

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

Aplicación de arquitectura en la nube con Amazon Web Services (AWS) para el desarrollo de aplicaciones web contenerizadas mediante Docker y Nginx.

Corporación Universitaria Remington.  
Facultad de Ingeniería  
Ingeniería de Sistemas

Alber Mauricio Arrieta Arrieta  
Brayan Joseph Rueda Preciado  
Moisés Guzmán Tovar  
Tania Toro García  
Ing. Juan Pablo Berrio.  
Opción de Trabajo de grado Seminario-Diplomado.  
2025

## **Dedicatoria**

Dedicamos este trabajo de grado primeramente a Dios, quién ha sido nuestra guía en sabiduría y perseverancia en este proceso de formación; a nuestras familias: padres, esposos, hijos, hermanos, tíos, abuelos..., quienes nos han acompañado incondicionalmente, brindándonos su apoyo emocional, económico y la motivación permanente en todo el proceso de formación profesional en Ingenierías de Sistemas. A nuestro docente, quién han propiciado, con sus enseñanzas, la pasión por esta carrera profesional. También a nuestros compañeros de virtualidad, quienes han dedicado tiempo, sacrificio y han sido pilares de conocimientos significativos y compartidos. Nuestros sueños pronto serán de ustedes

### **Agradecimientos**

Agradecemos inmensamente al tutor del seminario: Juan Pablo Berrío, por su preciada orientación, que consideramos constante; su dedicación y compromiso en el acompañamiento de este proceso académico. Su amplia experiencia y valiosas sugerencias han sido fundamentales para el desarrollo de este trabajo de grado, enfocado en servicios de computación en la nube, brindándonos una asesoría clara y oportuna en cada etapa del proyecto.

## Tabla de Contenidos

Resumen.....	6
Palabras clave.....	7
Marco conceptual y contextual .....	8
¿Qué es AWS? .....	8
VPC AWS.....	8
Máquinas virtuales EC2.....	9
AMI.....	10
El Auto Scaling.....	11
Contenedores.....	12
Docker.....	12
Desarrollo e implementación del aprendizaje.....	14
Introducción .....	14
Introducción .....	28
Servicio s3.....	28
Ilustración 4 .....	30
Ilustración 5 .....	30
Ilustración 6 .....	31
Ilustración 7 .....	32
Ilustración 8 .....	32
Ilustración 9 .....	33
Ilustración 10 .....	33
Ilustración 11 .....	34
Ilustración 15 .....	35
Ilustración 16 .....	36
Ilustración 17 .....	37
Ilustración 18 .....	37
Ilustración 19 .....	38
Ilustración 20 .....	39
Ilustración 21 .....	40
Ilustración 22 .....	40
Ilustración 23 .....	41
Ilustración 24 .....	41
Ilustración 25 .....	42
Ilustración 26 .....	42
Ilustración 27 .....	43
Ilustración 28 .....	43
Ilustración 29 .....	44
Ilustración 30 .....	44
Ilustración 31 .....	45

	5
Ilustración 32 .....	45
Ilustración 33 .....	46
Ilustración 34 .....	47
Ilustración 35 .....	47
Ilustración 36 .....	48
Ilustración 37 .....	48
Ilustración 38 .....	50
Ilustración 39 .....	50
Ilustración 40 .....	51
Ilustración 41 .....	52
Ilustración 42 .....	53
Ilustración 43 .....	53
Ilustración 44 .....	54
Conclusiones .....	57
Referencias.....	59

## ***Resumen***

En el mundo real de la computación en la nube, también conocido como "Cloud Computation", la contenerización es un fenómeno que ha revolucionado el desarrollo y dispersión de aplicaciones en la nube, al permitir que estas se ejecuten en cualquier entorno con consistencia, eficiencia y portabilidad. El desarrollo aplicativo de la arquitectura de nube sobre Amazon Web Services (AWS) facilita de manera ágil la dispersión de aplicaciones web en contenedores virtuales, empleando recursos o herramientas eficaces como la tecnología Docker que sirve para la creación y administración de contenedores SO LINUX; Nginx como servidor proxy de código abierto para direccionar solicitudes, y, Amazon EC2 para crear instancias, de tal modo que ejecuta un entorno escalable y seguro que permite administrar servicios modulares y automatizados.

El trabajo de grado de este seminario hace referencia al contexto de servicios Web Amazon, para la creación de una Virtual Private Cloud personalizada, la configuración de subredes, reglas de seguridad y volúmenes de almacenamiento Elastic Block Store, así como el uso de imágenes AMI para lanzar instancias con las configuraciones necesarias. Se hace énfasis en la automatización del ciclo de vida de las aplicaciones, desde su construcción en contenedores hasta su ejecución dentro de la nube. La arquitectura se fortalece con el uso de MobaXterm para conexiones remotas seguras y comandos CLI para gestión de servicios. Como resultado, se demuestra que la contenerización, combinada con los servicios de AWS, representa una solución eficiente y moderna para desplegar aplicaciones de forma continua, con alta disponibilidad y bajo costo operativo. Este enfoque no solo optimiza recursos, sino que también sienta las bases para una infraestructura preparada para escalar y adaptarse a las demandas actuales del desarrollo web.

De manera resumida, se crearon varios contenedores con la tecnología Docker, con el objetivo de optimizar el uso de recursos y aumentar la portabilidad de la aplicación. Se obtuvieron resultados que permitieron contenerizar todos los elementos necesarios para su correcta ejecución como una aplicación distribuida.

***Palabras clave***

Virtual Private Cloud, Elastic Block Store, Amazon Elastic Compute Cloud, Amazon Machine Image, Contenedor de Docker

## ***Marco conceptual y contextual***

### **¿Qué es AWS?**

Es la plataforma de nube más empleada ya que proporciona servicios de tecnología tales como, mejoramiento en informática, soporte en la infraestructura, almacenamiento, bases de datos e incluso con inteligencia artificial. Su amplia variedad nos brinda más de 200 alternativas de datos, lo que la convierte en la plataforma más ágil, innovadora y accesible, que beneficia principalmente a creadores u entidades en la ejecución en desarrollo de las aplicaciones de forma segura y asequible, dado que proporciona ciertos recursos y niveles sin costo y solo abonas el uso requerido. Al procesar los precios resultan accesibles debido a su bajo costo, y si quieres optimizar los recursos, solo debes detener las instancias que no estás empleando. (AWS LATAM, 2021)

### **VPC AWS**

“Con Amazon Virtual Private Cloud (Amazon VPC), 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”. (*¿Qué Es Amazon VPC? - Amazon Virtual Private Cloud*, s. f.)

Dentro de una VPC podemos realizar distintas funciones las cuales ayudarán con las distintas configuraciones que nuestros servicios requieran para su conectividad. Entre ellas encontramos la creación de subredes para organizar los recursos y mejorar la seguridad de estos, el direccionamiento de IP con el que podremos asignar direcciones IPv4 e IPv6 y establecer tablas de rutas con las que podremos determinar el flujo de tráfico dentro y fuera de la red (*¿Qué Es Amazon VPC? - Amazon Virtual Private Cloud*, s. f.).

Para la conectividad a Internet, Amazon VPC proporciona puertas de enlace, puntos finales para la comunicación entre los servicios de AWS sin abandonar la red privada y conexiones peering para unir diferentes VPCs enrutando su tráfico. Además, incluye opciones

como VPNs para la integración de redes locales. (*¿Qué Es Amazon VPC? - Amazon Virtual Private Cloud, s. f.*).

