



**TRABAJO DE GRADO**  
**Opción Seminario-Diplomado.**

**Implementación de Soluciones Cloud con Amazon AWS: alta disponibilidad,  
escalabilidad y elasticidad en infraestructura**

Corporación Universitaria Remington.

Ingeniería de Sistemas

SEMINARIO AMAZON AWS

Astrid Daniela Castellanos Mantilla.

Juan Pablo Berrios López

Seminario de grado

2024

## Tabla de contenido

<b>Resumen</b> .....	1
<b>Palabras clave</b> .....	2
<b>1. Marco conceptual y contextual</b> .....	2
<b>2. Desarrollo e implementación del aprendizaje</b> .....	5
2.1. Creación de una instancia.....	5
a) Registro y creación de cuenta ROOT (raíz) en AWS.....	5
b) Registro y creación de cuenta de cuanta por medio del servicio Identity and Access Management IAM.....	9
c) Crear un VPC.....	12
d) Crear una instancia .....	14
2.2. Pruebas de la instancia que se crea automática y aplicación correctamente con Auto Scaling Groups (ASG) y Target Groups .....	19
Pruebas.....	20
2.3. Trabajo practico: Implementación de Soluciones Cloud con Amazon AWS: alta disponibilidad, escalabilidad y elasticidad en infraestructura .....	23
□ Objetivo general: .....	24
□ Objetivos específicos: .....	24
2.3.1. Diagrama de recursos .....	25
2.3.2. Pruebas de trabajo práctico.....	25
<b>Figuras y tablas</b> .....	27
<b>Conclusiones</b> .....	30
<b>Referencias</b> .....	31

## Resumen

Este diplomado se estructuró mediante un hilo secuencial para el uso de herramientas mediante el diseño y ejecución de laboratorios propuestos para la aplicación de conceptos fundamentales tecno-prácticos de alta **disponibilidad**, **escalabilidad** y **elasticidad** en una infraestructura basada en **AMAZON Web Services (AWS)**.

Se desarrolló habilidades avanzadas en la configuración y gestión de instancias virtuales en **Amazon EC2**, diseño y despliegue de redes virtuales utilizando **Amazon Virtual Private Cloud (VPC)**, creación y segmentación de subredes públicas y privadas, configuración de Internet **Gateways NAT**.

Uso de herramientas como **AWS Identity and Access Management (IAM)** para la administración segura, la implementación de mecanismos de seguridad como **Security Groups**. La integración y gestión de contenedores mediante **Docker**, explorando cómo optimizar el desarrollo y la implementación de aplicaciones en entornos distribuidos.

Se abarcó la configuración e implementación de **Auto Scaling Groups (ASG)** para ajustar dinámicamente la capacidad de los recursos según la demanda.

Asimismo, se trabajó con **Target Groups** para optimizar la asignación de instancias a los balanceadores de carga, asegurando la estabilidad y eficiencia de las aplicaciones desplegadas en la nube.

## Palabras clave

1. Infraestructura en la nube.
2. Almacenamiento.
3. Servidores.
4. Zona de disponibilidad.
5. Contenedores.
6. Seguridad.

### 1. Marco conceptual y contextual

- **Computación en la nube (o cloud computing)** es un modelo de servicios tecnológicos a través de internet, donde los recursos como servidores, almacenamiento, bases de datos, redes, software y análisis, están disponibles bajo la necesidad del usuario. En lugar de poseer y gestionar infraestructura física, las organizaciones pueden acceder a estos recursos desde plataformas pago por uso.
- **Amazon Web Services (AWS)** uno de los principales proveedores de **computación en la nube** que contribuye significativamente al desarrollo y mejora de las infraestructuras tecnológicas en las áreas de TI, ofrece una amplia gama en servicios tecnológicos bajo un modelo de pago por uso. **AWS** se ha convertido en una de las plataformas más utilizadas a nivel global debido a su escalabilidad, seguridad, fiabilidad y flexibilidad.

- **Red global de regiones de AWS:** AWS cuenta con la infraestructura en la nube más amplia a nivel mundial. El modelo de regiones y zonas de disponibilidad de AWS es reconocido como el enfoque recomendado para ejecutar aplicaciones empresariales que requieren de una alta disponibilidad. (Amazon Web Services (AWS), 2024)
- **Amazon Virtual Private Cloud (VPC):** Es un servicio que puede lanzar recursos de AWS en una red virtual aislada de manera lógica que haya definido. Esta red virtual es muy similar a la red tradicional que usaría en su propio centro de datos, pero con los beneficios que supone utilizar la infraestructura escalable de AWS. (AWS, 2024)
- **Amazon EC2:** Una instancia de EC2 es un servidor virtual en la nube de AWS. (AWS, 2024)
- **Gateways NAT:** Es un servicio de traducción de direcciones de red (NAT). Puede utilizar una gateway NAT para que las instancias de una subred privada puedan conectarse a servicios fuera de la VPC, pero los servicios externos no pueden iniciar una conexión con esas instancias. (AWS, 2024)
- **AWS Identity and Access Management (IAM):** Es un Servicio de AWS que ayuda a los gestores a controlar de forma segura el acceso a los recursos de AWS. Los administradores de IAM controlan quién está *autenticado* (ha iniciado sesión) y *autorizado* (tiene permisos) para utilizar recursos de Amazon VPC. (AWS, 2024)

- **Grupos de seguridad en AWS:** Es una función de seguridad integrada dentro de varios Importancia en **EC2** y **VPC**. Actúan como **firewalls virtuales** para controlar el tráfico entrante y saliente de las instancias asociadas. (AWS, 2024)
- **Elastic Load Balancing (ELB)** es un servicio que distribuye automáticamente el tráfico de red entrante entre múltiples instancias de servidores (como EC2) para garantizar una alta disponibilidad y escalabilidad de las aplicaciones. (AWS, 2024)
- **Docker** es una plataforma de software que permite automatizar el despliegue, escalado y gestión de aplicaciones mediante el uso de contenedores. Un contenedor es una unidad estándar de software que empaqueta el código y todas sus dependencias (como bibliotecas, herramientas y configuraciones) para que la aplicación pueda ejecutarse de manera rápida y confiable en cualquier entorno, ya sea en desarrollo, prueba o producción. (DOCKER, 2024)

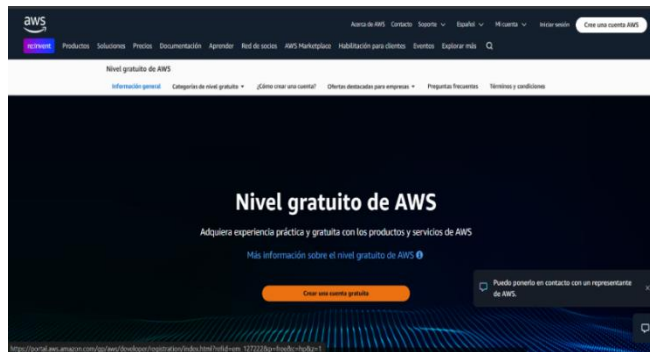
## 2. Desarrollo e implementación del aprendizaje

### Entrega uno

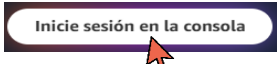
#### 2.1. Creación de una instancia

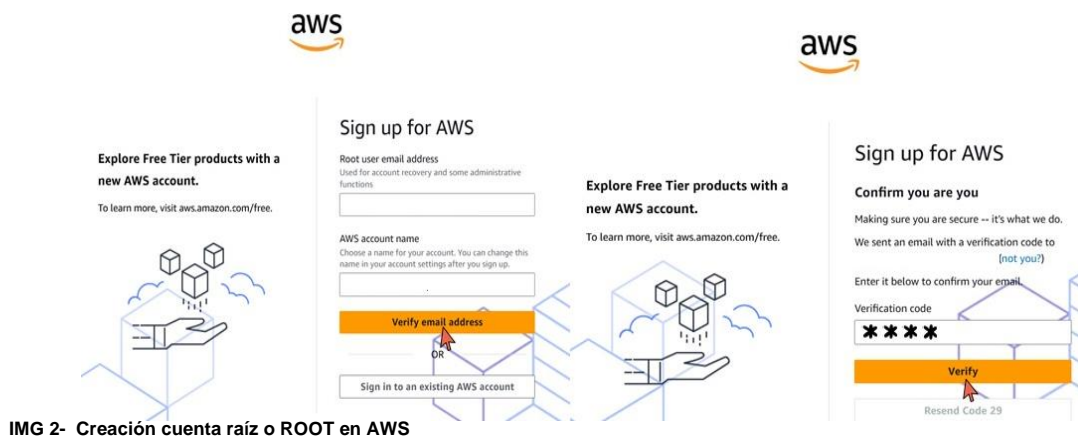
##### a) Registro y creación de cuenta ROOT (raíz) en AWS

