

TRABAJO DE GRADO
Opción Seminario-Diplomado.

1

Despliegue de infraestructura escalable en la nube con AWS

Corporación Universitaria Remington.
Nombre de la facultad: Ingeniería
Nombre del programa académico: Ingeniería de Sistemas

Kevin Samir Oliveros Bustamante – Robin Eriberto Ramírez Daza.
Juan Pablo Berrio López
Seminario-Diplomado.
2025

Tabla de Contenidos

Resumen.....	3
Marco conceptual y contextual	3
Desarrollo e implementación del aprendizaje.....	5
Conclusiones	19
Referencias.....	20
Ilustración 1 Cuadro comparativo Servicios en la nube	6
Ilustración 2 Línea de tiempo - Orígenes de la virtualización	7
Ilustración 3 Consola Amazon Web Service -VPC	7
Ilustración 4 Consola Amazon Web Service -EC2.....	8
Ilustración 5 Conexión instancia AWS por medio de Mobaxterm	9
Ilustración 6 Instalación Apache por medio de Mobaxterm	10
Ilustración 7 Consola Amazon Web Service -EC2.....	10
Ilustración 8 Creación Snapshot desde la AMI.....	11
Ilustración 9 Creación de un Target Group.....	11
Ilustración 10 consola de AWS validación de ELB.....	12
Ilustración 11 Consola Amazon Web Service -EC2.....	12
Ilustración 12 Validación desde un móvil de la IP generada desde AWS	13
Ilustración 13 Creación de Launch Template	13
Ilustración 14 Creación de auto scaling groups	14
Ilustración 15 Validación de creación automática de instancias	14
Ilustración 16 Detención instancia inutilizadas	15
Ilustración 17 Creación de auto scaling groups	15
Ilustración 18 Utilización de servicios S3	16
Ilustración 19 utilización de los servicios de RDS	16
Ilustración 20 Programa heidisql	17
Ilustración 21 desinstalación del apache e instalación del Docker en Mobaxterm.....	17
Ilustración 22 pruebas de funcionamiento desde navegador	18
Ilustración 23 Creación de clúster.....	18
Ilustración 24 Validación de puertos	19
Ilustración 25 Funcionamiento desde navegador.....	19

Resumen

Durante el seminario, implementamos una infraestructura en la nube utilizando Amazon Web Services (AWS) como plataforma principal. El propósito fue adquirir experiencia práctica con los servicios que ofrece AWS y demostrar cómo pueden utilizarse para construir entornos escalables, seguros y eficientes, aplicando principios clave de seguridad como la disponibilidad, integridad y confidencialidad.

A lo largo del proyecto, trabajamos con distintos servicios de AWS: usamos Amazon EC2 para crear y administrar instancias virtuales; Amazon VPC para diseñar redes privadas; Elastic Load Balancing para distribuir el tráfico de manera equilibrada; Amazon S3 para almacenamiento; y Amazon RDS para gestionar bases de datos relacionales. Estos servicios nos permitieron construir una arquitectura sólida y funcional, capaz de soportar aplicaciones que requieren alta disponibilidad y buen rendimiento.

El desarrollo del proyecto requirió conocimientos básicos en redes, sistemas operativos (Windows y Linux), y uso de comandos en consola. Además, fue clave mantener una actitud y disposición para el aprendizaje autónomo. La documentación oficial de AWS, junto con su consola fácil de usar y los laboratorios prácticos, ofreció un entorno ideal para desarrollar habilidades técnicas, incluso para quienes no tenían experiencia previa en la nube.

Este trabajo no solo permitió aprender a manejar servicios cloud, sino que también ayudó a entender la importancia de la automatización, la escalabilidad y la seguridad en los entornos actuales de TI. Asimismo, mostró cómo la nube puede ser una aliada estratégica para optimizar recursos, reducir costos operativos y acelerar la implementación de soluciones tecnológicas.

Palabras clave

- Amazon Web Services (AWS)
- Disponibilidad
- Virtualización
- Automatización
- Escalabilidad

Marco conceptual y contextual

El presente informe resume los resultados del seminario realizado sobre la implementación de infraestructura en la nube utilizando Amazon Web Services (AWS). El propósito de este seminario de formación fue brindar a los estudiantes los conocimientos teóricos y prácticos necesarios para entender, diseñar e implementar arquitecturas modernas de TI en la nube. Durante el proceso, abordamos temas clave como redes, almacenamiento, procesamiento, bases de datos y, por supuesto, seguridad.

La computación en la nube se ha consolidado como una de las soluciones más eficaces para enfrentar los desafíos de escalabilidad, disponibilidad y gestión de recursos tecnológicos en las organizaciones. Gracias a este modelo, las empresas pueden acceder a una infraestructura robusta sin necesidad de realizar grandes inversiones hardware físico. Además, tiene como ventajas la automatización, la alta disponibilidad y la continuidad operativa.

Como parte del seminario, desarrollamos un ejercicio práctico en el que diseñamos e implementamos una infraestructura básica en AWS, simulando un entorno de producción real.

A continuación, se presentan los principales conceptos y servicios utilizados durante el desarrollo del proyecto:

Amazon Web Services (AWS)

Es la plataforma de servicios en la nube de Amazon, que ofrece soluciones bajo demanda para cómputo, redes, almacenamiento, bases de datos, seguridad, análisis y mucho más. Gracias a su flexibilidad y eficiencia, muchas empresas la eligen para montar infraestructuras seguras y de alto rendimiento con un costo operativo optimizado (Amazon Web Services, n.d.).

Computación en la Nube (Cloud Computing)

Es un modelo que permite acceder a recursos tecnológicos a través de Internet, sin necesidad de tener los equipos físicamente. Funciona bajo el principio de “pago por uso” y se clasifica en modelos como IaaS (Infraestructura como Servicio), PaaS (Plataforma como Servicio) y SaaS (Software como Servicio). AWS se destaca principalmente como proveedor de IaaS (Amazon Web Services, n.d.).

Amazon EC2 (Elastic Compute Cloud)

Este servicio permite lanzar máquinas virtuales (instancias) con diferentes configuraciones de sistema operativo, capacidad de almacenamiento, CPU y red, según lo que necesite el proyecto. Es ideal para aplicaciones que requieren escalabilidad y flexibilidad (Amazon Web Services, n.d.).