También ofrece otras funciones, como la duplicación de tráfico para el monitoreo, registros de flujo de VPC, que capturan la información que entra y sale de la VPC y con las puertas de tránsito, las cuales actúan como un concentrador central para gestionar la comunicación de múltiples VPCs, así como redes externas. (*¿Qué Es Amazon VPC? - Amazon Virtual Private Cloud, s. f.*).

## **Máquinas virtuales EC2**

las máquinas virtuales (VM) son la simulación de equipos físicos dentro de otros esto gracias a la asignación de recursos CPU, memoria, red y otros componentes para su funcionamiento permitiendo así la ejecución de sistemas operativos y aplicaciones como si se estuviera en un hardware real, pero de manera aislada en otro equipo.

Entre los servicios que AWS nos ofrece nos encontramos un servicio llamado servicio Elastic Compute Cloud (Amazon EC2) en donde podemos crear y administrar máquinas virtuales que tienen el nombre de instancias, este servicio también nos ofrece una capacidad de cómputo escalable y bajo de manda lo cual permite hacer ajustes según la demanda, disminuyendo o aumentado los recursos según sea necesario, esto puede ayudar ahorrar costos o mejorar la capacidad de alguna tarea que lo requiera. con este servicio también tenemos la ventaja de que ahorramos los costos de tener un equipo físico el cual nos consume electricidad y hay que hacerle mantenimiento cada cierto tiempo, esto pasa ya que administramos las instancias de manera remota, este modo facilita el desarrollo e implementación de aplicaciones. Podemos crear y disponer de varios servidores o máquinas virtuales como sean necesarias para nuestro uso, así como poder configurar la seguridad, la red y gestionar el almacenamiento. (*¿Qué Es Amazon EC2? - Amazon Elastic Compute Cloud, s. f.*).

una instancia EC2 es un servidor en la nube, es decir una máquina virtual que vamos a tener a nuestra disposición, al momento de crearla podremos especificar los recursos que esta va a poseer con plantillas preconfiguradas que ya traen un sistema operativo las cuales son llamadas

AMIs, para luego poder acceder y hacer uso de la instancia desde nuestro equipo de forma remota. (*¿Qué Es Amazon EC2? - Amazon Elastic Compute Cloud*, s. f.).

Hay varios de tipos de instancias con distintas capacidades de recursos, es decir CPU, memoria y almacenamiento. para almacenar datos en ellas se usan los volúmenes de datos llamados EBS en el que los datos son persistentes o el almacén de instancias en el que los datos son temporales y se borran al detener o apagar la maquina o servidor. Para el apartado de seguridad se usan pares claves para un acceso seguro y los grupos de seguridad que funcionan como firewall determinado que conexiones son permitidas. (*¿Qué Es Amazon EC2? - Amazon Elastic Compute Cloud*, s. f.)

## **AMI**

Las imágenes de máquina de Amazon o (AMI) en Amazon ec2, son una imagen suministrada por el software, que necesita ser configurada para comenzar una instancia de ec2. En AWS, nos proporcionan múltiples imágenes de base para diferentes sistemas operativos, estas AMI proporcionan los recursos de edición para facilitar los datos de uso. De esta forma, tenemos la posibilidad de usar nuestra propia imagen, lo que nos proporciona beneficios, como paquetes ya instalados o la posibilidad de utilizar la AMI de otra persona para optimizar los recursos. Cada AMI incluye una asignación concreta de los dispositivos en los bloques que deben estar vinculados con cada instancia para su lanzamiento, y debe ser compatible con la instancia seleccionada. Para establecer el inicio de una AMI, es necesario utilizar un parámetro que se basa en la configuración realizada por el sistema operativo y el tipo de instancia. Por ejemplo, una AMI con un parámetro de modo de arranque de uefi genera una instancia con un parámetro llamado currentInstanceBootMode. Con este parámetro podemos establecer cómo deseamos que comience la instancia, ya sea desde la BIOS o UEFI, la utilidad del parámetro depende del tipo de instancia y la configuración, por otro lado, si no cuando no especificamos un valor para el modo arranque, la AWS fijara un modo de iniciación desde la arquitectura de la AMI y el tipo de instancia. (AWS, 2025).

## **Equilibrador de carga**

El equilibrador de carga, o load balancer, es el encargado de monitorear y redistribuir el tráfico en la red. Gracias a su servicio, podemos trazar el camino de las solicitudes, lo que facilita la distribución de las peticiones de los usuarios entre diferentes instancias del servidor, cada una con rutas variadas hacia distintos destinos de manera eficiente. Su objetivo principal es equilibrar la carga de trabajo para garantizar que las aplicaciones sean escalables y que el funcionamiento de las instancias sea óptimo. Existen varios tipos de balanceadores de carga, como los de aplicaciones, red y software. Por ejemplo, el balanceador de carga de aplicación está diseñado para operar en la capa de aplicación, que corresponde a la etapa 7 del modelo OSI. Este balanceador se encarga de recibir las solicitudes HTTP en el puerto 80, permitiendo que estas peticiones se distribuyan entre las instancias dentro del entorno de Elastic Beanstalk. Así, se puede gestionar eficazmente el tráfico de entrada, manejando el aumento del tráfico sin necesidad de intervención y supervisando las instancias. Además, podemos añadir un agente de escucha seguro en el puerto 80 para permitir el tráfico HTTPS. Los listeners verifican las solicitudes entrantes y las dirigen al puerto configurado según el protocolo, permitiendo el paso del tráfico a las instancias o contenedores correspondientes. Las verificaciones del estado de las instancias se realizan mediante solicitudes en un camino específico durante períodos establecidos, lo que determina si el registro se realizó con éxito; las instancias deben responder y estar disponibles. También se pueden habilitar los registros de acceso en el balanceador de carga de aplicaciones para documentar información que puede ser almacenada en un contenedor de Amazon S3.

## **El Auto Scaling**

El Auto Scaling simplifica la gestión de recursos en la interfaz, asegurando que nuestras aplicaciones funcionen de manera óptima en las instancias de Amazon EC2. Este proceso aumenta la eficiencia, rentabilidad y accesibilidad, gracias a su bajo costo el cual solo se genera el pago por los recursos utilizados, Al implementar este recurso, podemos automatizar la gestión de nuestras instancias, estableciendo tanto un límite máximo como mínimo de instancias. Además, tenemos la posibilidad de configurar una plantilla de instancia que incluya elementos como la AMI, grupos de seguridad, pares de claves, direcciones IP y otras configuraciones

definidas. De este modo, cada vez que creamos nuevas instancias, podemos asociarlas con las plantillas ya preestablecidas. También contamos con diversas opciones de escalado, ya sea manual o automatizado, lo que nos permite monitorear constantemente nuestras instancias y adaptarnos a las fluctuaciones en la demanda diaria. (Barr, 2018)

## Contenedores

Los contenedores son una forma de empaquetar aplicaciones ejecutables donde estas traen el código, las dependencias, bibliotecas y todas las configuraciones, es decir trae todo lo que necesita la aplicación para funcionar y con esto se logra que puedan ser ejecutadas en cualquier entorno informático como lo pueden ser una computadora normal, un servidor tradicional o en la nube. (*¿Qué Son los Contenedores? Google Cloud*, s. f.)

Los contenedores funcionan en cualquier entorno gracias a que aprovechan una forma de virtualización en el que usan el kernel del sistema operativo, por ejemplo, los silos y objetos de trabajo de Windows, los espacios de nombres y groups de Linux, Estas herramientas hacen la función de aislar los procesos y el número de recursos que va a estar usando cada contenedor para que no interfieran entre sí, es decir se va a estar ejecutando la aplicación en un espacio aislado, pero compartiendo el mismo sistema operativo. (*¿Qué Son los Contenedores? Google Cloud*, s. f.).

