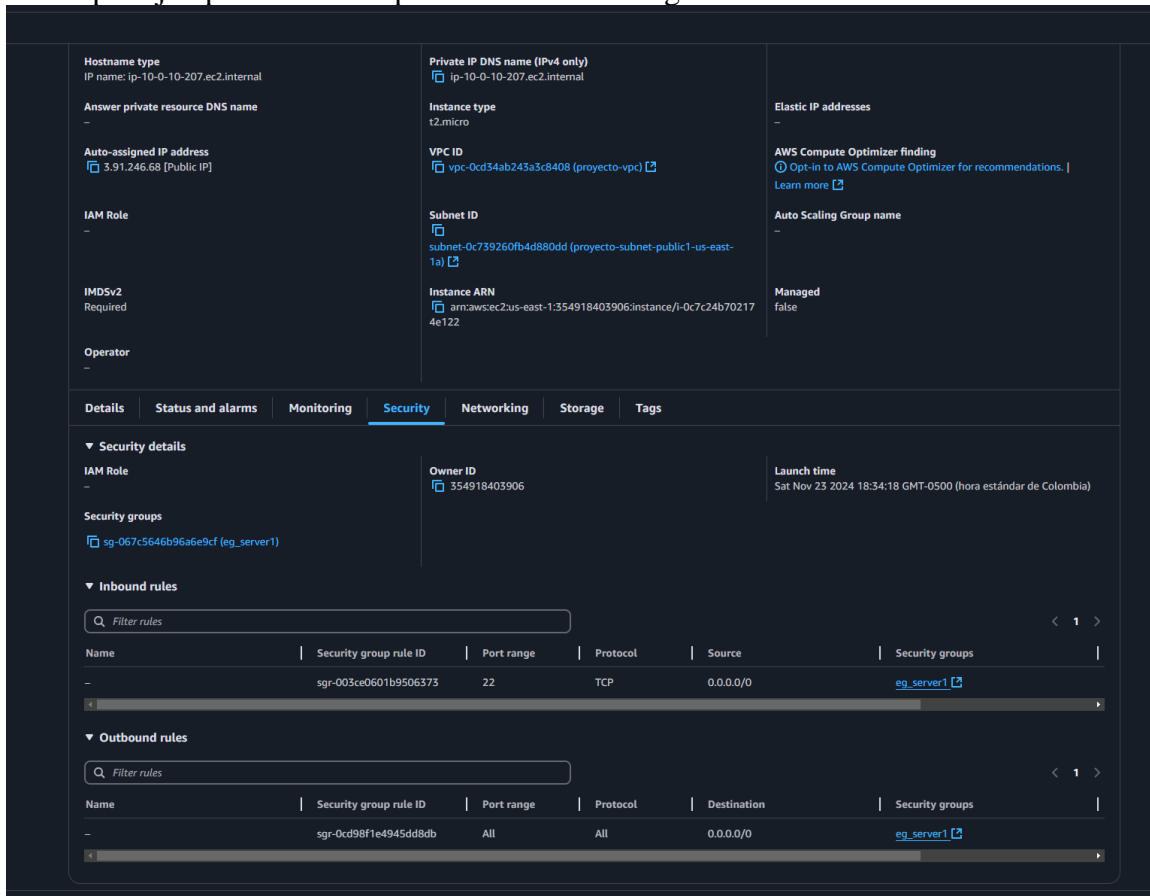
[Add security group rule](#)

Figura 5. lanzamos la instancia



Figura 6. Ingresando al id de la instancia podemos obtener datos como los grupos de seguridad

Como por ejemplo revisar los puertos de salida e ingreso



Ejemplo desde afuera hacia el interior por el puerto 22 recibo desde cualquier lugar conexiones.

Figura 7. Procedemos a conectar a la maquina.

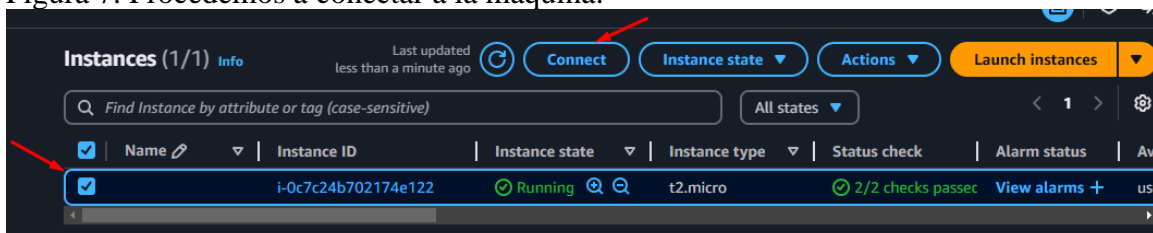


Figura 8. Procedemos a conectar desde Putty.

Connect to your instance i-0c7c24b702174e122 using any of these options

EC2 Instance Connect Session Manager **SSH client** EC2 serial console

Instance ID
i-0c7c24b702174e122

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is Server1.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 "Server1.pem"
4. Connect to your instance using its Public DNS:
ec2-3-91-246-68.compute-1.amazonaws.com

Example:
ssh -i "Server1.pem" ec2-user@ec2-3-91-246-68.compute-1.amazonaws.com

Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel

Figura 9. Abrir aplicación putty.

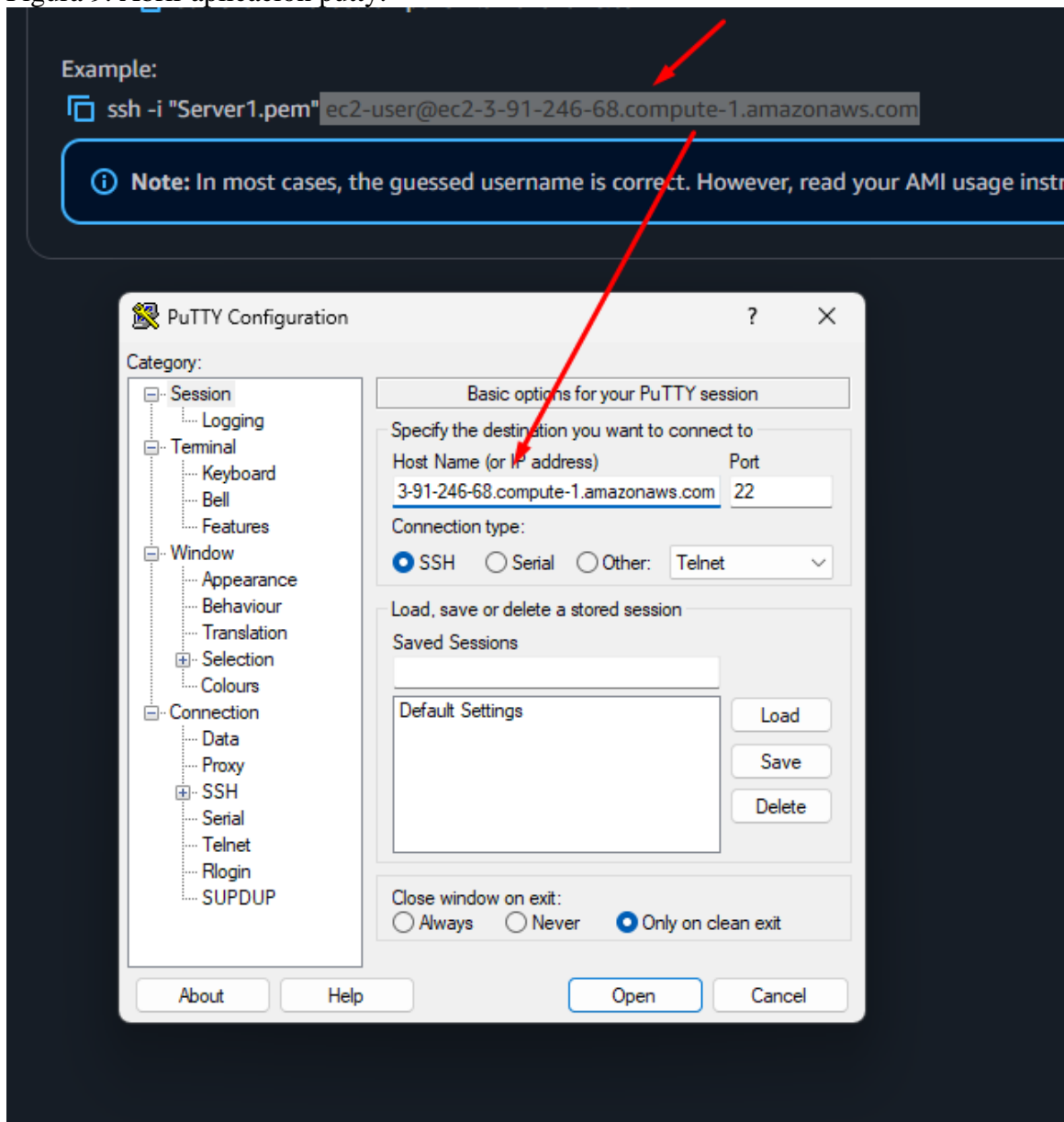


Figura 10. Conectamos por medio de certificado.

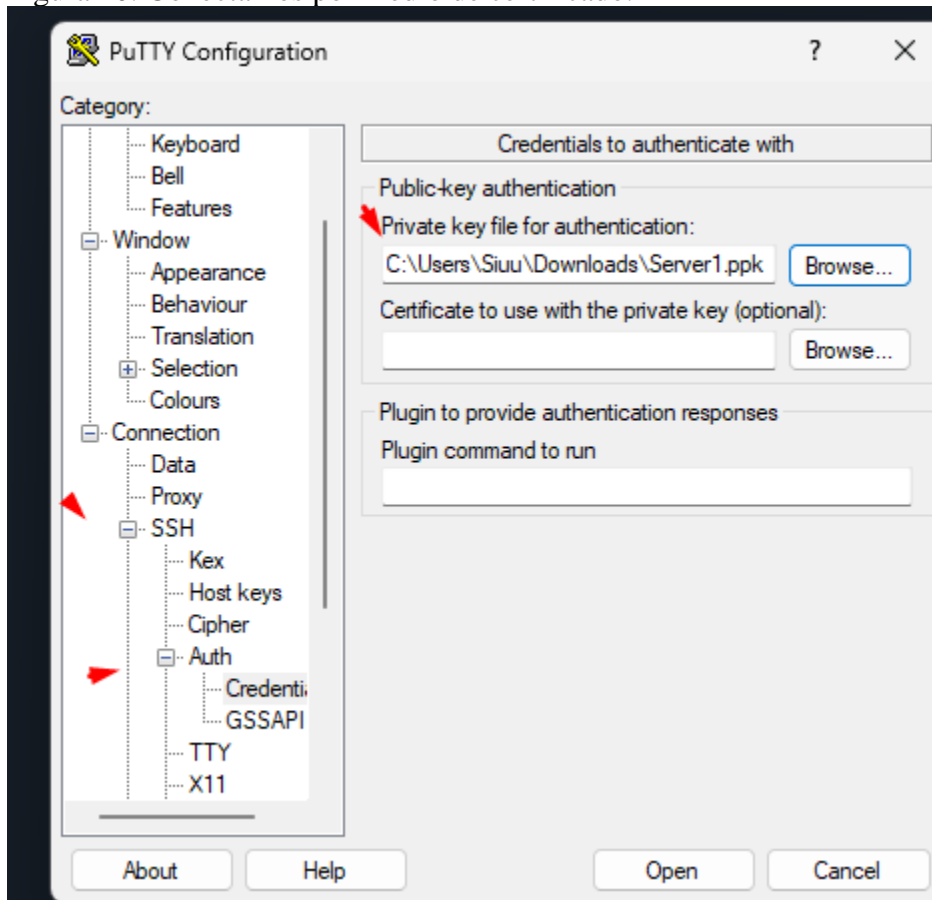


Figura 12. Instalación paquete apache le damos yes.

```
[ec2-user@ip-10-0-10-207 ~]$ sudo su
[root@ip-10-0-10-207 ec2-user]# yum install httpd
Last metadata expiration check: 2:16:51 ago on Sat Nov 23 23:34:51 2024.
Dependencies resolved.
=====
Package                Arch      Version                Repository             Size
=====
Installing:
httpd                  x86_64    2.4.62-1.amzn2023     amazonlinux            48 k
Installing dependencies:
apr                    x86_64    1.7.2-2.amzn2023.0.2  amazonlinux            129 k
apr-util               x86_64    1.6.3-1.amzn2023.0.1  amazonlinux            98 k
generic-logos-httpd   noarch    18.0.0-12.amzn2023.0.3 amazonlinux            19 k
httpd-core             x86_64    2.4.62-1.amzn2023     amazonlinux            1.4 M
httpd-filesystem      noarch    2.4.62-1.amzn2023     amazonlinux            14 k
httpd-tools           x86_64    2.4.62-1.amzn2023     amazonlinux            81 k
libbrotli              x86_64    1.0.9-4.amzn2023.0.2  amazonlinux            315 k
mailcap               noarch    2.1.49-3.amzn2023.0.3 amazonlinux            33 k
Installing weak dependencies:
apr-util-openssl      x86_64    1.6.3-1.amzn2023.0.1  amazonlinux            17 k
mod_http2             x86_64    2.0.27-1.amzn2023.0.3 amazonlinux            166 k
mod_lua               x86_64    2.4.62-1.amzn2023     amazonlinux            61 k

Transaction Summary
=====
Install 12 Packages

Total download size: 2.3 M
Installed size: 6.9 M
Is this ok [y/N]: 
```

