TRABAJO DE GRADO
Opción Seminario-Diplomado.

Seminario AWS

Corporación Universitaria Remington.
Ingenierías:
Ingeniería de Sistemas:

Estudiante: Andrés Felipe Alarcón Guerrero
Docente: Juan Pablo Berrio López

Opción de Trabajo de grado Seminario-Diplomado.
2025

Dedicatoria

Bubalú gracias a tu compañía y amor incondicional.

Agradecimientos

Gracias Juan Pablo Berrio López por tus enseñanzas y estilo pedagógico. Estas sembrando un conocimiento invaluable a cada profesional. Agradecimiento extendido a la Remington por ofrecer educación de extensión de buena calidad y con relevancia en el mercado.

Tabla de Contenidos

Resumen

Este seminario de grado aborda los fundamentos de Amazon Web Services (AWS), plataforma de computación en la nube que ofrece servicios dirigidos tanto a empresas como a desarrolladores. El objetivo principal del seminario es brindar una comprensión clave de los servicios en AWS, buenas prácticas recomendadas para garantizar la seguridad, la alta disponibilidad y la escalabilidad de las aplicaciones implementadas en la nube.

Palabras clave

AWS, computación en la nube, servicios, seguridad, escalabilidad

Marco conceptual y contextual

El presente trabajo describe el proceso de aprendizaje y uso de recursos de cómputo en la nube para AWS. Trabajo guiado por la formación del Seminario AWS de la Remington, donde se abordan conceptos y servicios claves para el uso de AWS en la capa gratuita.

Para la implementación de este trabajo es importante comprender una serie de conceptos que se utilizan en la plataforma AWS.

Conceptos

A continuación, se pondrá en contexto conceptos clave necesarios para comprender el siguiente trabajo. Información tomada de Amazon Web Services. (s.f.) y de Seminario AWS (2025).

VPC

Virtual Private Cloud. Funciona de manera análoga a un centro de datos. Sirve para la creación de redes privadas. En la capa gratuita se permite tener hasta dos VPC sin cobro adicional.

Dentro de un VPC se define todo el direccionamiento IP que utilizarán los recursos asociados, como las máquinas virtuales. Para organizar esta red, el VPC se segmenta en subredes, que pueden establecerse para distintos propósitos, como separar los entornos de desarrollo y administración, o definirse por departamentos.

Conectividad y Seguridad de Red

Las subredes se clasifican en públicas y privadas. Las subredes públicas tienen acceso a internet mediante un Internet Gateway (IGW), que funciona como un enrutador para dar salida y entrada a las conexiones.

Por defecto, las subredes privadas no tienen conexión a internet. Sin embargo, en ocasiones se necesita acceso para tareas como la actualización de software de seguridad. Para solucionar esto sin exponer los recursos a conexiones entrantes, se utiliza un NAT Gateway. Este dispositivo permite que los recursos en la subred privada inicien conexiones hacia internet (salida), pero impide que desde internet se pueda acceder a ellos. Cabe mencionar que el NAT Gateway es un recurso con un costo mensual.

Para optimizar costos y seguridad, se pueden usar VPC Endpoints. Estos permiten que un recurso dentro del VPC acceda a otros servicios de AWS, como el almacenamiento de objetos S3, de forma directa y sin pasar por internet, lo que puede reducir los cargos asociados al NAT Gateway.

Recursos de Cómputo y Gestión

En AWS, una máquina virtual se denomina instancia. Su seguridad perimetral se gestiona a través de Security Groups, que actúan como un firewall. En sus reglas se puede definir el origen o destino del tráfico; por ejemplo, una regla de salida con destino 0.0.0.0/0 indica que se permite la conexión hacia cualquier lugar en internet.

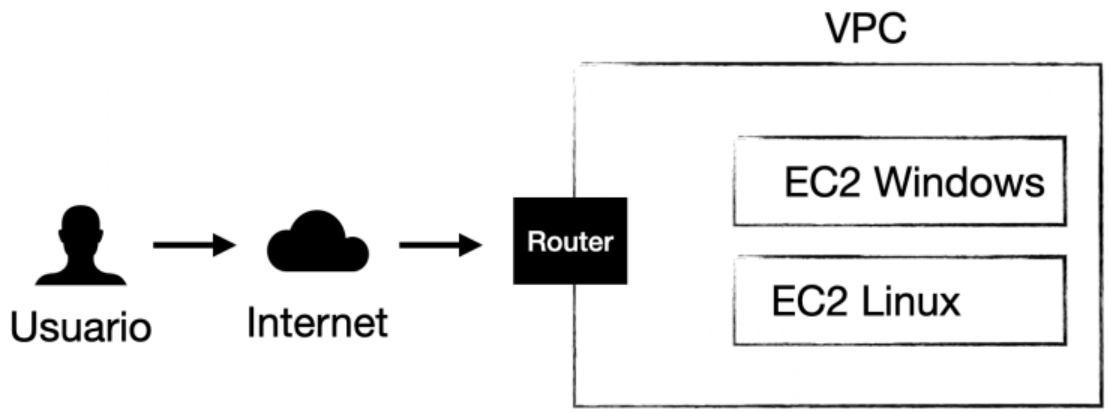
Finalmente, la gestión de costos es un aspecto crítico. A través de AWS Budgets, es posible crear presupuestos que funcionan como una alerta; el sistema notifica cuando el uso está a punto de superar un umbral predefinido. En cuanto al monitoreo, la capa gratuita ofrece métricas con una frecuencia de actualización de 5 minutos, un tiempo de respuesta considerado no apto para entornos productivos, mientras que las opciones de pago ofrecen datos en tiempo real.

ELB (Elastic Load Balancer)

En AWS, es un recurso que recibe las peticiones y las envía a las instancias de EC2. Requiere un Recurso llamado Listening, que recibe las peticiones, mismas peticiones que entran al ELB, y finalmente el ELB las envía al TargetGroup, que es el que contiene las instancias que hacen parte del balanceador de carga. El TargetGroup para considerar que una maquina es apta para registrarse tiene que pasar una validación que el servidor web está funcionando (200 http).

Documentación técnica

Figura 1. VPC Base



Autoría propia.

Descripción de la arquitectura

La base de la infraestructura es una Virtual Private Cloud (VPC) personalizada para garantizar un entorno de red aislado.

1. VPC: Se creó una VPC con el nombre seminario-vpc y un bloque de direcciones IP de 10.0.0.0/16.

2. Subredes: La VPC contiene múltiples subredes para segmentar la red. Las instancias se desplegaron en subredes públicas, como seminario-subnet-public1-us-east-1a con el CIDR 10.0.10.0/20, para permitir el acceso a Internet.

3. Conectividad a Internet: Un Internet Gateway (seminario-igw) se adjuntó a la VPC y se configuraron tablas de enrutamiento (seminario-rtb-public) para dirigir el tráfico desde y hacia internet a las subredes públicas.

Se desplegaron dos máquinas virtuales (instancias EC2) de tipo t2.micro en la zona de disponibilidad us-east-1a, cada una con un sistema operativo diferente.

Server1Windows

IP Pública: 98.81.36.30

IP Privada: 10.0.13.102

Server2Linux

IP Pública: 54.236.55.37

IP Privada: 10.0.12.218

Configuración de Seguridad

La seguridad y el acceso a las instancias se controlan mediante Grupos de Seguridad, que actúan como firewalls virtuales.

sgWindowsServer: Permite el tráfico entrante para:

RDP (puerto 3389) para administración remota.

HTTP (puerto 80) para el servidor web.

ICMP para permitir pruebas de conectividad (ping).

sgLinuxServer: Permite el tráfico entrante para:

SSH (puerto 22) para administración remota.

HTTP (puerto 80) para el servidor web.

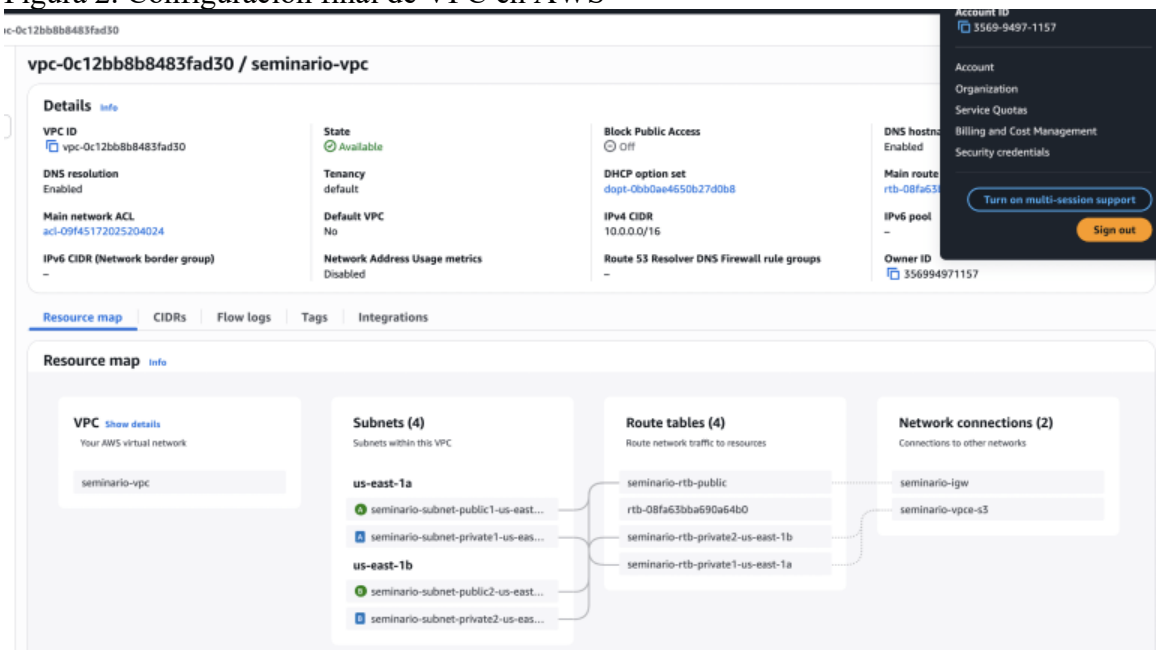
ICMP para permitir pruebas de conectividad (ping).

Estas reglas garantizan que cada servidor sea accesible públicamente a través de la web y que se puedan administrar de forma remota, al tiempo que permiten la comunicación entre ellos.

Configuración realizada

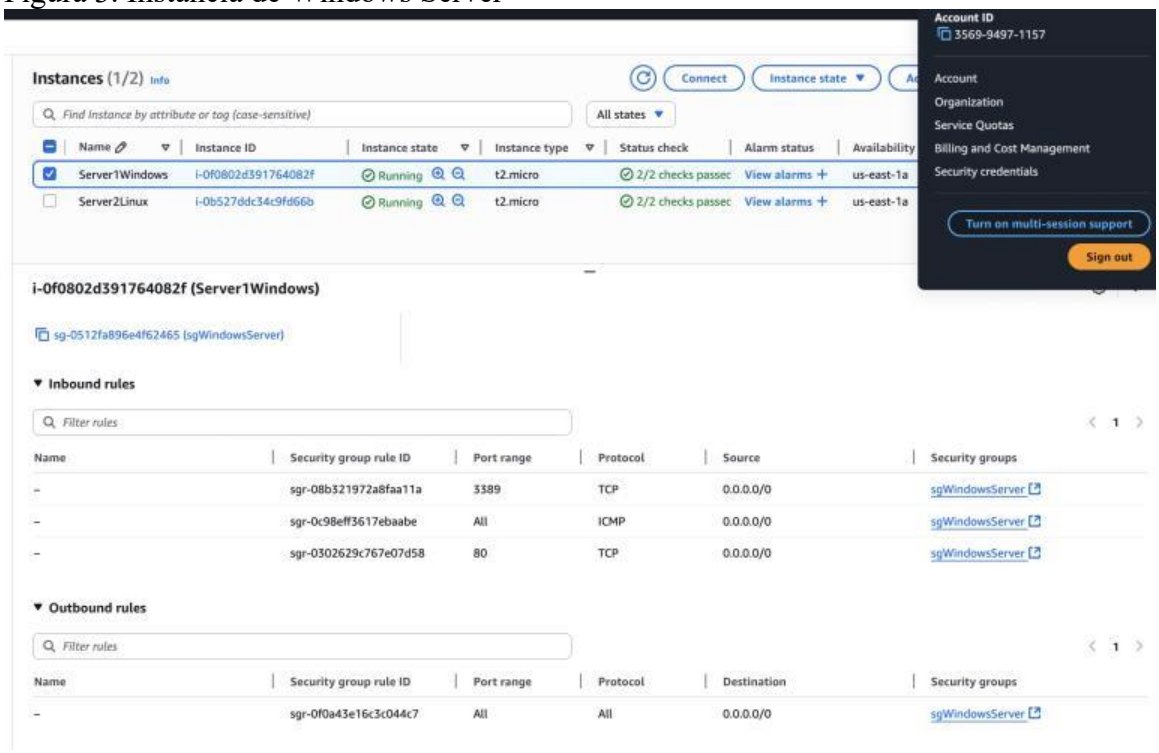
VPC

Figura 2. Configuración final de VPC en AWS



Autoría propia.