VPC (Virtual Private Cloud)

Con este servicio se pueden crear redes privadas dentro de AWS. VPC permite controlar aspectos como el direccionamiento IP, las subredes, las rutas de red y las reglas de acceso, creando un entorno aislado y seguro (Amazon Web Services, n.d.).

Elastic Load Balancer (ELB)

ELB distribuye automáticamente el tráfico de red entre varias instancias de EC2, ayudando a que las aplicaciones sigan funcionando incluso si una de las instancias falla. Hay varios tipos de balanceadores según el tipo de tráfico o necesidad (Amazon Web Services, n.d.).

Amazon RDS (Relational Database Service)

Se trata de un servicio administrado para bases de datos relacionales. Con RDS no hace falta preocuparse por tareas como respaldos o actualizaciones, ya que AWS se encarga de automatizarlas, permitiendo que el equipo se enfoque en el desarrollo y el uso de los datos (Amazon Web Services, n.d.).

Amazon S3 (Simple Storage Service)

Es un servicio de almacenamiento de objetos que permite guardar archivos de forma segura, duradera y accesible. Se usa mucho para copias de seguridad, almacenamiento de datos estáticos, registros del sistema, archivos multimedia, entre otros (Amazon Web Services, n.d.).

Desarrollo e implementación del aprendizaje

A continuación, se presenta un cuadro comparativo entre tres de las principales plataformas de servicios en la nube: Amazon Web Services (AWS), Microsoft Azure y Google Cloud Platform (GCP). En él se destacan algunas de las características más relevantes de cada proveedor en diferentes categorías de servicio, como cómputo,

almacenamiento, bases de datos, redes, seguridad, análisis de datos, contenedores e inteligencia artificial. Esta comparación permite visualizar de manera resumida las capacidades y fortalezas de cada plataforma, facilitando su análisis y selección según las necesidades del entorno tecnológico.

Servicio	AWS (Amazon Web Services)	Azure (Microsoft)	GCP (Google Cloud Platform)
Cómputo	EC2, Lambda, Lightsail	Virtual Machines, Functions	Compute Engine, Cloud Functions
Almacenamiento	S3, EBS, Glacier	Blob Storage, Disk Storage	Cloud Storage, Persistent Disk
Base de datos	RDS, DynamoDB, Aurora	SQL Database, Cosmos DB	Cloud SQL, Firestore, Bigtable
Redes	VPC, Route 53, CloudFront	Virtual Network, Traffic Manager	VPC, Cloud Load Balancing
Contenedores	ECS, EKS, Fargate	AKS (Azure Kubernetes Service)	GKE (Google Kubernetes Engine)
Inteligencia Artificial y Machine Learning	SageMaker, Rekognition, Comprehend	Azure ML, Cognitive Services	Vertex AI, AutoML, Vision AI
Seguridad	IAM, GuardDuty, Shield, WAF	Azure AD, Security Center, Defender	IAM, Cloud Armor, Security Command Center
Analítica	Redshift, Athena, QuickSight	Azure Synapse Analytics, Power BI	BigQuery, Data Studio, Looker

Fuente: Elaboración propia con base en la documentación oficial de Amazon Web Services (n.d.), Microsoft Azure (n.d.) y Google Cloud Platform (n.d.).

La siguiente línea de tiempo presenta los hitos más relevantes en la evolución de la computación en la nube, desde los primeros conceptos de virtualización hasta la consolidación de modelos híbridos, inteligencia artificial y Edge Computing. Esta recopilación histórica tiene como objetivo ofrecer una visión general del desarrollo tecnológico que ha permitido la transformación de la infraestructura de TI tradicional hacia soluciones más flexibles, escalables y orientadas a servicios.

La información presentada se basa en fuentes oficiales como la documentación de IBM, AWS, Google Cloud, Microsoft Azure, Salesforce, así como recursos de referencia como Timetoast y el sitio web del CERN.



Ilustración 3 Consola Amazon Web Service -VPC



A continuación, se describe el proceso para crear una arquitectura básica en AWS. El primer paso consiste en acceder a la consola de administración de AWS y crear una VPC (red privada en la nube).



Captura de pantalla propia tomada desde la consola de Amazon Web Services (AWS), 2025.

El VPC nos permite definir un rango de direcciones IP, crear subredes, añadir puertas de enlace, definir rutas y aplicar reglas de seguridad, entre otras funcionalidades. En este caso, se creó una VPC básica que generó cuatro subredes: dos privadas y dos públicas, distribuidas en diferentes zonas de disponibilidad.

Una vez configurada la VPC, se accedió al servicio EC2 (Elastic Compute Cloud), el cual permite crear servidores virtuales dentro de nuestra infraestructura en la nube. Aquí entra un concepto clave, las AMI (Amazon Machine Images), esto es básicamente una imagen de un sistema operativo provisto por Amazon Web Service. Estas imágenes permiten escoger múltiples opciones de máquinas que nos ofrecen diferentes opciones en cuanto a memoria RAM, disco duro, red, entre otros recursos.

Para este ejercicio, se seleccionó una imagen de Linux.

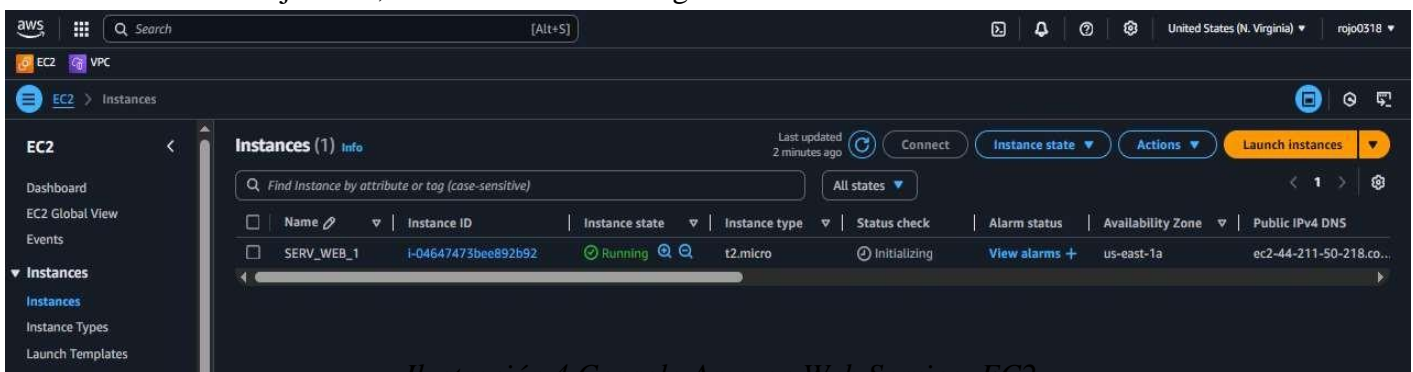


Ilustración 4 Consola Amazon Web Service -EC2

Captura de pantalla propia tomada desde la consola de Amazon Web Services (AWS), 2025.

