



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

**AWS CONTAINER SERVICE  
WITH LOAD BALANCER AND AUTO SCALING**

Corporación Universitaria Remington.  
Ingeniería de sistemas  
SEMINARIO AMAZON AWS

Cristhiam Taniry Romero Florez  
Gustavo Adolfo Gonzalez Santiago

Tutor  
Ing. Juan Pablo Berrio Lopez  
Seminario de grado  
2024.

## Tabla de Contenidos

Resumen.....	3
Marco conceptual y contextual .....	3
Desarrollo e implementación del aprendizaje.....	7
Figuras y tablas .....	10
Conclusiones .....	27
Referencias.....	28

## **Resumen**

### **Propuesta de Implementación de AWS Container Service con Load Balancer y Auto Scaling para TITAN BUILD PC STORE**

En nuestra empresa, tenemos soluciones de infraestructura en la nube enfocadas para satisfacer las necesidades de escalabilidad, rendimiento y disponibilidad de TITAN BUILD PC STORE. nuestra solución más efectiva es el servicio de AWS Container Service con Load Balancer y Auto Scaling, que permite a TITAN BUILD PC STORE mejorar su infraestructura de TI y manejar de manera eficiente el tráfico web, asegurando una experiencia de usuario sin interrupciones.

Con Amazon Web Services (AWS), creamos un entorno de nube seguro, escalable y flexible que se adapta a las necesidades específicas de TITAN BUILD PC STORE. Este servicio incluye la implementación de un Virtual Private Cloud (VPC), en el cual configuramos subredes públicas y privadas, garantizando una estructura de red eficiente y segura. Utilizamos instancias EC2 con Amazon Linux y Docker, lo que le permitirá ejecutar aplicaciones en contenedores de manera rápida y eficiente.

El Load Balancer distribuye el tráfico entre las instancias de su servidor, asegurando alta disponibilidad y optimizando el rendimiento de las aplicaciones, incluso en picos de tráfico. Además, configuramos Auto Scaling para que su infraestructura se ajuste automáticamente a las variaciones del tráfico, lo que le permite ahorrar costos al pagar solo por lo que realmente utiliza.

El servicio le proporciona una solución de infraestructura totalmente administrada, reduciendo la complejidad de gestión de servidores y aumentando la eficiencia operativa. Con nuestro soporte y monitoreo continuo, su empresa podrá disfrutar de una infraestructura robusta, flexible y de alto rendimiento sin preocupaciones técnicas.

## **Palabras clave**

- 1) Alta disponibilidad
- 2) Escalabilidad
- 3) Optimización de Rendimiento
- 4) Reducción de costos
- 5) Infraestructura flexible

## **Marco conceptual y contextual**

El marco conceptual de esta propuesta se basa en el uso de servicios avanzados en la nube proporcionados por Amazon Web Services (AWS), orientados a optimizar la infraestructura de TI de la empresa mediante la automatización, escalabilidad y distribución eficiente del tráfico. A continuación, se presentan los conceptos clave involucrados:

**Amazon Web Services (AWS):** Es la nube más completa y ampliamente adoptada del mundo, e incluye ofertas de infraestructura como servicio (IaaS) y plataforma como servicio (PaaS). Los servicios de AWS ofrecen soluciones escalables para la computación, el almacenamiento, las bases de datos, el análisis y mucho más.

**Virtual Private Cloud (VPC):** Le brinda control total sobre su entorno de redes virtuales, incluidas la ubicación de los recursos, la conectividad y la seguridad. Comience configurando su VPC en la consola de los servicios de AWS. Luego, agréguele recursos, como instancias de Amazon Elastic Compute Cloud (EC2) y Amazon Relational Database Service (RDS). Por último, defina cómo se comunican sus VPC entre sí, entre cuentas, zonas de disponibilidad o regiones de AWS. En el siguiente ejemplo, el tráfico de red se comparte entre dos VPC dentro de cada región.

**EC2 Instance:** Una instancia en la nube abstrae la infraestructura de computación física mediante la tecnología de máquinas virtuales. Es similar a tener su propio servidor en la nube. Básicamente, crea y administra su propia instancia de servidor virtual en el entorno de computación en la nube.

**Amazon Linux 2023:** Es la próxima generación de Amazon Linux de Amazon Web Services (AWS). Con AL2023, puede desarrollar y ejecutar aplicaciones empresariales y en la nube en un entorno de ejecución seguro, estable y de alto rendimiento. Además, dispondrá de un entorno de aplicaciones que ofrece soporte a largo plazo con acceso a las últimas innovaciones de Linux. AL2023 se ofrece sin cargo adicional.

**Snapshot:** Las instantáneas de EBS son una copia puntual de sus datos y se pueden utilizar para permitir la recuperación de desastres, migrar datos entre regiones y cuentas y mejorar la conformidad de las copias de seguridad.

**AMI:** Una imagen de máquina de Amazon (AMI) es una imagen que ofrece el software que se necesita para configurar y arrancar una instancia de Amazon EC2. Cada AMI también contiene una asignación de dispositivos de bloques que especifica los dispositivos de bloques que se deben asociar a las instancias que se lancen.

**Security group:** Un grupo de seguridad controla el tráfico al que se permite llegar y dejar los recursos a los que está asociado. Por ejemplo, después de asociar un grupo de seguridad a una instancia de EC2, controla el tráfico de entrada y salida de la instancia.

Al crear una VPC, incluye un grupo de seguridad predeterminado. Puede crear grupos de seguridad adicionales para una VPC, cada uno con sus propias reglas de entrada y salida.

Puede especificar el origen, el rango de puertos y el protocolo de cada regla de entrada. Puede especificar el destino, el rango de puertos y el protocolo de cada regla de salida.

**Target group:** Dirigen las solicitudes a destinos individuales registrados, como EC2 instancias, mediante el protocolo y el número de puerto que especifique. Puede registrar un destino en varios grupos de destino. Puede configurar las comprobaciones de estado de cada grupo de destino. Las comprobaciones de estado se llevan a cabo en todos los destinos registrados en un grupo de destino especificado en la regla del oyente del equilibrador de carga.

**Load Balancer:** Elastic Load Balancing distribuye automáticamente el tráfico entrante entre varios destinos, por ejemplo, instancias EC2, contenedores y direcciones IP en una o varias zonas de disponibilidad. Monitorea el estado de los destinos registrados y enruta el tráfico solamente a destinos en buen estado. Elastic Load Balancing escala el equilibrador de carga a medida que el tráfico entrante va cambiando con el tiempo. Puede escalarse automáticamente para adaptarse a la mayoría de las cargas de trabajo.

**Launch template:** Las plantillas de lanzamiento son una función nueva que facilita la creación de solicitudes de lanzamiento. Las plantillas de lanzamiento agilizan y simplifican el proceso de lanzamiento para las instancias Auto Scaling, de subasta, de flota de subasta y bajo demanda.

Las plantillas de lanzamiento reducen el número de pasos necesarios para crear una instancia al capturar todos los parámetros de lanzamiento en un recurso. Esto hace que el proceso sea fácil de reproducir. Además, gracias a la compatibilidad con las instancias Auto Scaling, de flota de subasta, de subasta y bajo demanda, las plantillas de lanzamiento facilitan la implementación de estándares y prácticas recomendadas, lo que permite administrar mejor los costos, mejorar el nivel de seguridad y minimizar el riesgo de errores de implementación.