Hoy en día gracias a que estos son más portables y ligeros que las máquinas virtuales, al arrancar más rápido y por su capacidad de consumir menos recursos son ideales para entornos donde se necesite velocidad es por ello por lo que son la forma estándar de ejecutar aplicaciones modernas. (*¿Qué Son los Contenedores? Google Cloud*, s. f.)

## Docker

Docker es una plataforma abierta que permite crear, distribuir y ejecutar aplicaciones dentro de contenedores. Esto permite lograr que el tiempo transcurrido entre la escritura del código y su ejecución disminuya significativamente al ser aprovechadas sus funciones para

distribuir probar e implementar código en cualquier entorno informático. (¿«*What Is Docker?*», 2024).

“Docker es ligero y rápido. Ofrece una alternativa viable y rentable a las máquinas virtuales basadas en hipervisor, lo que le permite aprovechar al máximo la capacidad de su servidor para alcanzar sus objetivos empresariales. Docker es perfecto para entornos de alta densidad y para implementaciones pequeñas y medianas donde necesita optimizar recursos”. (¿«*What Is Docker?*», 2024).

Docker funciona con una arquitectura cliente-servidor en donde el cliente se comunica con dockerd(servidor) el cual administra los contenedores en términos de creación, ejecución y distribución. El cliente y el dockerd pueden estar en el mismo equipo o en equipos separados ya que su comunicación se hace a través de API REST. (¿«*What Is Docker?*», 2024).

Se construyen las aplicaciones a partir de imágenes las cuales son planillas de solo lectura creadas con un archivo que se llama Dockerfile en donde se indican los pasos para la configuración de la aplicación. Estas imágenes se pueden compartir a través de registros Docker Hub y se pueden descargar como lo hicimos al momento de descargar una en nuestra instancia de Linux y pudimos usarla para crear varios contenedores con esa imagen de HTTPD. (¿«*What Is Docker?*», 2024).

Docker como vemos resulta super importante en las arquitecturas de nube como aws gracias a su portabilidad y uniformidad, facilitando el traslado de aplicaciones entre estos ambientes sin tener que realizar cambios en el código o la configuración. (¿«*What Is Docker?*», 2024)

## *Desarrollo e implementación del aprendizaje*

### **Introducción**

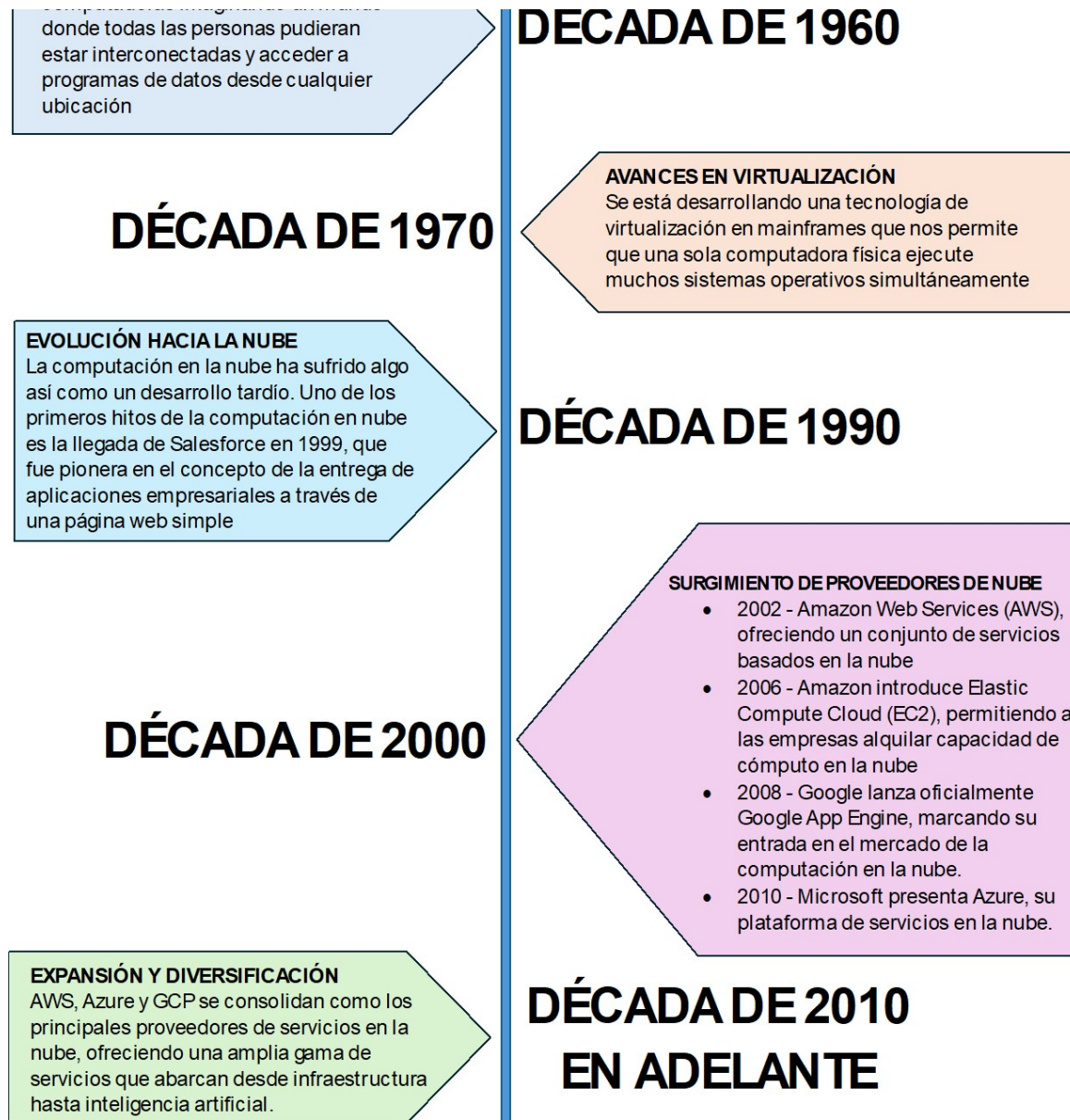
A lo largo de la historia se ha venido hablando de la virtualización y computación en la nube, quizás estas metodologías han transformado la manera que se automatizan los recursos informáticos. La realización de este primer parcial explora sus orígenes, analiza los avances mediante la línea de tiempo, compara algunos servicios de los principales proveedores en la nube como: AWS, Azure, GCP e implementa un balanceador de carga con un autoscaling y la creación de política.

- Orígenes de la virtualización y la computación en la nube, haga una línea de tiempo con los diferentes hitos para que hoy en día exista el nivel tecnológico que hemos alcanzado en la nube.

1. Comparación de nombres de servicios entre AWS, Azure y GCP.

### *Ilustración 1.*

*línea de tiempo de la computación en la nube desde de la década 1960 hasta la actualidad.*



*Nota: Adaptado del seminario AWS. Elaboración propia.*

Tabla 1

SERVICIOS	AWS	Azure	GCP	Características
Categoría				
<b>Computación</b>	EC2	VIRTUAL MACHINES	COMPUTE ENGINE	Estos servicios permiten la creación de máquinas virtuales en la nube con la posibilidad de ajustar la CPU, la memoria y el almacenamiento.
	LAMBDA	FUNTIONS	CLOUD FUNTION	Estos servicios sin servidor administrando los recursos informáticos ejecuta el código en respuesta a los eventos es decir cuando un usuario hace una solicitud.
	EKS	AKS	GKE	Estos servicios administran aplicaciones basados en contenedores utilizando kubernetes.
<b>Almacenamiento</b>	S3	BLOB STORAGE	CLOUD STORAGE	Estos son servicios de almacenamiento escalables y gestionados de grandes cantidades de datos
<b>Bases de Datos</b>	RDS	SQL DATABSE	CLOUD SQL	Estos servicios proporcionan bases de datos relacionales listas para ser usadas.
	DYNAMODB	COSMOSDB	FISTORE /BIGTABLE	Estos servicios nos proveen de bases de datos no relacionales diseñadas para el manejo de grandes volúmenes de datos con alta velocidad y disponibilidad.