Una vez seleccionada la AMI, se procede a inicializar y se establece conexión mediante SSH a la consola utilizando un cliente de conexión de preferencia.

Para lograr esta conexión, es necesario contar con ciertos datos, entre ellos el nombre y la clave privada. Estos elementos son esenciales para autenticar el acceso de forma segura.

Después de completar estos pasos y establecer la conexión correctamente, se podrá visualizar la interfaz de la consola, lo cual indica que se ha accedido exitosamente.

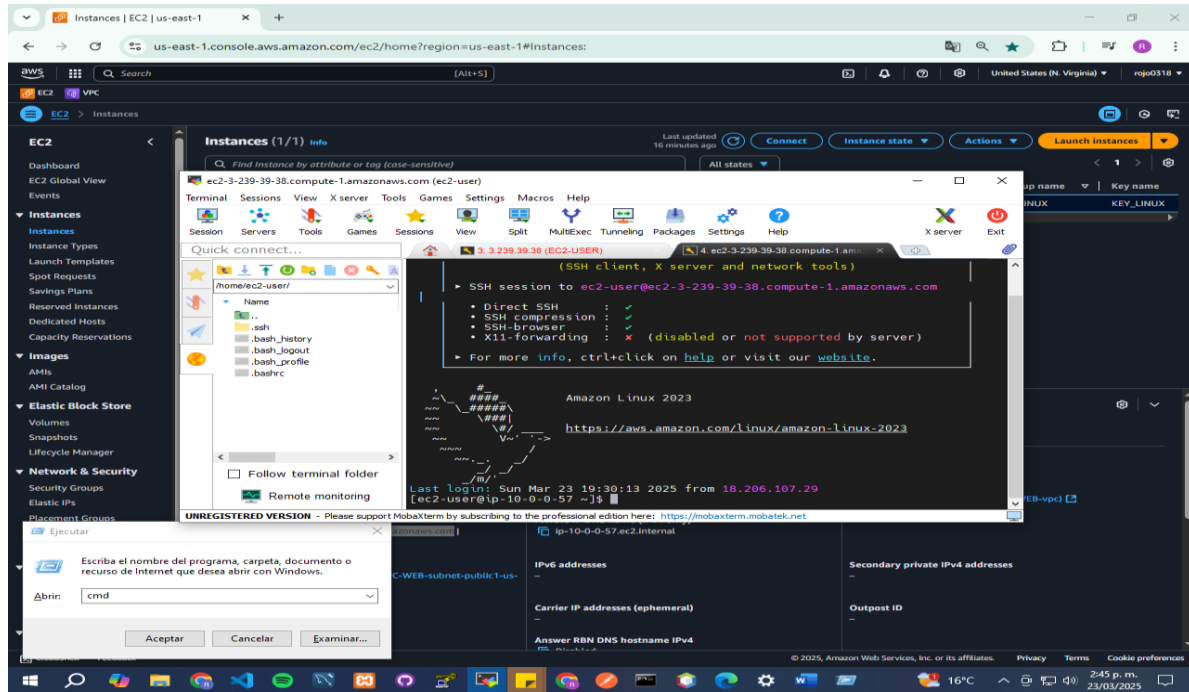


Ilustración 5 Conexión instancia AWS por medio de MobaXterm

Captura de pantalla propia tomada desde la consola de Amazon Web Services (AWS), 2025.

A continuación, comenzamos la instalación de los recursos necesarios. En este caso, se instaló un apache con el comando `dnf install httpd`. Una vez instalado, se procedió a iniciar el servicio con el comando `systemctl start httpd`. Finalmente, se validó el estado del servicio con el comando `systemctl status httpd`

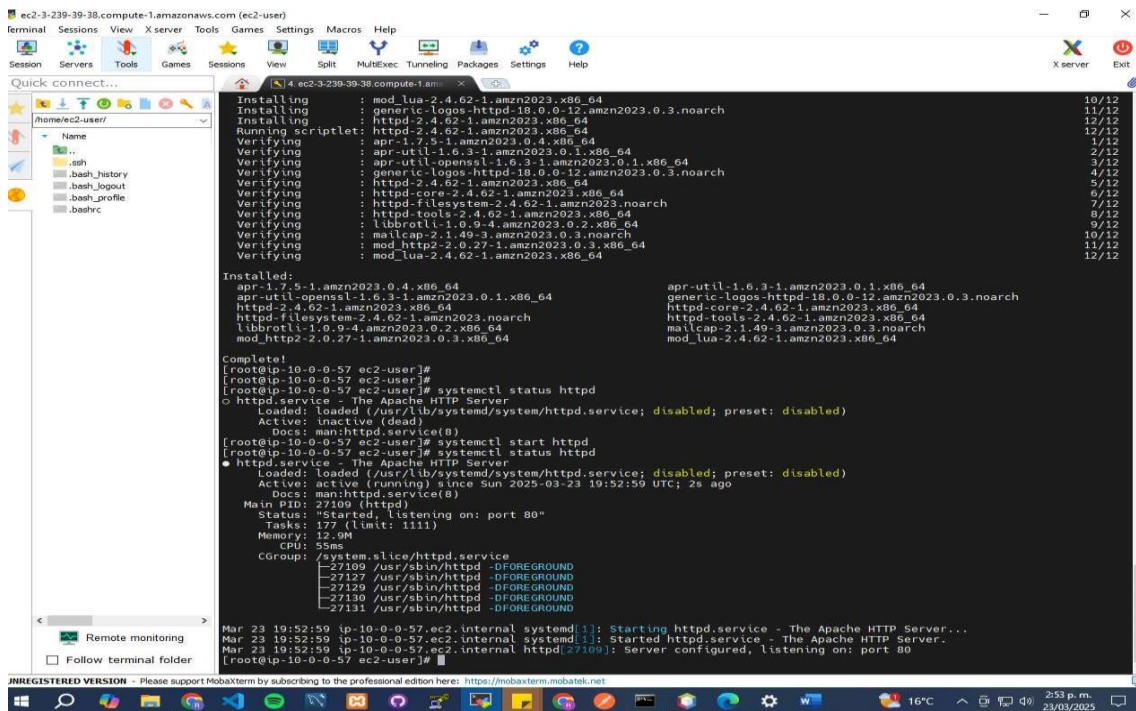


Ilustración 6 Instalación Apache por medio de MobaXterm

Captura de pantalla propia tomada desde la consola de Amazon Web Services (AWS), 2025

Regresamos al servicio EC2 y agregamos una regla en el grupo de seguridad, permitiendo acceso al puerto 80.