**Auto Scaling group:** AWS Auto Scaling le permite crear planes de escalado que automatizan la manera en la que diferentes recursos responden ante los cambios que se producen en la demanda. Puede optimizar la disponibilidad, los costos, o lograr un equilibrio entre ambos. AWS Auto Scaling crea todas las políticas de escalado y define los valores meta por usted automáticamente en función de sus preferencias. AWS Auto Scaling monitoriza su aplicación y añade o reduce automáticamente capacidad en los grupos de recursos en tiempo real a medida que las demandas cambian.

**NGINX:** Es un famoso software de servidor web de código abierto. En su versión inicial, funcionaba en servidores web HTTP. Sin embargo, hoy en día también sirve como proxy inverso, balanceador de carga HTTP y proxy de correo electrónico para IMAP, POP3 y SMTP.

**Docker:** Docker es una plataforma de software que le permite crear, probar e implementar aplicaciones rápidamente. Docker empaqueta software en unidades estandarizadas llamadas contenedores que incluyen todo lo necesario para que el software se ejecute, incluidas bibliotecas, herramientas de sistema, código y versión ejecutable. Con Docker,

puede implementar y ajustar la escala de aplicaciones rápidamente en cualquier entorno con la certeza de saber que su código se ejecutará.

**Contenedor:** Un contenedor es una unidad estándar de software que empaqueta el código y todas sus dependencias para que la aplicación se ejecute de manera rápida y confiable de un entorno informático a otro. Una imagen de contenedor Docker es un paquete de software ejecutable, independiente y liviano que incluye todo lo necesario para ejecutar una aplicación: código, entorno de ejecución, herramientas del sistema, bibliotecas del sistema y configuraciones.

## Desarrollo e implementación del aprendizaje

### Entrega 1

link del video "SERVIDOR WEB AMAZON LINUX AWS:  
<https://youtu.be/NuPVmscL5xY>

#### Recursos de AWS para ejecutar instancias

Amazon Web Services (AWS) tiene una variedad de recursos que nos permiten crear máquinas virtuales, llamadas instancias. Estas instancias se usan para tareas como alojar aplicaciones, procesar datos o hacer simulaciones.

#### 1. Amazon EC2 (Elastic Compute Cloud)

Este es el servicio principal que nos permite lanzar y manejar instancias. EC2 es como alquilar una computadora en la nube.

Tipos de instancias: AWS tiene diferentes tipos de máquinas según lo que necesites (más rápidas, con más memoria o más económicas). Por ejemplo:

t2.micro: Una máquina básica y barata.

m5.large: Una máquina más poderosa.

AMIs (Amazon Machine Images): Son como una copia o plantilla que incluye el sistema operativo y el software que necesitas en tu instancia (por ejemplo, Windows o Linux).

#### 2. Almacenamiento para las instancias

Una máquina necesita almacenamiento para guardar datos. AWS tiene varias opciones:

EBS (Elastic Block Store): Es un disco duro que puedes conectar a tu instancia. Lo bueno es que, si apagas la máquina, los datos se quedan guardados.

Instance Store: Es almacenamiento rápido que viene con la máquina, pero los datos se pierden si la apagas.

S3 (Simple Storage Service): Sirve para guardar grandes cantidades de datos, como archivos o backups, pero no es un disco duro como EBS.

#### 3. Redes y conexión

Para que las máquinas se conecten entre sí o con internet, necesitas configurar una red:

VPC (Virtual Private Cloud): Es como una red privada que creas para tus máquinas.

Security Groups: Son como las reglas de un cortafuegos que dicen qué conexiones están permitidas (por ejemplo, quién puede entrar o salir de la máquina).

Elastic IP: Si necesitas que tu máquina tenga una dirección IP fija para conectarte desde cualquier lugar.

#### Resumen

Para ejecutar una instancia en AWS necesitas:  
Amazon EC2 para lanzar la máquina.  
Almacenamiento (EBS o Instance Store) para guardar los datos.  
VPC y Security Groups para conectarla a una red.

## Entrega 2

DNS

LbServerUni-1930524815.us-east-2.elb.amazonaws.com

Vinculo video YOUTUBE

<https://youtu.be/i-lzLqnziEA>

Implemente un servidor web en amazon Linux, que ejecute una aplicación en una instancia, haga pruebas mostrando como las instancias se crean automáticamente y la aplicación queda funcionando correctamente:

Se ingreso a la consola de amazon AWS donde ya teníamos un servidor funcionando de la entrega 1 y a partir de allí iniciamos a instalar el servicio web de alta disponibilidad para eso se requirió de los siguientes pasos:

- 1) Create snapshot:  
Ya teniendo un servidor funcionando con un servicio web se inicia la creación de un snapshot (una copia completa del volumen de esa instancia).
- 2) Create AMI:  
Teniendo el snapshot iniciamos a montar una imagen a partir del mismo snapshot con las mismas características del servidor inicial. (t2.micro).
- 3) Create Security Group:  
Se necesito crear un Security Group para vincularlo al Load Balancer que necesitábamos crear.
- 4) Create Target Group:  
Se necesito crear un Target Group para vincularlo al Load Balancer que necesitábamos crear.
- 5) Create Load Balancer:  
Ya teniendo el Security Group y Target Group iniciamos a crear el Load Balancer con las características que necesitamos para este servicio.
- 6) Create Launch Template:  
Se necesito crear un Launch Template para vincularlo al Auto Scaling Group que necesitábamos crear.

7) Create Auto Scaling Group:

Ya teniendo el Launch Template iniciamos a crear el Auto Scaling Group con las características que necesitamos para este servicio.