Para dar inicio al proceso ingresa desde cualquier buscador a la página oficial de AWS <https://aws.amazon.com/es/>



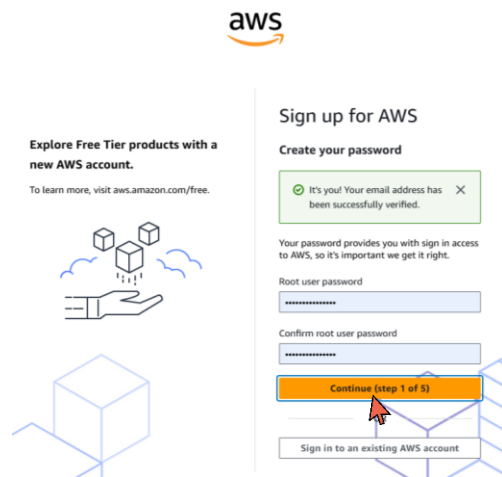
IMG 1- página inicio AWS

- **Paso 1:** Se ubica en la parte superior de la ventana al lado izquierdo en el botón 
- **Paso 2:** Se abrirá un formulario que solicita un correo electrónico, nombre del usuario y verificación mediante un código enviado al correo registrado



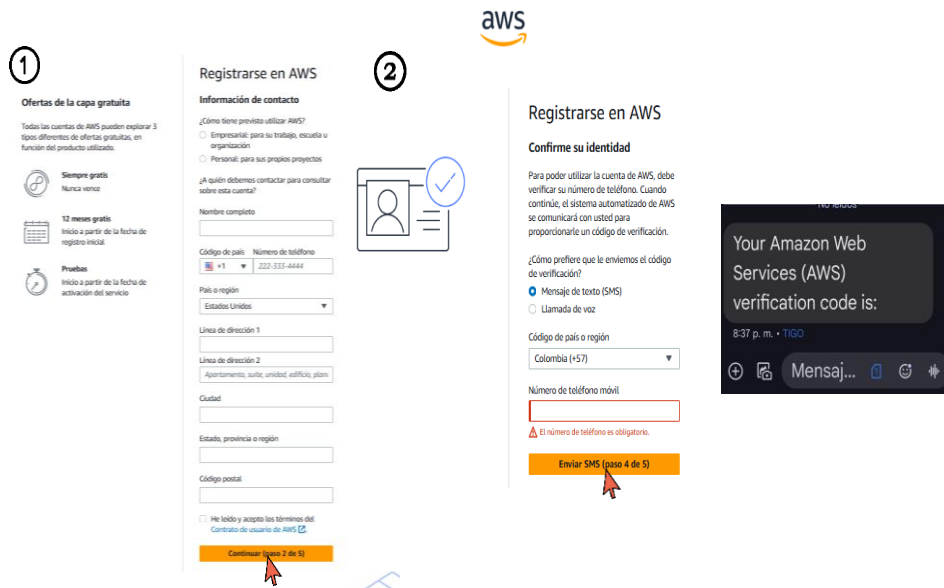
IMG 2- Creación cuenta raíz o ROOT en AWS

- **Paso 3:** Se asigna una clave de cuenta



IMG 3- Creación contraseña cuenta raíz o ROOT en AWS

- **Paso 4:** Se proporciona la información personal del usuario y confirmación de identidad mediante un mensaje de texto (SMS)



IMG 4- Autenticación de cuenta raíz o ROOT en AWS por SMS




- **Paso 5:** Selecciona un plan de soporte y se finaliza el proceso de registro y creación de cuenta ROOT (raíz) en **AWS**



## Registrarse en AWS

### Seleccionar un plan de soporte

Elija un plan de soporte para su cuenta personal o empresarial. [Compare planes y ejemplos de precio](#). Puede cambiar su plan en cualquier momento desde la consola de administración de AWS.

<input checked="" type="radio"/> <b>Soporte de nivel Basic: gratis</b> <ul style="list-style-type: none"><li>Recomendado para los usuarios nuevos que recién comienzan a utilizar AWS</li><li>Acceso de autoserivicio las 24 horas del día, los 7 días de la semana a los recursos de AWS</li><li>Solo para problemas de facturación y cuentas</li><li>Acceso a Personal Health Dashboard y Trusted Advisor</li></ul> 	<input type="radio"/> <b>Soporte Developer: a partir de 29 USD al mes</b> <ul style="list-style-type: none"><li>Recomendado para desarrolladores que experimentan con AWS</li><li>Acceso por correo electrónico a AWS Support durante el horario laboral</li><li>Tiempos de respuesta de 12 horas (horario laboral)</li></ul> 	<input type="radio"/> <b>Soporte Business: a partir de 100 USD al mes</b> <ul style="list-style-type: none"><li>Recomendado para ejecutar cargas de trabajo de producción en AWS</li><li>Soporte técnico las 24 horas, los 7 días de la semana por correo electrónico, teléfono y chat</li><li>Tiempos de respuesta de 1 hora</li><li>Conjunto completo de recomendaciones de prácticas de Trusted Advisor</li></ul> 
---	---	--



#### ¿Necesita soporte de nivel Enterprise?

A partir de los 15 000 USD por mes, tendrá tiempos de respuesta de 15 minutos y una experiencia de consejería con un director técnico de cuenta asignado. [Más información](#)



## ¡Felicitaciones!

Gracias por registrarse en AWS.


Estamos activando la cuenta. Tardaremos solo unos minutos. Recibirá un correo electrónico cuando hayamos acabado.


[Ir a la consola de administración de AWS](#)

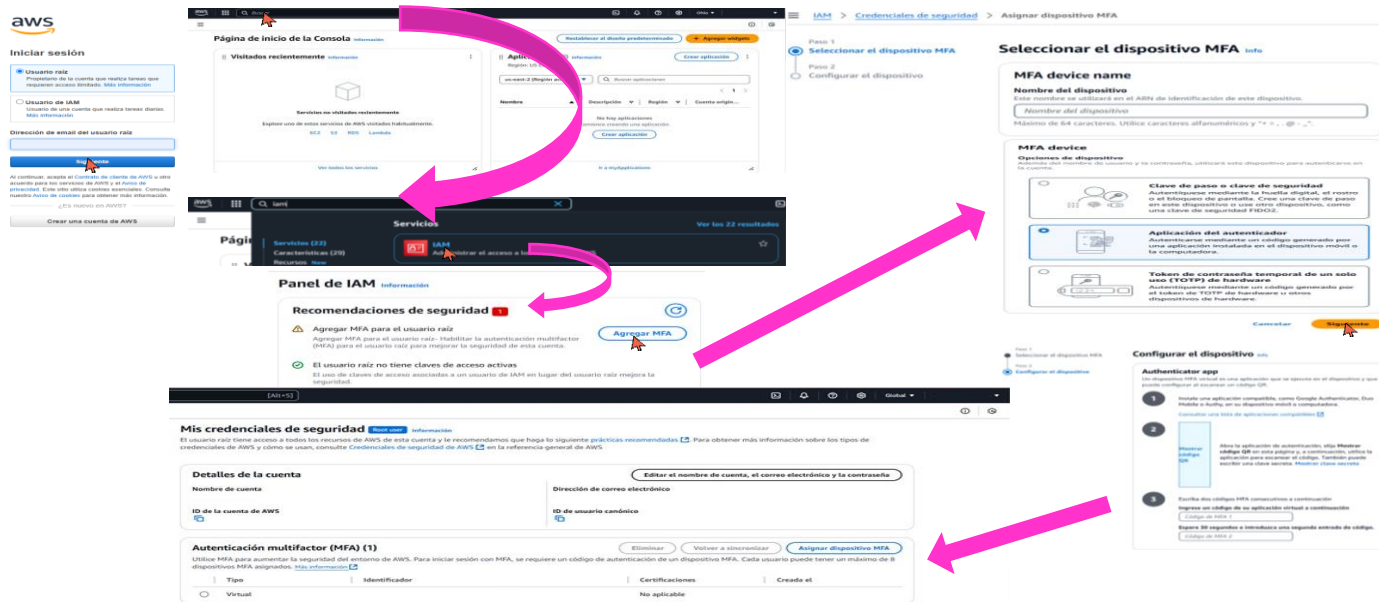
[Registrarse en otra cuenta](#) o [contactar con ventas](#)

[Finalizar registro](#)

IMG 5- creación exitosa de la cuenta raíz o ROOT en AWS por SMS

**Nota:** Proceso recomendado de seguridad para la cuenta ROOT (raíz) en **AWS**, por medio **MFA (Multi Factor Authentication)** o factor de autenticación de dos pasos, el cual se vinculara mediante un android por medio de una aplicación en este caso daremos uso a  google Authenticator.