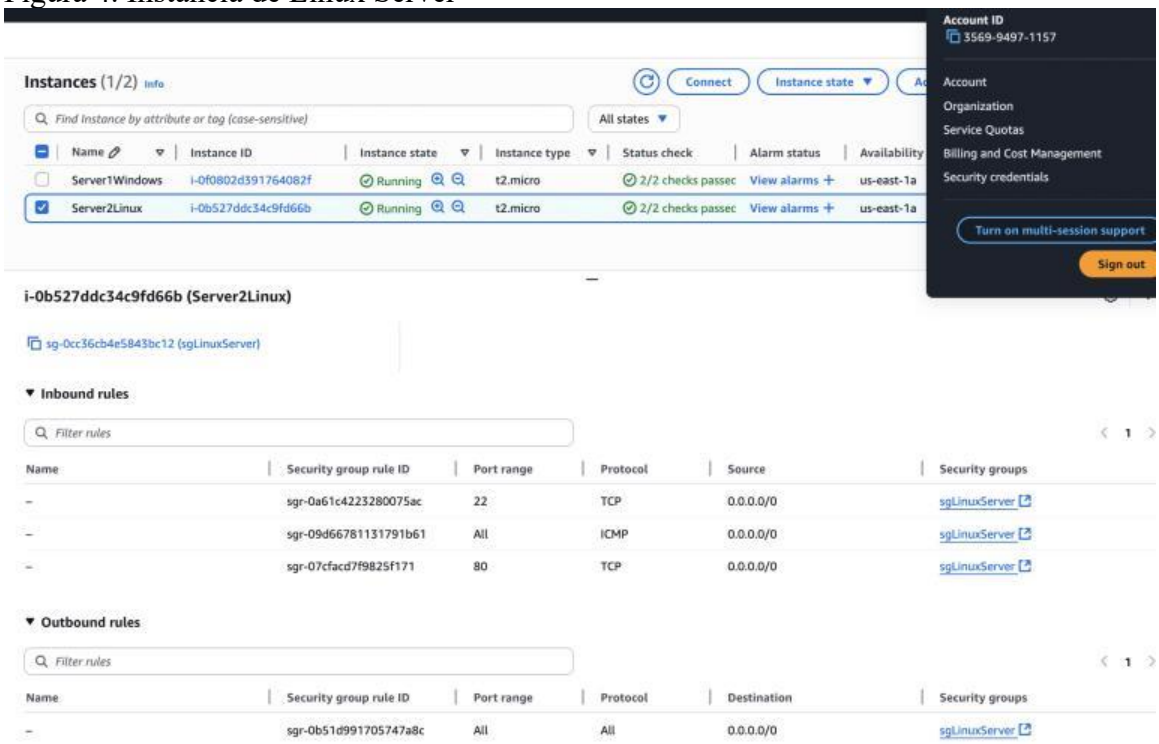
Figura 3. Instancia de Windows Server



Autoría propia.

Linux Server

Figura 4. Instancia de Linux Server



Autoría propia.

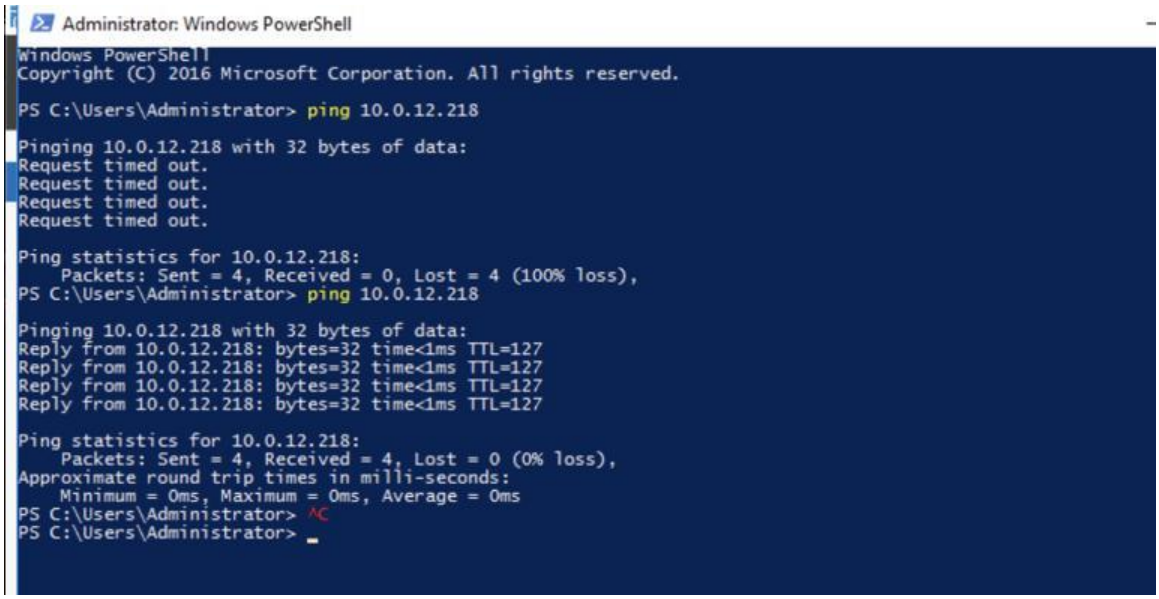
Procedimiento de Acceso

Se validó el cumplimiento de los requisitos de conectividad.

Accesibilidad Pública: La asignación de IPs públicas y las reglas de los grupos de seguridad en el puerto 80 aseguran que los servidores web instalados en cada instancia sean accesibles desde internet.

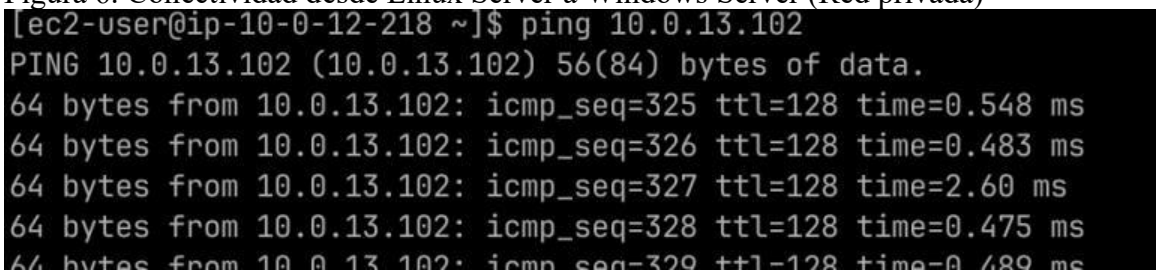
Conectividad Interna: Las pruebas de ping entre las instancias utilizando sus direcciones IP privadas fueron exitosas. La máquina Windows (10.0.13.102) pudo comunicarse con la máquina Linux (10.0.12.218) y viceversa, confirmando que la red interna es funcional.

Figura 5. Conectividad desde Windows Server a Linux Server (Red privada)



Autoría propia.

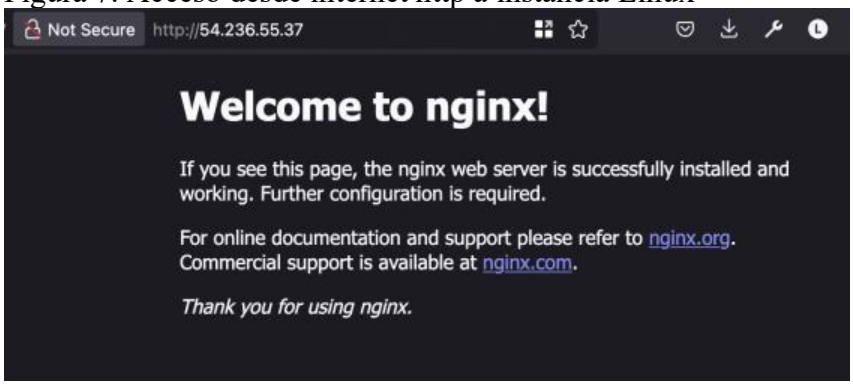
Figura 6. Conectividad desde Linux Server a Windows Server (Red privada)



Autoría propia.

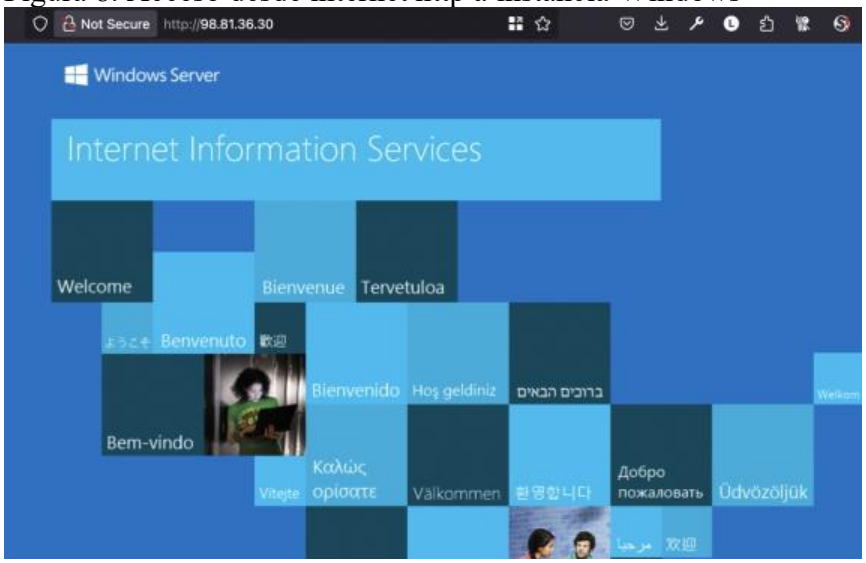
Internet

Figura 7. Acceso desde internet http a instancia Linux



Autoría propia.