### Entrega 3

DNS

LbServerUni-1930524815.us-east-2.elb.amazonaws.com

Vinculo video YOUTUBE

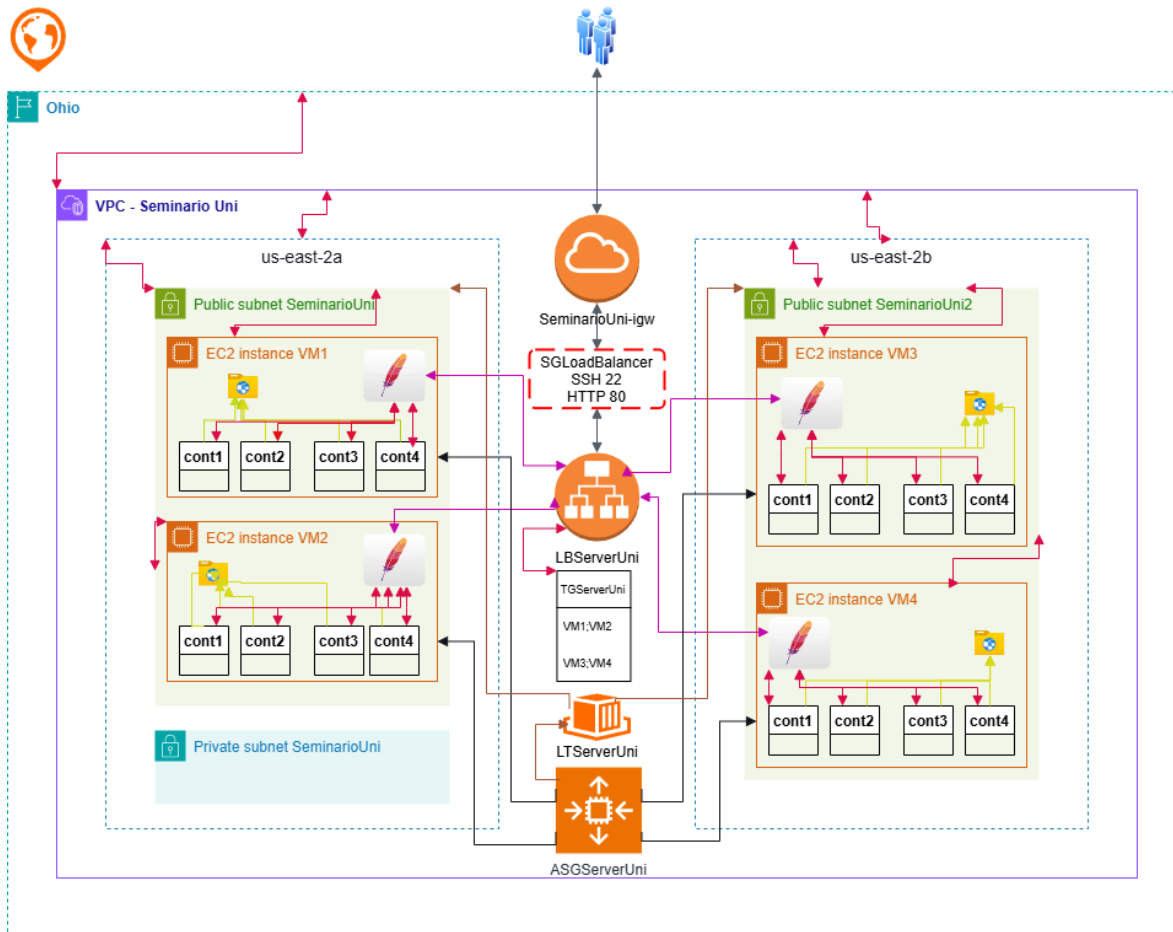
<https://youtu.be/bj4qoZiOq6I>

#### AWS CONTAINER SERVICE WITH LOAD BALANCER AND AUTO SCALING

Para poder implementar este servicio de alta disponibilidad aún más mejorado que la entrega 2 fue necesario cambiar configuraciones en la instancia ya creada y con la misma que hemos trabajado hasta ahora, entramos a su sistema operativo amazon Linux 2023 en este momento debimos parar por completo el servicio web APACHE (httpd), para instalar el servicio web de NGINX ya que este servicio nos permite configurarlo como balanceador de cargas en la misma instancia, se creó un folder en la raíz del sistema operativo donde se va a copiar y crear todos los archivos del sitio que de dicha empresa, luego se procede con la instalación del servicio DOCKER que es necesario para crear contenedores que son los servicios que se conectaran al NGINX y buscaran los archivos del sitio web en el folder para mostrar dicho sitio web con esta instancia inicializada e habilitado el inicio automático de todos sus servicios y dándole permisos de todo tipo ya está terminada.

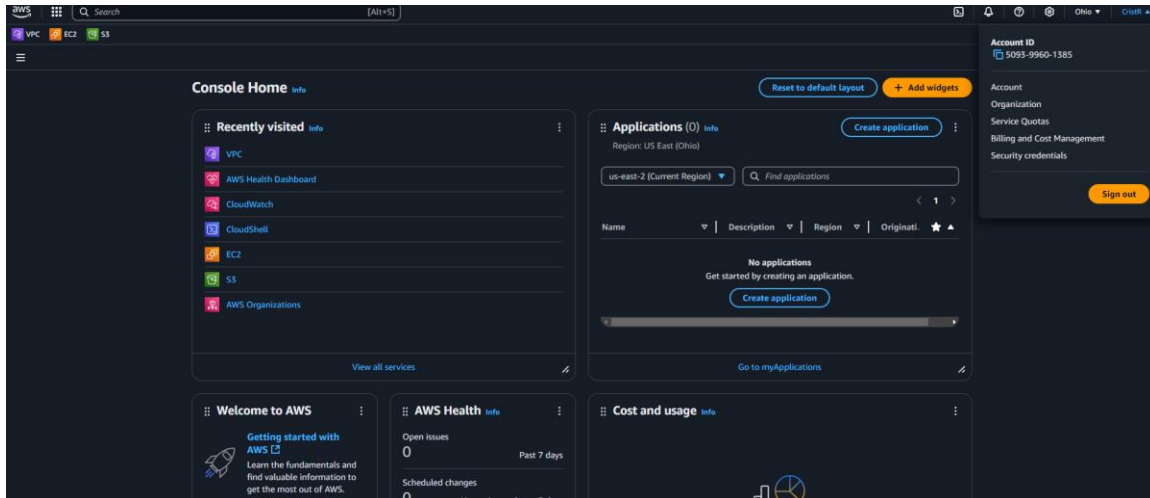
Con esta instancia maestra podemos iniciar el proceso de SNAPSHOT para así crear una AMI de esta misma y luego fuimos a el LAUNCH TEMPLATE que se había vinculado con el AUTO SCALING GROUP y así cuando fuese necesario implementaría más instancias idénticas según las configuraciones del AUTO SCALING GROUP, ya en este momento está listo el servicio de ALTA DIPONIBILIDAD con balanceador de cargas en AWS.