Figura 13. Revisamos estatus.

```
root@ip-10-0-10-207:/home/ec2-user
Verifying      : apr-util-1.6.3-1.amzn2023.0.1.x86_64      2/12
Verifying      : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64 3/12
Verifying      : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 4/12
Verifying      : httpd-2.4.62-1.amzn2023.x86_64        5/12
Verifying      : httpd-core-2.4.62-1.amzn2023.x86_64    6/12
Verifying      : httpd-filesystem-2.4.62-1.amzn2023.noarch 7/12
Verifying      : httpd-tools-2.4.62-1.amzn2023.x86_64   8/12
Verifying      : libbrotli-1.0.9-4.amzn2023.0.2.x86_64  9/12
Verifying      : mailcap-2.1.49-3.amzn2023.0.3.noarch   10/12
Verifying      : mod_http2-2.0.27-1.amzn2023.0.3.x86_64 11/12
Verifying      : mod_lua-2.4.62-1.amzn2023.x86_64      12/12

Installed:
apr-1.7.2-2.amzn2023.0.2.x86_64
apr-util-1.6.3-1.amzn2023.0.1.x86_64
apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
httpd-2.4.62-1.amzn2023.x86_64
httpd-core-2.4.62-1.amzn2023.x86_64
httpd-filesystem-2.4.62-1.amzn2023.noarch
httpd-tools-2.4.62-1.amzn2023.x86_64
libbrotli-1.0.9-4.amzn2023.0.2.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch
mod_http2-2.0.27-1.amzn2023.0.3.x86_64
mod_lua-2.4.62-1.amzn2023.x86_64

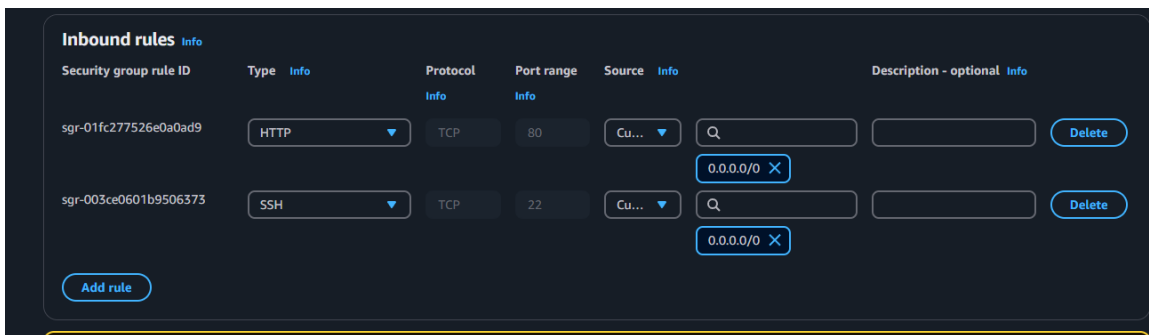
Complete!
[root@ip-10-0-10-207 ec2-user]# systemctl status httpd
o httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: inactive (dead)
     Docs: man:httpd.service(8)
lines 1-4/4 (END) ...skipping...
o httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: inactive (dead)
     Docs: man:httpd.service(8)
~
~
~
```

Figura 14. Activamos el servicio.

```
lines 1-4/4 (END)
^C
[root@ip-10-0-10-207 ec2-user]# systemctl start httpd
[root@ip-10-0-10-207 ec2-user]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: active (running) since Sun 2024-11-24 01:54:02 UTC; 3s ago
     Docs: man:httpd.service(8)
  Main PID: 31340 (httpd)
   Status: "Started, listening on: port 80"
    Tasks: 177 (limit: 1111)
  Memory: 12.9M
     CPU: 57ms
  CGroup: /system.slice/httpd.service
          └─31340 /usr/sbin/httpd -DFOREGROUND
            └─31341 /usr/sbin/httpd -DFOREGROUND
              └─31342 /usr/sbin/httpd -DFOREGROUND
                └─31343 /usr/sbin/httpd -DFOREGROUND
                  └─31344 /usr/sbin/httpd -DFOREGROUND

Nov 24 01:54:02 ip-10-0-10-207.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server:
Nov 24 01:54:02 ip-10-0-10-207.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server:
Nov 24 01:54:02 ip-10-0-10-207.ec2.internal httpd[31340]: Server configured, listening on: 0.0.0.0:80
lines 1-19/19 (END)
```

Figura 15. Agregamos puerto 80.



Details Status and alarms Monitoring **Security** Networking Storage Tags

▼ Security details

IAM Role: -

Owner ID: 354918403906

Launch time: Sat Nov 23 2024 18:34:18 GMT-0500 (hora estándar de Colombia)

Security groups: sg-067c5646b96a6e9cf (eg_server1)

▼ Inbound rules

Filter rules

Name	Security group rule ID	Port range	Protocol	Source	Security groups
-	sgr-01fc277526e0a0ad9	80	TCP	0.0.0.0/0	eg_server1
-	sgr-003ce0601b9506373	22	TCP	0.0.0.0/0	eg_server1

▼ Outbound rules

Filter rules

Name	Security group rule ID	Port range	Protocol	Destination	Security groups
-	sgr-0cd98f1e4945dd8db	All	All	0.0.0.0/0	eg_server1

Figura 16. Conectamos a la ip publica.

Instance summary for i-0c7c24b702174e122

Updated less than a minute ago

Connect Instance state Actions

Instance ID: i-0c7c24b702174e122

Public IPv4 address: 3.91.246.68 | open address

Private IPv4 address: 10.0.10.207

Instance state: Running

Public IPv4 DNS: ec2-3-91-246-68.compute-1.amazonaws.com | open address

Private IP DNS name (IPv4 only): ip-10-0-10-207.ec2.internal

Instance type: t2.micro

VPC ID: vpc-0cd34ab243a3c8408 (proyecto-vpc)

Private IP DNS name (IPv6 only): -

Instance type: -

Elastic IP addresses: -

Auto-assigned IP address: 3.91.246.68 [Public IP]

IAM Role: -

Subnet ID: subnet-0c739260fb4d880dd (proyecto-subnet-public1-us-east-1a)

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Auto Scaling Group name: -

IMDSv2: Required

Instance ARN: arn:aws:ec2:us-east-1:354918403906:instance/i-0c7c24b702174e122

Managed: false

Operator: -

Details Status and alarms Monitoring **Security** Networking Storage Tags

Figura 16. Conectar a ip.

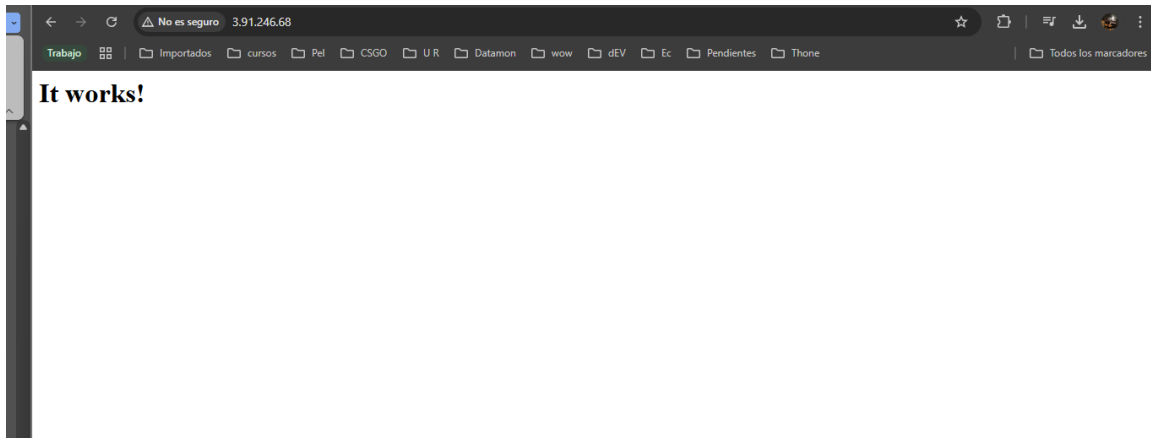


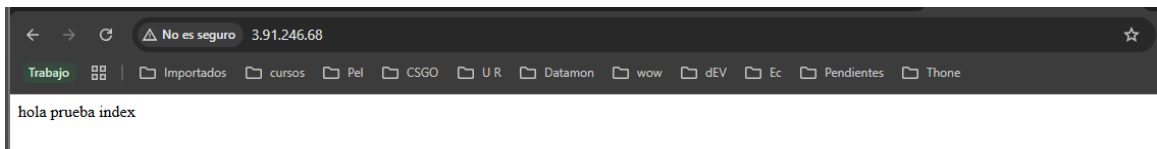
Figura 17. Ingresamos desde el Linux a la carpeta de la web y creamos un índice.

```
[root@ip-10-0-10-207 ec2-user]# cd /var/www/html/  
[root@ip-10-0-10-207 html]# nano index.html
```

Figura 18. Le damos permisos al archivo.

```
[root@ip-10-0-10-207 ec2-user]# cd /var/www/html/  
[root@ip-10-0-10-207 html]# nano index.html  
[root@ip-10-0-10-207 html]# chmod 777 index.html  
[root@ip-10-0-10-207 html]#
```

Figura 19. Verificamos cambios en la nube.