Figura 8. Acceso desde internet http a instancia Windows



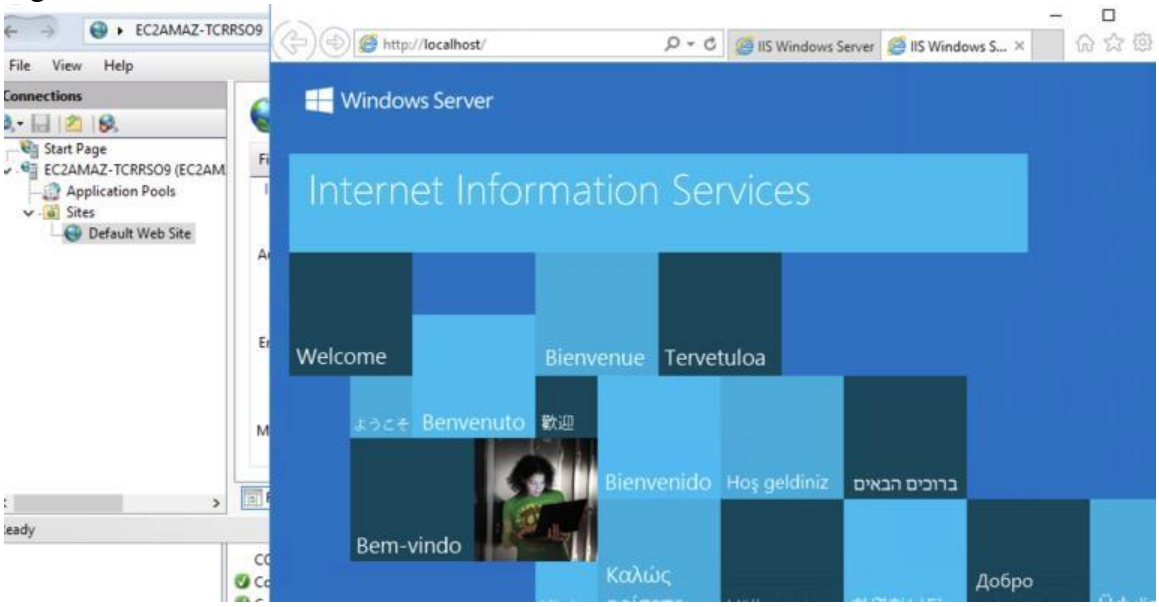
Autoría propia.

Configuración del Servidor

Figura 9. Configuración IIS en instancia EC2 Windows

Autoría propia.

Figura 10. Validación local de IIS en instancia EC2 Windows

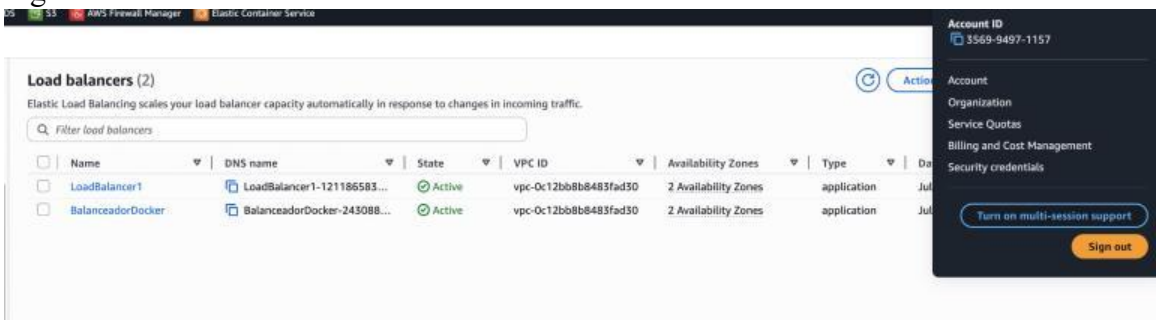


Autoría propia.