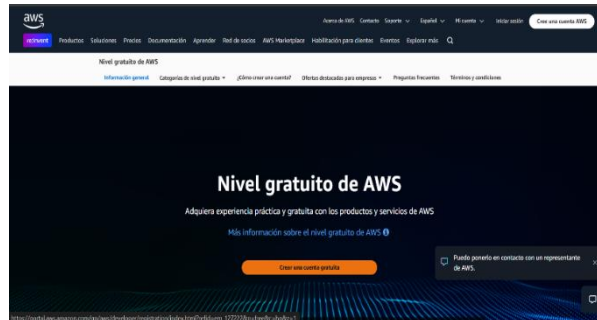
Luego de iniciar sesión desde la cuenta ROOT (raíz), en la barra de búsqueda se ingresa **IAM**, donde se despliega la vista del panel de **IAM**, en el  botón se seleccionara una opción de autenticación por medio de un dispositivo para **MFA**, es como se genera una credencial de seguridad.



IMG 6- Asignación de Multi Factor Authentication a cuenta raíz o ROOT en AWS por SMS

## b) Registro y creación de cuenta de cuenta por medio del servicio Identity and Access Management IAM

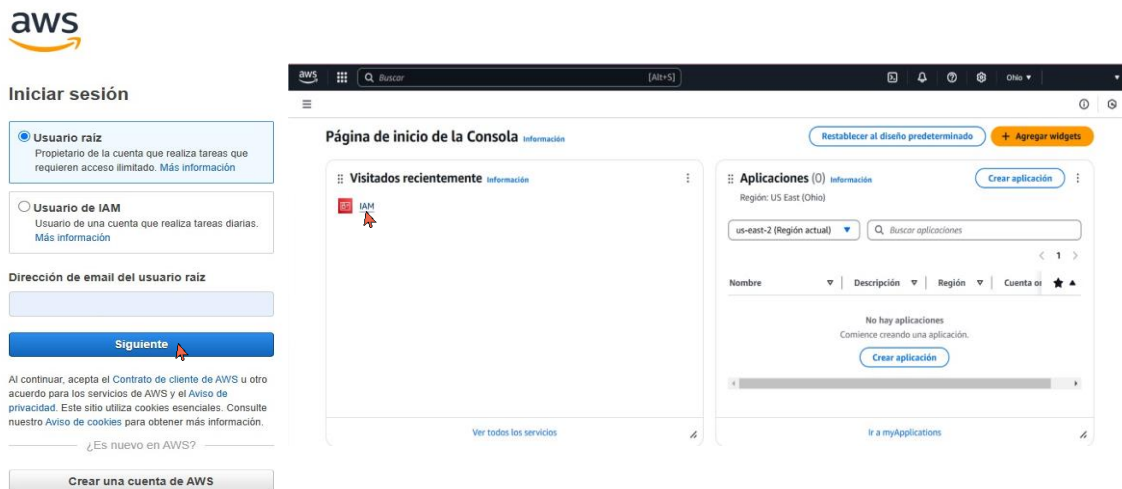
Para dar inicio al proceso ingresa desde cualquier buscador a la página oficial de AWS <https://aws.amazon.com/es/>



IMG 1- página inicio AWS

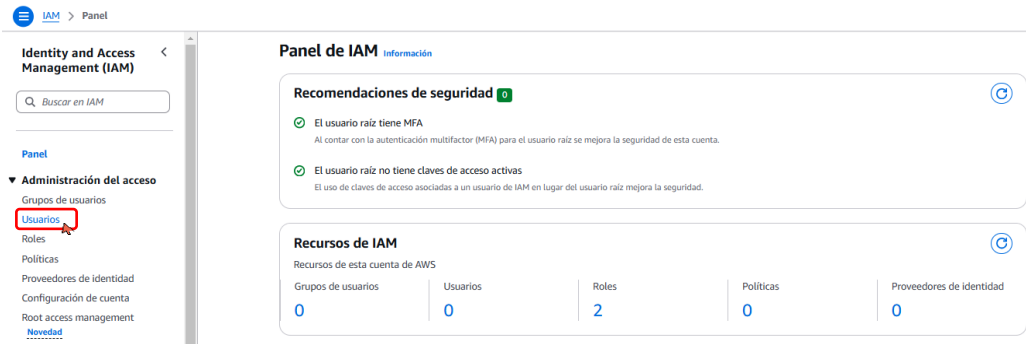
Se ubica en la parte superior de la ventana al lado izquierdo en el Inicie sesión en la consola

- **Paso 1:** Luego del ingreso a la cuenta raíz mediante la consola de AWS se ingresa al servicio IAM



IMG 7- Búsqueda de la IAM

- **Paso 2:** Seleccionar la opción de usuarios en el menú ubicado al lado izquierdo del portal



IMG 8 - Búsqueda opción usuario para la creación de la IAM

- **Paso 3:** Seleccionar la opción de crear nuevo usuario ubicado al lado derecho **Crear usuario**

- **Paso 4:** se diligencia el formulario con la información solicitada este paso consta de cuatro pasos

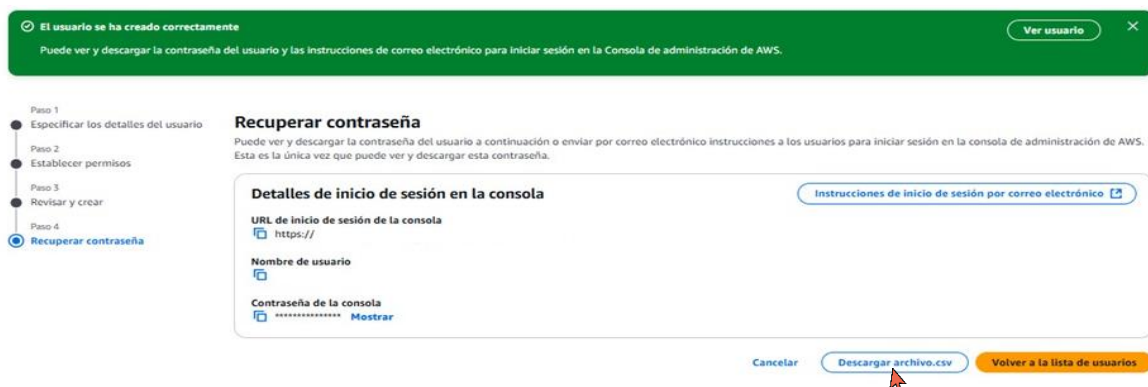
➤ **Paso 1:** Especificar los detalles del usuario



IMG 9 - Especificación de detalles de usuario IAM



- **Paso 4:** Recuperar contraseña, en este paso es muy importante guardar de forma segura la llave privada SSH (ya que esta contiene los permisos para poder acceder a nuestra cuenta IAM) la cual se descarga en formato .PPK ya que en esta ocasión el sistema operativo del pc es Windows es el formato que corresponde para usar junto con la herramienta PUTTY.
- **Nota:** si se pierde la llave SSH, se pierde el acceso y por ende no tenemos forma de recuperar el acceso a la cuenta.