Video Entrega 1:

<https://youtu.be/qm5SF8yIAMo>

Entrega 2

Figura 20. Iniciaremos creando un snapshot para crear luego la imagen

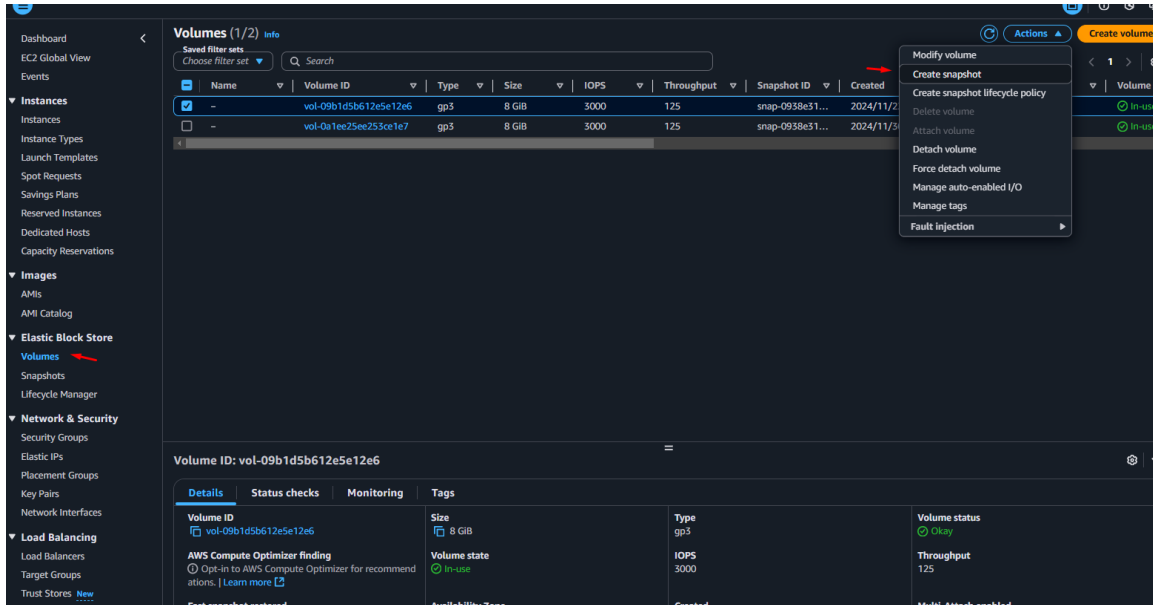


Figura 21. Luego creamos una imagen (ami)

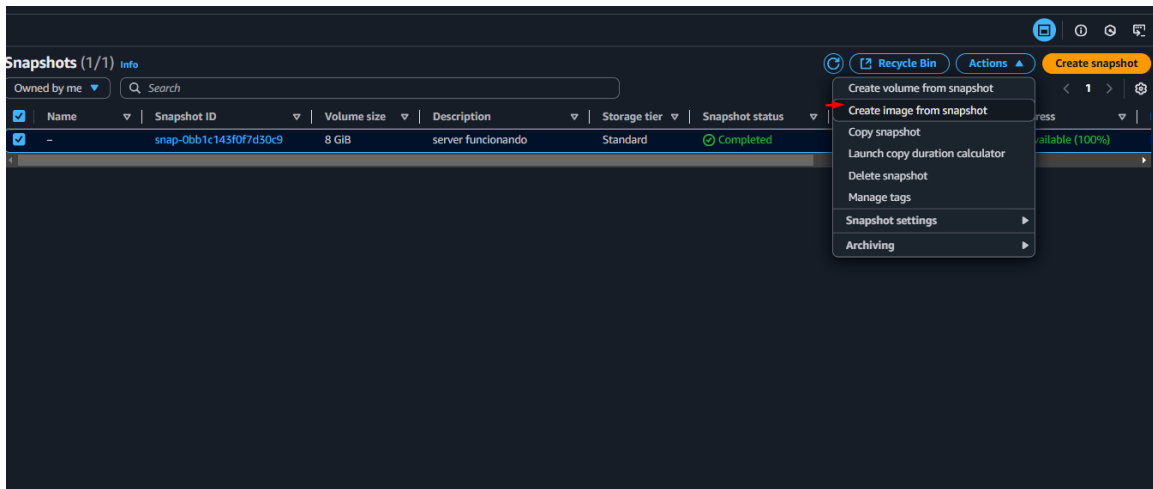


Figura 22.

nap-00b1c143f07d30c9 > Create image from snapshots

Snapshot ID
snap-00b1c143f07d30c9

Image name
A descriptive name for the image.
AmazonLinuxAndres

Description
A description for the image.
My image description

Architecture
Select i386 for 32-bit or x86_64 for 64-bit.
x86_64

Root device name
The device name that is reserved for the root volume.
/dev/sda1

Virtualization type
The virtualization type to be used by instances launched from this image.
Hardware-assisted virtualization

Kernel ID
The operating system kernel for the AMI.
Use default

RAM disk ID
The RAM disk for the image.
Use default

Boot mode
Use default

Block device mappings - optional

▼ Volume 1

Device type Root	Device name Snapshot snap-00b1c143f07d30c9
Size (GiB) 8	Volume type General Purpose SSD (gp3)
Throughput (MB/s) 125	IOPS 3000
Add volume	Termination behavior <input checked="" type="checkbox"/> Delete on termination
	Encryption <input type="checkbox"/> Encrypt volume

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.
No tags associated with the resource.
Add tag
You can add 50 more tags.

Cancel **Create image**

Figura 23 . verificamos la ami

Dashboard < Amazon Machine Images (AMIs) (1) info

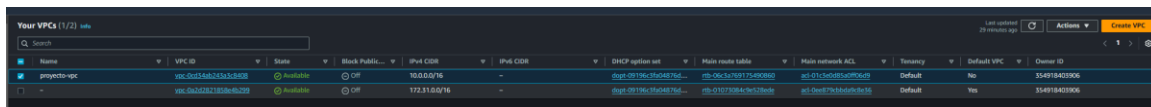
Owned by me Find AMI by attribute or tag

<input type="checkbox"/>	Name	AMI name	AMI ID	Source	Owner	Visibility
<input type="checkbox"/>		AmazonLinuxAndres	ami-08c117691de57c7f3	354918403906/AmazonLinuxAndres	354918403906	Private

Instances > Images > AMIs

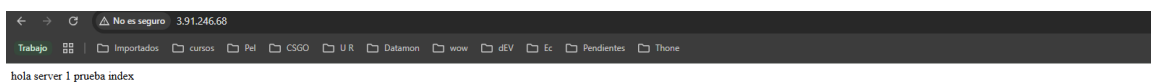
Balancedores de carga

Figura 24. Crear otra instancia de Linux con las mismas configuraciones anteriores

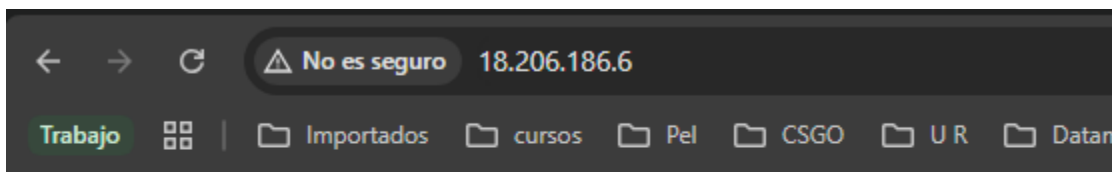


Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR	DHCP options set	Main route table	Main network ACL	Tenancy	Default VPC	Owner ID
pmpeto-ec2	vpc-0a15ab25a3538908	Running	Off	10.0.0.0/16	-	dhpt-0919a13f048875d...	rtb-03c2a705172d90800	acl-01c3e005e0d0790c0	Default	No	354918402906
ec2-0a202221805e6b229	vpc-0a202221805e6b229	Running	Off	172.31.0.0/16	-	dhpt-0919a13f048875d...	rtb-0107308445ec138dc	acl-0a6d77a160a6b0c36	Default	Yes	354918402906

Figura 25. Valíamos el entrono web de las dos instancias



hola server 1 prueba index



server 222222222222222222

Figura 26. Iniciamos proceso de crear balanceador de carga

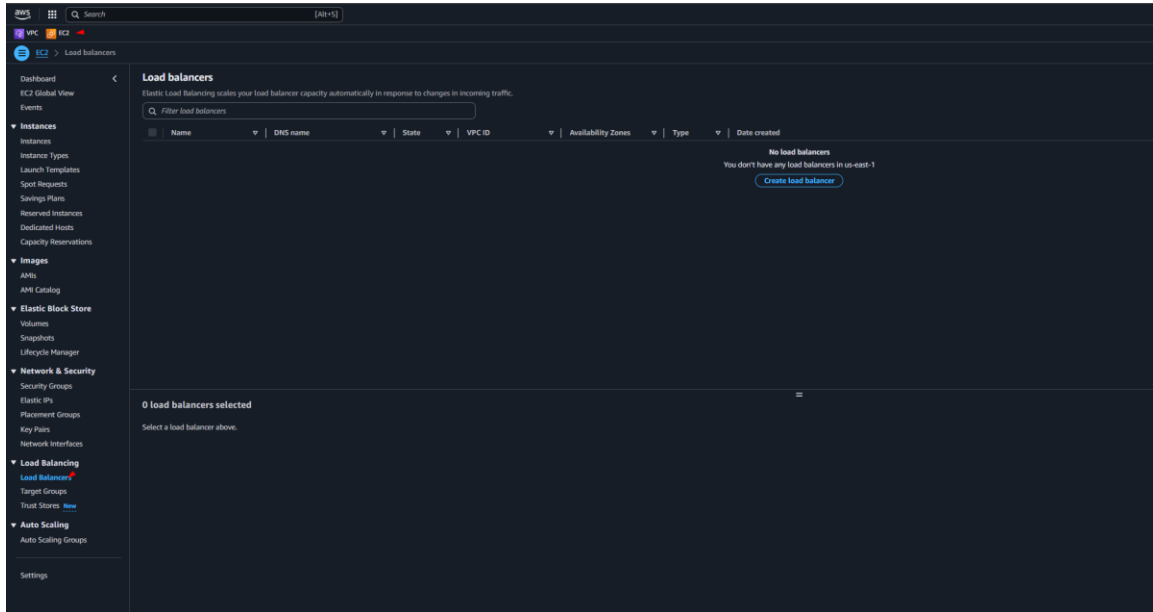


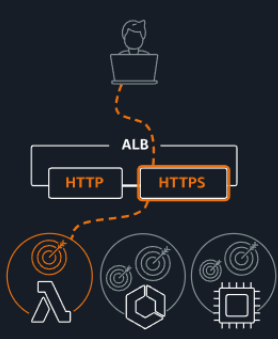
Figura 27. Seleccionamos **Application Load Balancer**

Compare and select load balancer type

A complete feature-by-feature comparison along with detailed highlights is also available. [Learn more](#)

Load balancer types

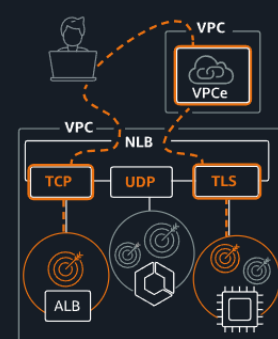
Application Load Balancer Info



Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

[Create](#)


Network Load Balancer Info



Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

[Create](#)

Gateway Load Balancer Info



Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.

[Create](#)

► **Classic Load Balancer - previous generation**

[Close](#)

Figura 28. Configuramos el balanceador importante seleccionar el mismo vpc

EC2 > Load balancers > Create Application Load Balancer

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 can't be changed after the load balancer is created.

IBPruebas

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

Scheme Info
Scheme can't be changed after the load balancer is created.

Internet-facing

- Serves internet-facing traffic.
- Has public IP addresses.
- DNS name is publicly resolvable.
- Requires a public subnet.

Internal

- Serves internal traffic.
- Has private IP addresses.
- DNS name is not publicly resolvable.
- Compatible with the IPv4 and Dualstack IP address types.

Load balancer IP address type Info
Select the front-end IP address type to assign to the load balancer. The VPC and subnets mapped to this load balancer must include the selected IP address types. Public IPv4 addresses have an additional cost.

IPv4
Includes only IPv4 addresses.

Dualstack
Includes IPv4 and IPv6 addresses.

Dualstack without public IPv4
Includes a public IPv6 address, and private IPv4 and IPv6 addresses. Compatible with **Internet-facing** load balancers only.

Network mapping Info

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC Info
The load balancer will exist and scale within the selected VPC. The selected VPC is also where the load balancer targets must be hosted unless routing to Lambda or on-premises targets, or if using VPC peering. To confirm the VPC for your targets, view [target groups](#). For a new VPC, [create a VPC](#).

projecto-vpc
vpc-0cd34ab243a3c8408
IPv4 VPC CIDR: 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.

Availability zones

us-east-1a (use1-az6)

Subnet

subnet-0c739260fb4d880dd proyecto-subnet-public1-us-east-1a

IPv4 address
Assigned by AWS

Figura 29. Se debe crear su propio grupo de seguridad.

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

Application Load Balancers require at least one security group. If none are selected, the VPC's default security group will be applied.

Create security group Info
A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name Info
SG_BAlanceadorCG
Name cannot be edited after creation.

Description Info
Bc

VPC Info
vpc-0cd34ab243a3c8408 (proyecto-vpc)

Inbound rules Info

Type	Protocol	Port range	Source	Description - optional
Custom TCP	TCP	80	Any... 0.0.0.0	

Outbound rules Info

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	Any... 0.0.0.0	

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.
No tags associated with the resource.
Add new tag
You can add up to 50 more tags.