Linux

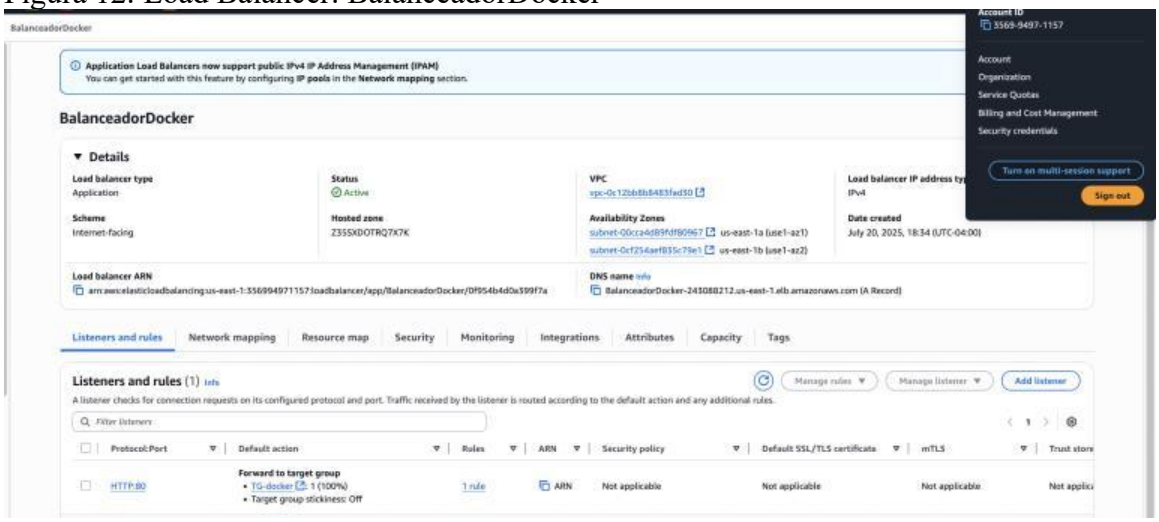
```
sudo dnf update –y
sudo dnf install nginx –y
sudo systemctl start nginx
sudo systemctl enable nginx

[ec2-user@ip-10-0-12-218 ~]$ curl http://localhost
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

Figura 11. Load Balancers



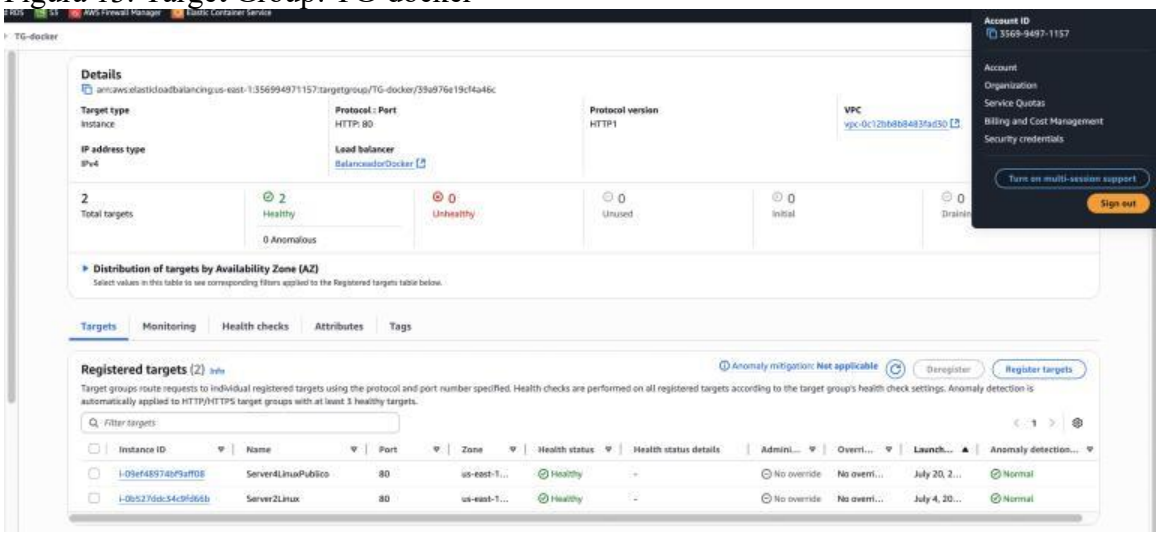
Autoría propia.

Figura 12. Load Balancer: BalanceadorDocker



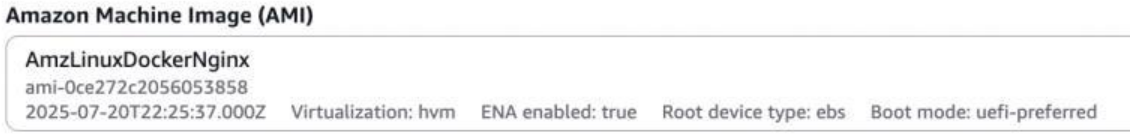
Autoría propia.

Figura 13. Target Group: TG-docker



Autoría propia.

Figura 14. AMI Base Docker

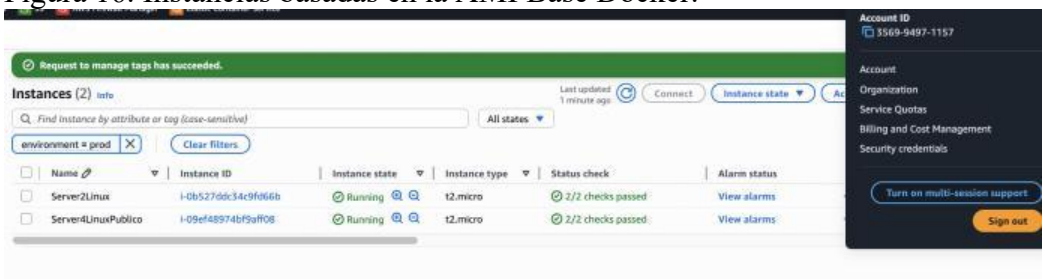


Autoría propia.

Figura 15. Docker containers

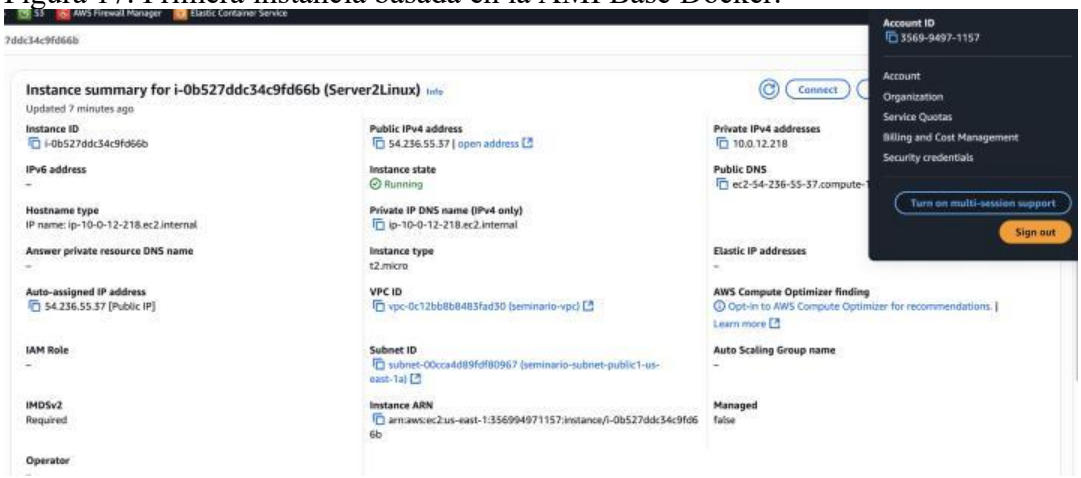
Autoría propia.

Figura 16. Instancias basadas en la AMI Base Docker.



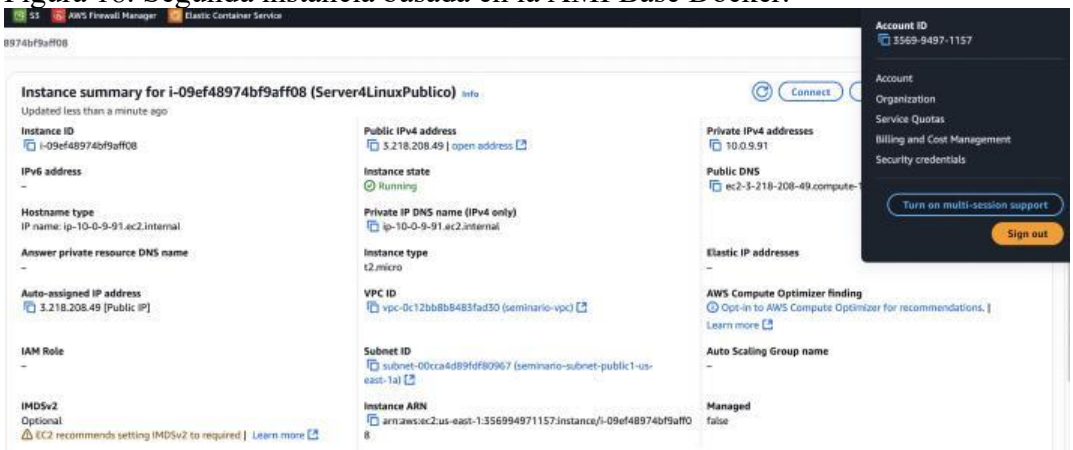
Autoría propia.

Figura 17. Primera instancia basada en la AMI Base Docker.



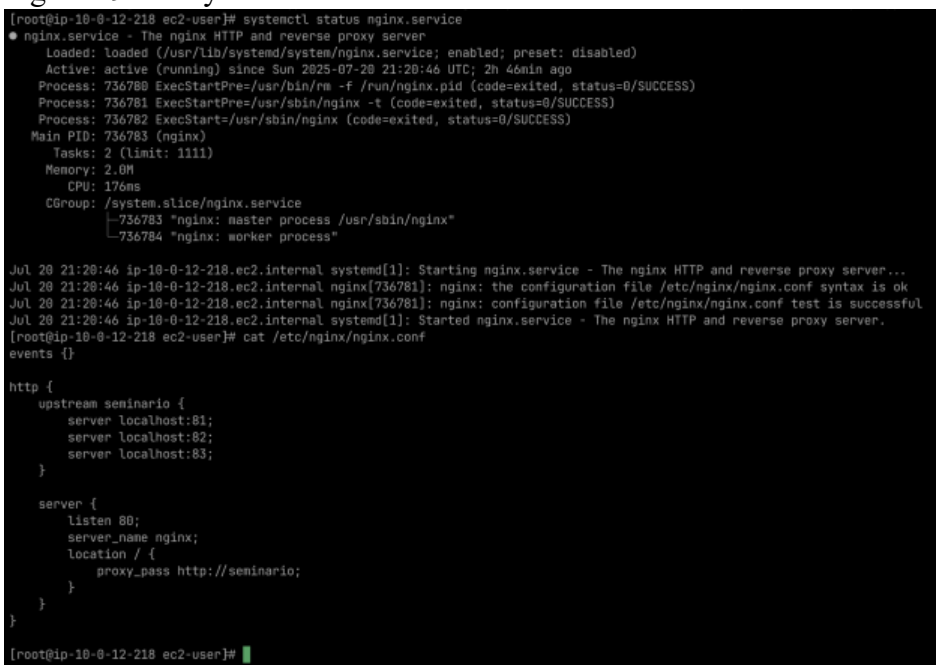
Autoría propia.

Figura 18. Segunda instancia basada en la AMI Base Docker.



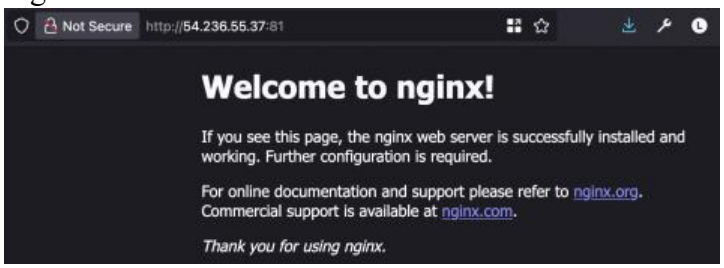
Autoría propia.

Figura 19. Proxy Reverse a Contenedores Docker



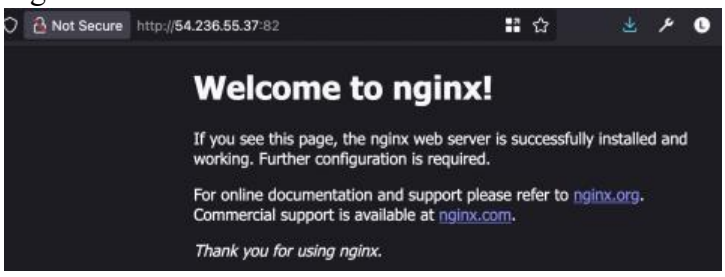