IMG 12 - creación de la llave SSH

### c) Crear un VPC

Para dar inicio al proceso ingresa desde cualquier buscador a la página oficial de AWS <https://aws.amazon.com/es/>

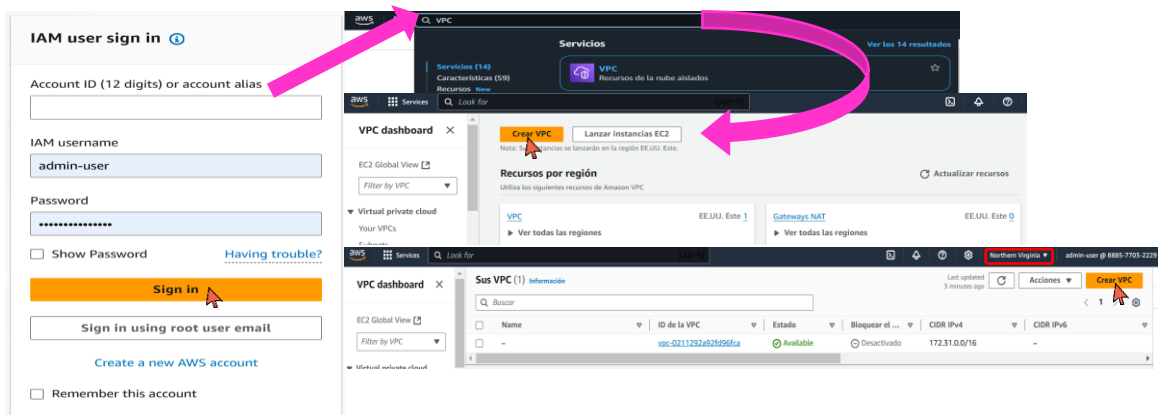


IMG 1- página inicio AWS

Se ubica en la parte superior de la ventana al lado izquierdo en el botón

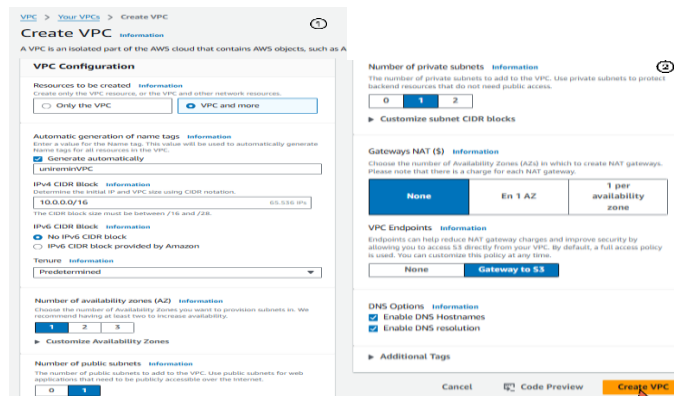
Inicie sesión en la consola

- **Paso 1:** Luego de dar inicio de sesión en una cuenta AWS existente en la barra de búsqueda **VPC** (Recursos de nube aislados), ya en el **VPC dashboard** damos clic en **crear VPC**, tenemos en cuenta la región donde estamos localizados y damos nuevamente clic en **crear VPC**.

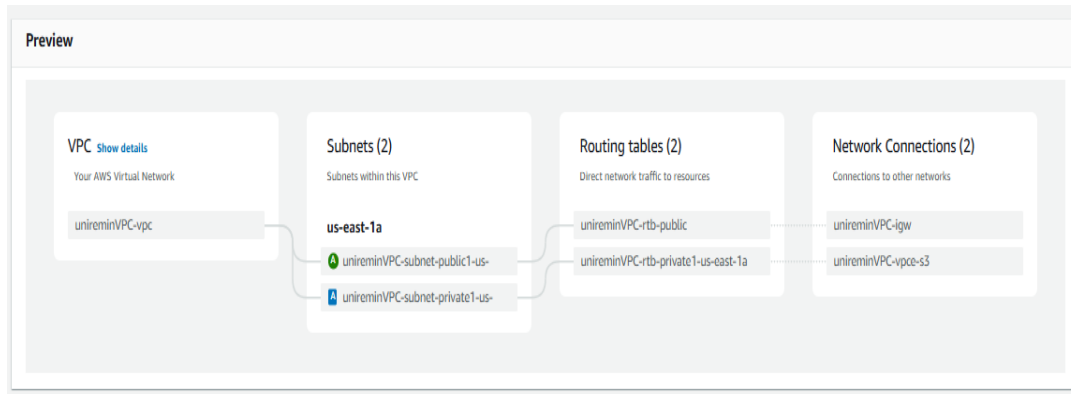


IMG 13 - Búsqueda del VPC

- **Paso 2:** Configuramos la VPC teniendo en cuenta su número de zona de disponibilidad, subredes públicas y/o privadas, tablas de enrutamiento, conexiones de red y puntos de enlaces a la VPC.



IMG 14 - Configuración del VPC

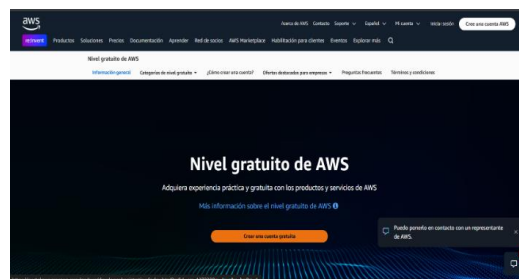


IMG 15 - Vista previa del PREVIEW

**PREVIEW:** Observamos la VPC, con las dos subnets que establecimos (una pública y una privada) cada una con una route table, la route table de la subnet pública nos permitirá conectarla con un internet Gateway que a su vez hará de router para conectarse a internet y desde internet, la route table de la subnet privada nos permitirá conectarla a un VPCE que a su vez nos permitirá conectarnos a la subnet privada mediante una conexión VPN.

#### d) Crear una instancia

Para dar inicio al proceso ingresa desde cualquier buscador a la página oficial de AWS <https://aws.amazon.com/es/>

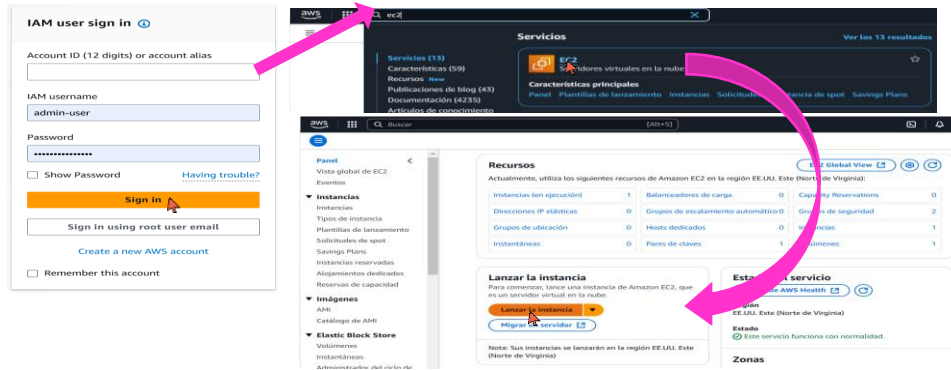


IMG 1- página inicio AWS

Se ubica en la parte superior de la ventana al lado izquierdo en el botón



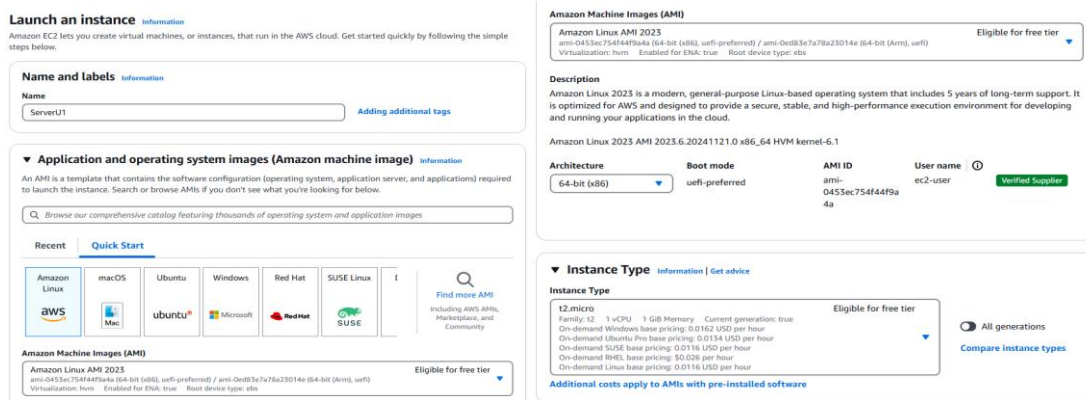
- **Paso 1:** Luego de dar inicio de sesión en una cuenta AWS existente en la barra de búsqueda **EC2** (servidores virtuales en la nube), ya en recursos damos clic en el botón lanzar la instancia.