<b>Redes y seguridad</b>	VPC	VIRTUAL NETWORK	VIRTUAL PRIVATE CLOUD	Estos servicios permiten crear y gestionar redes privadas dentro de la nube para conectar distintos recursos de forma segura.
	IAM	AZURE AD	IAM	Estos servicios permiten la administración de identidades y permisos de usuarios definiendo quien puede entrar y a qué servicio.
<b>Entrega de contenidos</b>	CLOUDWATCH	MONITOR	CLOUD MONITORING	Estos servicios brindan la supervisión del rendimiento de las aplicaciones y recursos en la nube.
<b>Mensajería y comunicación</b>	SNS	EVENT GRID	PUB/SUB	Estos servicios envían mensajes a través de eventos de un sistema a otro en tiempo real, como alertas de seguridad o actualizaciones de pedidos.
<b>Integración y procesamiento de datos</b>	GLUE	DATA FACTORY	DATAFLOW	Estos servicios son herramientas que automatizan el movimiento y transformación de datos entre diferentes sistemas, permitiendo mayor facilidad para análisis o almacenamiento.

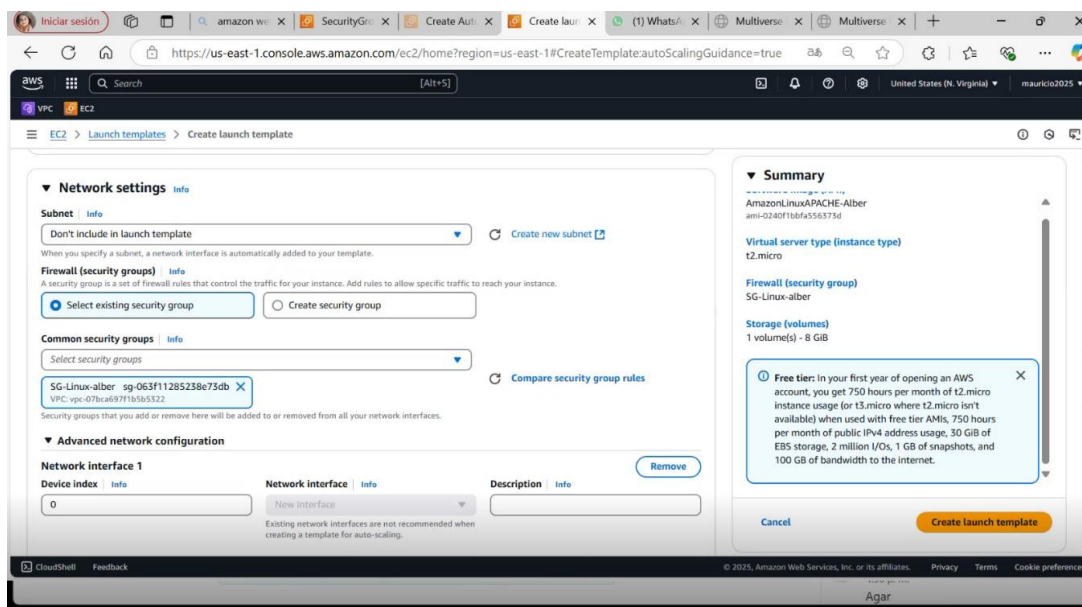
<b>Análisis y big data</b>	REDSHIFT	SYNAPSE	BIGQUERY	Estos servicios permiten almacenar y analizar un gran volumen de datos de manera rápida, permitiendo tomar decisiones basados en información detallada.
<b>Machine learning e inteligencia artificial</b>	SAGEMARKE	MACHINE LEARNING	VERTEX AI	Estos servicios brindan la facilidad de creación, entrenamiento, implementación y gestión de modelos de inteligencia artificial.

*Nota: Esta tabla resume de manera comparativa, los servicios más populares de computación en la nube ofrecidos por Amazon Web Services, Azure y Google Cloud Products en distintas categorías, ayudando a interpretar la comparación de sus funcionalidades y aplicaciones. Por medio de esta comparación, es posible la identificación de fortalezas y diferencias de cada proveedor, lo que permite la toma de decisiones estratégicas para la implementación de soluciones en la nube. Por otro lado, se hace relevante los recursos informáticos que ofrece la computación en la nube como una herramienta popular para la escalabilidad, optimización de costos y seguridad en ambientes empresariales.*

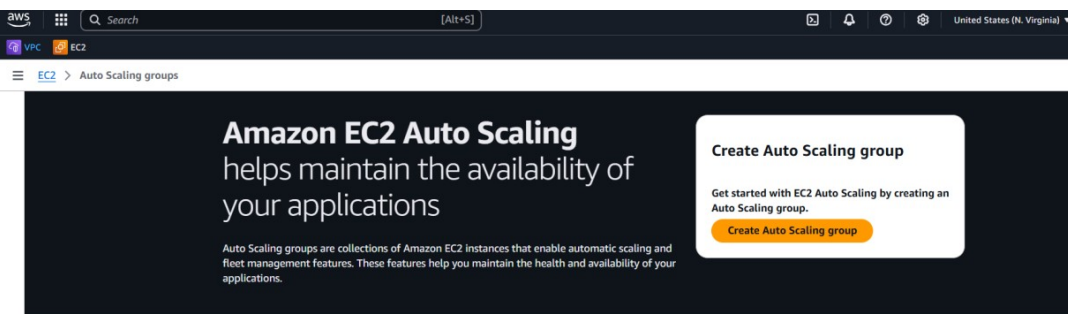
Seminario AWS



2. Implemente un balanceador de carga con un autoscaling, las instancias deben ser Linux, debe tener una aplicación web básica que cargue automáticamente al crearse la instancia. Se debe mostrar cómo funciona el autoscaling al terminar instancias manualmente.



Entramos a la sección de Auto Scaling groups en donde vamos a dar clic en Create Auto Scaling group. Con esto haremos configuraciones para balancear la carga de nuestras instancias automáticamente, lo que permitirá que se creen o terminen en función a la demanda.



#### How it works



#### Pricing

Amazon EC2 Auto Scaling features have no additional fees beyond the service fees for Amazon EC2, CloudWatch (for scaling policies), and the other AWS resources that you use. Visit the pricing page of each service to learn more.