Agregamos el la regla en el grupo de seguridad para acceso por el puerto 80

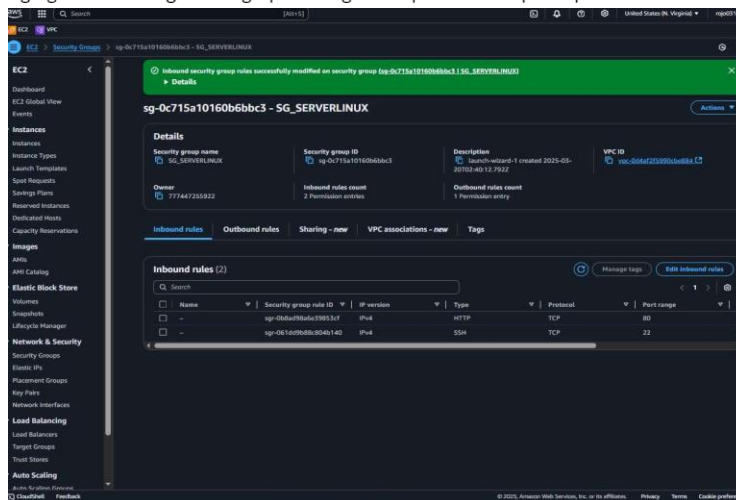
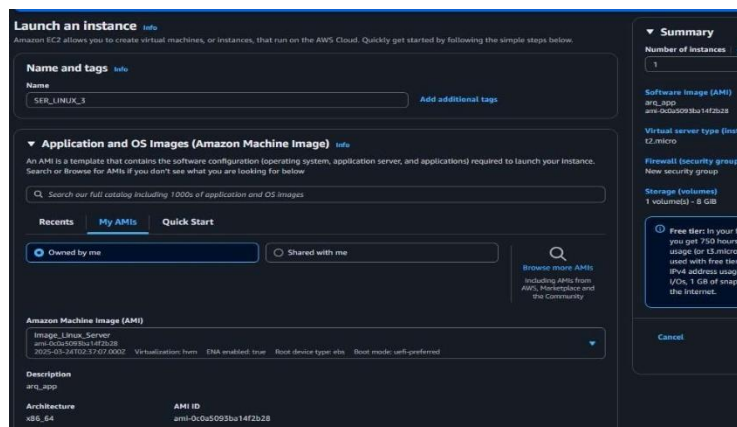


Ilustración 7 Consola Amazon Web Service -EC2

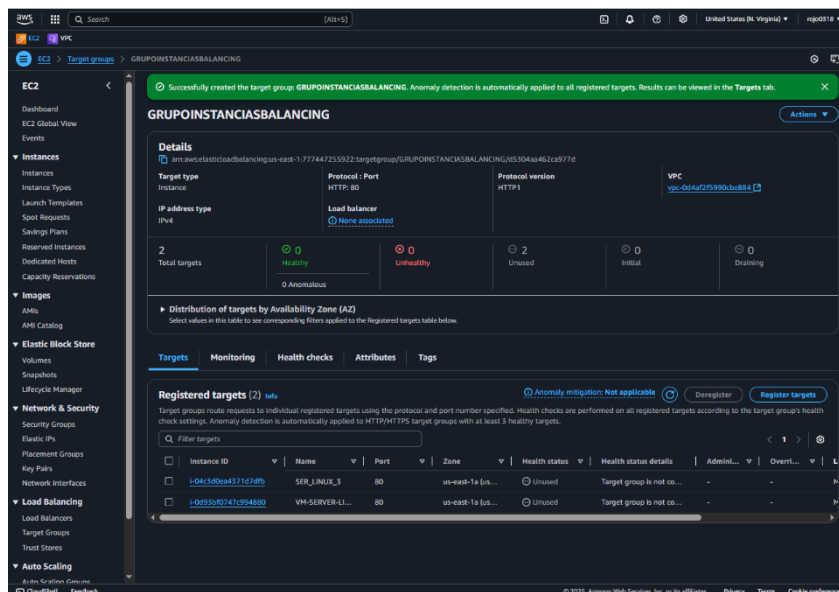
Captura de pantalla propia tomada desde la consola de Amazon Web Services (AWS), 2025.

Posteriormente, creamos un snapshot de nuestra AMI previamente configurada. Un snapshot es una copia de seguridad exacta de nuestro volumen de almacenamiento EBS (Elastic Block Store), que nos permite conservar el estado actual del sistema y, en caso necesario, restaurar o replicar la instancia con la misma configuración.



*Ilustración 8 Creación Snapshot desde la AMI
Captura de pantalla propia tomada desde la consola de Amazon Web Services (AWS), 2025.*

El siguiente paso es crear nuestro target group (grupo de destinos).



*Ilustración 9 Creación de un Target Group
Captura de pantalla propia tomada desde la consola de Amazon Web Services (AWS), 2025.*

La creación del target group es esencial porque, al implementar nuestro Load Balancer (ELB) este va a recibir el tráfico que llega del (ELB) y redirige el tráfico, garantizando que las solicitudes se distribuyan de manera equilibrada y eficiente, según la disponibilidad.