IMG 16 - Búsqueda de EC2

**Paso 2:** Para lanzar una instancia se le asigna un nombre, Seleccionamos la imagen del sistema operativo (AMI) para la imagen, en este caso Amazon Linux de capa gratuita (Free tier eligible), **Tipo de instancia:** Seleccionamos t2.micro para este ejemplo debido a que es de capa gratuita (Free tier eligible) **Key par** (login): Para este ejemplo creamos unas llaves de tipo ppk (para conectarnos con Putty desde Windows), también podemos seleccionar unas llaves actuales si queremos **Network Settings:** Aquí debemos seleccionar la opción editar y seleccionar la VPC creada anteriormente, también la asignamos a la subnet pública de esa VPC para exponer la instancia a internet. Marcamos la opción Auto-assign Public IP en Enable para que se asigne una IP pública y pueda recibir tráfico desde internet. En Security Group creamos uno, le asignamos un nombre y agregamos en reglas de entrada los puertos 80 y 443, igualmente colocamos una descripción corta al grupo de seguridad.

**Configure Storage:** En esta opción dejaremos el valor por defecto de almacenamiento (8 GB) para evitar superar la capa gratuita Finalmente damos en Launch Instance para finalizar la creación de la instancia EC2



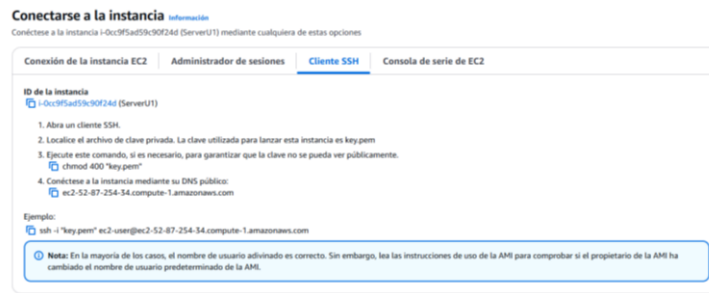
IMG 17 - Configuración del Storage

- **Paso 3:** Luego de lanzada la instancia Espera unos minutos hasta que la instancia cambie a estado **Running**.

Name	ID de la instancia	Estado de la i...	Tipo de inst...	Comprobación de	Estado de la al...	Zona de dispon...	DNS de IPv4 pública	Dirección IP...	IP elástico
ServerU1	i-0cc9f5ad59c90f24d	En ejecución	t2.micro	2/2 comprobacion	Ver alarmas +	us-east-1a	ec2-52-87-254-34.com...	52.87.254.34	-

IMG 18 - Lanzamiento de la instancia

- **Paso 4:** Conéctate a la instancia Localiza la dirección **Public IPaddress** de tu instancia en el panel de EC2, damos un uso de una terminal PuTTY en Windows para conectarnos



IMG 19 - Conexión a la instancia Localiza la dirección Public IPaddress

- **Paso 5:** Configura tu sistema Linux, autenticamos con el documento .PPK, damos clic en SSH, AUTH, Credentials y damos clic en open, ya conectados vamos actualizar e instalar NGINX, damos uso a los siguientes instrucciones

**sudo yum update -y**

**sudo amazon-linux-extras enable  
nginx1**

**sudo yum install nginx -y**

**sudo systemctl start nginx**

**sudo systemctl enable nginx**

**sudo systemctl status nginx**

```

ec2-user@ip-10-0-0-78:~
nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

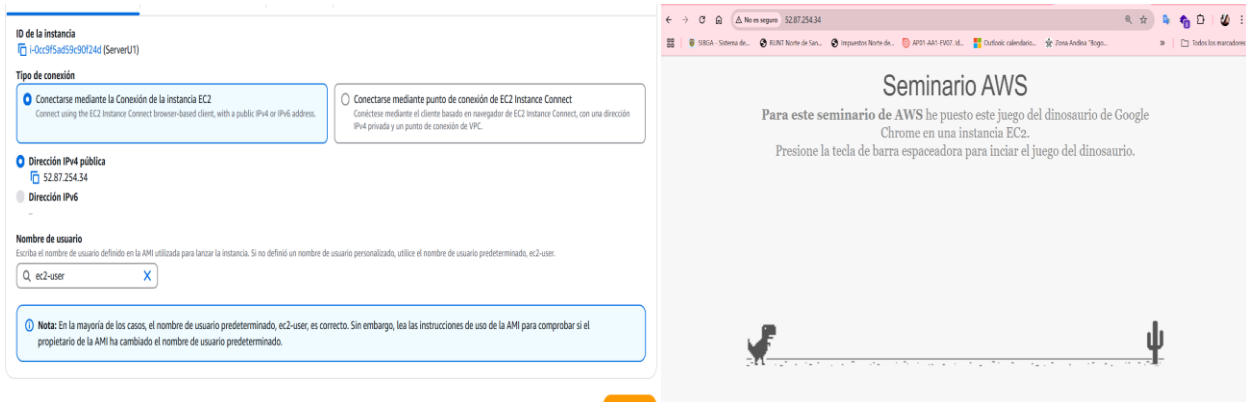
Complete!
[ec2-user@ip-10-0-0-78 ~]$ sudo systemctl start nginx
[ec2-user@ip-10-0-0-78 ~]$ sudo systemctl enable nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
[ec2-user@ip-10-0-0-78 ~]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
   Active: active (running) since Tue 2024-11-26 04:57:28 UTC; 43s ago
     Main PID: 28208 (nginx)
       Tasks: 2 (limit: 1111)
      Memory: 2.4M
         CPU: 43ms
    CGroup: /system.slice/nginx.service
            └─28208 "nginx: master process /usr/sbin/nginx"
              └─28212 "nginx: worker process"

Nov 26 04:57:28 ip-10-0-0-78.ec2.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server: [main]...
Nov 26 04:57:28 ip-10-0-0-78.ec2.internal nginx[28187]: nginx: the configuration file /etc/nginx/nginx.conf is not open (yet)
Nov 26 04:57:28 ip-10-0-0-78.ec2.internal nginx[28187]: nginx: configuration file /etc/nginx/nginx.conf successfully loaded
Nov 26 04:57:28 ip-10-0-0-78.ec2.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server: [main]...
lines 1-15/15 (END)

```

IMG 20- Herramienta PUTTYN

- **Paso 6:** ya con la IP publica 52.87.254.34 podemos verificar la conexión



IMG 21- Verificación de la conexión entrega 1

## Entrega 2

### 2.2. Pruebas de la instancia que se crea automática y aplicación correctamente con Auto Scaling Groups (ASG) y Target Groups

**Auto Scaling Groups (ASG):** es un componente de Amazon EC2 Auto Scaling que permite gestionar automáticamente el escalado de instancias EC2 en función de políticas predefinidas, métricas de monitoreo o eventos específicos. Con ASG, puedes garantizar que el número adecuado de instancias EC2 esté siempre disponible para manejar la carga de trabajo, ajustándose dinámicamente a picos o caídas de demanda.

#### Usos:

- **Escalado automático vertical y horizontal:** Permite agregar o eliminar instancias según métricas como CPU, tráfico de red o solicitudes HTTP, optimizando costos y rendimiento.
- **Alta disponibilidad:** Garantiza que siempre haya un número mínimo de instancias en ejecución, incluso en caso de fallos.
- **Elasticidad:** Facilita el ajuste de capacidad durante horarios pico o temporadas de alta demanda.

**Target Groups:** son componentes esenciales en Elastic Load Balancing (ELB) que agrupan las instancias o recursos backend (como instancias EC2, contenedores o funciones Lambda) a los que el balanceador de carga distribuirá el tráfico. Cada Target Group contiene las direcciones de destino (targets) y las reglas que definen cómo se distribuye el tráfico hacia ellos.