Diagrama  
AWS CONTAINER SERVICE  
WITH LOAD BALANCER AND AUTO SCALING

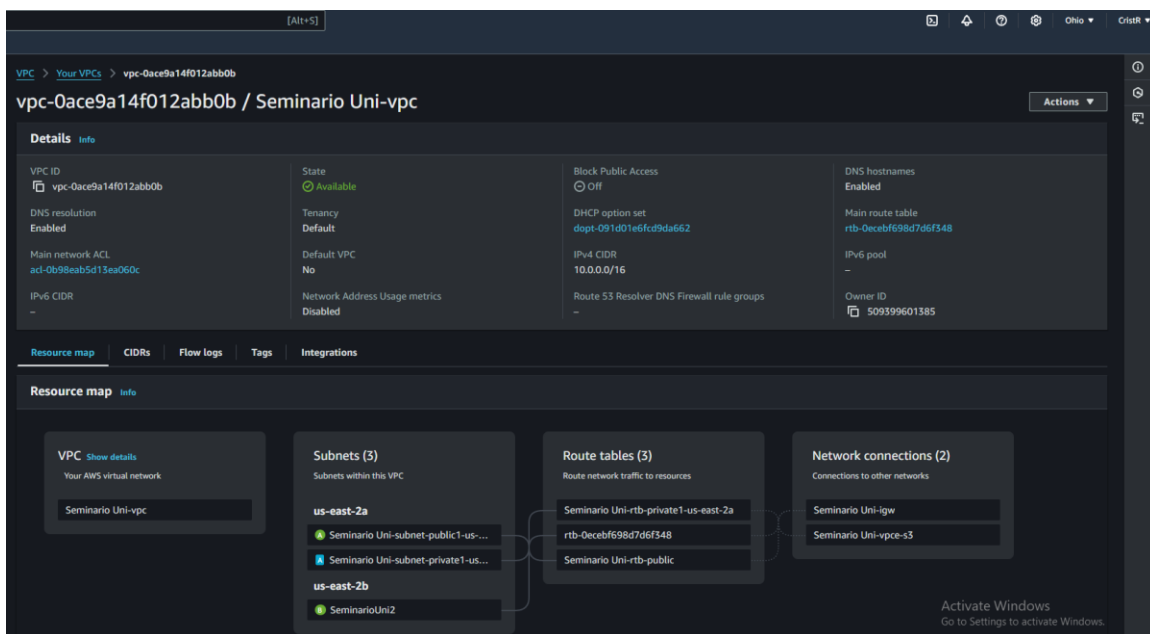


## Figuras y tablas

Registrado e iniciado sesión en Amazon AWS



## Virtual Private Cloud (VPC)



## Subnets

Subnets (3) info

Last updated 1 minute ago

Find resources by attribute or tag

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR
SeminarioUni2	subnet-0b63e7b76aa1cd0d1	Available	vpc-0ace9a14f012abb0b   Semi...	Off	10.0.192.0/20	-
Seminario Uni-subnet-private1-us-east-2a	subnet-0373120c56091a03c	Available	vpc-0ace9a14f012abb0b   Semi...	Off	10.0.128.0/20	-
Seminario Uni-subnet-public1-us-east-2a	subnet-0f450f97eebd6bc58	Available	vpc-0ace9a14f012abb0b   Semi...	Off	10.0.0.0/20	-

## Subnet configuración

subnet-0f450f97eebd6bc58 / Seminario Uni-subnet-public1-us-east-2a

Details

Subnet ID subnet-0f450f97eebd6bc58	Subnet ARN arn:aws:ec2:us-east-2:509399601385:subnet/subnet-0f450f97eebd6bc58	State Available	Block Public Access Off
IPv4 CIDR 10.0.0.0/20	Available IPv4 addresses 4088	IPv6 CIDR -	IPv6 CIDR association ID -
Availability Zone us-east-2a	Availability Zone ID use2-az1	VPC vpc-0ace9a14f012abb0b   Seminario Uni-vpc	Route table rtb-0127e194c79f29904   Seminario Uni-rtb-public
Network ACL acl-0b98eab5d13ea060c	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 509399601385	

Flow logs

Route table

Network ACL

CIDR reservations

Sharing

Tags

Flow logs

No matching resource found

## Route tables

The screenshot displays the AWS Management Console interface for a Route Table. The breadcrumb navigation shows 'VPC > Route tables > rtb-0127e194c79f29904'. The title is 'rtb-0127e194c79f29904 / Seminario Uni-rtb-public'. The 'Details' section includes:

- Route table ID: rtb-0127e194c79f29904
- VPC: vpc-0ace9a14f012abb0b | Seminario Uni-vpc
- Main: No
- Owner ID: 509399601385
- Explicit subnet associations: subnet-0f450f977eebd0bc58 / Seminario Uni-subnet-public1-us-east-2a
- Edge associations: -

The 'Routes' section shows 2 routes:

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0d8264eeff02be15d	Active	No
10.0.0.0/16	local	Active	No

## Internet gateway

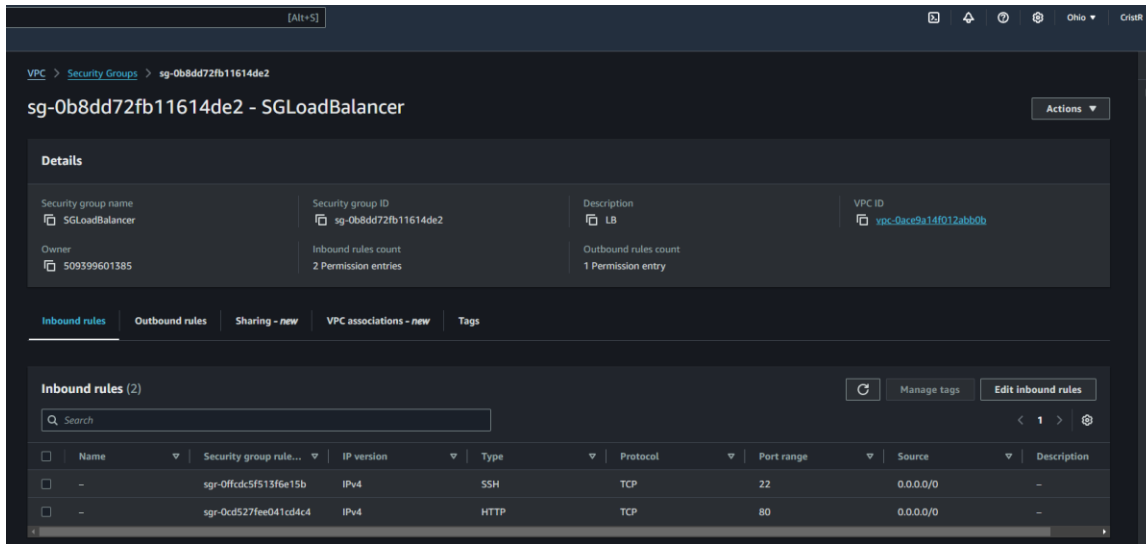
The screenshot displays the AWS Management Console interface for an Internet Gateway. The breadcrumb navigation shows 'VPC > Internet gateways > igw-0d8264eeff02be15d'. The title is 'igw-0d8264eeff02be15d / Seminario Uni-igw'. The 'Details' section includes:

- Internet gateway ID: igw-0d8264eeff02be15d
- State: Attached
- VPC ID: vpc-0ace9a14f012abb0b | Seminario Uni-vpc
- Owner: 509399601385

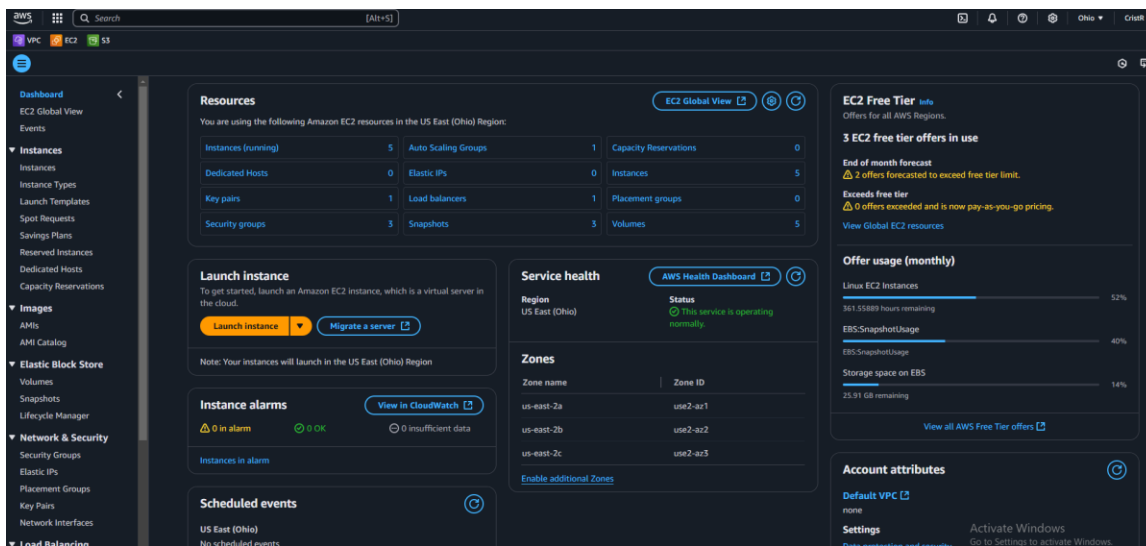
The 'Tags' section shows one tag:

Key	Value
Name	Seminario Uni-igw

## Security Groups



## EC2 Dashboard



## Instances

The screenshot shows the AWS Management Console 'Instances' page. At the top, there are buttons for 'Connect', 'Instance state', 'Actions', and 'Launch instances'. Below these is a search bar and a table of instances. The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, Public IPv4 address, and Elastic IP. Five instances are listed, all in a 'Running' state. The instance 'Linux Docker' is highlighted.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
	i-07c05be152c0c810a	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2a	-	-	-
	i-0db1042db0275912d	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2a	ec2-3-15-225-99.us-eas...	3.15.225.99	-
	i-01182d4957af37d54	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2b	-	-	-
Linux Docker	i-0324a92fe2e9ac23b	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2b	ec2-3-15-158-77.us-eas...	3.15.158.77	-
	i-009446918f514aac2	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2b	ec2-3-137-195-93.us-e...	3.137.195.93	-

## Instancia summary

The screenshot shows the 'Instance summary' page for the instance 'i-0324a92fe2e9ac23b (Linux Docker)'. The page is updated 'less than a minute ago'. It features a 'Connect' button, 'Instance state' dropdown, and 'Actions' dropdown. The summary is organized into several sections:

- Instance ID:** i-0324a92fe2e9ac23b
- Instance state:** Running
- Public IPv4 address:** 3.15.158.77
- Private IP DNS name (IPv4 only):** ip-10-0-203-41.us-east-2.compute.internal
- Instance type:** t2.micro
- VPC ID:** vpc-0ace9a14f012abb0b (Seminarior Uni-vpc)
- Subnet ID:** subnet-0b63e7b76aa1cd0d1 (Seminarior Uni2)
- Instance ARN:** arn:aws:ec2:us-east-2:509399601385:instance/i-0324a92fe2e9ac23b
- Public IPv4 DNS:** ec2-3-15-158-77.us-east-2.compute.amazonaws.com
- Auto Scaling Group name:** -
- Managed:** false

At the bottom, there are tabs for 'Details', 'Status and alarms', 'Monitoring', 'Security', 'Networking', 'Storage', and 'Tags'. The 'Details' tab is currently selected.

## Instance detail

**Instance details**

- AMI ID: ami-08bfe9f4b1b69da1
- AMI name: ServerUni2
- Stop protection: Disabled
- Instance auto-recovery: Default
- AMI Launch Index: 0
- Credit specification: standard
- Usage operation: RunInstances
- Enclaves Support: -
- Allow tags in instance metadata: Disabled

**Monitoring**

- Monitoring: disabled
- Allowed image: -
- Launch time: Mon Dec 09 2024 11:17:37 GMT-0500 (hora estándar de Colombia) (1 day)
- Lifecycle: normal
- Key pair assigned at launch: ServerUni1
- Kernel ID: -
- RAM disk ID: -
- Boot mode: uefi-preferred
- Use RBN as guest OS hostname: Disabled

**Platform details**

- Linux/UNIX
- Termination protection: Disabled
- AMI location: 509399601385/ServerUni2
- Stop-hibernate behavior: Disabled
- State transition reason: -
- State transition message: -
- Owner: 509399601385
- Current instance boot mode: legacy-bios
- Answer RBN DNS hostname IPv4: Disabled

**Host and placement group**

- Host ID: -
- Host resource group name: -
- Virtualization type: hvm
- Affinity: -
- Tenancy: default
- Reservation: r-0487a0fe4d187e0a1
- Placement group: -
- Placement group ID: -
- Partition number: -

Activate Windows  
Go to Settings to activate Windows.

## Instance monitoring

**CloudWatch agent metrics**

The monitoring tab will now include metrics related to a single instance in the CWAgent namespace. If you want metrics that are emitted from the CloudWatch agent to be displayed, include them in the CWAgent namespace.

Include metrics in the CWAgent namespace [Learn more](#)