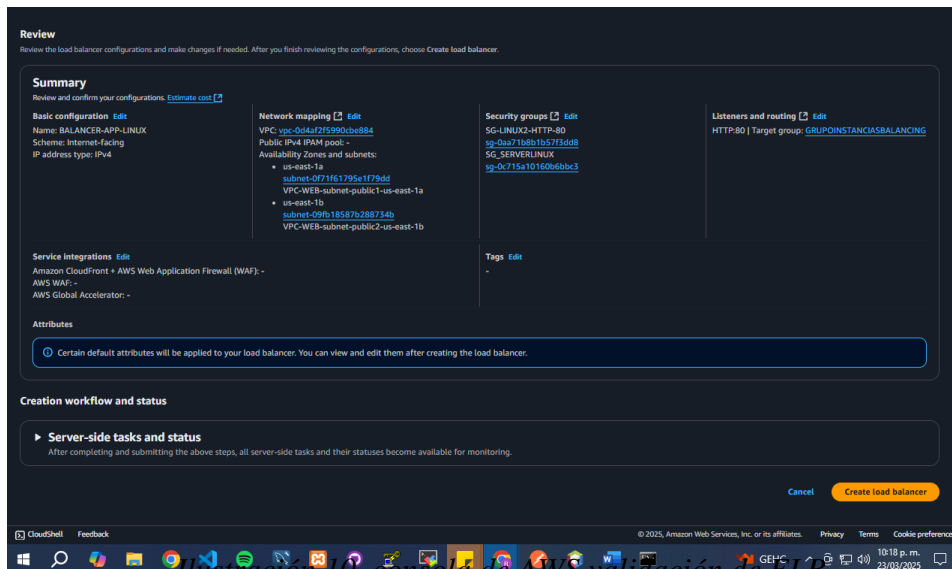


Ilustración 10 consola de AWS validación de ELB
Captura de pantalla propia tomada desde la consola de Amazon Web Services (AWS), 2025.

Una vez creado el Load Balancer, podemos verificar que los servicios se encuentran activos.

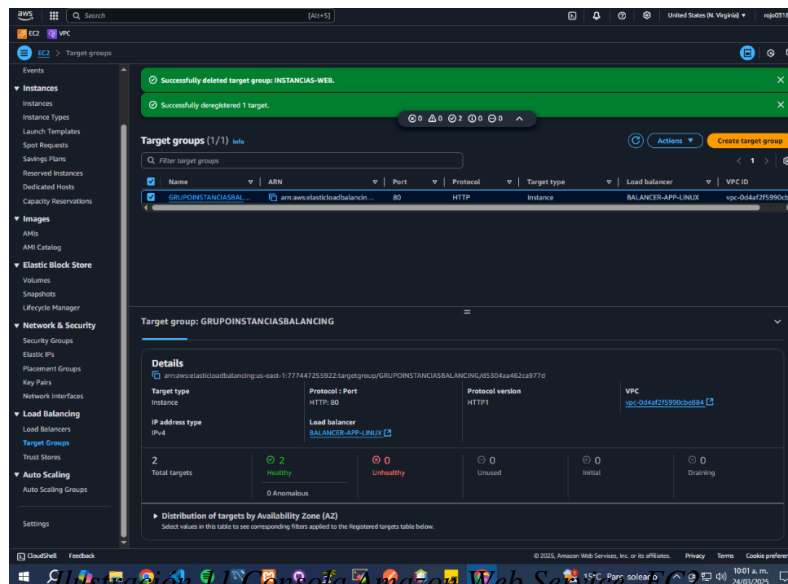
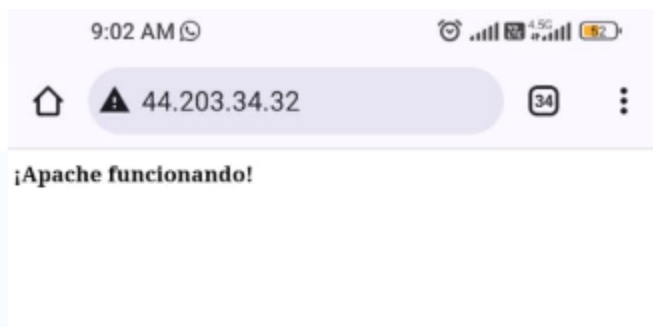


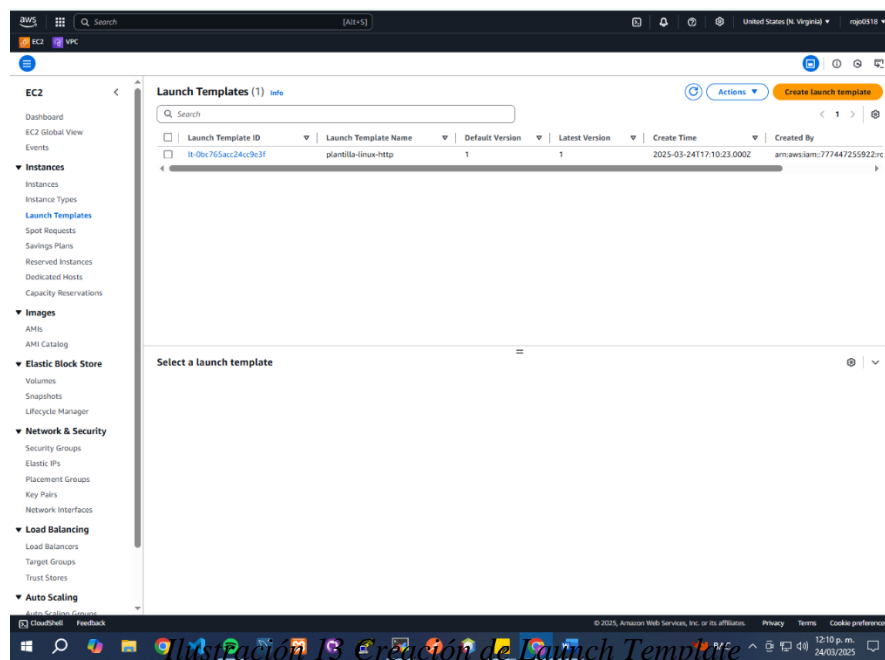
Ilustración 11 Consola Amazon Web Service EC2
Captura de pantalla propia tomada desde la consola de Amazon Web Services (AWS), 2025.

Cuando accedamos al servicio desde una red pública, podemos confirmar como nuestro servicio está funcionando correctamente.