#### Usos:

- **Enrutamiento de tráfico específico:** Permiten enrutar solicitudes según configuraciones como puertos, protocolos y condiciones (por ejemplo, patrones de URL en balanceadores de tipo Application Load Balancer - ALB).

- **Monitoreo de salud (Health Checks):** Evalúan regularmente la disponibilidad de los destinos (targets), asegurando que solo los recursos saludables reciban tráfico.
- **Escalabilidad:** Trabajan en conjunto con ASG para registrar dinámicamente nuevas instancias o eliminarlas según los cambios en el grupo de escalado.
- **Compatibilidad con múltiples servicios:** Permiten configurar balanceadores de carga para aplicaciones que tienen diferentes endpoints, como APIs y sitios web en un mismo entorno.

## Pruebas

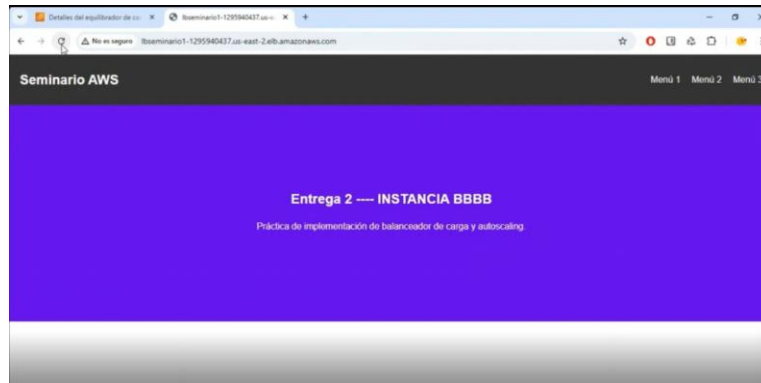
Para el desarrollo de estas pruebas disponemos del Balanceador de carga LBSeminaro1, el balanceador está expuesto a internet y recibe tráfico desde el puerto 80 (HTTP).

The screenshot displays the AWS Management Console interface for Load Balancing. It is divided into three main sections:

- Balanced Load Balancers (1):** Shows a table with columns for Name, DNS Name, State, VPC ID, Availability Zones, Type, and Creation Date. One balancer, 'LBSeminaro1', is listed as 'Active'.
- Listeners and Rules (1):** Shows a table with columns for Protocol:Port, Action, Rules, ARN, Security Policy, and SSL Certificate. A listener for 'HTTP:80' is shown, which routes traffic to a target group 'TGWeb' (100%) with a '1 rule'.
- Security Groups (1):** Shows a table with columns for ID, Name, and Description. A security group 'sg-065afe400a2b7e5ff' is listed with the name 'SGLoadBalancer' and description 'Acceso Web'.

IMG 22- Prueba del balanceador expuesto a internet

Por este motivo tiene el grupo de seguridad **SGLoadBalancer** que solo tiene una regla de entrada para el puerto 80, Accederemos al enlace DNS del balanceador de carga, es importante mencionar que previamente accedimos a las dos instancias para modificar el código HTML y mostrar el correcto funcionamiento del balanceador.



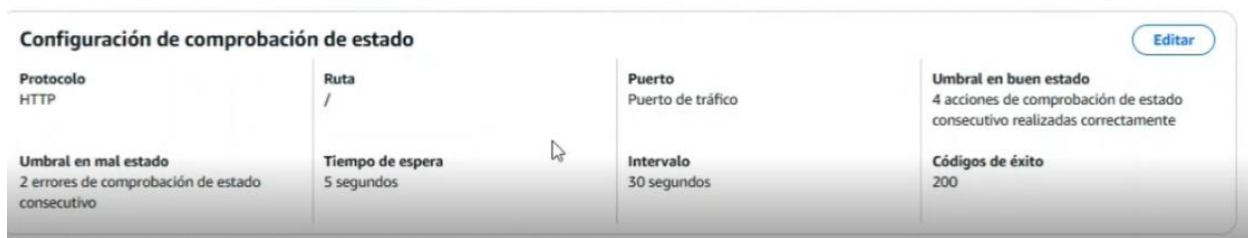
IMG 23- Acceso al DNS balanceador

Este balanceador tiene asociado el **Target Group TGWeb**, tiene dos instancias asociadas, ya que la capacidad deseada del Auto **Scaling Group** es de 2.



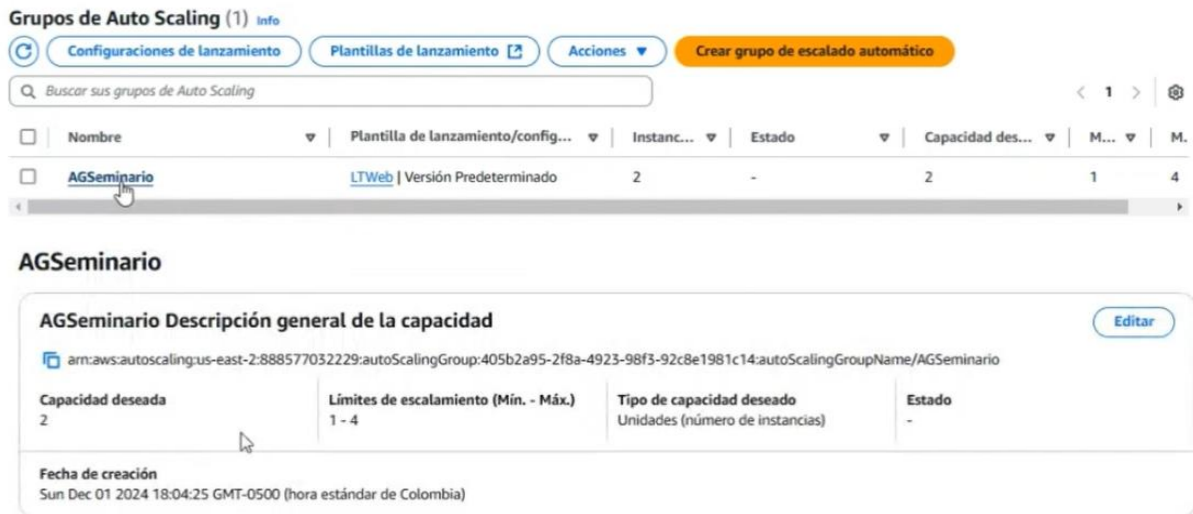
IMG 24 – Target group TGWEB asociado al Balanceador

Con una comprobación de estado de salud con esta configuración.



IMG 25 – Estado de salud de la configuración

El **Auto Scaling Group** se llama **AGSeminaro** y tiene él tiene el **Launch Template** LTWEB con el que hemos configurado previamente nuestra AMI personalizada con una configuración de capacidad deseada 2, Capacidad mínima 1 y capacidad máxima 4.



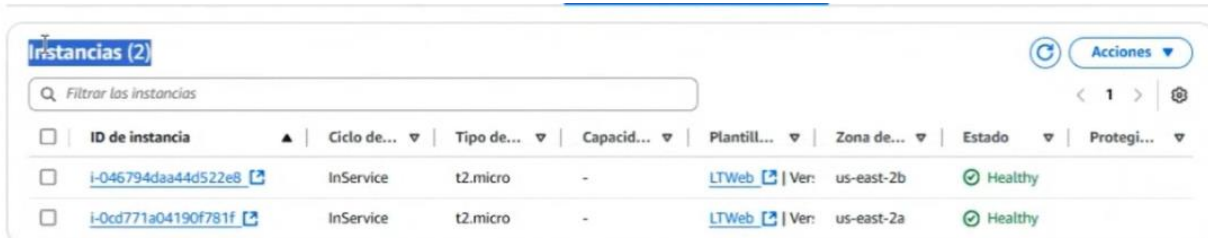
IMG 26 – Descripción general del auto scaling group

Se ha incluido en la comprobación de estado del balanceador de carga para sustituir las instancias de ser necesario cuando el servicio web falle