Autoría propia.

Figura 20. Conectividad Contenedor 1



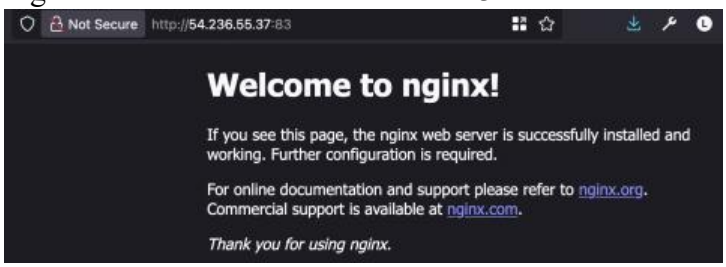
Autoría propia.

Figura 21. Conectividad Contenedor 2



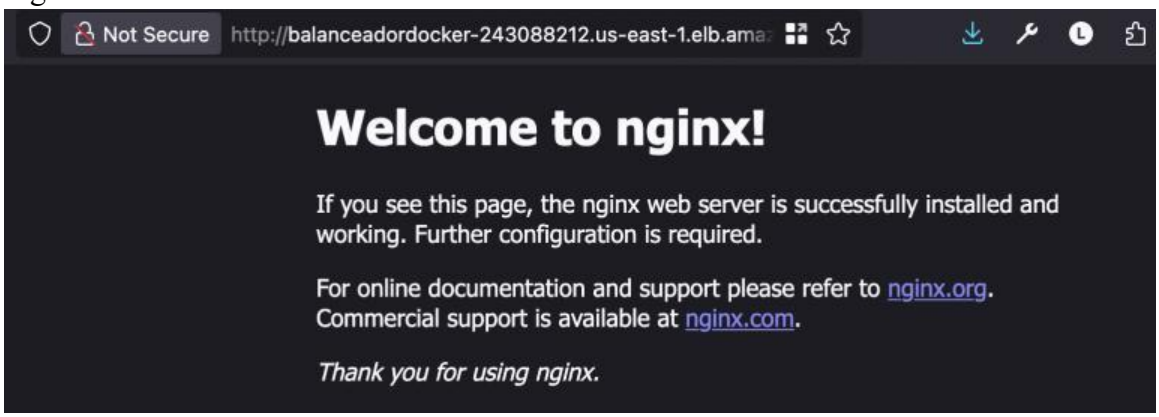
Autoría propia.

Figura 22. Conectividad Contenedor 3



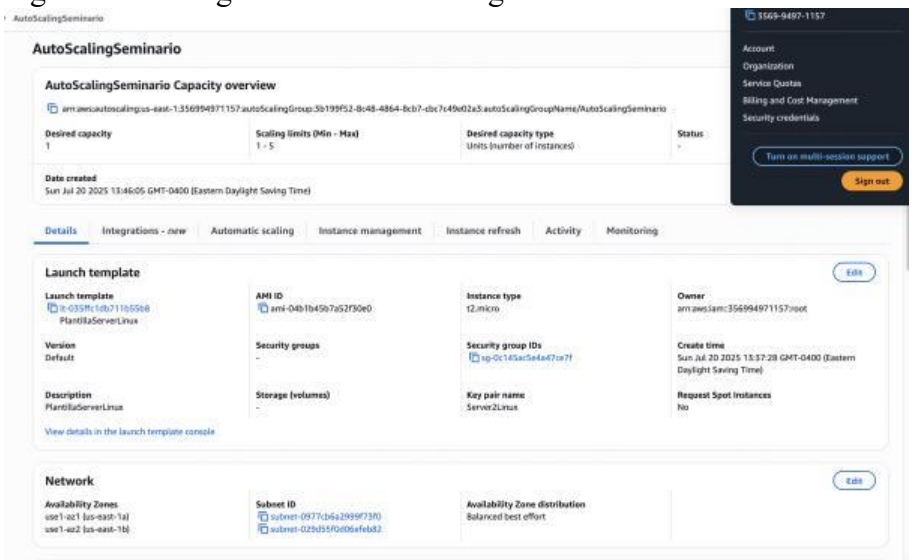
Autoría propia.

Figura 23. Conectividad DNS Load Balancer



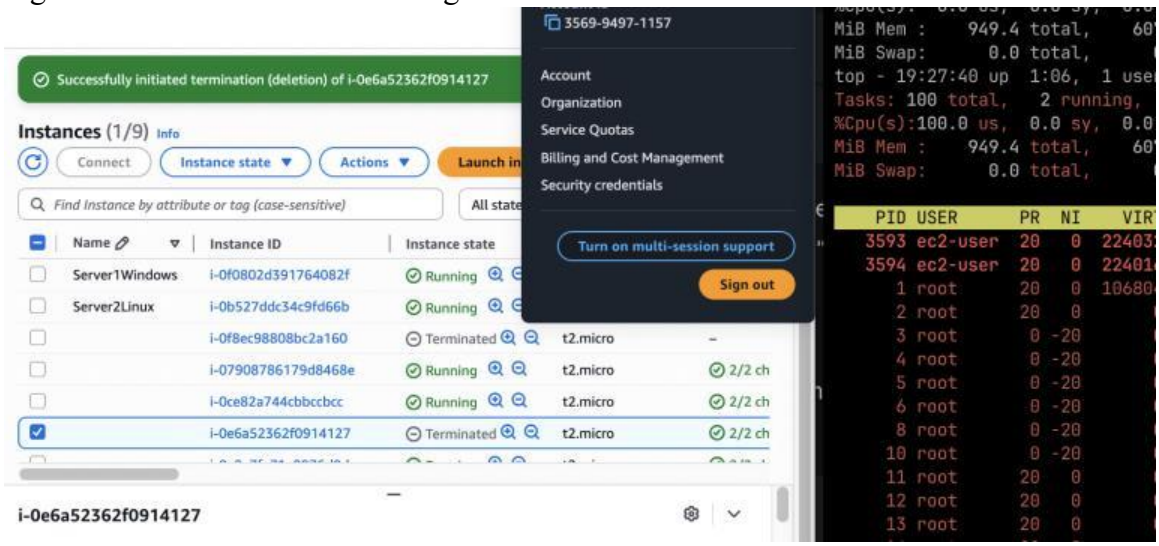
Autoría propia.

Figura 24. Configuración Auto Scaling



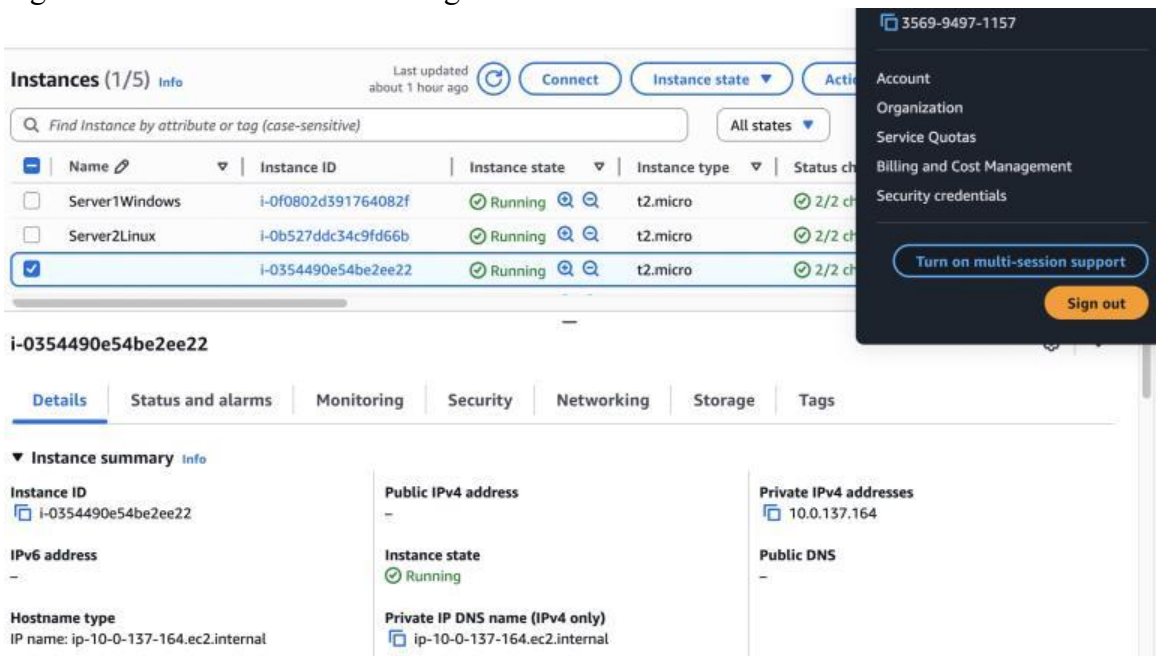
Autoría propia.

Figura 25. Validación Auto Scaling



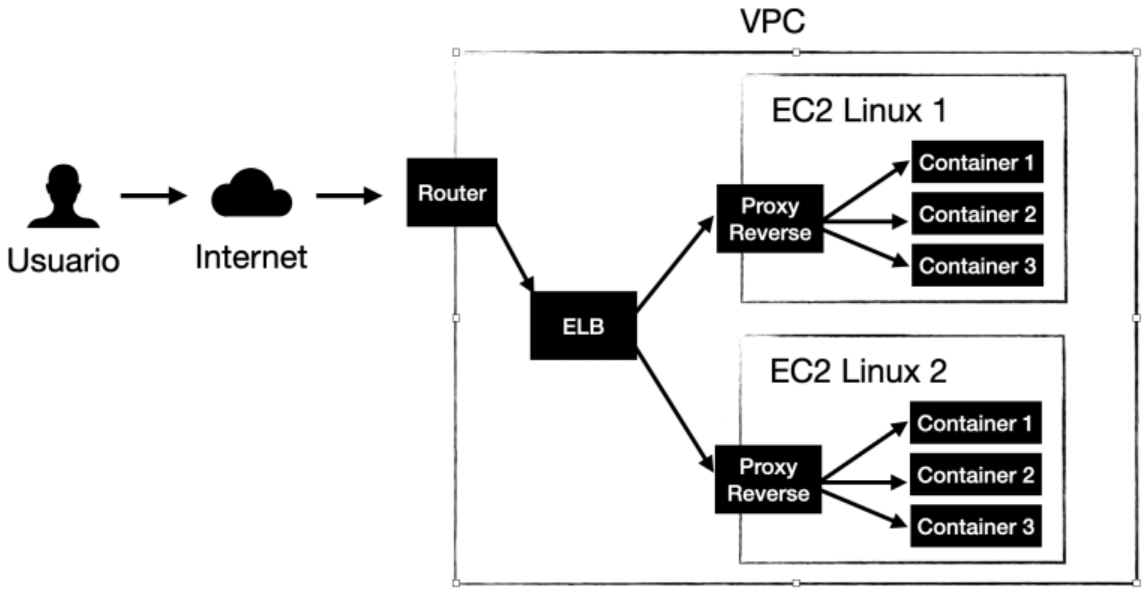
Autoría propia.

Figura 26. Validación Downscaling



Autoría propia.

Figura 27. Diagrama Seguridad, alta disponibilidad y escalabilidad de las aplicaciones.



Autoría propia.

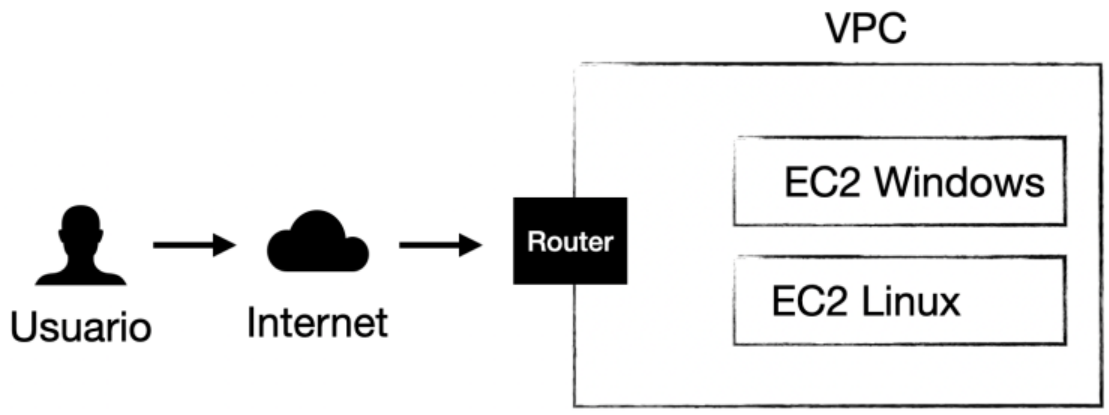
Conclusiones

El presente trabajo concluye con la implementación de una arquitectura en la nube de las más necesarias y fundamentales en AWS, con un enfoque en la seguridad, alta disponibilidad y escalabilidad de dichas aplicaciones.