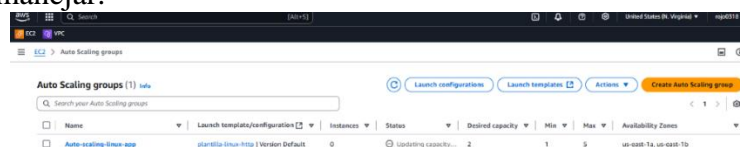
*Ilustración 12 Validación desde un móvil de la IP generada desde AWS
Captura de pantalla propia tomada desde la consola de Amazon Web Services (AWS),
2025.*

Hasta este punto, se realizó una implementación manual de como implementar los servicios de AWS. Sin embargo, se busca automatizar este proceso. Para lograrlo, se empleó un Launch Template (plantilla de lanzamiento), que nos permite predefinir las condiciones de activación de las instancias.



*Ilustración 13 Creación de Launch Template
Captura de pantalla propia tomada desde la consola de Amazon Web Services (AWS),
2025.*

Esta plantilla de lanzamiento busca que cuando se utilice el servicio de Auto Scaling Group (ASG) en conjunto al Load Balancer se defina como se lanzaran las nuevas instalaciones y el tráfico a manejar.



*Ilustración 14 Creación de auto scaling groups
Captura de pantalla propia tomada desde la consola de Amazon Web Services (AWS),
2025.*

A continuación, se observa cómo se inicializaron las máquinas de forma automática

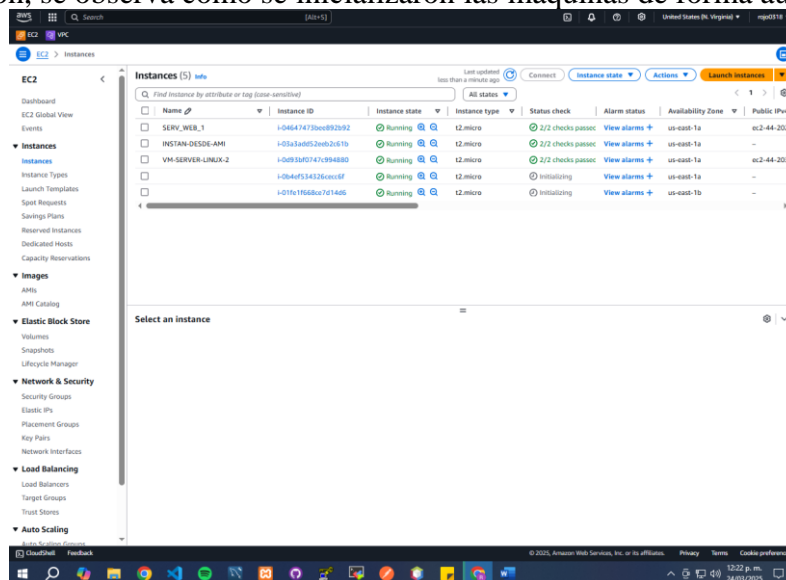
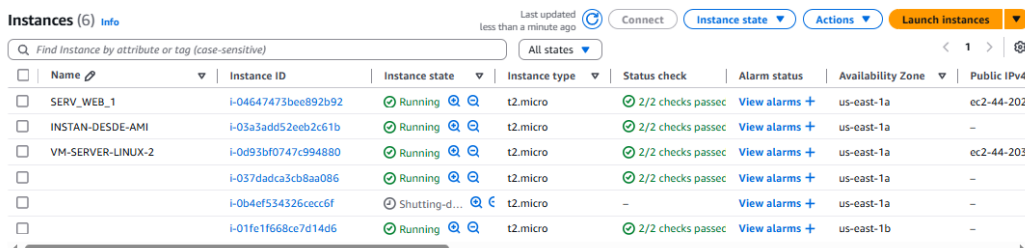


Ilustración 15 Validación de creación automática de instancias

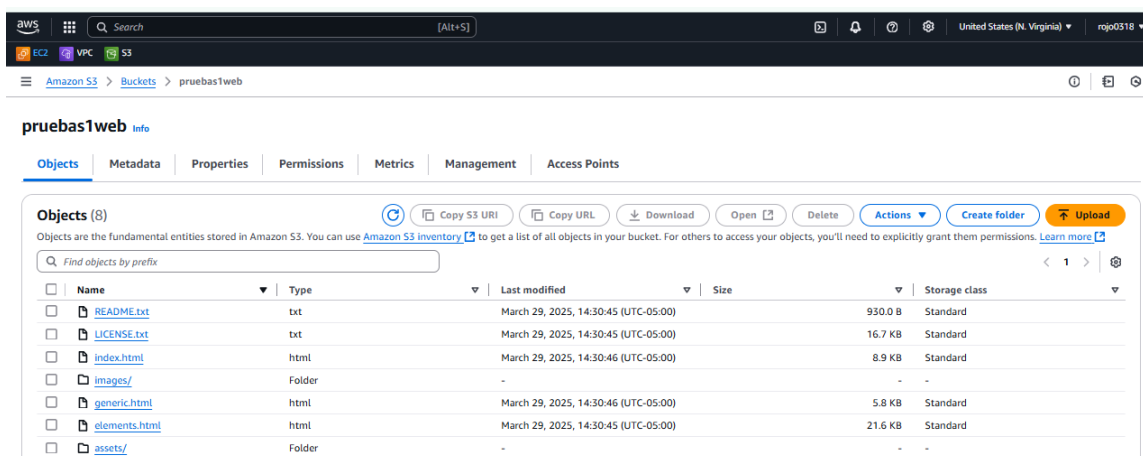
*Captura de pantalla propia tomada desde la consola de Amazon Web Services (AWS),
2025.*

La ventaja de automatizar este tipo de procesos, es que cuando el identifica que no necesita el recurso de la máquina, esta se elimina de forma automática. Esto permite una gestión eficiente de los recursos.