Figura 30. En el listener configuramos para que entre desde internet y las maquinas objetivo

The screenshot shows the 'Create Application Load Balancer' configuration page in the AWS Management Console. The page is dark-themed and contains the following sections:

- VPC**: Shows the selected VPC as 'proyecto-vpc' (vpc-0cd34ab243a3c8408) with IPv4 CIDR: 10.0.0.0/16.
- Mappings**: Shows the selected Availability Zone as 'us-east-1a (use1-az6)' and the Subnet as 'subnet-0c739260fb4d880dd' (IPv4 subnet CIDR: 10.0.0.0/20).
- Security groups**: Shows the selected security group as 'SG_BAlanceadorCG' (sg-075d19bb334dcd9e1).
- Listeners and routing**: This section is highlighted with a red box. It shows a listener named 'Listener HTTP:80' with the following configuration:
 - Protocol: HTTP
 - Port: 80
 - Default action: Forward to (with a dropdown menu for selecting a target group)
 - Link: [Create target group](#)
- Load balancer tags - optional**: A section for adding tags to the load balancer.

Figura 31. Configuramos el Grupo

EC2 > Target groups > Create target group

Step 1
● Specify group details
Step 2
○ Register targets

Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration

Settings in this section can't be changed after the target group is created.

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.

IP addresses

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

Lambda function

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

Application Load Balancer

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name

TG_Reming1

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

Protocol : Port

Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation.

HTTP 80
1-65535

IP address type

Only targets with the indicated IP address type can be registered to this target group.

IPv4

Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

IPv6

Each instance you register must have an assigned primary IPv6 address. This is configured on the Instance's default network interface (eth0). [Learn more](#)

VPC

Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

projecto-vpc
vpc-0cd34ab243a3c8408
IPv4 VPC CIDR: 10.0.0.0/16

Protocol version

HTTP1

Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

Figura 32. Seleccionar vpc y las rutas de diagnostico

target group

Protocol : Port
Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation

HTTP 80
1-65535

IP address type
Only targets with the indicated IP address type can be registered to this target group.

IPv4
Each Instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

IPv6
Each Instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)

VPC
Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

proyecto-vpc
vpc-0cd34ab243a3c8408
IPv4 VPC CIDR: 10.0.0.0/16

Protocol version

HTTP1
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

HTTP2
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

gRPC
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

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 perform health checks on the root, or specify a custom path if preferred.
/
Up to 1024 characters allowed.

▶ **Advanced health check settings**

Attributes
Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

▶ **Tags - optional**
Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

Cancel Next

Figura 33. Seleccionamos las instancias

Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (2/2)

Filter instances

<input checked="" type="checkbox"/>	Instance ID	Name	State	Security groups
<input checked="" type="checkbox"/>	i-0c3e30885ac684b45	server2	Running	eg_server1
<input checked="" type="checkbox"/>	i-0c7c24b702174e122	server1	Running	eg_server1

2 selected

Ports for the selected instances
Ports for routing traffic to the selected instances.

1-65535 (separate multiple ports with commas)

Include as pending below

Review targets

Review targets

Targets (2)

Filter targets

Show only pending

Instance ID	Name	Port	State	Security groups	Zone	Private IPv4 address	Subnet ID
i-0c3e30885ac684b45	server2	80	Running	eg_server1	us-east-1a	10.0.6.6	subnet-0c73926a
i-0c7c24b702174e122	server1	80	Running	eg_server1	us-east-1a	10.0.10.207	subnet-0c73926a

2 pending

Cancel Previous **Create target group**

Figura 34. En este momento empezar a validar las instancias creadas y el Healthy debe ser igual a las cantidades de instancias

The screenshot displays the AWS Management Console interface for a Target Group named "TGReming1". At the top, a green notification bar states: "Successfully created the target group: TGReming1. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the Targets tab." Below this, the "Details" section shows the following information:

- Target type:** Instance
- Protocol : Port:** HTTP: 80
- Protocol version:** HTTP1
- VPC:** vpc-0cd34ab243a3c8408
- IP address type:** IPv4
- Load balancer:** None associated

A red arrow points to the "IP address type" field. Below the details, a summary bar indicates:

- Total targets:** 2
- Healthy:** 0
- Unhealthy:** 0
- Unused:** 2
- Initial:** 0
- Draining:** 0
- Anomalous:** 0

Below the summary bar, there is a section for "Distribution of targets by Availability Zone (AZ)" with a note: "Select values in this table to see corresponding filters applied to the Registered targets table below."

The "Targets" tab is selected, showing a table of registered targets:

Instance ID	Name	Port	Zone	Health status	Health status details	Admini...
i-0c3e30885ac684b45	server2	80	us-east-1a (us...)	Unused	Target group is not co...	-
i-0c7c24b702174e122	server1	80	us-east-1a (us...)	Unused	Target group is not co...	-

Figura 35.
Mientras carga el TG seguimos configurando el balanceador

IPV4 VPC CIDR: 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.

Availability Zones
 us-east-1a (use1-az6)
Subnet
subnet-0c739260fb4d880dd IPv4 subnet CIDR: 10.0.0.0/20 proyecto-subnet-public1-us-east-1a

IPv4 address
Assigned by AWS

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

SG_BALANCEADORCG sg-075d19bb334dc9e1 VPC: vpc-0cd54ab243a3c8408

Listeners and routing [Info](#)
A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener HTTP:80 Remove

Protocol: HTTP Port: 80 (1-65535)

Default action [Info](#)
Forward to: TGReming1 (Target type: Instance, IPv4) HTTP Refresh
[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.

[Add listener](#)

► **Load balancer tags - optional**
Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them. The 'Key' is required, but 'Value' is optional. For example, you can have Key = production-webserver, or Key = webserver, and Value = production.

Optimize with service integrations - optional [Info](#)
Optimize your load balancing architecture by integrating AWS services with this load balancer at launch. You can also add these and other services after your load balancer is created by reviewing the load balancer's "Integrations" tab.

Figura 36. Importante de debe configurar la vpc para que de dos subredes publicas funcionales

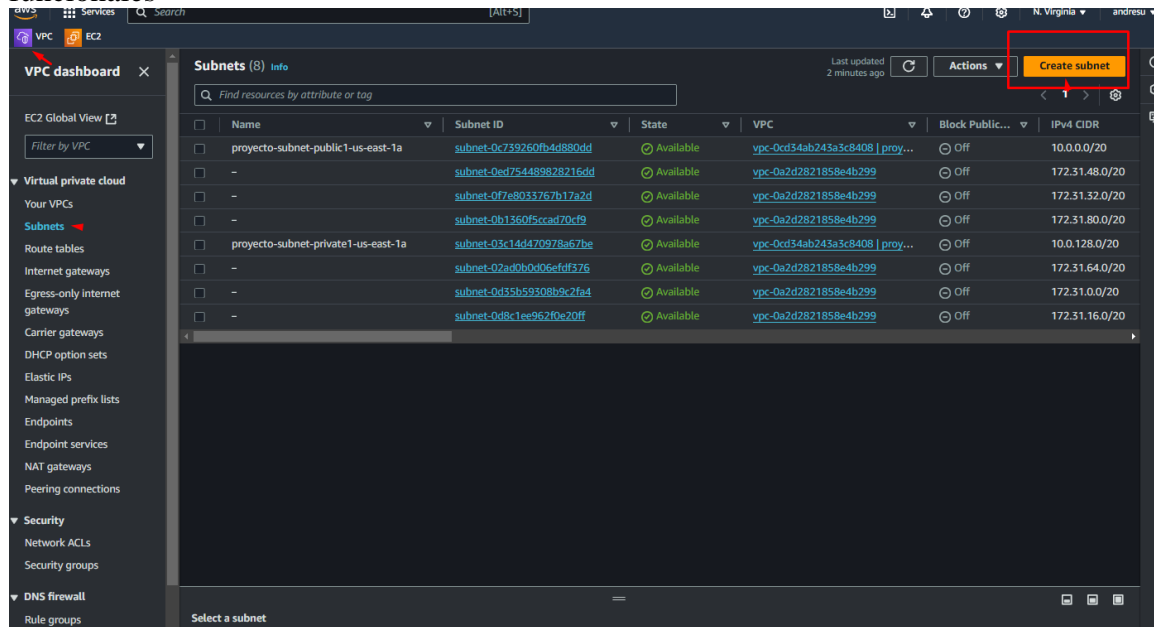


Figura 37. Seleccionamos las subredes en el balanceador

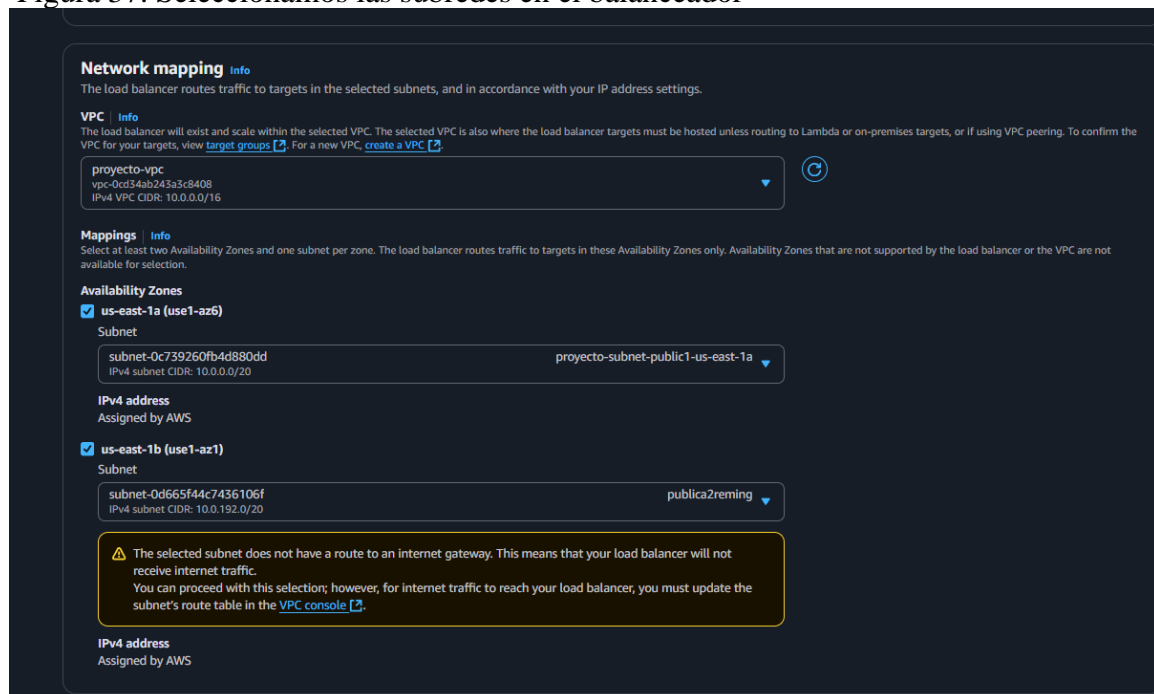


Figura 37. Configuración subredes

VPC

VPC ID
Create subnets in this VPC.
vpc-0cd34ab243a3c8408 (proyecto-vpc)

Associated VPC CIDRs

IPv4 CIDRs
10.0.0.0/16

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
publica2reming
The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US East (N. Virginia) / us-east-1b

IPv4 VPC CIDR block [Info](#)
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
10.0.0.0/16

IPv4 subnet CIDR block
10.0.192.0/20 4096 IPs
< > ^ v

Tags - optional

Key	Value - optional	
Q Name	Q publica2reming	Remove

Add new tag
You can add 49 more tags.