Implementación que hace uso de conceptos clave en implementaciones maduras puestas en producción como lo son el uso de contenedores, balanceadores de carga, máquinas virtuales, centro de datos, firewalls, subredes, proxy inverso, redundancia, servidores de aplicaciones, control de costos y notificaciones, por mencionar los más relevantes.

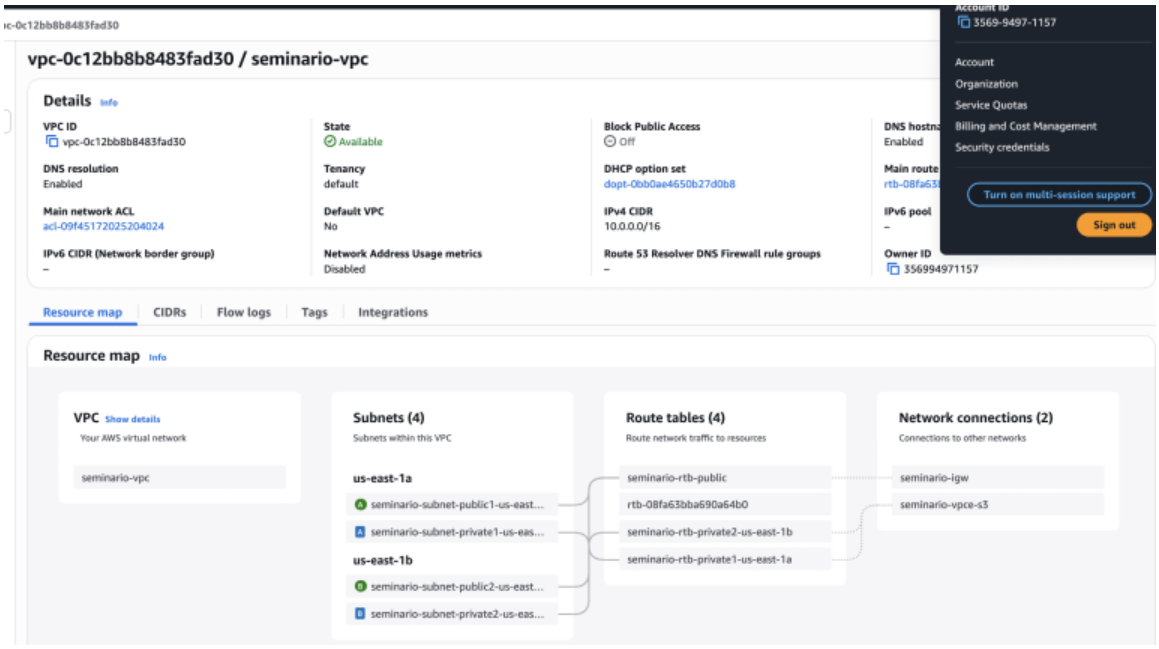
Figuras

Figura 1. VPC Base



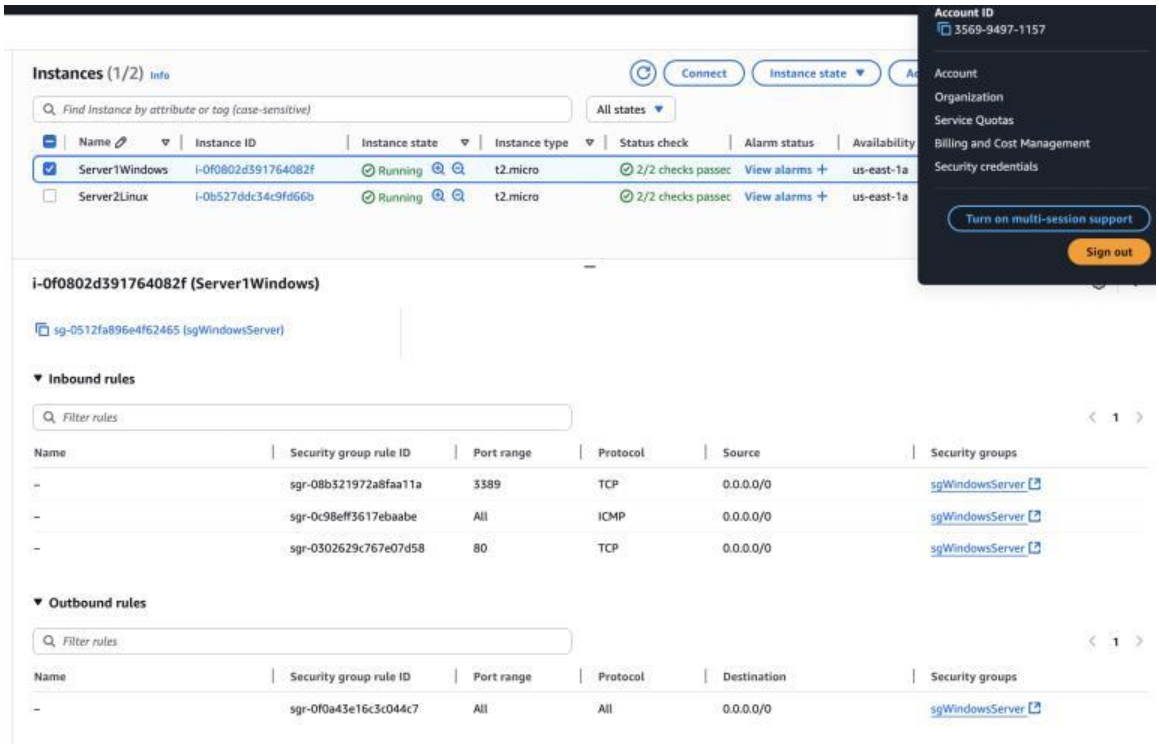
Autoría propia.

Figura 2. Configuración final de VPC en AWS



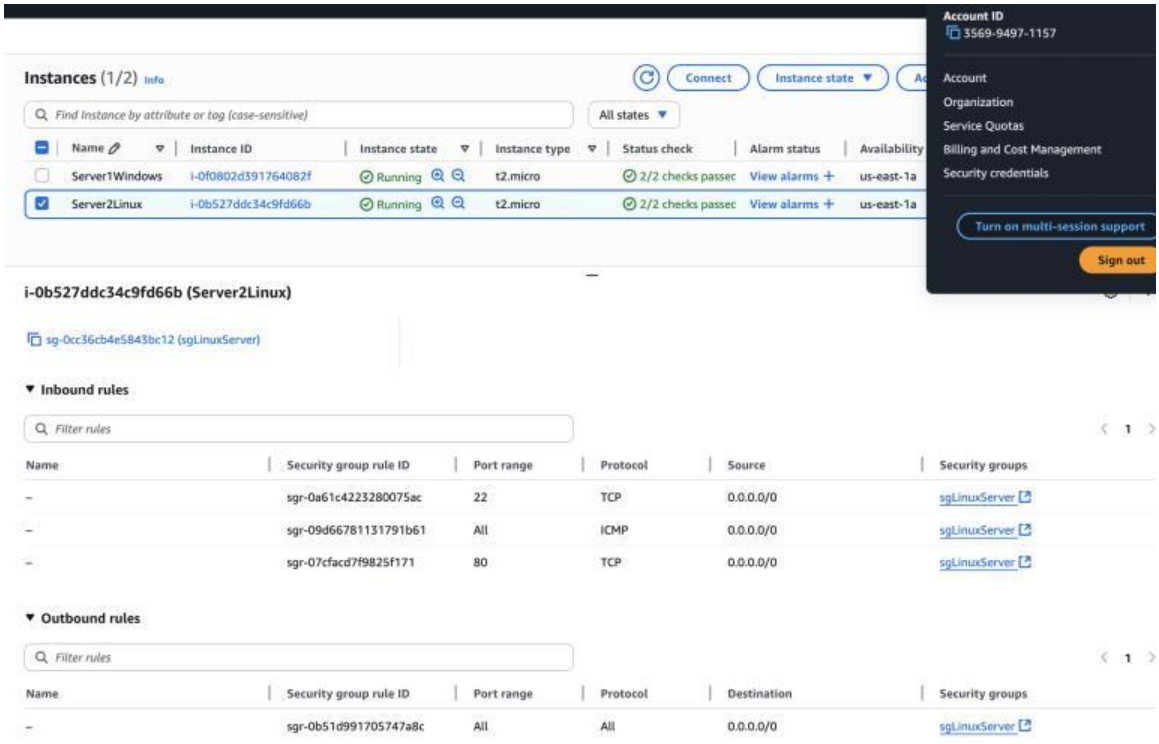
Autoría propia.

Figura 3. Instancia de Windows Server



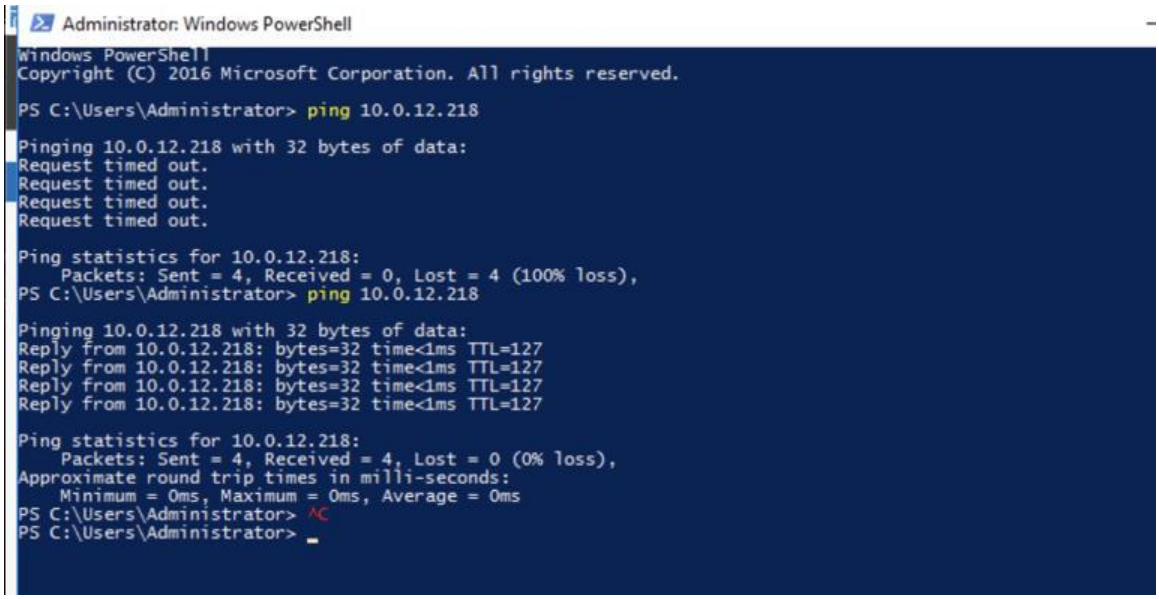
Autoría propia.

Figura 4. Instancia de Linux Server



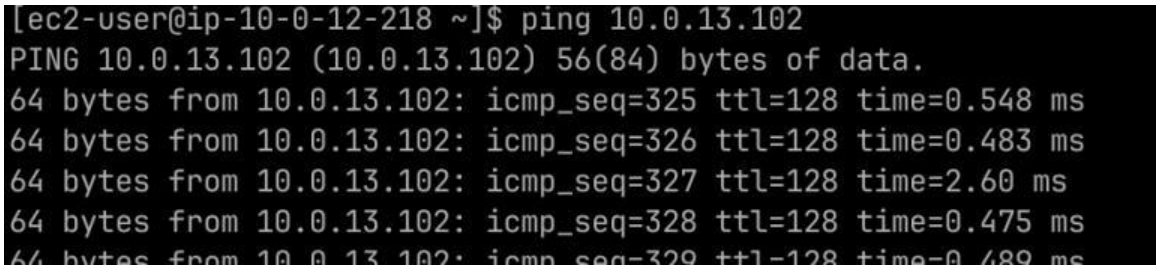
Autoría propia.

Figura 5. Conectividad desde Windows Server a Linux Server (Red privada)



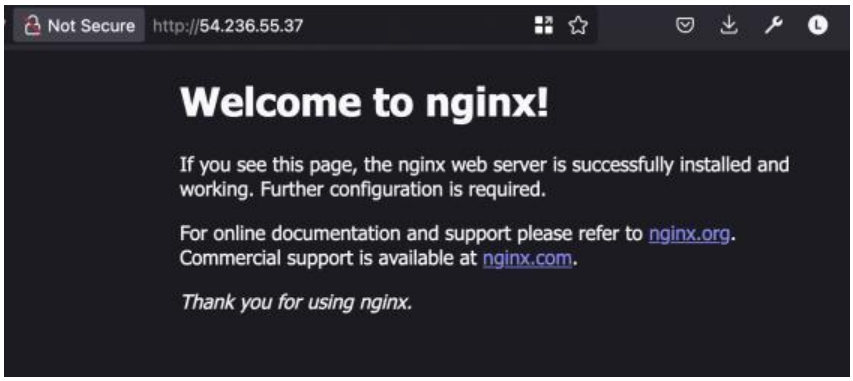
Autoría propia.

Figura 6. Conectividad desde Linux Server a Windows Server (Red privada)



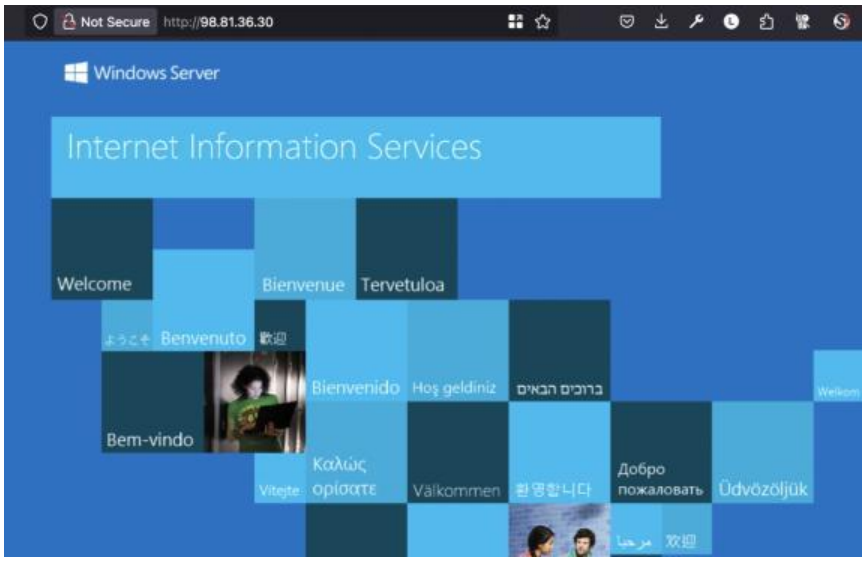
Autoría propia.

Figura 7. Acceso desde internet http a instancia Linux



Autoría propia.

Figura 8. Acceso desde internet http a instancia Windows