Alarm recommendations

1h 3h 12h 1d 3d 1w Custom UTC timezone Add to dashboard

**CPU utilization (%)**

Percent

3.44

1.72

0

22:45 23:45

**Network in (bytes)**

Bytes

1.80k

900

0

22:45 23:45

**Network out (bytes)**

Bytes

1.40k

698

0

22:45 23:45

**Network packets in (count)**

Count

5.6

2.8

0

22:45 23:45

**Network packets out (count)**

Count

7.6

3.8

0

22:45 23:45

**Metadata no token (count)**

No unit

1

0.5

0

22:45 23:45

**CPU credit usage (count)**

Count

0.026

0.013

0

22:45 23:45

**CPU credit balance (count)**

Count

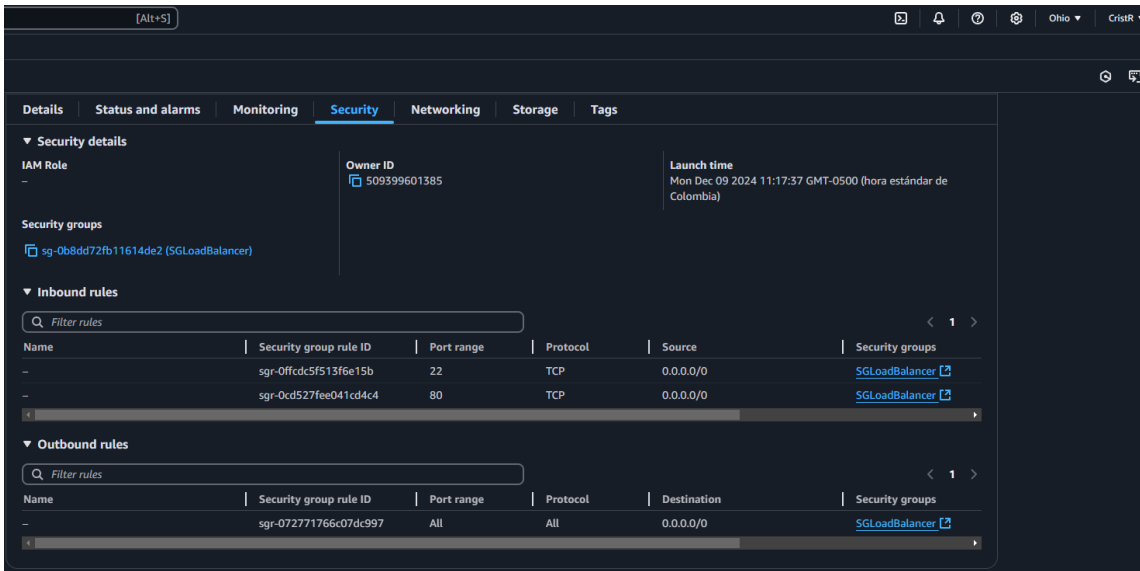
163

81.7

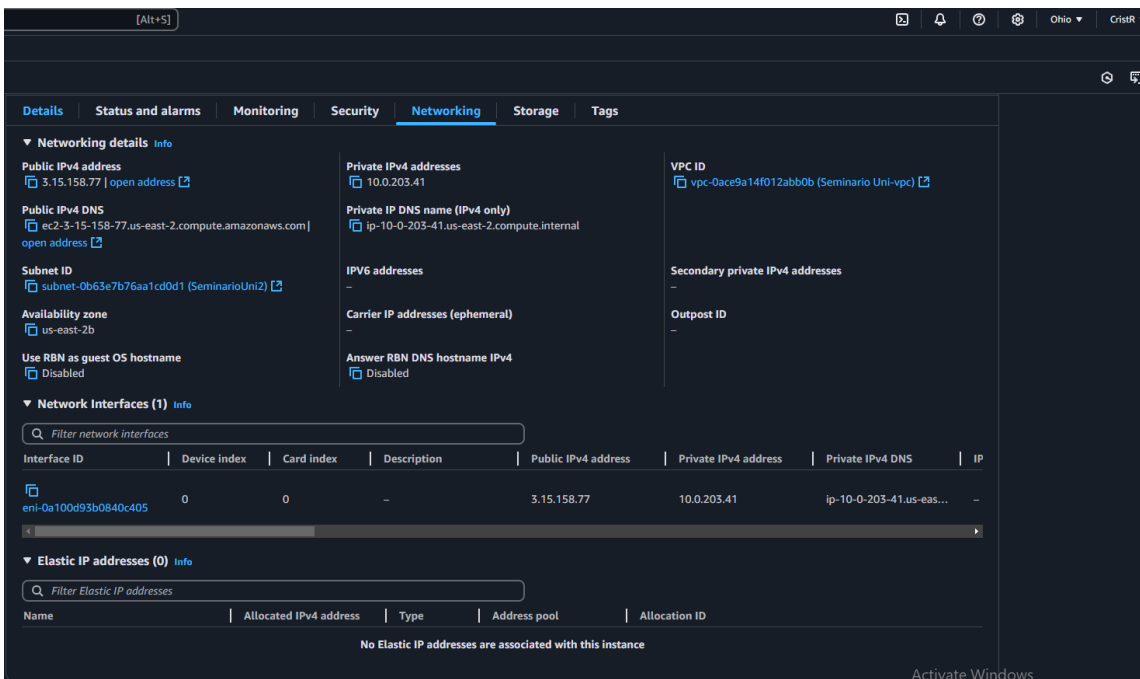
0

22:45 23:45

## Instance security



## Instance networking



## Instance storage



```

root@ip-10-0-203-41:/seminario
GNU nano 5.8
<p>SEMINARIO AMAZON AWS</p>
<p> </p>
  <p>ENTREGA 01</p>
  <p>DEPLOY A WEB SERVICE ON AMAZON AWS</p>
<p> </p>
  <p>1) CREATE VPC</p>
  <p>2) CREATE INSTANCE</p>
  <p>3) CREATE SERVICE WEB</p>
<p> </p>
  <p>ENTREGA 02</p>
  <p>DEPLOY A LOAD BALANCER SERVICE AND AUTO SCALING ON AMAZON AWS</p>
<p> </p>
  <p>1) CREATE SNAPSHOT</p>
  <p>2) CREATE AMIs</p>
  <p>3) CREATE SECURITY GROUP</p>
  <p>4) CREATE TARGET GROUP</p>
  <p>5) CREATE LOAD BALANCER</p>
  <p>6) CONFIGURE LOAD BALANCER</p>
  <p>7) CREATE LAUNCH TEMPLATE</p>
  <p>8) CREATE AUNTO SCALING</p>
  <p>9) CONFIGURE AUTO SCALING</p>
<p> </p>
  <p>ENTREGA 03</p>
  <p>AWS CONTAINER SERVICE WITH LOAD BALANCER AND AUTO SCALING</p>
<p> </p>
  <p>1) MODIFY INSTANCE</p>
  <p>2) DISABLE AND STOP HTTPD</p>
  <p>3) INSTALL AND START AND ENABLE NGINX</p>
  <p>4) CREATE FOLDER AND INDEX</p>
  <p>5) INSTALL AND START AND ENABLE DOCKER</p>
  <p>6) DOWNLOAD IMAGE DOCKER APACHE2 HTTPD</p>
  <p>7) CREATE 4 CONTAINER AND VERIFY</p>
  <p>8) MODIFY NGINX.CONF</p>
  <p>9) MODIFY LOAD BALANCER AND AUTO SCALING FROM THIS IMAGE</p>
<p> </p>
  <p>INTEGRANTES</p>
<p> </p>
  <p>- CRISTHIAM TANIRY ROMERO FLOREZ</p>
  <p>- GUSTAVO ADOLFO GONZALEZ SANTIAGO</p>

```

Docker containers

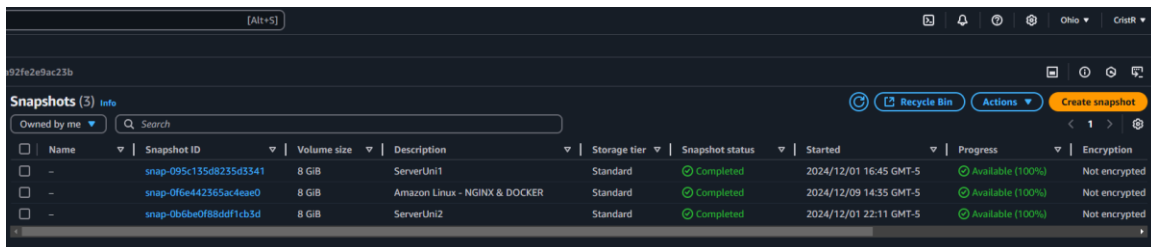


```
root@ip-10-0-203-41:/etc/nginx
GNU nano 5.8
events {}

http {
    upstream container {
        server localhost:8081;
        server localhost:8082;
        server localhost:8083;
        server localhost:8084;
    }

    server {
        listen 80;
        server_name nginx;
        location / {
            proxy_pass http://container;
        }
    }
}
```

## Snapshots



<input type="checkbox"/>	Name	Snapshot ID	Volume size	Description	Storage tier	Snapshot status	Started	Progress	Encryption
<input type="checkbox"/>	-	snap-095c135d8235d3341	8 GiB	ServerUnit1	Standard	Completed	2024/12/01 16:45 GMT-5	Available (100%)	Not encrypted
<input type="checkbox"/>	-	snap-0f6e42365ac4eae0	8 GiB	Amazon Linux - NGINX & DOCKER	Standard	Completed	2024/12/09 14:35 GMT-5	Available (100%)	Not encrypted
<input type="checkbox"/>	-	snap-0b6be0f88ddf1cb3d	8 GiB	ServerUnit2	Standard	Completed	2024/12/01 22:11 GMT-5	Available (100%)	Not encrypted