Remove

Add new subnet

Cancel **Create subnet**

Figura 38. Configuramos el internet Gateway

Subnets (1/9) Info Last updated 2 minutes ago ↻ **Actions** ▼ **Create subnet**

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
publica1reming	subnet-0c739260fb4d880dd	Available	vpc-0cd34ab243a3c8408 proy...	Off	10.0.0.0/20
-	subnet-0ed754489828216dd	Available	vpc-0a2d2821858e4b299	Off	172.31.48.0/20
-	subnet-0f7e8033767b17a2d	Available	vpc-0a2d2821858e4b299	Off	172.31.32.0/20
-	subnet-0b1360f5ccad70cf9	Available	vpc-0a2d2821858e4b299	Off	172.31.80.0/20
proyecto-subnet-private1-us-east-1a	subnet-03c14d470978a67be	Available	vpc-0cd34ab243a3c8408 proy...	Off	10.0.128.0/20
-	subnet-02ad0b0d066fd376	Available	vpc-0a2d2821858e4b299	Off	172.31.64.0/20
-	subnet-0d35b59308b9c2fa4	Available	vpc-0a2d2821858e4b299	Off	172.31.0.0/20
-	subnet-0d8c1ee962f0e20ff	Available	vpc-0a2d2821858e4b299	Off	172.31.16.0/20
publica2reming	subnet-0d665f44c7436106f	Available	vpc-0cd34ab243a3c8408 proy...	Off	10.0.192.0/20

subnet-0d665f44c7436106f / publica2reming **Actions** ▼

Details

Subnet ID subnet-0d665f44c7436106f	Subnet ARN arn:aws:ec2:us-east-1:354918403906:subnet/subnet-0d665f44c7436106f	State Available	Block Public Access Off
IPv4 CIDR 10.0.192.0/20	Available IPv4 addresses 4091	IPv6 CIDR -	IPv6 CIDR association ID -
Availability Zone us-east-1b	Availability Zone ID use1-az1	Network border group us-east-1	VPC vpc-0cd34ab243a3c8408 proyecto-vpc
Route table rtb-06c3a769175490860	Network ACL acl-01c3e0d85a0ff06d9	Default subnet No	Auto-assign public IPv4 address No
Auto-assign IPv6 address No	Auto-assign customer-owned IPv4 address No	Customer-owned IPv4 pool -	Outpost ID -
IPv4 CIDR reservations -	IPv6 CIDR reservations -	IPv6-only No	Hostname type IP name
Resource name DNS A record Disabled	Resource name DNS AAAA record Disabled	DNS64 Disabled	Owner 354918403906

Flow logs | **Route table** | **Network ACL** | **CIDR reservations** | **Sharing** | **Tags**

Route table: rtb-06c3a769175490860 **Edit route table association**

Routes (2)

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-09804a0ebda954166

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway	-	No
	igw-09804a0ebda954166		

Figura 39. Continuamos con la creación del balanceador
Seleccionamos las subredes

Network mapping Info

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC Info
The load balancer will exist and scale within the selected VPC. The selected VPC is also where the load balancer targets must be hosted unless routing to Lambda or on-premises targets, or if using VPC peering. To confirm the VPC for your targets, view [target groups](#). For a new VPC, [create a VPC](#).

projecto-vcpc
vpc-0cd34ab243a3c8408
IPv4 VPC CIDR: 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.

Availability Zones

us-east-1a (use1-az6)
Subnet
subnet-0c739260fb4d880dd public1reming
IPv4 subnet CIDR: 10.0.0.0/20

IPv4 address
Assigned by AWS

us-east-1b (use1-az1)
Subnet
subnet-0d665f44c7436106f public2reming
IPv4 subnet CIDR: 10.0.192.0/20

IPv4 address
Assigned by AWS

Figura 40. Bajamos y le damos crear

The screenshot shows the 'Create Application Load Balancer' page in the AWS Management Console. The page is divided into several sections:

- Optimizes security:** A checkbox for 'Apply application layer security protections - in front of targets' is unchecked. Below it, a note states: 'Your choice of either a pre-defined security configuration with basic recommended AWS WAF security protections, or associate any of your existing WAF configurations for custom protections. [Additional charges apply](#)'.
- Benefits and considerations:** A section with a right-pointing arrow.
- AWS Global Accelerator:** A section with an 'Info' icon and the text 'Optimizes: Performance, Availability'. A checkbox for 'Apply global load balancing across multiple regions' is unchecked. Below it, a note states: 'Creates an accelerator in your account with two global static IPs that act as a fixed entry point to your load balancer. If you do not need global static IPs or traffic management across multiple regions, select Amazon CloudFront. [Additional charges apply](#)'.
- Review:** A section with the heading 'Review' and a sub-heading 'Review the load balancer configurations and make changes if needed. After you finish reviewing the configurations, choose **Create load balancer**.' Below this is a 'Summary' section with four columns:
 - Basic configuration:** IBPruebas, Internet-facing, IPv4.
 - Security groups:** default (sq-0b48eb2d14f587e41).
 - Network mapping:** VPC: vpc-0cd34ab243a3c8408 (projecto-vc), subnets: us-east-1a (subnet-0c739260fb4d880dd), us-east-1b (subnet-0d665f44c7436106f).
 - Listeners and routing:** HTTP:80 defaults to TGReming1.
- Service integrations:** Amazon CloudFront + AWS Web Application Firewall (WAF): None, AWS WAF: None, AWS Global Accelerator: None.
- Attributes:** A note: 'Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.'
- Creation workflow and status:** A section with a right-pointing arrow and the heading 'Server-side tasks and status'. Below it, a note states: 'After completing and submitting the above steps, all server-side tasks and their statuses become available for monitoring.'

At the bottom right, there are two buttons: 'Cancel' and 'Create load balancer'. The 'Create load balancer' button is highlighted with a red box.

Figura 4. Se verifican los datos y la creación.

The screenshot displays the AWS Management Console interface for a newly created Application Load Balancer. At the top, a green notification banner states: "Successfully created load balancer: IBPruebas. It might take a few minutes for your load balancer to fully set up and route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks." Below this, the main content area is titled "IBPruebas" and includes a "Details" section with the following information:

- Load balancer type:** Application
- Scheme:** Internet-facing
- Status:** Provisioning
- Hosted zone:** Z35SXDOTRQ7X7K
- VPC:** vpc-0cd34ab243a3c8408
- Availability Zones:**
 - subnet-0c739260fb4d880dd (use1-azg) us-east-1a
 - subnet-0d665f44c7436106f (use1-az1) us-east-1b
- Load balancer IP address type:** IPv4
- Date created:** November 30, 2024, 18:13 (UTC-05:00)
- Load balancer ARN:** arn:aws:elasticloadbalancing:us-east-1:354918403906:loadbalancer/app/IBPruebas/ba67e427295fc72d
- DNS name info:** IBPruebas-1843630471.us-east-1.elb.amazonaws.com (A Record)

Below the details, there are tabs for "Listeners and rules", "Network mapping", "Resource map - new", "Security", "Monitoring", "Integrations", "Attributes", "Capacity - new", and "Tags". The "Listeners and rules" tab is active, showing a table with one listener:

Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certifica
HTTP:80	Forward to target group <ul style="list-style-type: none"> TGReming1 (100%) Target group stickiness: Off 	1 rule	ARN	Not applicable	Not applicable

Figura 41. Revisamos el grupo de seguridad y seleccionamos el del balanceador

The screenshot shows the "Edit security groups" dialog box in the AWS Management Console. The dialog is titled "Load balancer details: IBPruebas" and contains the following information:

- Security groups:** A section explaining that a security group is a set of firewall rules that control traffic to the load balancer. It includes a link to "create a new security group".
- Security groups:** A dropdown menu with a search icon and a refresh icon. Below it, a search bar contains the text "Select up to 5 security groups".
- Selected security group:** A card showing the selected security group: "SG_IBBalanceadorCG" with ID "sg-075d19bb3346c9e1" and VPC "vpc-0cd34ab243a3c8408".
- Buttons:** "Cancel" and "Save changes" buttons are located at the bottom right.

Figura 42.
Validamos la conexión y en el momento conecta a la maquina servidor 2

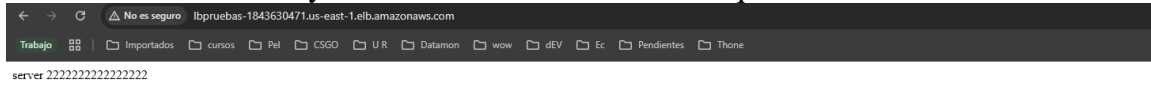


Figura 43.
Ahora se automatiza el balanceador.

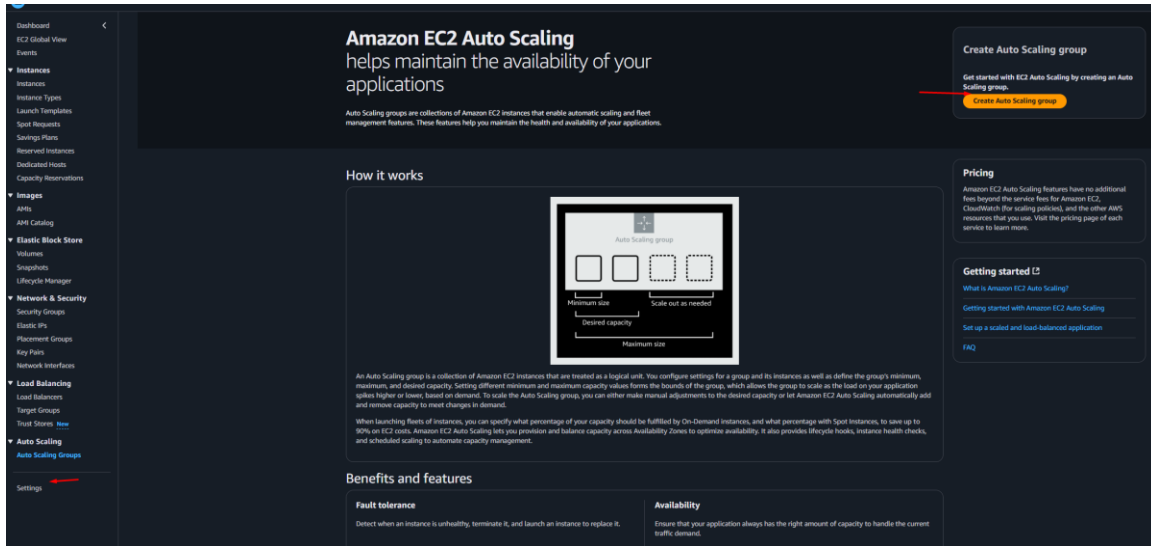


Figura 44.
Creamos el template

EC2 / Auto Scaling groups / Create Auto Scaling group

Step 1 **Choose launch template**
Step 2 Choose instance launch options
Step 3 - optional Integrate with other services
Step 4 - optional Configure group size and scaling
Step 5 - optional Add notifications
Step 6 - optional Add tags
Step 7 Review

Choose launch template Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

Name

Auto Scaling group name
Enter a name to identify the group.

AGRenington1

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

Launch template Info

Info For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

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.

Select a launch template

Create a launch template [↗](#)

Cancel **Next**

Figura 45 le damos un nombre y seleccionamos la ami creada en los primeros paso.

EC2 > Launch templates > Create launch template

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - *required*

LTReming

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

Template version description

A prod webserver for MyApp

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

Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

▼ Application and OS Images (Amazon Machine Image) - required [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Q Search our full catalog including 1000s of application and OS images

Recents **My AMIs** Quick Start

Owned by me Shared with me

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

AmazonLinuxAndres
ami-08c117691de57c7f3
2024-11-30T23:44:37.000Z Virtualization: hvm ENA enabled: true Root device type: ebs

Figura 46. Seleccionamos el key, y grupo creados anteriormente y de damos crear

Amazon Machine Image (AMI)

AmazonLinuxAndres
ami-08c117691de57c7f3
2024-11-30T23:44:37.000Z Virtualization: hvm ENA enabled: true Root device type: ebs

Description

-

Architecture **AMI ID**
x86_64 ami-08c117691de57c7f3

Instance type [Info](#) | [Get advice](#) Advanced

Instance type

t2.micro Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Windows base pricing: 0.0162 USD per Hour

On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour On-Demand SUSE base pricing: 0.0116 USD per Hour

On-Demand RHEL base pricing: 0.026 USD per Hour On-Demand Linux base pricing: 0.0116 USD per Hour

All generations [Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

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 ↗

Server1 [Create new key pair](#)

Network settings [Info](#)

Subnet [Info](#)

Don't include in launch template [Create new subnet](#)

When you specify a subnet, a network interface is automatically added to your template.

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

Common security groups [Info](#)

Select security groups

SG_BALanceadorCG sg-075d19bb334dcd9e1 ↗ [Compare security group rules](#)

VPC: vpc-0cd34ab243a3c8408

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

Advanced network configuration

Summary

Software Image (AMI)
AmazonLinuxAndres
ami-08c117691de57c7f3

Virtual server type (instance type)
t2.micro

Firewall (security group)
SG_BALanceadorCG

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

Free tier: In your first month, 750 hours of t2.micro (or t3.micro) is unavailable) instance hours, 30 GiB of EBS snapshots, and 100 GB of Amazon S3 Standard-IA storage.