*Ilustración 16 Detención instancia inutilizadas
Captura de pantalla propia tomada desde la consola de Amazon Web Services (AWS),
2025.*

Ahora vamos hablar del servicio de S3 (Simple Storage Service), que básicamente es un sistema de almacenamiento de objetos en la nube. A diferencia de EBS, en donde se paga por capacidad asignada, por ejemplo, 30 GB independientemente de que solo estemos utilizando 1 GB, S3 permite guardar objetos y permite pagar únicamente por el espacio que efectivamente se utiliza, es decir, que si tengo si tengo solamente un video, solo pago por este y no por todo el espacio



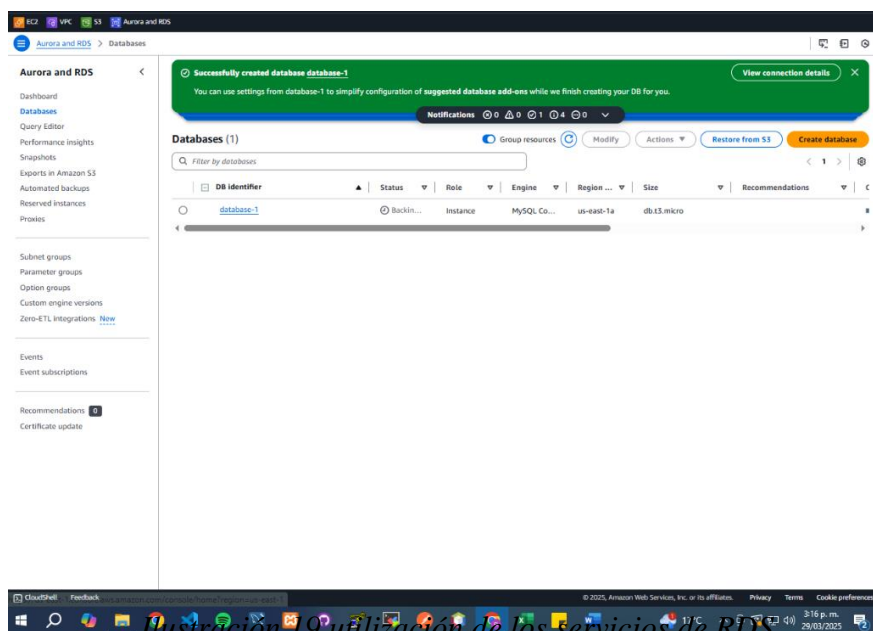
*Ilustración 17 Creación de auto scaling groups
Captura de pantalla propia tomada desde la consola de Amazon Web Services (AWS),
2025.*

Aclarado esto, se procedió a crear un Bucket, esto es un contenedor donde se guarda cualquier tipo de archivo, estos archivos se denominan objetos. Adicionalmente, este permite que cuando cargamos nuestra información, esta se pueda hacer pública o privada según la necesidad, para este caso puntualmente, se creó pública



*Ilustración 18 Utilización de servicios S3
Captura de pantalla propia tomada desde la consola de Amazon Web Services (AWS),
2025.*

A continuación, se implementó el servicio de RDS (Relational Database Service), este servicio nos permite crear, gestionar y escalar nuestras bases de datos relacionales, pero de forma automática. procesemos a realizar la creación de nuestro servicio



*Ilustración 19 utilización de los servicios de RDS
Captura de pantalla propia tomada desde la consola de Amazon Web Services (AWS),
2025.*

Posterior a ello, se realizó la instalación de cliente de administración de bases de datos, en este caso utilizamos heidisql.

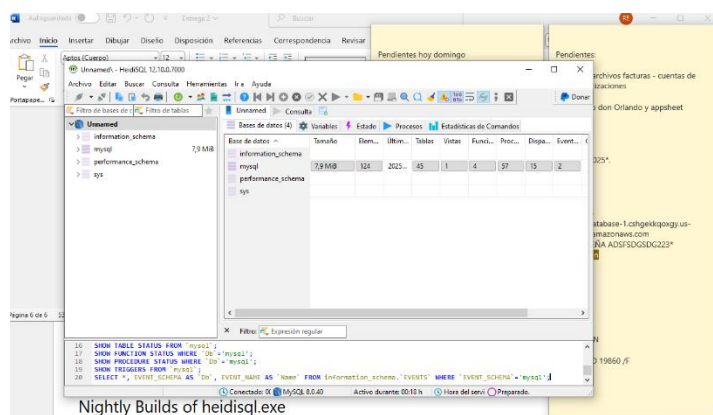


Ilustración 20 Programa heidisql Captura de pantalla propia tomada desde la consola de Amazon Web Services (AWS), 2025.

Comenzamos con la desinstalación del apache que montamos con anterioridad en nuestro servidor de Linux, y así comenzamos con la instalación del Docker. Los Dockers son contenedores, para que estos funcionen de forma correcta, debemos agregar en nuestro firewall los puertos en los grupos de seguridad, para este caso, montamos dos: el puerto 8080 y el puerto 8081.

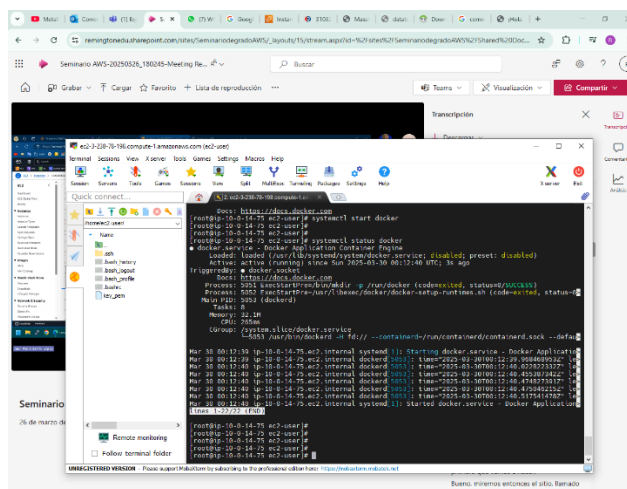
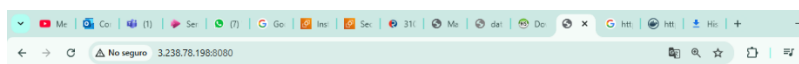
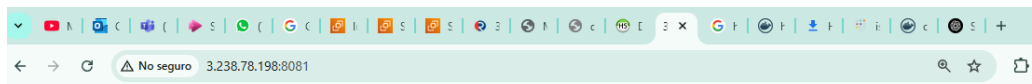


Ilustración 21 desinstalación del apache e instalación del Docker en Mobaxterm Captura de pantalla propia tomada desde la consola de Amazon Web Services (AWS), 2025.

En este paso, se realizaron pruebas de funcionamiento desde navegador. En cada contenedor, se replicaron unas páginas, el objetivo de esto, era verificar el funcionamiento del balanceador de carga, y que se encontrará la disponibilidad del que se encontrará activo.