## Snapshot details

**snap-0f6e442365ac4eae0**

Snapshot ID: snap-0f6e442365ac4eae0

Progress: Available (100%)

Snapshot status: Completed

Owner: 509399601385

Started: Mon Dec 09 2024 14:35:07 GMT-0500 (hora estándar de Colombia)

Product codes: -

Fast snapshot restore: -

Description: Amazon Linux - NGINX & DOCKER

Source volume: vol-022aaabe565bf8822

Volume size: 8 GiB

Encryption: Not encrypted

KMS key ID: -

KMS key alias: -

KMS key ARN: -

Snapshot settings | Storage tier | Tags

Snapshot Lock - new

Lock mode: Not locked

Share permissions

Snapshot share permissions: Private

Modify permissions

Shared accounts (0)

The snapshot is shared with the following AWS accounts:

## AMI

Amazon Machine Images (AMIs) (3)

Name	AMI name	AMI ID	Source	Owner	Visibility	Status	Creation date
AmzLinuxCrist	ami-0ae1315190080c816	509399601385/AmzLinuxCrist	509399601385	Private	Available	2024/12/01 16:55 GMT-5	
ServerUnit2	ami-08bfe9f4b1b69da1	509399601385/ServerUnit2	509399601385	Private	Available	2024/12/01 22:14 GMT-5	
AmazonLinuxNginxDocker	ami-024326b9f5cea27ad	509399601385/AmazonLinuxNginxDoc...	509399601385	Private	Available	2024/12/09 14:51 GMT-5	

## AMI detail

**Image summary for ami-024326b9f5cea27ad**

<b>AMI ID</b> ami-024326b9f5cea27ad	<b>Image type</b> machine	<b>Platform details</b> Linux/UNIX	<b>Root device type</b> EBS
<b>AMI name</b> AmazonLinuxNgInxDocker	<b>Owner account ID</b> 509399601385	<b>Architecture</b> x86_64	<b>Usage operation</b> RunInstances
<b>Root device name</b> /dev/sda1	<b>Status</b> Available	<b>Source</b> 509399601385/AmazonLinuxNgInxDocker	<b>Virtualization type</b> hvm
<b>Boot mode</b> uefi-preferred	<b>State reason</b> -	<b>Creation date</b> 2024-12-09T19:51:23.000Z	<b>Kernel ID</b> -
<b>Description</b> Amazon Linux - NGINX & DOCKER	<b>Product codes</b> -	<b>RAM disk ID</b> -	<b>Deprecation time</b> -
<b>Last launched time</b> Mon Dec 09 2024 16:06:07 GMT-0500 (hora estándar de Colombia)	<b>Block devices</b> /dev/sda1=snap-0f6e442365ac4eae0:8:tr ue:gp3	<b>Deregistration protection</b> Disabled	<b>Allowed image</b> -
<b>Source AMI ID</b> -	<b>Source AMI Region</b> -		

## Target Groups

**Target groups (1)**

Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
TGServerUni	arn:aws:elasticloadbalancing...	80	HTTP	Instance	LBServerUni	vpc-0ace9a14f012abb0b

## Target group detail

**TGServerUni** Actions ▾

**Details**  
 am:aws:elasticloadbalancing:us-east-2:509399601385:targetgroup/TGServerUni/10d4f6e61337345b

<b>Target type</b> Instance	<b>Protocol : Port</b> HTTP: 80	<b>Protocol version</b> HTTP1	<b>VPC</b> <a href="#">vpc-0ace9a14f012abb0b</a>
<b>IP address type</b> IPv4	<b>Load balancer</b> <a href="#">LbServerUni</a>		

4 Total targets 4 Healthy 0 Unhealthy 0 Unused 0 Initial 0 Draining  
 0 Anomalous

► **Distribution of targets by Availability Zone (AZ)**  
 Select values in this table to see corresponding filters applied to the Registered targets table below.

**Targets** | Monitoring | Health checks | Attributes | Tags

**Registered targets (4)** Anomaly mitigation: Not applicable | Deregister | Register targets

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

Filter targets

Instance ID	Name	Port	Zone	Health status	Health status details	Admini...
<a href="#">i-009446918f514aac2</a>		80	us-east-2b (use2-az2)	Healthy	-	No override, Nc

## Targets

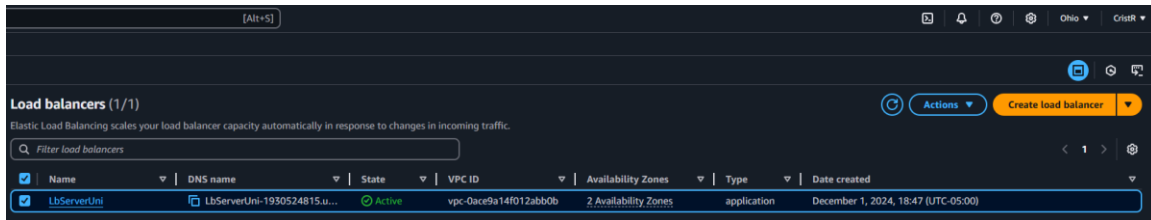
**Registered targets (4)** Anomaly mitigation: Not applicable | Deregister | Register targets