[Cancel](#)

Figura 47. procedemos con la configuración

Step 1 **Choose launch template**

Step 2 Choose instance launch options

Step 3 - optional Integrate with other services

Step 4 - optional Configure group size and scaling

Step 5 - optional Add notifications

Step 6 - optional Add tags

Step 7 Review

Choose launch template Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

Name

Auto Scaling group name
Enter a name to identify the group.

AGRenington1

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

Launch template Info

For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

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.

LTReming

Create a launch template

Version

Default (1)

Create a launch template version

Description -	Launch template LTReming lt-0f6eaffe53de5d084	Instance type t2.micro
AMI ID ami-08c117691de57c7f3	Security groups -	Request Spot Instances No
Key pair name Server1	Security group IDs sg-075d19bb334dcd9e1	
Additional details		
Storage (volumes) -	Date created Sat Nov 30 2024 19:03:54 GMT-0500 (hora estándar de Colombia)	

Cancel **Next**

Figura 48. Le damos siguiente

Step 1
● Choose launch template

Step 2
● **Choose instance launch options**

Step 3 - optional
● Integrate with other services

Step 4 - optional
● Configure group size and scaling

Step 5 - optional
● Add notifications

Step 6 - optional
● Add tags

Step 7
● Review

Choose instance launch options Info

Choose the VPC network environment that your instances are launched into, and customize the instance types and purchase options.

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
LTReming ↗ lt-0f6eaffe53de5d084	Default	-

Instance type
t2.micro

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-0cd34ab243a3c8408 (projecto-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-east-1a | subnet-0c739260fb4d880dd (publica1reming) [✕](#)
10.0.0.0/20

us-east-1b | subnet-0d665f44c7436106f (publica2reming) [✕](#)
10.0.192.0/20

[Create a subnet](#) [↗](#)

Availability Zone distribution - new
Auto Scaling automatically balances instances across Availability Zones. If launch failures occur in a zone, select a strategy.

Balanced best effort
If launches fail in one Availability Zone, Auto Scaling will attempt to launch in another healthy Availability Zone.

Balanced only
If launches fail in one Availability Zone, Auto Scaling will continue to attempt to launch in the unhealthy Availability Zone to preserve balanced distribution.

Cancel Skip to review Previous Next

Figura 49.

Seleccionamos el que realice el balanceador a maquinas existentes por ejemplo cuando se elimine una maquina.

Step 1
● Choose launch template

Step 2
● Choose instance launch options

Step 3 - optional
● **Integrate with other services**

Step 4 - optional
● Configure group size and scaling

Step 5 - optional
● Add notifications

Step 6 - optional
● Add tags

Step 7
● Review

Integrate with other services - optional [Info](#)

Use a load balancer to distribute network traffic across multiple servers. Enable service-to-service communications with VPC Lattice. Shift resources away from impaired Availability Zones with zonal shift. You can also customize health check replacements and monitoring.

Load balancing [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

TGReming1 | HTTP
Application Load Balancer: IBPruebas

VPC Lattice integration options [Info](#)

To improve networking capabilities and scalability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between AWS services and helps you connect and manage your applications across compute services in AWS.

Select VPC Lattice service to attach

No VPC Lattice service
VPC Lattice will not manage your Auto Scaling group's network access and connectivity with other services.

Attach to VPC Lattice service
Incoming requests associated with specified VPC Lattice target groups will be routed to your Auto Scaling group.

Create new VPC Lattice service [↗](#)

Application Recovery Controller (ARC) zonal shift - new [Info](#)

During an Availability Zone impairment, target instance launches towards other healthy Availability Zones.

Enable zonal shift
New instance launches will be retargeted towards healthy Availability Zones until the zonal shift is canceled.

Figura 50.
Le indicamos que cree dos maquinas

The screenshot shows the 'Configure group size and scaling' step in the AWS console. The left sidebar lists steps 1 through 7, with step 4, 'Configure group size and scaling', highlighted. The main content area is divided into several sections:

- Group size:** A section titled 'Group size' with an 'Info' icon. It explains that the initial size can be changed manually or automatically. Below this, there is a 'Desired capacity type' dropdown menu set to 'Units (number of instances)'. A 'Desired capacity' input field contains the number '2', with a red arrow pointing to it.
- Scaling:** A section titled 'Scaling' with an 'Info' icon. It includes 'Scaling limits' with 'Min desired capacity' set to '1' and 'Max desired capacity' set to '2'. Below this is the 'Automatic scaling' section, which has two radio button options: 'No scaling policies' (selected) and 'Target tracking scaling policy'.
- Instance maintenance policy:** A section titled 'Instance maintenance policy' with an 'Info' icon. It includes a sub-section 'Choose a replacement behavior depending on your availability requirements' with four radio button options: 'No policy' (selected), 'Launch before terminating', 'Terminate and launch', and 'Custom behavior'.

Figura 51.
Le damos siguiente

The screenshot shows the 'Add notifications' step in the AWS console. The left sidebar lists steps 1 through 7, with step 5, 'Add notifications', highlighted. The main content area includes:

- A section titled 'Add notifications' with an 'Info' icon. It states: 'Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group.' Below this is an 'Add notification' button.
- At the bottom right, there are four buttons: 'Cancel', 'Skip to review', 'Previous', and 'Next'.

Figura 52.
Siguiente

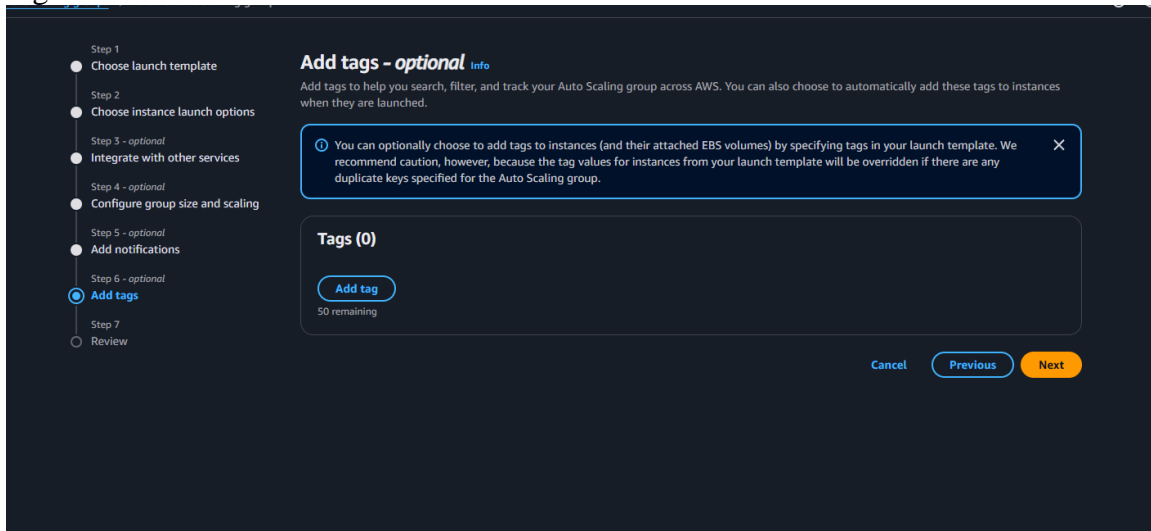


Figura 53. Y crear

Step 1: Desired capacity

Desired capacity	2	Desired capacity type	Units (number of instances)
------------------	---	-----------------------	-----------------------------

Step 2: Scaling

Minimum desired capacity	1	Maximum desired capacity	2
Target tracking policy	-		

Step 3: Instance maintenance policy

Replacement behavior	No policy	Min healthy percentage	-	Max healthy percentage	-
----------------------	-----------	------------------------	---	------------------------	---

Step 4: Additional settings

Instance scale-in protection	Disabled	Monitoring	Disabled	Default instance warmup	Disabled
------------------------------	----------	------------	----------	-------------------------	----------

Step 5: Capacity Reservation preference

Preference	Default	Capacity Reservation IDs	-	Resource Groups	-
------------	---------	--------------------------	---	-----------------	---

Step 5: Add notifications [Edit](#)

Notifications

No notifications

Step 6: Add tags [Edit](#)

Tags (0)

Key	Value	Tag new instances
No tags		

[Preview code](#) [Cancel](#) [Previous](#) [Create Auto Scaling group](#)

Figura 54. Ingresamos a revisar el AG

The screenshot shows the 'AGRenington1 Capacity overview' page in the AWS Management Console. The page is divided into several sections:

- Launch template:** Shows the AMI ID (ami-0bc11797f1dc7c7f3), Instance type (t2.micro), and Security group ID (sg-075d1968134d329e1).
- Network:** Shows the Availability Zone (us-east-1a, us-east-1b), Subnet ID (subnet-0c732626b-4d880d4, subnet-0d66c94e7455106f), and Availability Zone distribution (Balanced best effort).
- Health checks:** Shows the Health check type (EC2, ELB) and Health check grace period (200). A red arrow points to this section.

Figura 55.

Para que suba cuando detecte un servicio mal como la detención de la web y mal rendimientos de la maquina

The screenshot shows the 'Edit AGRenington1' page in the AWS Management Console. The page displays the 'Health checks - optional' section, which includes the following options:

- EC2 health checks:** Always enabled.
- Additional health check types - optional:**
 - Turn on Elastic Load Balancing health checks: This enables Elastic Load Balancing to report on unhealthy instances. EC2 Auto Scaling can replace it on its next periodic check.
 - Turn on Amazon EBS health checks: This enables Amazon EBS to report on unhealthy volumes. EC2 Auto Scaling can replace the instance on its next periodic health check.
- Health check grace period:** 200 seconds.

A red arrow points to the 'Turn on Elastic Load Balancing health checks' option.

Create dynamic scaling policy

Policy type
Target tracking scaling

Scaling policy name
Target Tracking Policy

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.
Average CPU utilization

Target value
50

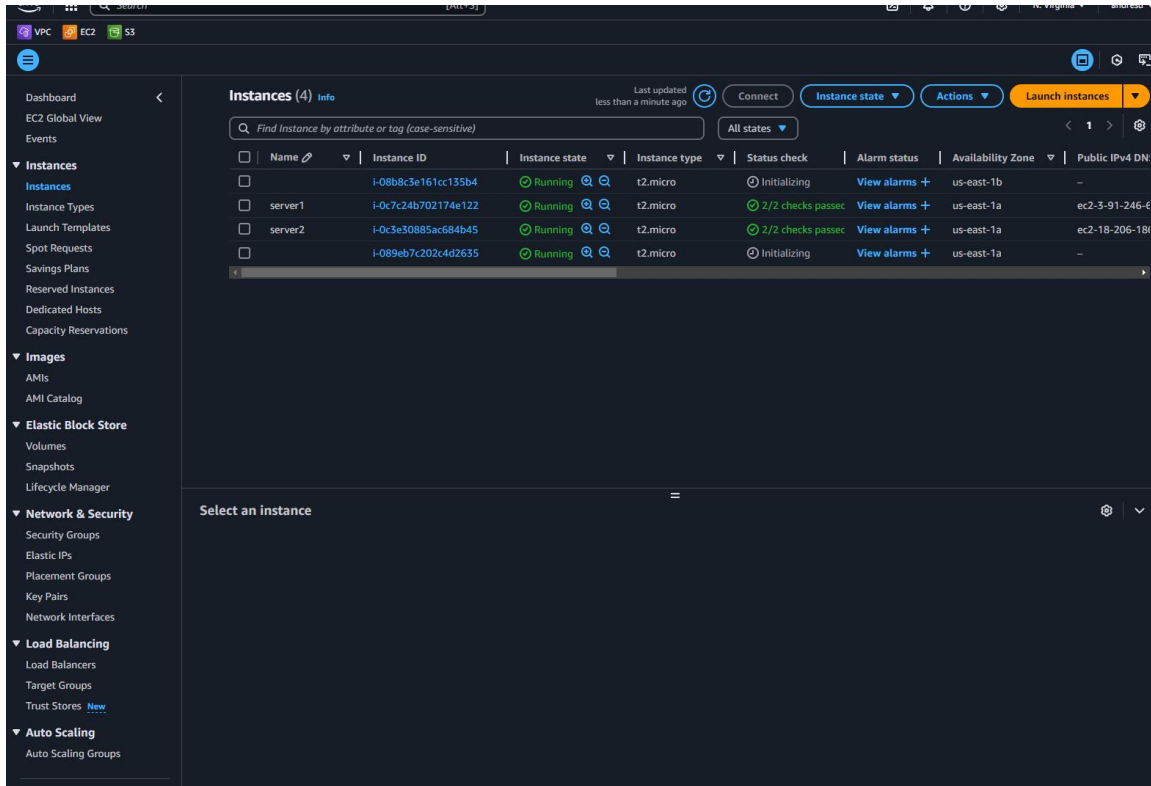
Instance warmup Info
300 seconds

Disable scale in to create only a scale-out policy

[Cancel](#) [Create](#)

Figura 56.