#### Getting started [↗](#)

Entramos a una sección sobre la selección de una plantilla de lanzamiento para un grupo de Auto Scaling de EC2 donde se nos pide un nombre para el auto scaling group, el cual le pusimos AS-WEB

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

### Choose launch template Info

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

**Name**

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

SC-WEB\_alber

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

**Launch template Info**

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

**Launch template**  
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

Select a launch template

Create a launch template [↗](#)

Cancel Next

En la parte de launch template lo mantenemos como viene predeterminadamente y le damos clic en next para proseguir con los demás ajustes

aws Search [Alt+S] United States (N. Virginia) mauricio2025

VPC EC2

EC2 > Auto Scaling groups > Create Auto Scaling group

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

### Choose launch template Info

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

**Name**

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

AS-WEB

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

**Launch template Info**

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

**Launch template**  
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

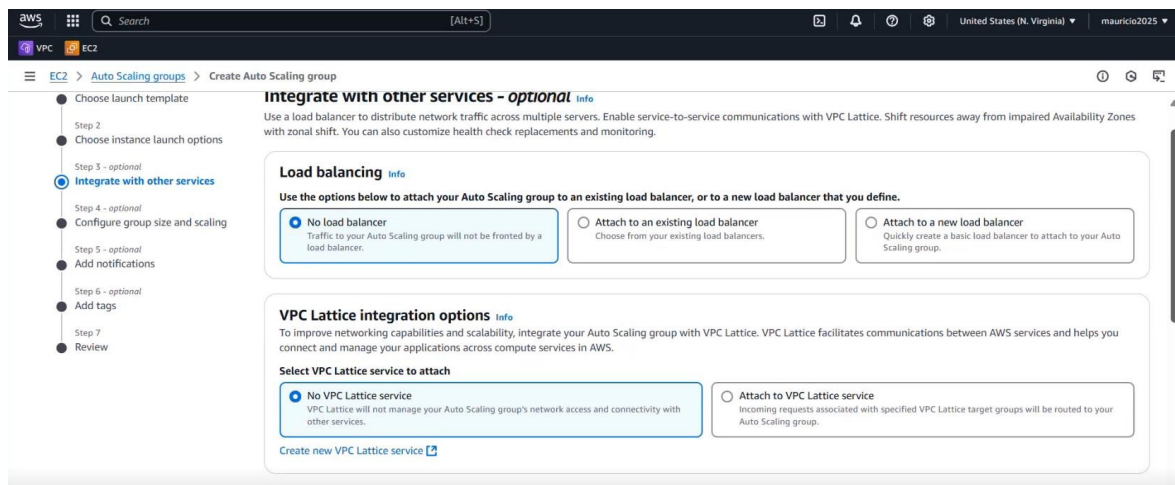
Select a launch template

Search launch templates

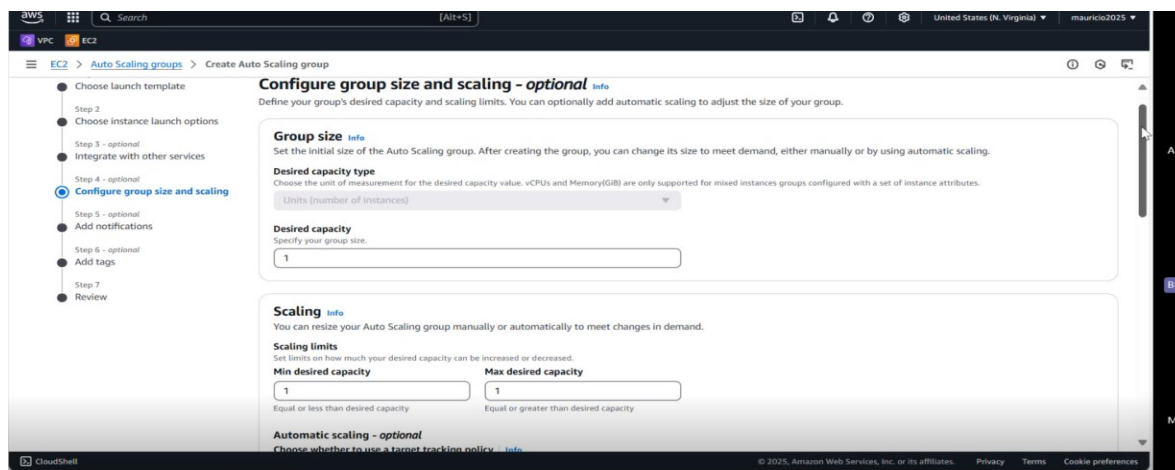
No launch templates

Cancel Next

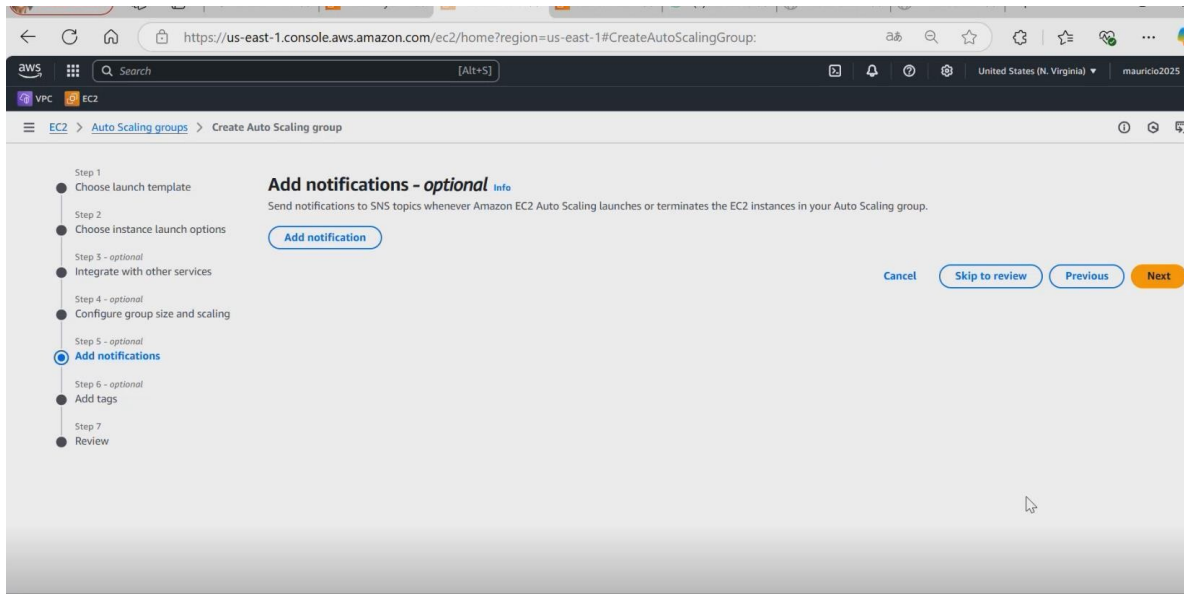
Se nos da la opción de integrarlo con otros servicios. En las opciones de balanceo de carga elegimos la que dice “no load balancer” esto va a significar que no vamos a integrar un balanceador de carga existente, luego se nos indica la opción de integración de un VPC lattice en la cual marcamos “No VPC lattice service” lo que significa que no nos vamos a comunicar con otras VPC.