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

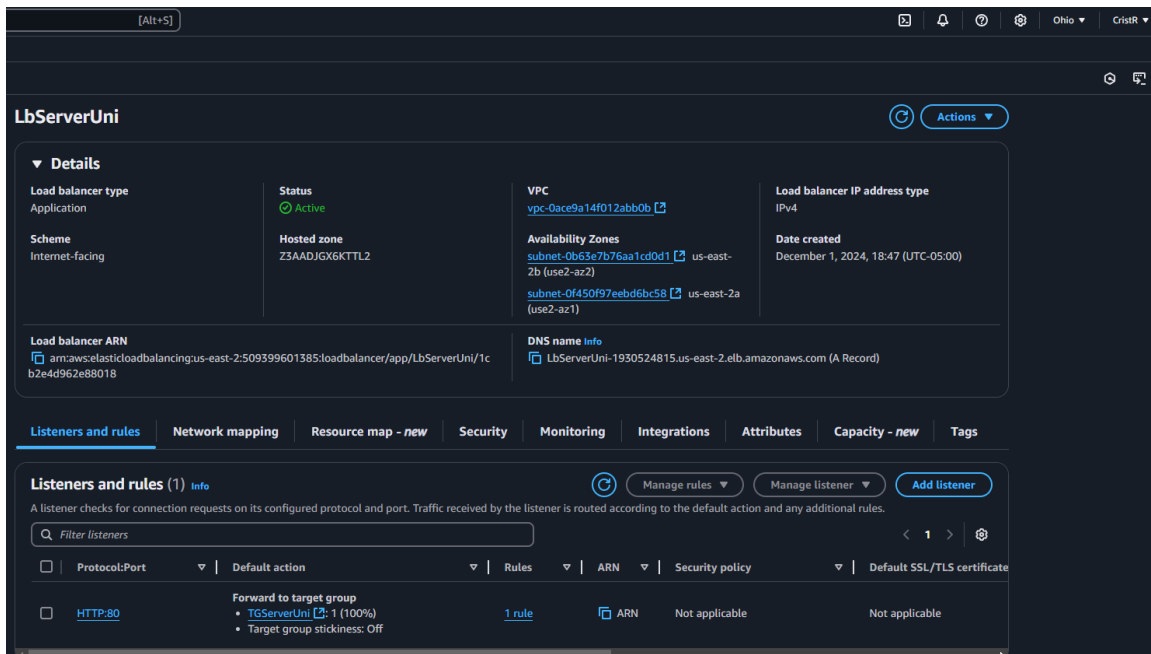
Filter targets

Instance ID	Name	Port	Zone	Health status	Health status details	Admini...
<a href="#">i-009446918f514aac2</a>		80	us-east-2b (use2-az2)	Healthy	-	No override, Nc
<a href="#">i-0db1042db0275912d</a>		80	us-east-2a (use2-az1)	Healthy	-	No override, Nc
<a href="#">i-01182d4957af37d54</a>		80	us-east-2b (use2-az2)	Healthy	-	No override, Nc
<a href="#">i-07c05be152c0c810a</a>		80	us-east-2a (use2-az1)	Healthy	-	No override, Nc

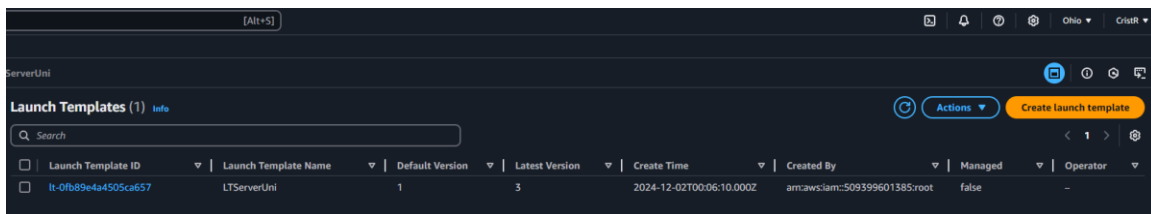
## Load Balancer



## Load Balancer details



## Launch Template



## Launch Template details

The screenshot displays the 'Launch template details' page for 'LTServerUni' (ID: lt-0fb89e4a4505ca657). The page is divided into two main sections: 'Launch template details' and 'Launch template version details'.

**Launch template details:**

- Launch template ID: lt-0fb89e4a4505ca657
- Launch template name: LTServerUni
- Default version: 1
- Owner: am:awsiam::509399601385:root

**Launch template version details (Version 1):**

- Description: LT
- Date created: 2024-12-02T00:06:10.000Z
- Created by: am:awsiam::509399601385:root

**Instance details:**

- AMI ID: ami-0ae1315190080c816
- Instance type: t2.micro
- Availability Zone: -
- Key pair name: ServerUni1
- Security groups: sg-0b8dd72fb11614de2

## Auto Scaling Group

The screenshot shows the 'Auto Scaling groups (1)' page in the AWS Management Console. It features a search bar and a table listing the Auto Scaling Group.

**Auto Scaling groups (1) Info**

Search your Auto Scaling groups

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
ASGServerUni	LTServerUni   Version Latest	4	-	4	2	5	us-east-2a, us-east-2b

## Auto Scaling Group details

[Alt+S] Ohio CristR

## ASGServerUni

### ASGServerUni Capacity overview Edit

arn:aws:autoscaling:us-east-2:509399601385:autoScalingGroup:60729bb5-818a-4fdd-8451-d3a49b96a092:autoScalingGroupName/ASGServerUni

<b>Desired capacity</b> 4	<b>Scaling limits (Min - Max)</b> 2 - 5	<b>Desired capacity type</b> Units (number of instances)	<b>Status</b> -
------------------------------	--	---	--------------------

**Date created**  
Sun Dec 01 2024 21:37:51 GMT-0500 (hora estándar de Colombia)

[Details](#) | [Integrations - new](#) | [Automatic scaling](#) | [Instance management](#) | [Instance refresh](#) | [Activity](#) | [Monitoring](#)

### Launch template Edit

<b>Launch template</b> lt-0fb89e4e4505ca657 LTServerUni	<b>AMI ID</b> ami-024326b9f5cea27ad	<b>Instance type</b> t2.micro	<b>Owner</b> arn:aws:iam::509399601385:root
<b>Version</b> Latest	<b>Security groups</b> -	<b>Security group IDs</b> sg-0b8dd72fb11614de2	<b>Create time</b> Mon Dec 09 2024 16:04:13 GMT-0500 (hora estándar de Colombia)
<b>Description</b> LTDoker <a href="#">View details in the launch template console</a>	<b>Storage (volumes)</b> -	<b>Key pair name</b> ServerUni1	<b>Request Spot Instances</b> No

### Network Edit

<b>Availability Zones</b> us-east-2a, us-east-2b	<b>Subnet ID</b> subnet-0b63e7b76aa1cd0d1, subnet-	<b>Availability Zone distribution</b> Balanced best effort	<b>Activate Windows</b> Go to Settings to activate Windows.
---	---	---	--

## Conclusiones

El Proyecto AWS CONTAINER SERVICE WITH LOAD BALANCER AND AUTO SCALING se implementó con total éxito en la empresa TITAN BUILD PC STORE, con este servicio esta empresa podrá gozar de un servicio web de alta disponibilidad y podrá dar atención a todas esos clientes en busca de hardware para equipos de cómputo que así mismo dará crecimiento en sus ventas y a su empresa sin que ningún cliente se pierda por falta de estabilidad en el servicio web.

### **Referencias**

- <https://aws.amazon.com/es/getting-started/>
- <https://aws.amazon.com/es/vpc/>

- <https://aws.amazon.com/es/what-is/cloud-instances/>
- [https://docs.aws.amazon.com/es\\_es/linux/al2023/ug/what-is-amazon-linux.html](https://docs.aws.amazon.com/es_es/linux/al2023/ug/what-is-amazon-linux.html)
- <https://aws.amazon.com/es/ebs/snapshots/>
- [https://docs.aws.amazon.com/es\\_es/AWSEC2/latest/UserGuide/AMIs.html](https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/AMIs.html)
- [https://docs.aws.amazon.com/es\\_es/vpc/latest/userguide/vpc-security-groups.html](https://docs.aws.amazon.com/es_es/vpc/latest/userguide/vpc-security-groups.html)
- [https://docs.aws.amazon.com/es\\_es/elasticloadbalancing/latest/application/load-balancer-target-groups.html](https://docs.aws.amazon.com/es_es/elasticloadbalancing/latest/application/load-balancer-target-groups.html)
- [https://docs.aws.amazon.com/es\\_es/elasticloadbalancing/latest/application/introduction.html](https://docs.aws.amazon.com/es_es/elasticloadbalancing/latest/application/introduction.html)
- <https://aws.amazon.com/es/about-aws/whats-new/2017/11/introducing-launch-templates-for-amazon-ec2-instances/>
- <https://aws.amazon.com/es/autoscaling/>
- <https://www.hostinger.co/tutoriales/que-es-nginx>
- <https://aws.amazon.com/es/docker/>
- <https://www.docker.com/resources/what-container/>