Luego vemos que en las instancias se crearon las dos que le indicamos en este caso 2 para un total de 4



The screenshot shows the AWS Management Console interface for the EC2 Instances page. The left sidebar contains navigation options such as Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, Load Balancers, Target Groups, Trust Stores, and Auto Scaling, Auto Scaling Groups.

The main content area displays the 'Instances (4) Info' page. At the top, there are buttons for 'Connect', 'Instance state', 'Actions', and 'Launch instances'. Below these is a search bar and a filter dropdown set to 'All states'. The table below shows the following data:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DN
	i-08b8c3e161cc135b4	Running	t2.micro	Initializing	View alarms +	us-east-1b	-
server1	i-0c7c24b702174e122	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-3-91-246-6
server2	i-0c3e30885ac684b45	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-18-206-18
	i-089eb7c202c4d2635	Running	t2.micro	Initializing	View alarms +	us-east-1a	-

Below the table, there is a 'Select an Instance' section with a search bar and a dropdown menu.

Figura 57.
En el balanceador de carga vemos las 4

The screenshot shows the AWS Management Console interface for a Target Group named 'TGReming1'. The 'Details' section indicates 4 healthy targets, 0 unhealthy, 0 unused, 0 initial, and 0 draining. Below this, a table lists the registered targets:

Instance ID	Name	Port	Zone	Health status	Health status details	Admini...	Overri...
i-089eb7c202c4d2635		80	us-east-1a (us...)	Healthy	-	No override	No overri...
i-08b8c3e161cc135b4		80	us-east-1b (us...)	Healthy	-	No override	No overri...
i-0c3e30885ac684b45	server2	80	us-east-1a (us...)	Healthy	-	No override	No overri...
i-0c7c24b702174e122	server1	80	us-east-1a (us...)	Healthy	-	No override	No overri...

Figura 58.
Ahora podemos eliminar las instancias creadas inicialmente por que ya tenemos el balanceador de carga ok

The screenshot shows the 'Instances' page for the Target Group. The table lists five instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring	Security group name
	i-08b8c3e161cc135b4	Terminated	t2.micro	-	View alarms	us-east-1b	-	-	-	-	disabled	-
	i-018da0fba3dfaa17	Running	t2.micro	-	View alarms	us-east-1b	-	-	-	-	disabled	SG_BalancedorCG
server1	i-0c7c24b702174e122	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	-	-	-	-	disabled	SG_BalancedorCG
server2	i-0c3e30885ac684b45	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	-	-	-	-	disabled	SG_BalancedorCG
	i-089eb7c202c4d2635	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	-	-	-	-	disabled	SG_BalancedorCG

Figura 59.
Creamos una maquina virtual para conectar a las instancias de balanceador

Figura 60.

Desde la instancia creada conectamos a las instancias de los balanceadores para ello pasamos el certificado a la maquina y le damos permisos y conectamos por ssh

```

ec2-user@ip-10-0-196-207:~
@
WARNING: UNPROTECTED PRIVATE KEY FILE!
Permissions 0644 for 'cer.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "cer.pem": bad permissions
ec2-user@10.0.196.207: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[root@ip-10-0-9-201 ec2-user]# chmod 444 cer.pem
[root@ip-10-0-9-201 ec2-user]# ssh -i cer.pem ec2-user@10.0.196.207
@
WARNING: UNPROTECTED PRIVATE KEY FILE!
Permissions 0444 for 'cer.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "cer.pem": bad permissions
ec2-user@10.0.196.207: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[root@ip-10-0-9-201 ec2-user]# chmod 400 cer.pem
[root@ip-10-0-9-201 ec2-user]# ssh -i cer.pem ec2-user@10.0.196.207

#_
~\  ###_      Amazon Linux 2023
~~\  #####\
~~  \###|
~~   \#/      https://aws.amazon.com/linux/amazon-linux-2023
~~    V~'  '->
~~~~
~~  .-.-
~~  /m/'

Last login: Sat Nov 30 22:01:18 2024 from 181.134.136.197
[ec2-user@ip-10-0-196-207 ~]#

```

Figura 61. Revisamos los servicios web y los detenemos

```

root@ip-10-0-196-207/home/ec2-user
[root@ip-10-0-196-207 ec2-user]# systemctl status httpd
bash: systemctl: command not found
[root@ip-10-0-196-207 ec2-user]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: active (running) since Sun 2024-12-01 00:31:10 UTC; 54min ago
     Docs: man:httpd.service(8)
  Main PID: 1998 (httpd)
   Status: "Total requests: 4106; Idle/Busy workers 100/0; Requests/sec: 1.25; Bytes served: 1000000"
    Tasks: 177 (limit: 1111)
  Memory: 21.4M
     CPU: 3.271s
   CGroup: /system.slice/httpd.service
           └─1998 /usr/sbin/httpd -DFOREGROUND
             └─2008 /usr/sbin/httpd -DFOREGROUND
               └─2009 /usr/sbin/httpd -DFOREGROUND
                 └─2010 /usr/sbin/httpd -DFOREGROUND
                   └─2011 /usr/sbin/httpd -DFOREGROUND

Dec 01 00:31:09 ip-10-0-10-207.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server:
Dec 01 00:31:10 ip-10-0-10-207.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server:
Dec 01 00:31:10 ip-10-0-10-207.ec2.internal httpd[1998]: Server configured, listening on: port 80
lines 1-19/19 (END)
^C
[root@ip-10-0-196-207 ec2-user]# systemctl sttop httpd
Unknown command verb sttop.
[root@ip-10-0-196-207 ec2-user]# systemctl stop httpd
[root@ip-10-0-196-207 ec2-user]#

```

Figura 62.

Vemos que detecta detenido y lo vuelve a subir

The screenshot shows the AWS Management Console interface for a Target Group. The 'Details' section indicates 3 total targets, with 2 healthy, 0 unhealthy, 0 unused, 0 initial, and 1 draining. Below this, the 'Distribution of targets by Availability Zone (AZ)' is shown. The 'Registered targets' table lists three targets:

Instance ID	Name	Port	Zone	Health status	Health status details	Adminis...	Overrid...	Launch...
i-0ecdf1806537dbb3		80	us-east-1b (us...)	Healthy		No override...	No overrid...	November...
i-075daf7e01c26011a		80	us-east-1b (us...)	Draining	Target deregistration I...	No override...	No overrid...	November...
i-0af7ab14e8f21d50		80	us-east-1a (us...)	Healthy		No override...	No overrid...	November...

Video entrega dos balanceador

https://youtu.be/9_f4AFg75PM

Entrega final 3

Docker

Figura 63

Instalando nginx

```
[root@ip-10-0-13-122 html]# dnf install nginx -y
```

Figura 64.

Subimos el servicio

```
[root@ip-10-0-13-122 html]# systemctl start nginx
```

Figura 65.

Comprobamos el servicio

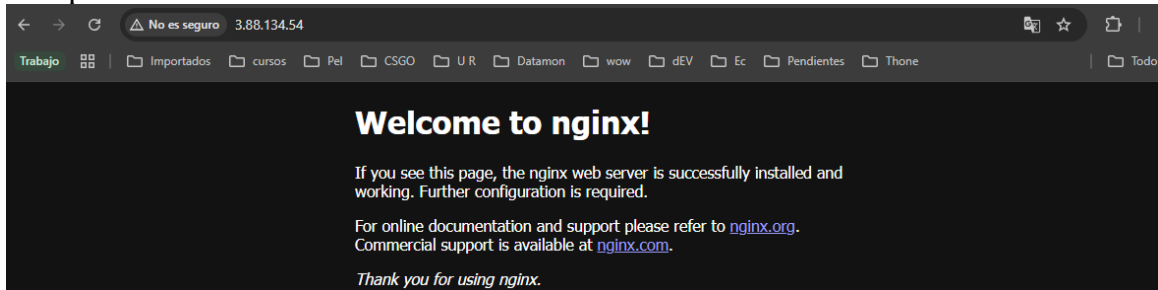


Figura 66.

Instalamos Docker

```
Complete!
[root@ip-10-0-13-122 html]# yum install docker -y
```

Figura 67.

Subimose el servicio

```
Complete!
[root@ip-10-0-13-122 html]# systemctl start docker
[root@ip-10-0-13-122 html]# systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service - /usr/lib/systemd/system/docker.service.
[root@ip-10-0-13-122 html]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: d
   Active: active (running) since Sun 2024-12-08 15:58:40 UTC; 23s ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 29419 (dockerd)
     Tasks: 7
    Memory: 38.6M
      CPU: 271ms
    CGroup: /system.slice/docker.service
```

Figura 68.

Creamos un contenedor por medio de una imagen de Docker se puede descargar desde un hub.

The screenshot shows the Docker Hub interface for the 'httpd' image. At the top, there's a navigation bar with the Docker Hub logo and a search bar containing 'httpd'. Below the navigation bar, the page title is 'httpd Docker Official Image' with a download count of '1B+' and a star count of '4.8K'. The main content area is divided into several sections:

- Quick reference:**
 - Maintained by: [the Docker Community](#)
 - Where to get help: [the Docker Community Slack](#), [Server Fault](#), [Unix & Linux](#), or [Stack Overflow](#)
- Supported tags and respective Dockerfile links:**
 - 2.4.62, 2.4, 2, latest, 2.4.62-bookworm, 2.4-bookworm, 2-bookworm, bookworm
 - 2.4.62-alpine, 2.4-alpine, 2-alpine, alpine, 2.4.62-alpine3.20, 2.4-alpine3.20, 2-alpine3.20, alpine3.20
- Quick reference (cont.):**
 - Where to file issues: <https://github.com/docker-library/httpd/issues>
 - Supported architectures: [\(more info\)](#)
 - amd64, arm32v5, arm32v6, arm32v7, arm64v8, i386, mips64le, ppc64le, riscv64, s390x
 - Published image artifact details:
- Recent tags:** latest, bookworm, 2.4.62-bookworm, 2.4.62, 2.4-bookworm, 2.4, 2-bookworm, 2, alpine3.20, alpine
- About Official Images:** Docker Official Images are a curated set of Docker open source and drop-in solution repositories.
- Why Official Images?:** These images have clear documentation, promote best practices, and are designed for the most common use cases.

Figura 69.

Descargamos la imagen

```
[root@ip-10-0-13-122 html]# docker pull httpd
Using default tag: latest
latest: Pulling from library/httpd
bc0965b23a04: Pull complete
d7ad38c6dd97: Pull complete
4f4fb700ef54: Pull complete
79b49624e34b: Pull complete
7d9f97915db2: Pull complete
9bd25d4f7b77: Pull complete
Digest: sha256:f4c5139eda466e45814122d9bd8b886d8ef6877296126c09b76dbad72b03
Status: Downloaded newer image for httpd:latest
docker.io/library/httpd:latest
```

Figura 70.

Comprobamos las imagenes

```
[root@ip-10-0-13-122 html]# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
httpd         latest   494b2b45fd74   4 months ago  147MB
[root@ip-10-0-13-122 html]# ~
```

Figura 71.

Creamos una carpeta para guardar la informacion

```
[root@ip-10-0-13-122 html]# cd /
[root@ip-10-0-13-122 /]# mkdir appl
```

Figura 72.

Iniciamos a crear el contenedor

Por medio del comando `-d` (segundo plano) `it` (sea interactivo) `-name` (para el nombre) `-p` (indicar puerto) `-v` volumen (donde indicamos la ruta local y la ruta del servicio) `--restart always` (para que suban solos los servicios)

```
[root@ip-10-0-13-122 /]# docker run -dit --name appl --restart always -p 8080:80 -v /appl:/usr/local/apache2/htdocs/ httpd
7a177d469368895c0ca555c3d36b9db10a1da482e08871e20acb4a1007d424576
[root@ip-10-0-13-122 /]#
```

Figura 73.

revisamos la instancia

```
[root@ip-10-0-13-122 /]# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
7a177d469368   httpd    "httpd-foreground"      23 seconds ago Up 22 seconds 0.0.0.0:8080->80/tcp, :::8080->80/tcp   appl
[root@ip-10-0-13-122 /]#
```

Figura 74.

Apagar servicio de nginx para evitar conflictos de puertos

```
[root@ip-10-0-13-122 appl]# systemctl stop nginx
```

Figura 75.

Reiniciamos También la instancia de Docker y creamos otra instancia con puerto 80:80

```
[root@ip-10-0-13-122 appl]# docker stop 7a177d469368
7a177d469368
-FWXFWXFWX. 1 root root 17 Dec 8 16:21 index.html
[root@ip-10-0-13-122 appl]# docker run -dit --name app2 --restart always -p 80:80 -v /appl:/usr/local/apache2/htdocs/ httpd
f4cec4356b0de8d8b0b1b492c8a01b6e426319bd5be849623b196c7f44db4294
[root@ip-10-0-13-122 appl]#
```

Figura 76.

Iniciamos proceso de configurar proxy reverso con nginx.