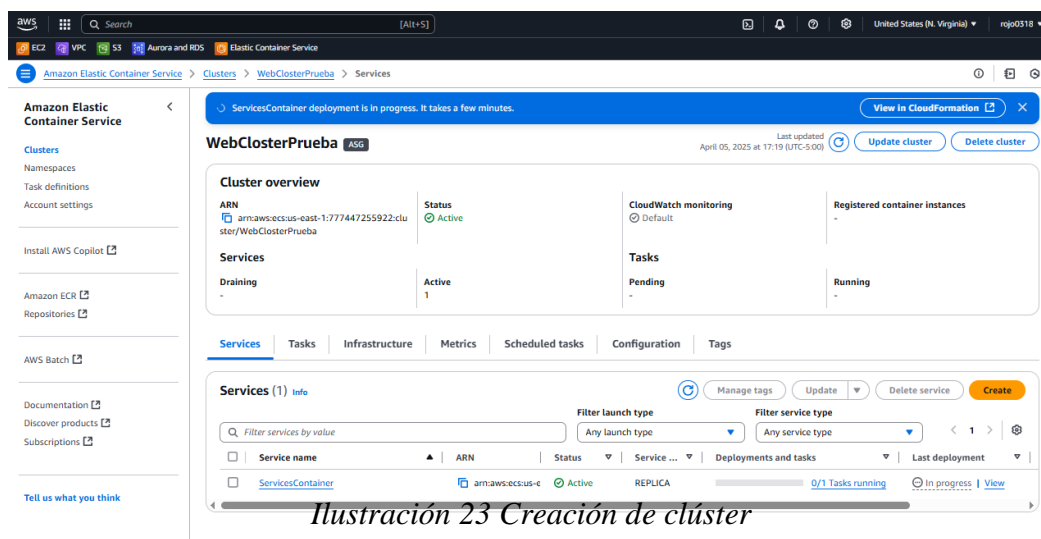
It works!



HOLA SOY ROBIN :)

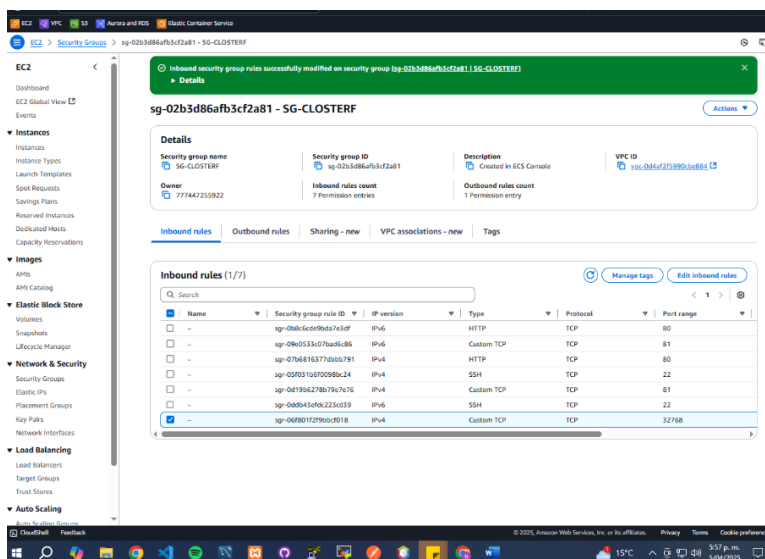
*Ilustración 22 pruebas de funcionamiento desde navegador
Captura de pantalla propia tomada desde la consola de Amazon Web Services (AWS),
2025.*

A continuación, se utilizó el servicio de ECS (Elastic Container Service), lo que nos permitió automatizar nuestro sistema de contenedores respecto a una configuración inicial. Se creó un clúster con el fin de unificar nuestros servicios para trabajarlos como una sola máquina.



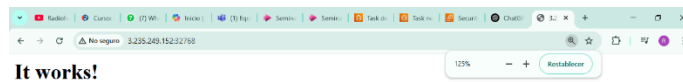
*Ilustración 23 Creación de clúster
Captura de pantalla propia tomada desde la consola de Amazon Web Services (AWS),
2025.*

Se validó que se haya creado el contenedor de forma correcta y se verificó que AWS asignara el puerto específico y que estuviera activo.



*Ilustración 24 Validación de puertos
Captura de pantalla propia tomada desde la consola de Amazon Web Services (AWS),
2025.*

Al validar la conexión, se verifica el funcionamiento adecuado del sistema.



*Ilustración 25 Funcionamiento desde navegador
Captura de pantalla propia tomada desde la consola de Amazon Web Services (AWS),
2025.*

Conclusiones

Lo que hemos aprendido en este seminario de Amazon Web Services (AWS) es que nos ofrece herramientas para almacenar, procesar, proteger y escalar aplicaciones en la nube. Cada servicio cumple una función específica, pero todos pueden integrarse para crear soluciones completas, seguras y eficientes.

Referencias

1. Amazon Web Services. (n.d.). *AWS Documentation*. <https://docs.aws.amazon.com/>
2. Microsoft. (n.d.). *Microsoft Azure Documentation*. <https://learn.microsoft.com/en-us/azure/>
3. Google Cloud. (n.d.). *Google Cloud Documentation*. <https://cloud.google.com/docs>
4. Amazon Web Services. (s.f.). *¿Qué es AWS?* <https://aws.amazon.com/es/executive-insights/cloud-strategy/what-is-aws/>
5. CERN. (s.f.). *The birth of the Web*. <https://home.cern/science/computing/birth-web>
6. Google Cloud. (s.f.). *About Google Cloud*. <https://cloud.google.com/about>
7. IBM. (s.f.). *IBM Archives: Timeline of IBM history*. <https://www.ibm.com/ibm/history/>
8. Microsoft Azure. (s.f.). *What is Azure?* <https://azure.microsoft.com/en-us/overview/what-is-azure/>

9. Salesforce. (s.f.). *Salesforce – About us / Historia*.
<https://www.salesforce.com/company/news-press/>
10. Timetoast. (s.f.). *Línea de tiempo de la evolución de la computación en la nube*.
<https://www.timetoast.com/timelines/linea-de-tiempo-de-la-evolucion-de-la-computacion-en-la-nube>