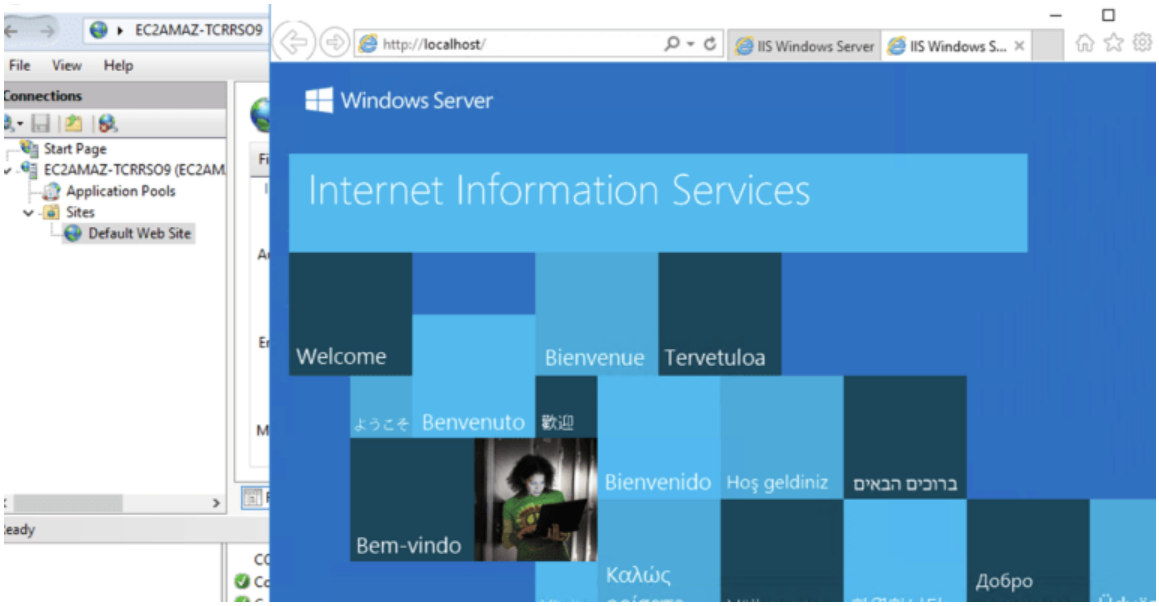


Autoría propia.

Figura 9. Configuración IIS en instancia EC2 Windows

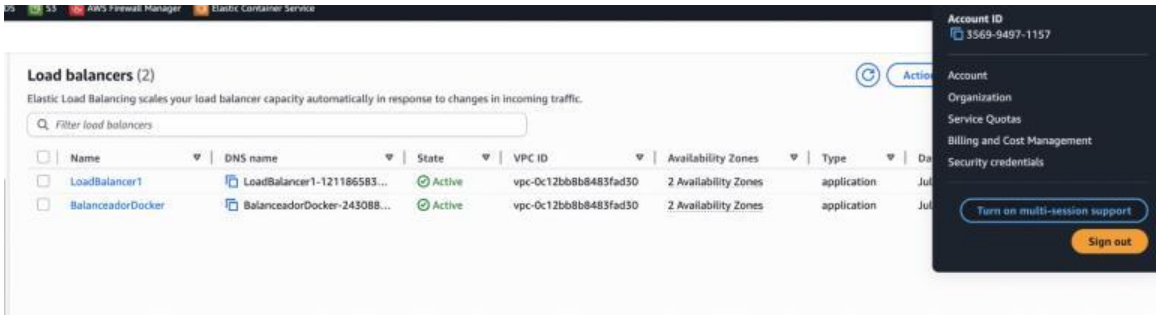
Autoría propia.

Figura 10. Validación local de IIS en instancia EC2 Windows



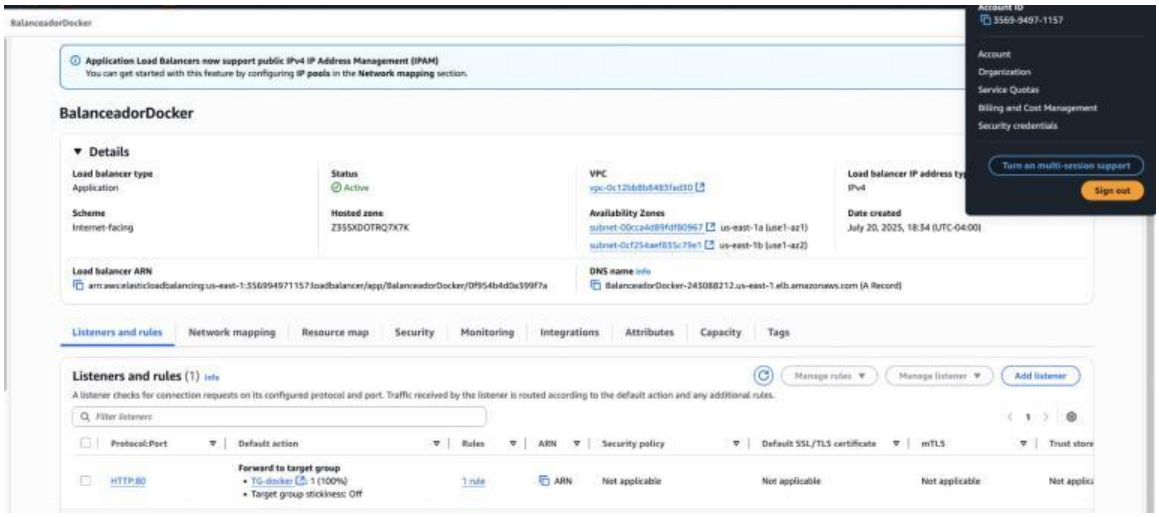
Autoría propia.

Figura 11. Load Balancers



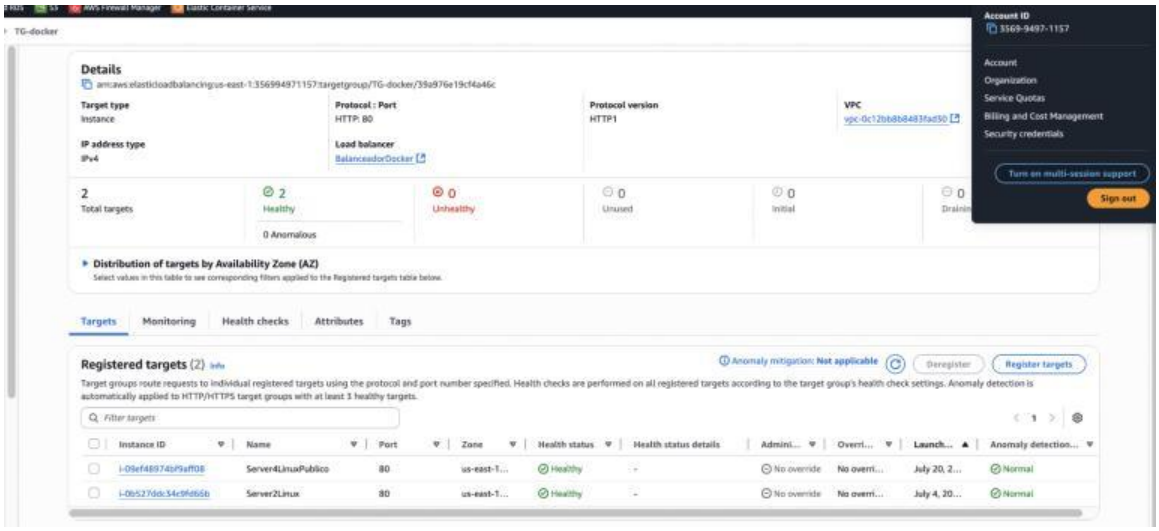
Autoría propia.

Figura 12. Load Balancer: BalanceadorDocker



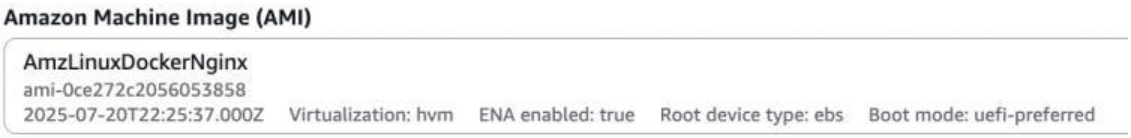
Autoría propia.

Figura 13. Target Group: TG-docker



Autoría propia.

Figura 14. AMI Base Docker

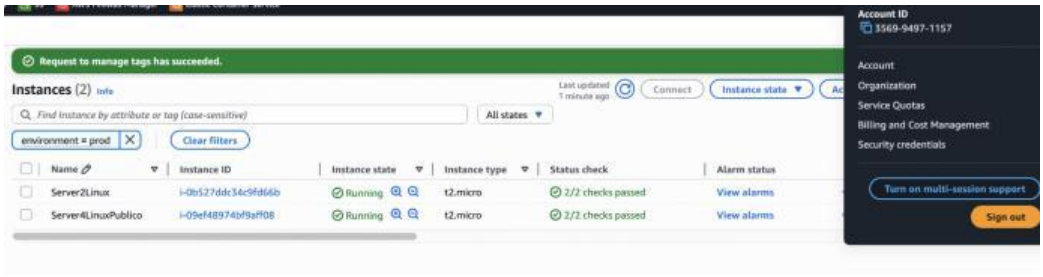


Autoría propia.

Figura 15. Docker containers

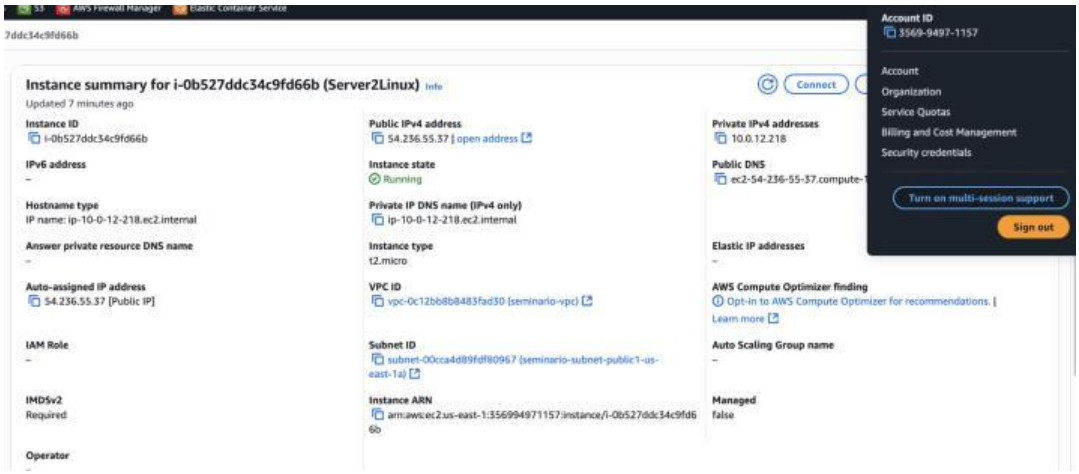
Autoría propia.

Figura 16. Instancias basadas en la AMI Base Docker.



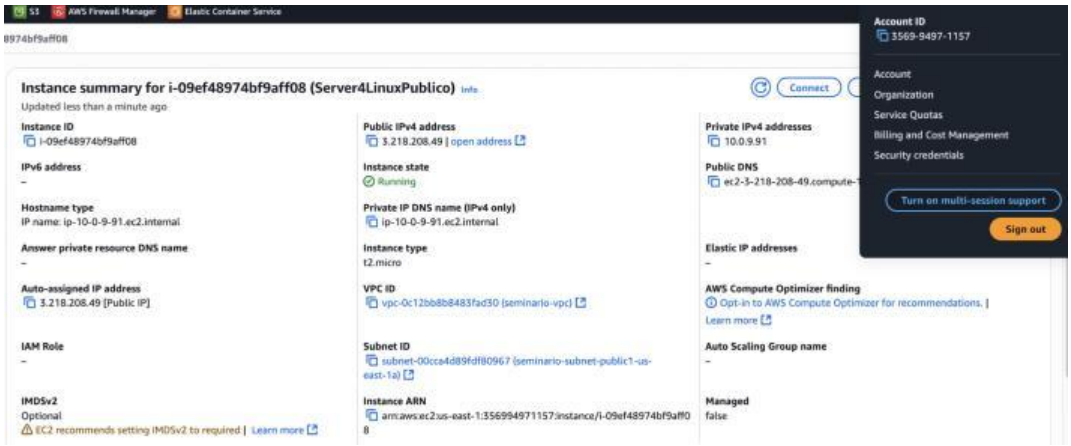
Autoría propia.

Figura 17. Primera instancia basada en la AMI Base Docker.



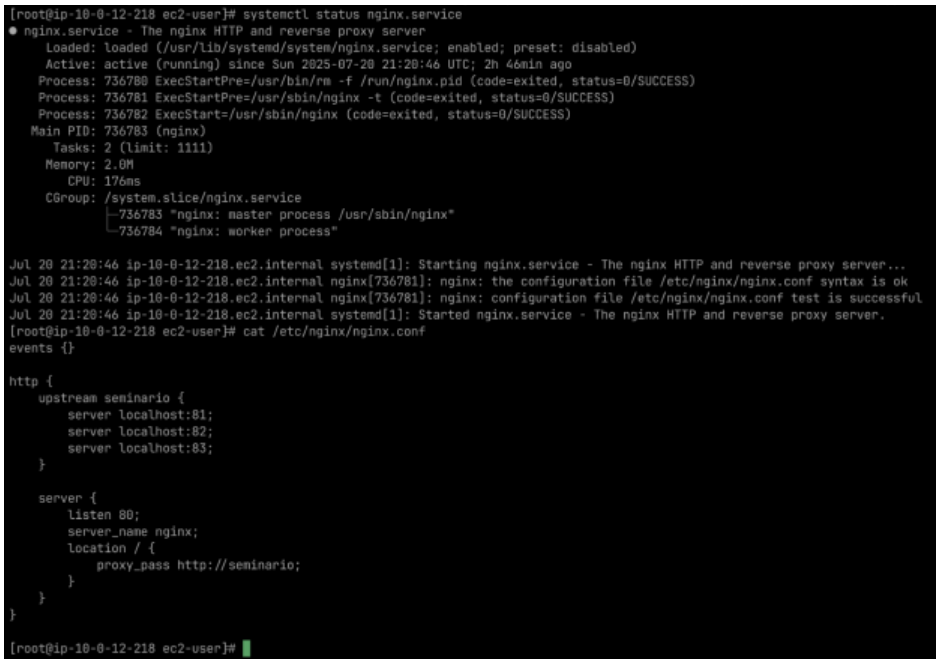
Autoría propia.

Figura 18. Segunda instancia basada en la AMI Base Docker.



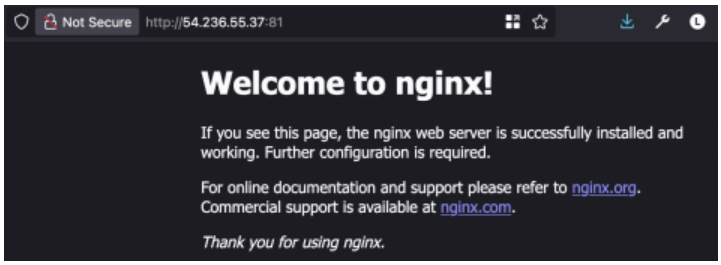
Autoría propia.

Figura 19. Proxy Reverse a Contenedores Docker