```
[root@ip-10-0-13-122 app2]# systemctl start nginx
[root@ip-10-0-13-122 app2]# cd /etc/nginx
[root@ip-10-0-13-122 nginx]# ls
mime.types  fastcgi.conf  fastcgi_params  koi-utf  mime.types  nginx.conf  scgi_params  uwsgi_params  win-utf
default.conf  fastcgi.conf.default  fastcgi_params.default  koi-win  mime.types.default  nginx.conf.default  scgi_params.default  uwsgi_params.default
[root@ip-10-0-13-122 nginx]#
```

Figura 77.

Indicamos los puertos y configuración del nginx

```

GNU nano 5.8                               nginx.conf
#
# * Official English Documentation: http://nginx.org/en/docs/
# * Official Russian Documentation: http://nginx.org/ru/docs/
#
events {}

http {
    upstream backend {
        server localhost:8080;
        server localhost:8082;
        server localhost:8083;
    }

    server {
        listen 80;
        server_name nginx;

        location / {
            proxy_pass http://backend;
        }
    }
}

# Settings for a TLS enabled server.
#
# server {
#     listen 443 ssl;
#     listen [::]:443 ssl;
#     http2 on;
#     server_name _;
#     root /usr/share/nginx/html;
#
#     ssl_certificate "/etc/pki/nginx/server.crt";
#     ssl_certificate_key "/etc/pki/nginx/private/server.key";
# }

```

Figura 78.

Verificamos funcionamiento

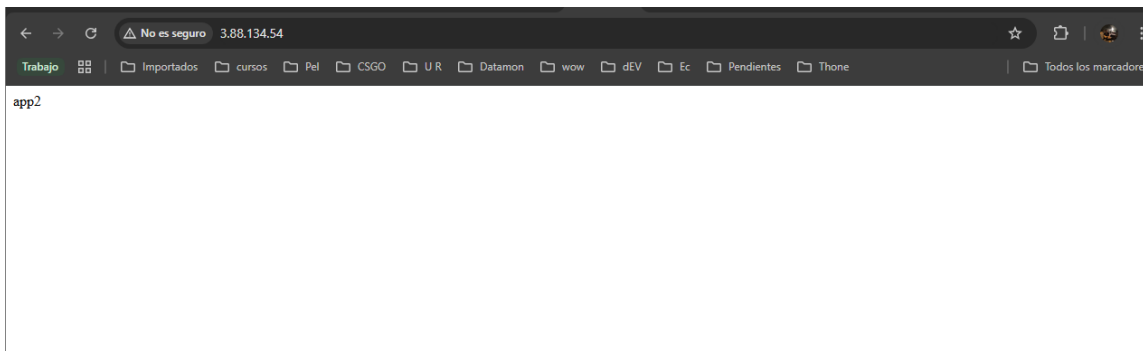


Figura 79.

Realizamos el proceso de crear nuevamente la ami (como la imagen 20)

Amazon Machine Images (AMIs) (2) Info							
Name	AMI name	AMI ID	Source	Owner	Visibility	Status	Creation date
<input type="checkbox"/>	Linux	AmazonLinuxAndres	ami-08c117691de57c7f3	354918403906/AmazonLinuxAndres	354918403906	Private	Available 2024/11/30 18:44 GMT-5
<input type="checkbox"/>	Docker	LinuxDocker	ami-0753b9c8dafda97f5	354918403906/LinuxDocker	354918403906	Private	Available 2024/12/08 13:11 GMT-5

Figura 80. Agregamos la ami para que acople al auto balanceador con los ajustes realizados el Docker

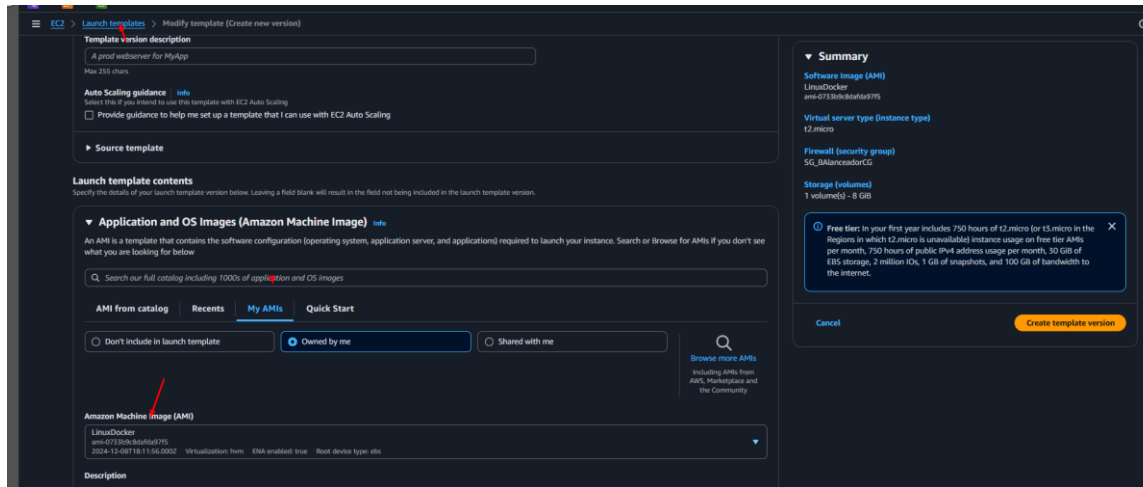


Figura 81.

Comprobamos que al detener los servicios internos de la instancias este nos registre y las vuelva a subir

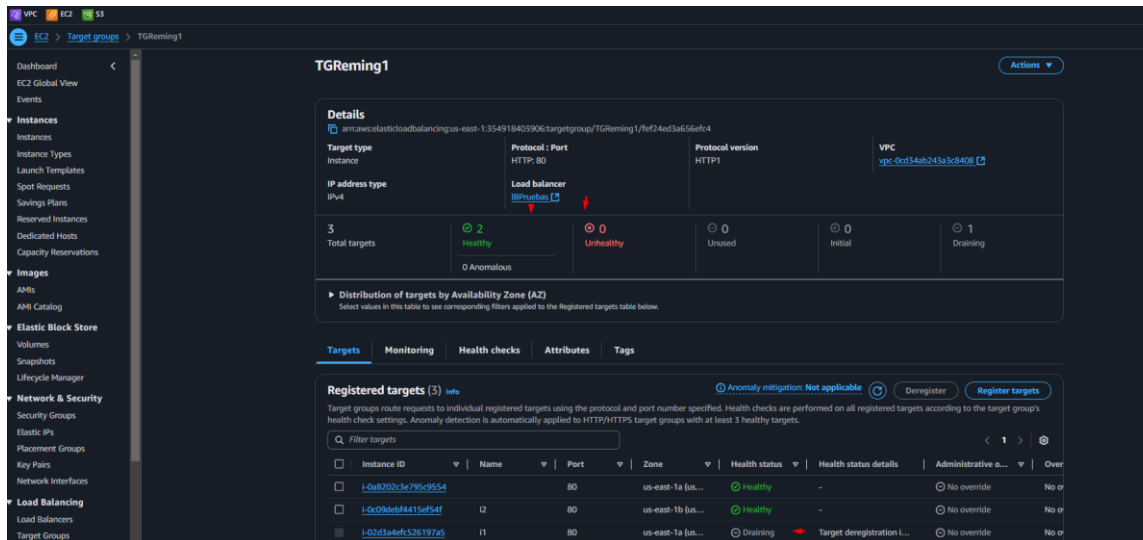


Figura 82.confirmamos siempre la alta disponibilidad en todo momento.

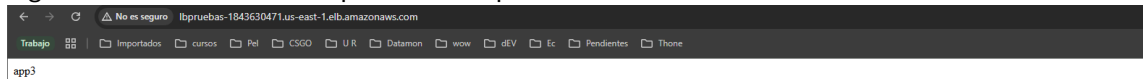
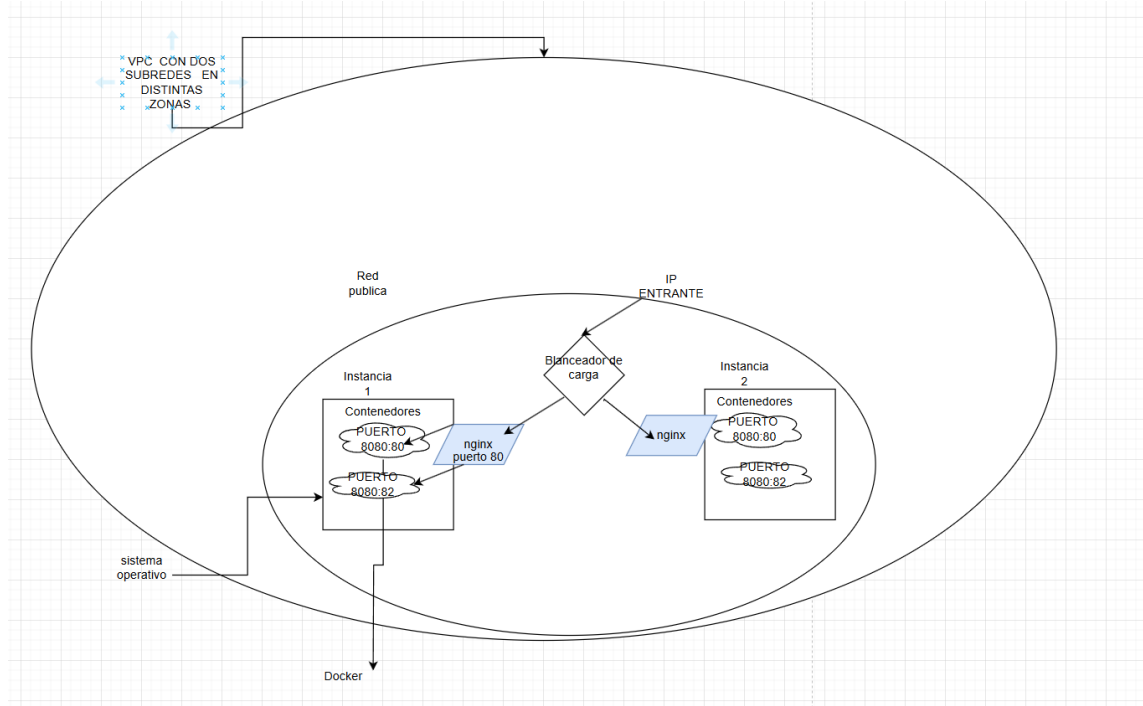


Diagrama Funcional



[Video entrega 3 Final](#)

Conclusiones

Encontramos durante el seminario la importancia de este modelo de alta disponibilidad permitiendo tener una serie de Backup de modo que si una instancia falla casi de manera instantánea automáticamente se sube otra instancia esto ayuda a optimizar la buena experiencia de usuarios y la gran calidad de conectividad ayudando de esta forma a las empresas o personas que usen este modelo de aws a mejorar sus ganancias y potenciar su calidad de estructura tecnológica ya que en este modelo las conectividades siempre están disponibles.

Referencias

(Aws, 2024)
(docker, 2024)