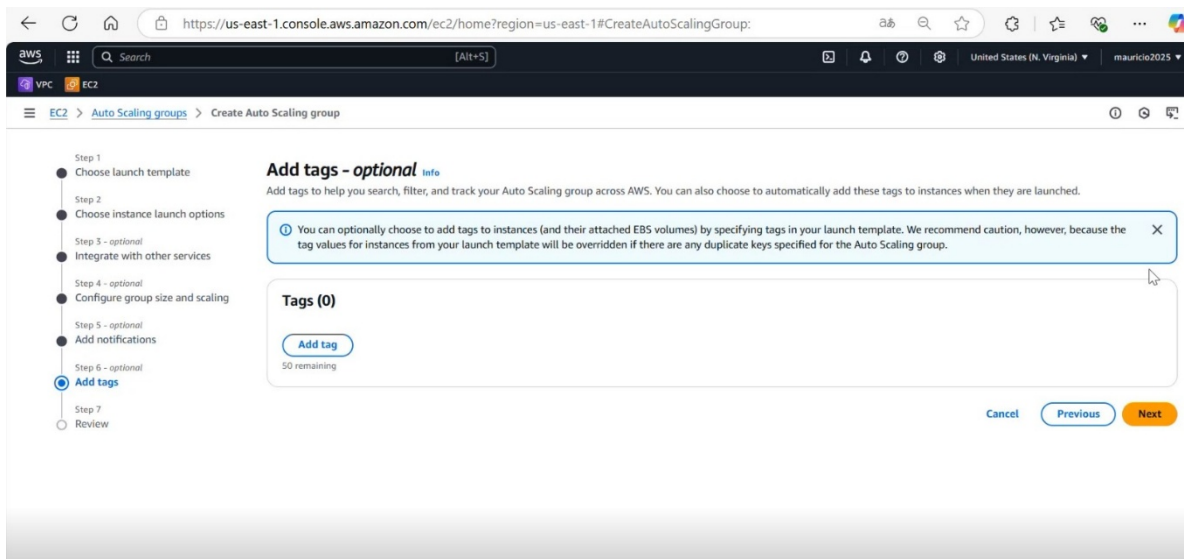
Entramos a ahora a la sección de configuración del tamaño de scaling group en donde vamos a determinar el número de instancias que el grupo va a tener en cualquier en la opción de “Desired capacity” momento la capacidad mínima es la opción “min desired capacity” la cual es el número de instancias que el grupo debe mantener y la capacidad máxima e la opción “max desired capacity” esta es el número máximo permitido de instancias



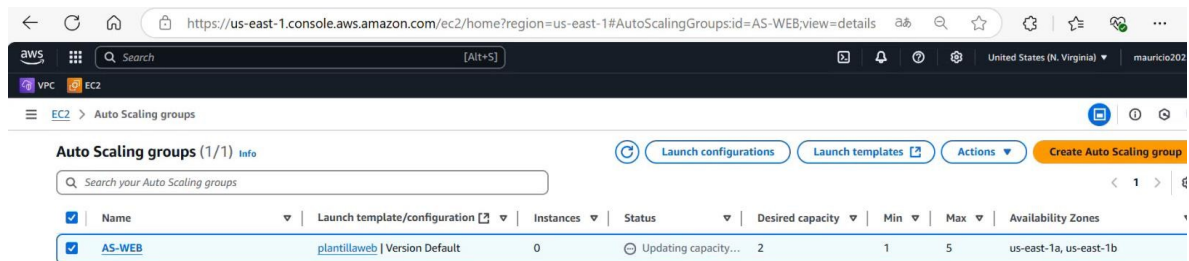
Entramos a la sección de add notifications en la cual podemos decidir si recibir notificaciones cuando el auto scaling cree una instancia o termine alguna en el grupo, elegimos no agregar notificaciones y damos clic en next



Entramos en la sección de add tags en donde decidiremos si queremos agregar etiquetas a las instancias lanzadas por el staling group, elegiremos que no y damos clic en next



Finalizamos la creación del group staling group mostrándonos nuestro grupo creado en la sección de Auto Scaling groups

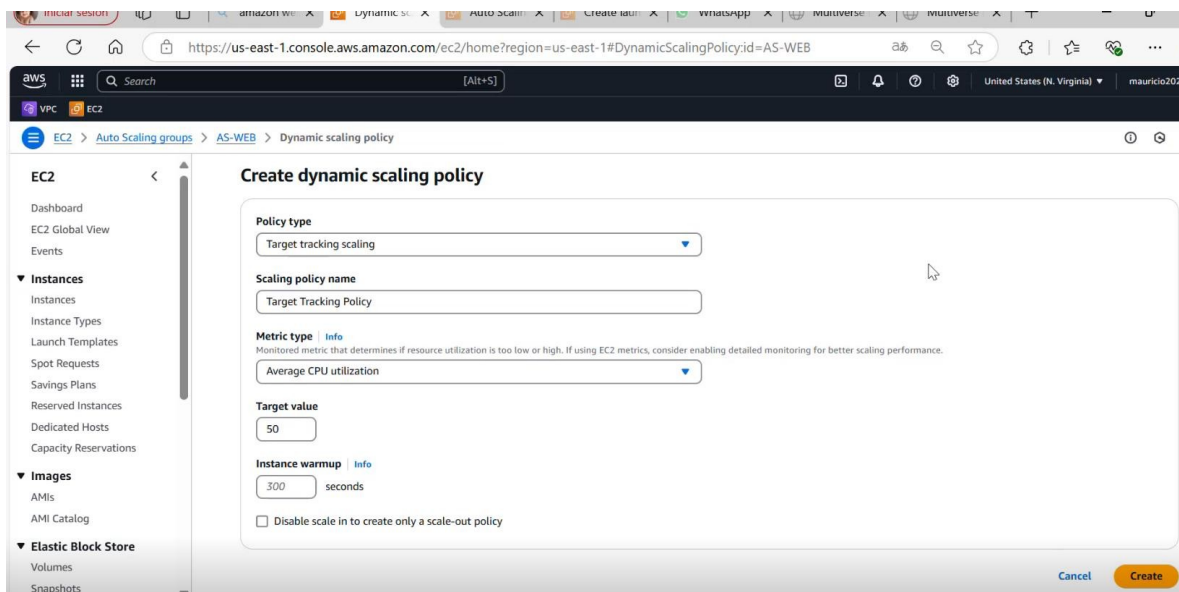


De manera resumida, the load balancer, o balanceo de carga ayuda a distribuir el tráfico entre varios servidores, garantizando estabilidad y eficiencia. Para ello, se implementa un Load Balancer en Amazon Web Services. Por otra parte, el autoscaling permite aumentar o reducir los recursos según la demanda.

3. Se debe crear una política de autoscaling al tener un aumento de cpu de un umbral definido. Ej: 50%

Entramos a la sección de nuestro auto staling group creado AS-WEB y vamos a la opción de Dinamic scaling policy donde definiermos una política que ajuste el número de instancias en función de la utilización promedio de la CPU

En la opción policy type dejamos como "Target tracking scaling" lo que significa que se ajustara automáticamente para mantener la métrica. La opción Scaling policy name le daremos el nombre de "Target Tracking Policy" y en la opción de metric type elegimos la métrica "Average CPU utilization" que significa el promedio de uso de CPU luego elegimos el valor en la opción target value y elegimos 50 lo que quiere decir que el promedio de CPU. Para finalizar le damos clic en el botón créate



The screenshot shows the AWS Management Console interface for creating a dynamic scaling policy. The browser address bar indicates the URL: `https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#DynamicScalingPolicy:id=AS-WEB`. The console navigation pane on the left shows the path: `EC2 > Auto Scaling groups > AS-WEB > Dynamic scaling policy`. The main content area is titled "Create dynamic scaling policy" and contains the following fields:

- Policy type:** A dropdown menu set to "Target tracking scaling".
- Scaling policy name:** A text input field containing "Target Tracking Policy".
- Metric type:** A dropdown menu set to "Average CPU utilization". Below this field is a small informational text: "Monitored metric that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance."
- Target value:** A text input field containing "50".
- Instance warmup:** A text input field containing "300" followed by the unit "seconds".
- Disable scale in:** A checkbox labeled "Disable scale in to create only a scale-out policy", which is currently unchecked.

At the bottom right of the form, there are two buttons: "Cancel" and "Create".

Se agrego satisfactoriamente las políticas de Dinamic scaling policy

Dynamic scaling policy created or edited successfully.