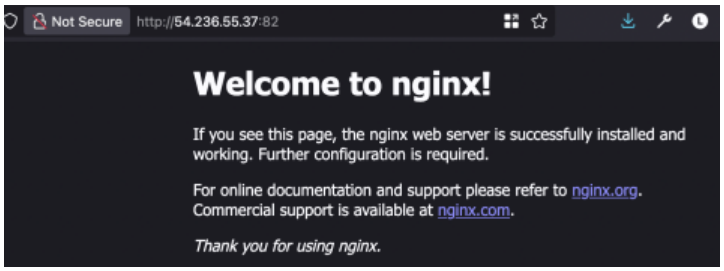
Autoría propia.

Figura 20. Conectividad Contenedor 1



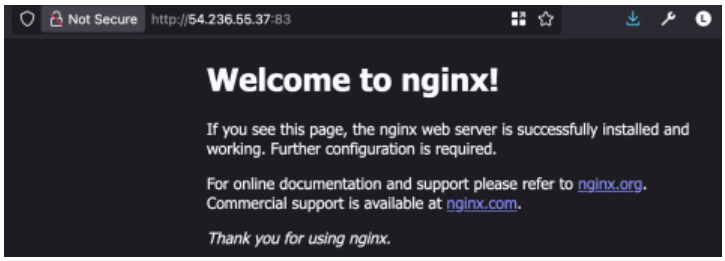
Autoría propia.

Figura 21. Conectividad Contenedor 2



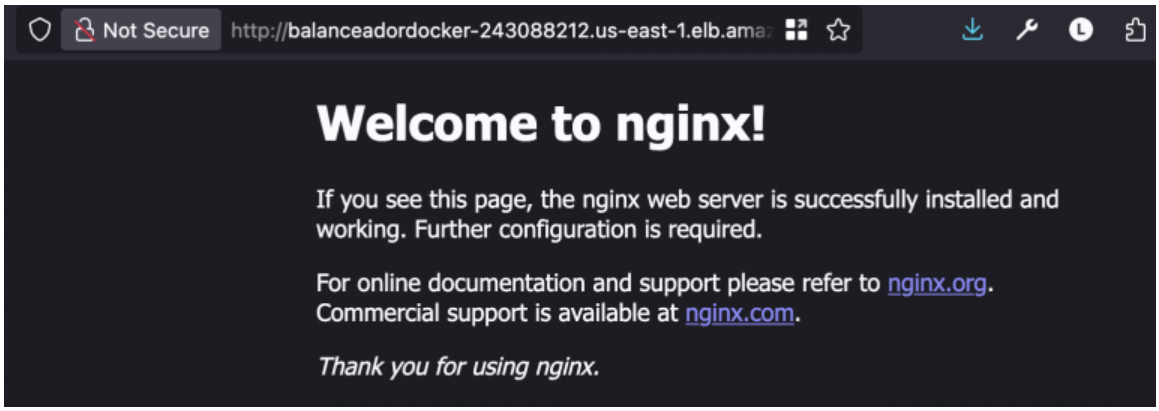
Autoría propia.

Figura 22. Conectividad Contenedor 3



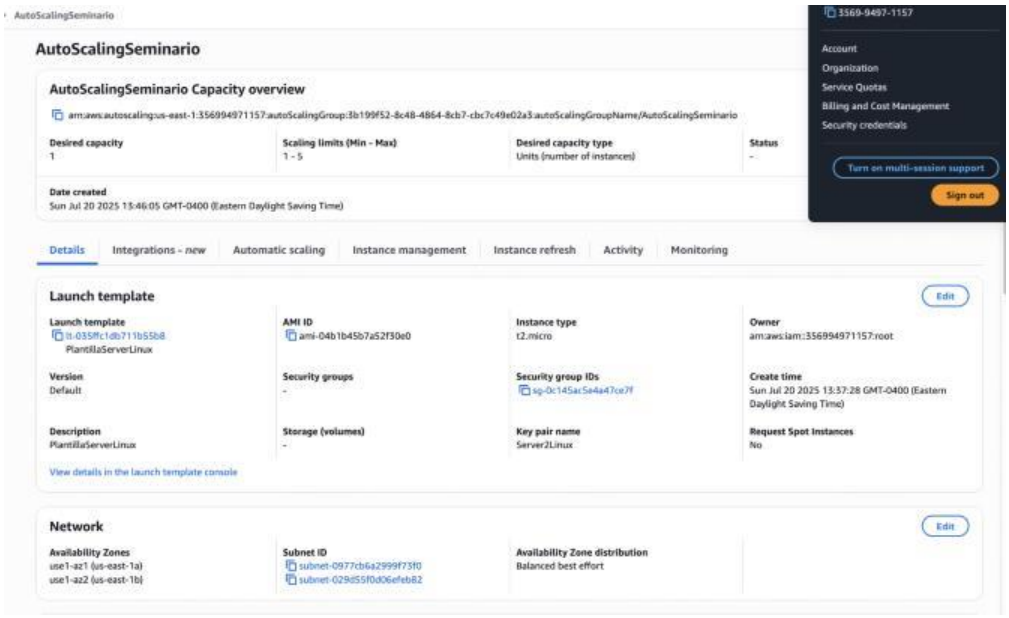
Autoría propia.

Figura 23. Conectividad DNS Load Balancer



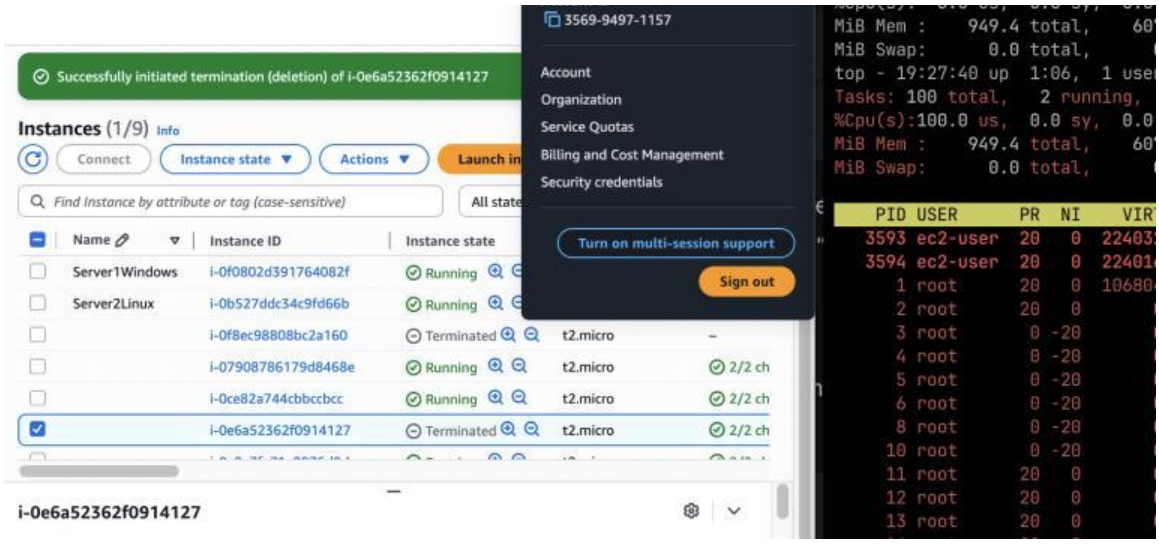
Autoría propia.

Figura 24. Configuración Auto Scaling



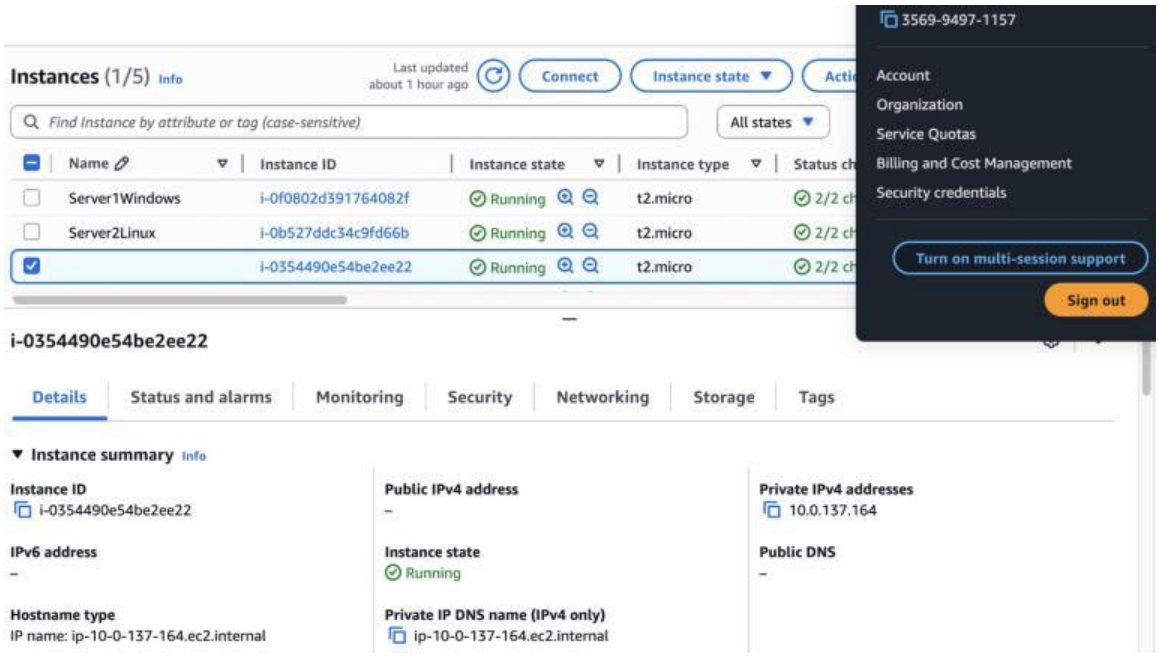
Autoría propia.

Figura 25. Validación Auto Scaling



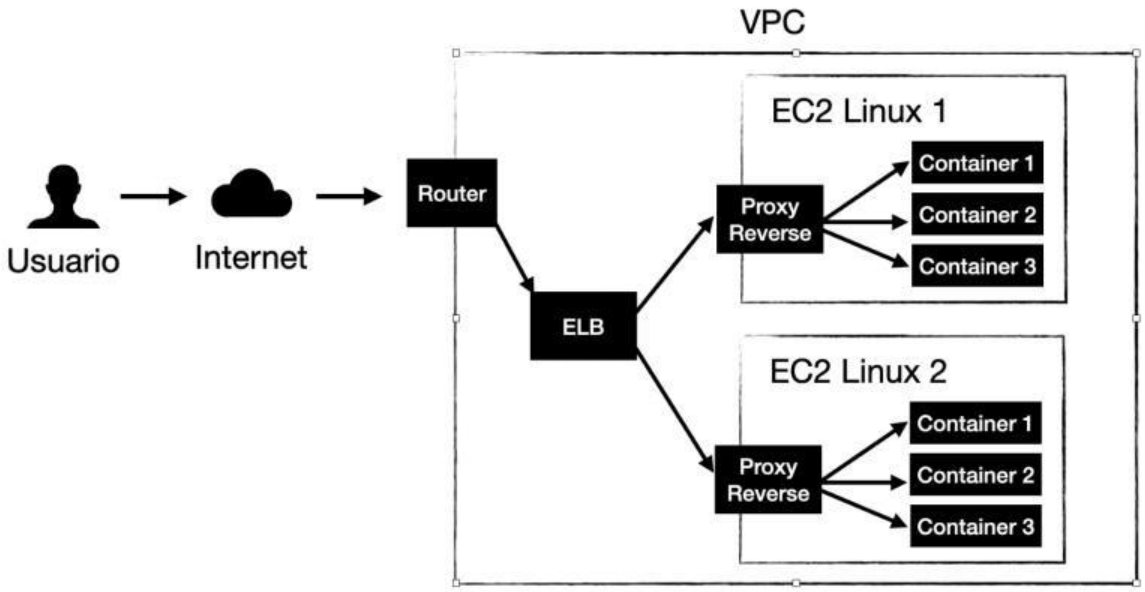
Autoría propia.

Figura 26. Validación Downscaling



Autoría propia.

Figura 27. Diagrama Seguridad, alta disponibilidad y escalabilidad de las aplicaciones.



Autoría propia.

Referencias

Amazon Web Services. (s.f.) Documentación de AWS. Recuperado el 20 de julio de 2005, de https://docs.aws.amazon.com/es_es/

Seminario AWS (2025), Seminario Opción-Grado AWS, Seguridad, Alta disponibilidad y escalabilidad de las aplicaciones en AWS. Juan Pablo Berrio López