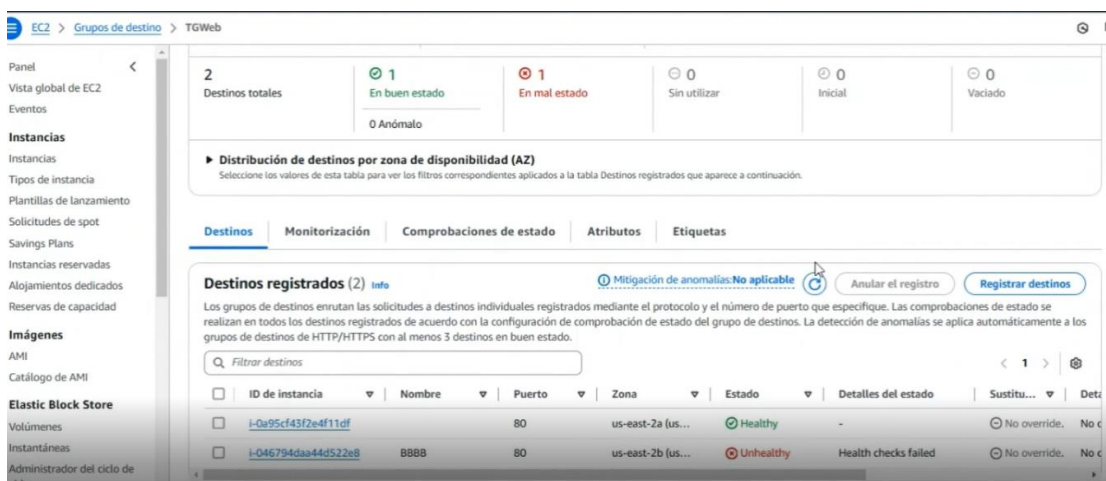
IMG 27 – Comprobación de estado del balanceador

En la opción de administración de instancias podemos ver el estado de las mismas



IMG 28 – Comprobación de estado de la instancia

Conectarnos desde la instancia de administración que no se encuentra incluida en el **target group**, para detener el servicio web de una de las instancias desplegadas por el **autoscaling group** y observaremos el comportamiento del **target group** para dejar de pasarle tráfico a la instancia y también observaremos como el **autoscaling group** se encarga de levantar una nueva instancia y terminar con la instancia que presenta problemas.



IMG 29 – Instancias desplegadas por el autoscaling group

### 2.3. Trabajo practico: Implementación de Soluciones Cloud con Amazon AWS: alta disponibilidad, escalabilidad y elasticidad en infraestructura

En los últimos meses, la empresa de tecnología denominada **TechNova Solutions** ha experimentado un crecimiento significativo en su volumen de operaciones, lo que generado la necesidad urgente de escalar su infraestructura tecnológica para mantener la calidad de sus servicios. Este crecimiento, acompañado de un aumento en la demanda de usuarios y datos, hace indispensable la implementación de una solución que garantice alta disponibilidad, escalabilidad y tiempos de respuesta optimizados.

Dentro de este proyecto se presenta a **Amazon Web Services (AWS)** como la plataforma que ofrece una solución de **arquitectura en la nube** que cumpla con los requisitos de flexibilidad, fiabilidad y eficiencia. **AWS** proporciona un conjunto de herramientas y servicios robustos que permiten a las empresas como **TechNova Solutions** adaptarse a las demandas crecientes sin la necesidad de realizar grandes inversiones en infraestructura física.

Se propone como objetivo diseñar e implementar una arquitectura en la nube utilizando los servicios de AWS que permita a **TechNova Solutions** escalar sus recursos de manera automática, garantizar la disponibilidad continua de los servicios y mejorar significativamente los tiempos de respuesta, brindando una solución que no solo satisfaga las necesidades actuales, sino que también se adapte a futuras expansiones.

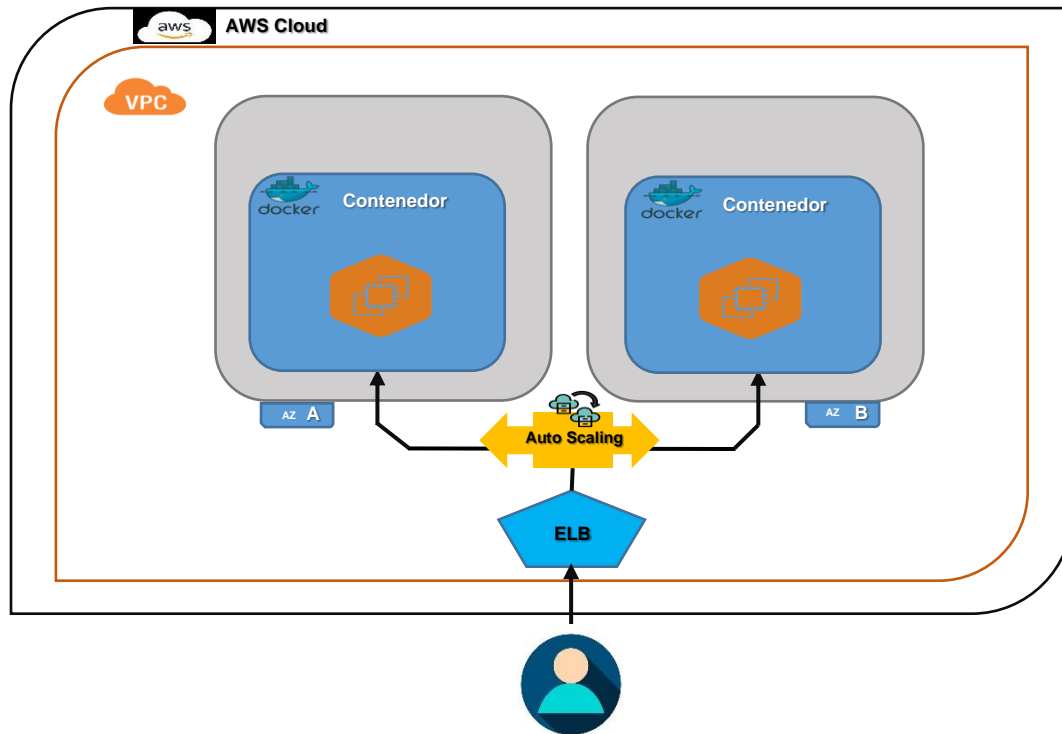
- **Objetivo general:**

Diseñar e implementar una arquitectura en la nube utilizando los servicios de AWS.

- **Objetivos específicos:**

1. Configurar un Balanceador de carga usando **Application Load Balancer (ALB)**, que permita distribuir el tráfico entrante a múltiples instancias **EC2**.
2. Implementar una **Instancia EC2** con una configuración multi-zona para garantizar la alta disponibilidad
3. Implementar dentro de Instancia EC2 un **proxy reverso** con Nginx, para el direccionamiento de las solicitudes a servicios internos
4. Configurar la política de **Autoescalado** para el aumento o disminución de los servicios internos según el uso requerido por la sociedad
5. Asegurar el tráfico utilizando **grupos de Seguridad** y habilitados HTTPS en el balanceador de carga.
6. Entregar Documentación.

### 2.3.1. Diagrama de recursos



IMG 30 – Diagrama de recursos

### 2.3.2. Pruebas de trabajo práctico

Para el desarrollo de estas pruebas disponemos de una instancia de administración utilizando la herramienta PUTTYN, verificamos si los balanceadores están enviando carga a las otras instancias, verificando que funciona, repetimos el proceso pero desde una de las otras instancias, verificamos de que la herramienta DOCKER corran listándolo y encontrado un contenedor llamado APP1, el cual recibe tráfico desde el puerto interno 80 que es direccionado desde el puerto 80 de la instancia, verificando el tráfico desde el navegador, para posteriormente detener el contenedor, evidenciando el mal estado de la instancia y evidenciamos como funciona correctamente.

aws [Alt+S] Ohio admin-user @ 8885-7703-2229

EC2 > Grupos de destino > TGWeb

Panel  
Vista global de EC2  
Eventos

▼ **Instancias**  
Instancias  
Tipos de instancia  
Plantillas de lanzamiento  
Solicitudes de spot

2 Destinos totales

1 En buen estado  
0 Anómalo

1 En mal estado

0 Sin utilizar

0 Inicial

0 Vacío

► **Distribución de destinos por zona de disponibilidad (AZ)**  
Seleccione los valores de esta tabla para ver los filtros correspondientes aplicados a la tabla Destinos registrados que aparece a continuación.