### AS-WEB

AS-WEB Capacity overview [Edit](#)

arn:aws:autoscaling:us-east-1:422719327954:autoScalingGroup:ffae0e14-ac56-4779-a5aa-35a6ccc6776a:autoScalingGroupName/AS-WEB

Desired capacity	Scaling limits (Min - Max)	Desired capacity type	Status
2	1 - 5	Units (number of Instances)	-

Date created  
Mon Mar 24 2025 16:42:30 GMT-0500 (hora estándar de Colombia)

Details | Integrations - new | **Automatic scaling** | Instance management | Instance refresh | Activity | Monitoring

Scaling policies resize your Auto Scaling group to meet changes in demand. With reactive dynamic scaling policies, you can track specific CloudWatch metrics and take action when the CloudWatch alarm threshold is met. Use predictive scaling policies along with dynamic scaling policies in the following situations: when your application demand changes quickly, but with a recurring pattern, or when your EC2 instances require more time to initialize.

Dynamic scaling policies (1) [Info](#) [Actions](#) [Create dynamic scaling policy](#)

Se aprecia que se crearon automáticamente las instancias definidas en el auto scaling group

Instances (8) [Info](#)

Find Instance by attribute or tag (case-sensitive) [All states](#)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
	i-0133c88cc767afeff	Terminated	t2.micro	-	View alarms +	us-east-1b	-
	i-0339a69f14657e759	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	-
Linux-Alber	i-042ffb8196ed7f44	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-34-230-8-115.c
	i-0bf7a3a63347996cd	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-54-158-9-124.c
	i-0e4118a82804394e4	Initializing	t2.micro	Initializing	View alarms +	us-east-1a	-

Vemos que muestra toda su información y que están corriendo normalmente

The screenshot displays the AWS Management Console interface for an EC2 instance. The left sidebar shows navigation options like Dashboard, EC2 Global View, and Instances. The main content area shows the instance details for ID i-0e4118a82804394e4. The instance is currently in a 'Running' state. Key configuration details include: Hostname type (IP name: ip-10-0-129-118.ec2.internal), Instance type (t2.micro), VPC ID (vpc-07bca697f1b5b5322), Subnet ID (subnet-04ca1c5f4eccc20e4), and IAM Role (Required). The instance is managed by the AS-WEB Auto Scaling Group. Below the details, there are tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. The 'Instance details' section shows the AMI ID (ami-0240f1bbfa556373d) and the platform details (Linux/UNIX).

Para finalizar se aprecia que nos podemos conectar normalmente a ellas

The screenshot shows the 'Connect to instance' page in the AWS Management Console for instance i-0c3cb9b3c271b4beb. The page provides three methods for connecting: EC2 Instance Connect, Session Manager, and SSH client. The 'SSH client' option is selected, and it provides a list of steps for connecting via SSH. The steps are: 1. Open an SSH client. 2. Locate your private key file. The key used to launch this instance is AlberLinux.pem. 3. Run this command, if necessary, to ensure your key is not publicly viewable. 4. Connect to your instance using its Private IP: 10.0.132.169. An example command is provided: `ssh -i "AlberLinux.pem" root@10.0.132.169`. A note states: "Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username." A 'Cancel' button is visible at the bottom right.

En las imágenes se evidencia la Create Dynamic Scaling Policy, también conocida como política de autoscaling, esta define las condiciones para escalar los recursos. Por ejemplo, se puede configurar para que aumente instancias cuando el uso de CPU supere el 50% y reduzca cuando esté por debajo del 30%. Lo anterior significa que, esto optimiza costos y garantiza un rendimiento eficiente.

## **Introducción**

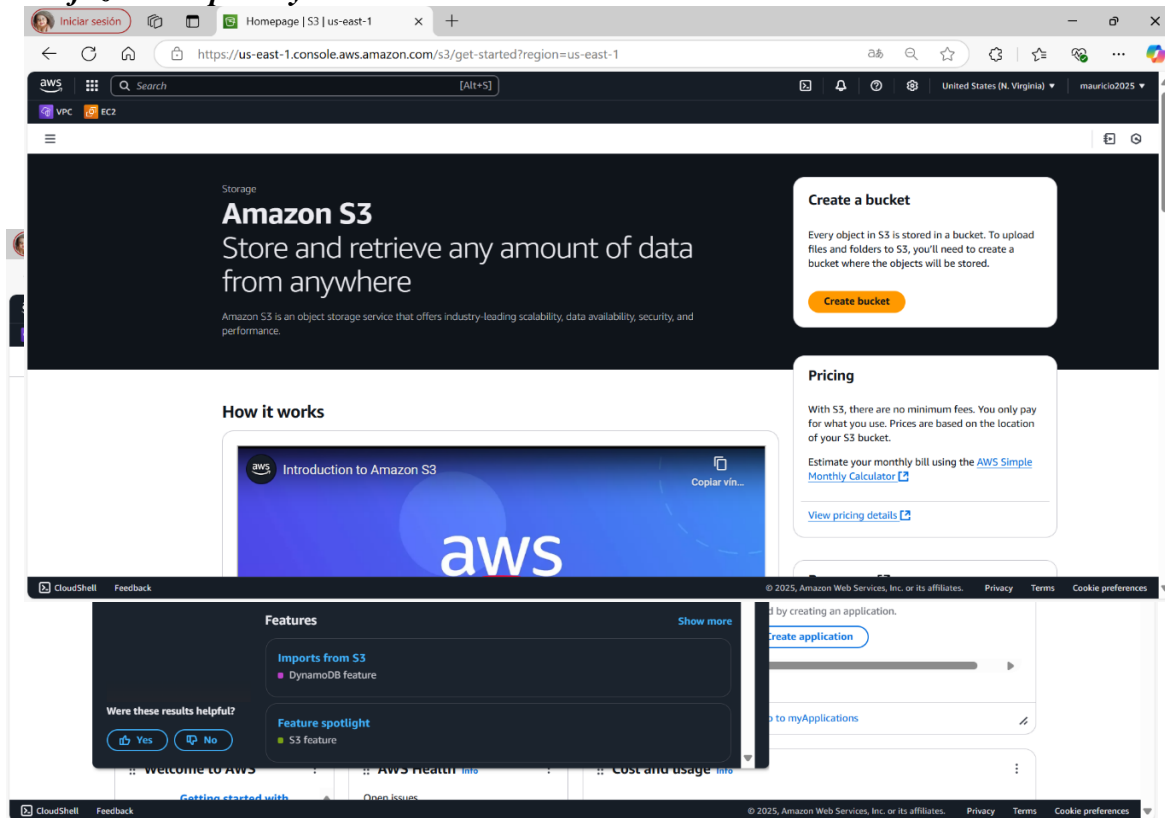
*En esta entrega segundo parcial se lleva a cabo la configuración y despliegue de un entorno en AWS utilizando varios servicios clave: la Creación de un Bucket en S3, este se configura un Amazon S3 Bucket para almacenar archivos estáticos o respaldos que puedan ser utilizados por las aplicaciones dentro de los contenedores. Segundo, la implementación de Contenedores Docker, los cuales se despliegan tres contenedores Docker, cada uno corriendo en un puerto diferente (8080, 8081, 8082), los cuales muestran diferentes pruebas de sitios web. Al igual se realiza la configuración de una Instancia ECS, que se utiliza Amazon ECS (Elastic Container Service) para administrar y escalar los contenedores de manera eficiente dentro de la infraestructura de AWS. Por otro lado, la configuración de Nginx como Reverse Proxy se implementa Nginx en una instancia EC2 para redirigir el tráfico a los contenedores con base en un DNS personalizado, permitiendo el acceso a través de dominios específicos como: (www.uniremington.edu, www.seminario.com, [www.carulla.com](http://www.carulla.com)). Por último, la modificación del Archivo Local Hosts en Windows se configura el archivo hosts en la máquina local para mapear los dominios personalizados a la dirección IP de la instancia EC2, facilitando la resolución de nombres y el acceso a las aplicaciones.*