Destinos | Monitorización | Comprobaciones de estado | Atributos | Etiquetas

aws [Alt+S] Ohio admin-user @ 8885-7703-2229

EC2 > Grupos de destino > TGWeb

▼ **Elastic Block Store**  
Volúmenes  
Instantáneas  
Administrador del ciclo de vida

▼ **Red y seguridad**  
Security Groups  
Direcciones IP elásticas  
Grupos de ubicación  
Pares de claves  
Interfaces de red

▼ **Equilibrio de carga**  
Balanceadores de carga  
Grupos de destino  
Trust Stores **Nuevo**

▼ **Auto Scaling**  
Grupos de Auto Scaling  
Configuración

0 Anómalo

► **Distribución de destinos por zona de disponibilidad (AZ)**  
Seleccione los valores de esta tabla para ver los filtros correspondientes aplicados a la tabla Destinos registrados que aparece a continuación.

Destinos | Monitorización | Comprobaciones de estado | Atributos | Etiquetas

**Destinos registrados (3)** Info [Mitigación de anomalías: No aplicable](#) [Anular el registro](#) [Registrar destinos](#)

Los grupos de destinos enrutan las solicitudes a destinos individuales registrados mediante el protocolo y el número de puerto que especifique. Las comprobaciones de estado se realizan en todos los destinos registrados de acuerdo con la configuración de comprobación de estado del grupo de destinos. La detección de anomalías se aplica automáticamente a los grupos de destinos de HTTP/HTTPS con al menos 3 destinos en buen estado.

Filtrar destinos

<input type="checkbox"/>	ID de instancia	Nombre	Puerto	Zona	Estado	Detalles del estado	Sustitu...	Det...
<input type="checkbox"/>	<a href="#">i-0462481b2c5b8b0d</a>		80	us-east-2a (us...	Healthy	-	<input type="radio"/> No override	No c
<input type="checkbox"/>	<a href="#">i-0fd17cd94e3841cd</a>		80	us-east-2a (us...	Healthy	-	<input type="radio"/> No override	No c
<input type="checkbox"/>	<a href="#">i-069dee875bcf22eea</a>		80	us-east-2a (us...	Draining	Target deregistration i...	<input type="radio"/> No override	No c

aws [Alt+S] Ohio admin-user @ 8885-7703-2229

EC2 > Grupos de Auto Scaling > ASGSeminarioDocker

Panel  
Vista global de EC2  
Eventos

▼ **Instancias**  
Instancias  
Tipos de instancia  
Plantillas de lanzamiento  
Solicitudes de spot  
Savings Plans  
Instancias reservadas  
Alojamientos dedicados  
Reservas de capacidad

▼ **Imágenes**  
AMI  
Catálogo de AMI

▼ **Elastic Block Store**  
Volúmenes  
Instantáneas  
Administrador del ciclo de vida

Integraciones - nueva | Escalado automático | **Administración de instancias** | Actualización de instancias | Actividad | Monito

**Instancias (3)** Info [Acciones](#)

Filtrar las instancias

<input type="checkbox"/>	ID de instancia	Ciclo de...	Tipo de...	Capacid...	Plantill...	Zona de...	Estado	Protegi...
<input type="checkbox"/>	<a href="#">i-0462481b2c5b8b0d</a>	InService	t2.micro	-	<a href="#">LTWeb</a>   Ver	us-east-2a	Healthy	
<input type="checkbox"/>	<a href="#">i-069dee875bcf22eea</a>	Terminating	t2.micro	-	<a href="#">LTWeb</a>   Ver	us-east-2a	Unhealthy...	
<input type="checkbox"/>	<a href="#">i-0fd17cd94e3841cd</a>	InService	t2.micro	-	<a href="#">LTWeb</a>   Ver	us-east-2a	Healthy	

**Enlaces de ciclo de vida (0)** Info [Acciones](#) [Crear enlace de ciclo de vida](#)

Filtrar enlaces de ciclo de vida

Nombre | Transición de cic... | Resultado prede... | Tiempo de esper... | ARN del destino ... | ARN del rol

Actualmente no hay enlaces de ciclo de vida configurados.  
Los enlaces de ciclo de vida le ayudan a realizar acciones personalizadas en las instancias a medida que se lanzan y antes de que terminen.

[Crear enlace de ciclo de vida](#)

CloudShell Comentarios © 2024, Amazon Web Services, Inc. o sus filiales. Privacidad Términos Preferencias de cookies

IMG 31- Prueba del contenedor

## Figuras y Tablas

**Tabla 1.** Imágenes.

<i>Imagen</i>	<i>Descripción</i>
IMG 1	Página inicio AWS
IMG 2	Creación cuenta raíz o ROOT en AWS
IMG 3	Creación contraseña cuenta raíz o ROOT en AWS
IMG 4	Autenticación de cuenta raíz o ROOT en AWS por SMS
IMG 5	Creación exitosa de la cuenta raíz o ROOT en AWS por SMS
IMG 6	Asignación de Multi Factor Authentication a cuenta raíz o ROOT en AWS por SMS
IMG 7	Búsqueda de la IAM
IMG 8	Búsqueda opción usuario para la creación de la IAM
IMG 9	Especificación de detalles de usuario IAM
IMG 10	Creación del grupo de usuario
IMG 11	Creación de usuario IAM
IMG 12	Creación de la llave SSH
IMG 13	Búsqueda del VPC
IMG 14	Configuración del VPC
IMG 15	Vista previa del PREVIEW

---

IMG 16	Búsqueda de EC2
IMG 17	Configuración del Storage Datos
IMG 18	Lanzamiento de la instancia
IMG 19	Conexión a la instancia Localiza la dirección <b>Public IPaddress</b>
IMG 20	Herramienta PUTTYN
IMG 21	Verificación de la conexión entrega 1
IMG 22	Prueba del balanceador expuesto a internet
IMG 23	Acceso al DNS balanceador
IMG 24	Target group TGWEB asociado al Balanceador
IMG 25	Estado de salud de la configuración
IMG 26	Descripción general del auto scaling group
IMG 27	Comprobación de estado del balanceador
IMG 28	Comprobación de estado de la instancia
IMG 29	Instancias desplegadas por el autoscaling group

---

---

IMG 30

Diagrama de recursos

IMG 31

Prueba del contenedor

---

## Conclusiones

- Este diplomado proporcionó habilidades avanzadas en el uso de herramientas y tecnologías de AWS, fortaleciendo la capacidad técnica para enfrentar desafíos relacionados con infraestructura en la nube, consolidando a los participantes como expertos en computación en la nube.
- AWS proporciona una guía completa sobre sus servicios en su documentación oficial. Puedes consultar temas específicos como **EC2, VPC, IAM, Auto Scaling Groups, y Application Load Balancer**, entre otros. Este seminario explica a detalle cómo implementar, configurar y gestionar infraestructuras en la nube usando AWS.
- El diseño e implementación de una arquitectura en la nube utilizando **Amazon Web Services (AWS)** permite escalar infraestructura tecnológica de manera eficiente. Esto asegura alta disponibilidad, tiempos de respuesta optimizados y capacidad para gestionar el aumento en la demanda de servicios sin comprometer la calidad.

## Referencias

(AWS), A. W. (2024). *Amazon Web Services (AWS)*. Obtenido de Amazon Web Services (AWS): [https://aws.amazon.com/es/what-is-aws/?nc1=f\\_cc](https://aws.amazon.com/es/what-is-aws/?nc1=f_cc)

(AWS), A. W. (2024). *AWS*. Obtenido de Documentacion - Amazon EC2 - Guía del usuario: [https://docs.aws.amazon.com/es\\_es/AWSEC2/latest/UserGuide/concepts.html](https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/concepts.html)

(AWS), A. W. (2024). *AWS*. Obtenido de AWS - Documentacion - Amazon VPC - Guía del Usuario: [https://docs.aws.amazon.com/es\\_es/vpc/latest/userguide/what-is-amazon-vpc.html](https://docs.aws.amazon.com/es_es/vpc/latest/userguide/what-is-amazon-vpc.html)

Amazon Web Services (AWS). (2024). *AWS*. Obtenido de AWS - Documentacion - amazon VPC - (IAM): [https://docs.aws.amazon.com/es\\_es/vpc/latest/userguide/security-iam.html](https://docs.aws.amazon.com/es_es/vpc/latest/userguide/security-iam.html)

DOCKER. (2024). *DOCKER*. Obtenido de DOCKER: <https://www.docker.com/resources/what-container>