## **Servicio s3**

Store and retrieve any amount of data from anywhere

## Ilustración 2

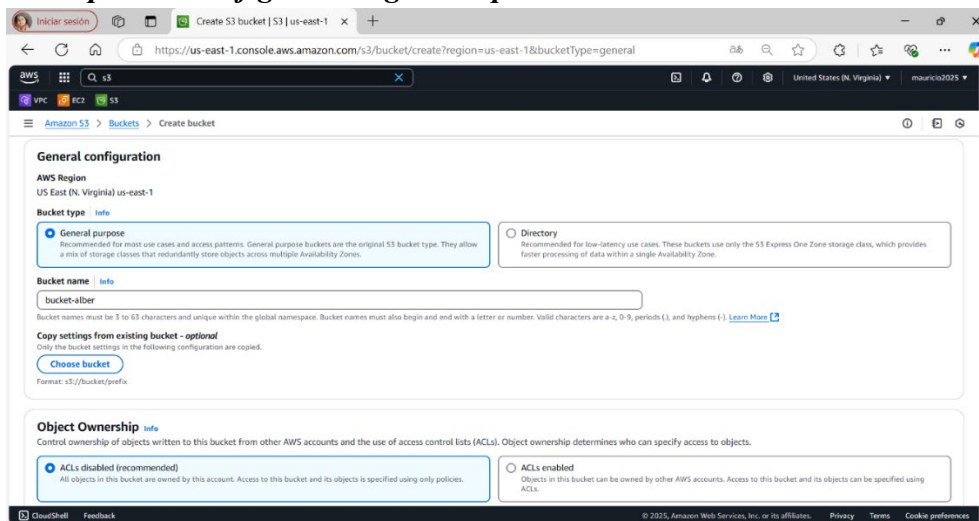
### Interfaz de búsqueda y acceso a servicios de almacenamiento en la consola de AWS



*Nota: Capture de pantalla de la consola de Amazon Web Services (AWS), Fuente propia, 2025.*

## Ilustración 3

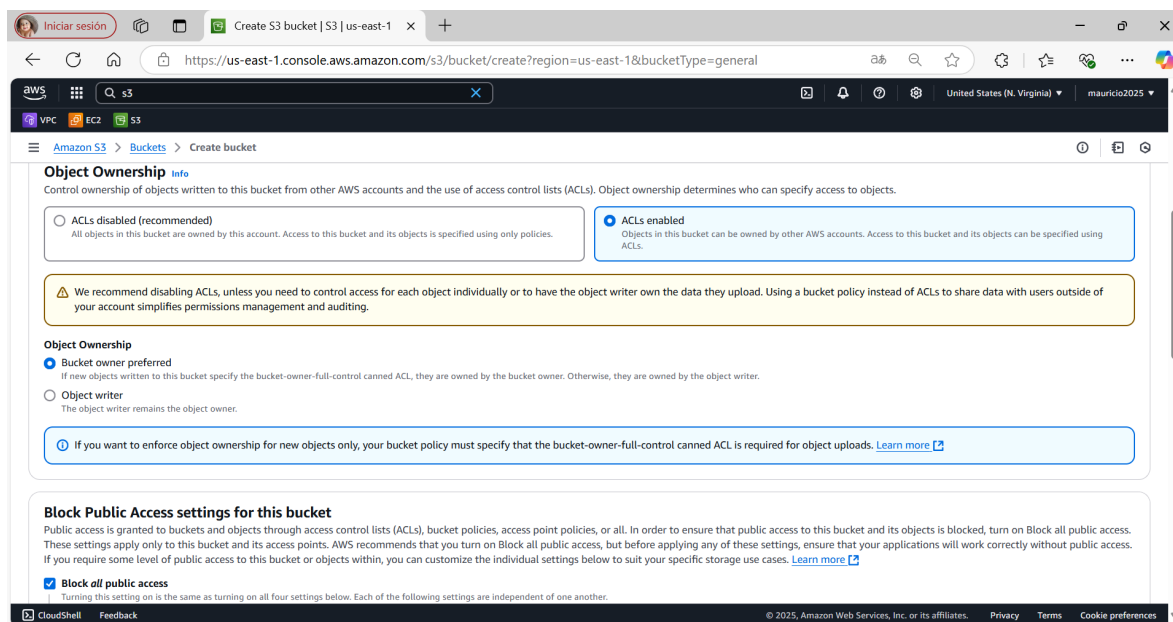
### Pasos para la configuración general para la creación de un bucket en Amazon S3.



*Nota: Lista de control de acceso ACL, permite asignar permiso a los buckets, se activan los ACLs para restringir acciones a los usuarios y solo modificados por el propietario. Fuente propia, 2025*

### **Ilustración 4**

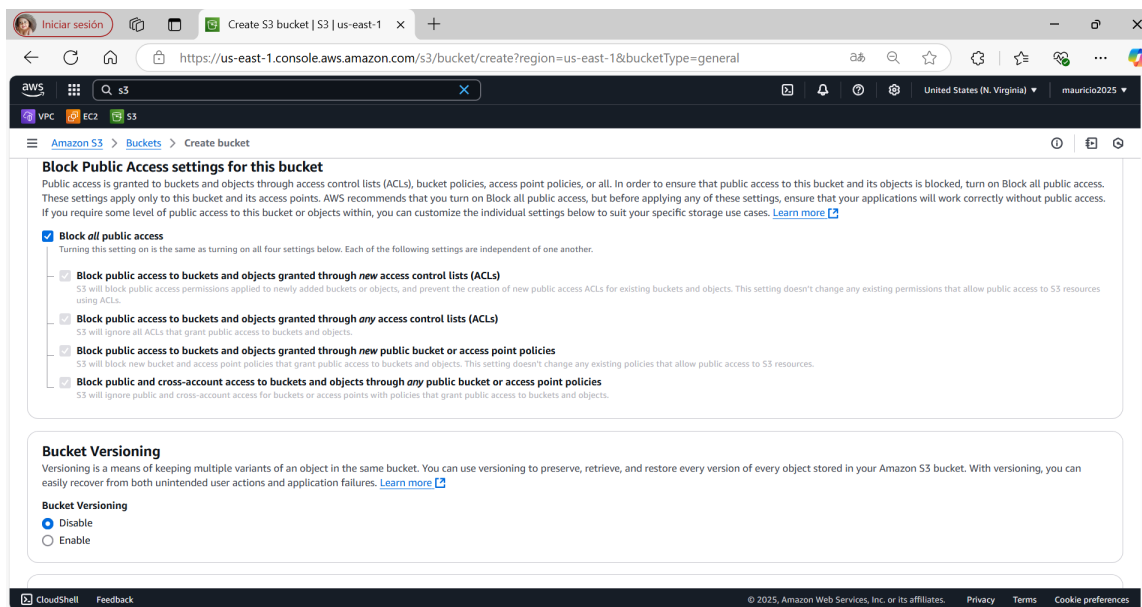
*Establecimiento de la propiedad de los objetos (ACLs) y opciones para bloquear el acceso público durante la creación de un bucket.*



*Nota: Bloquear acceso público a los bucket, que por defectos son acceso privado. Activar El versionamiento de Bucket permitir almacenar archivos con el mismo nombre, pero con diferentes versiones. Es también utilizado para copias de seguridad*

### **Ilustración 5**

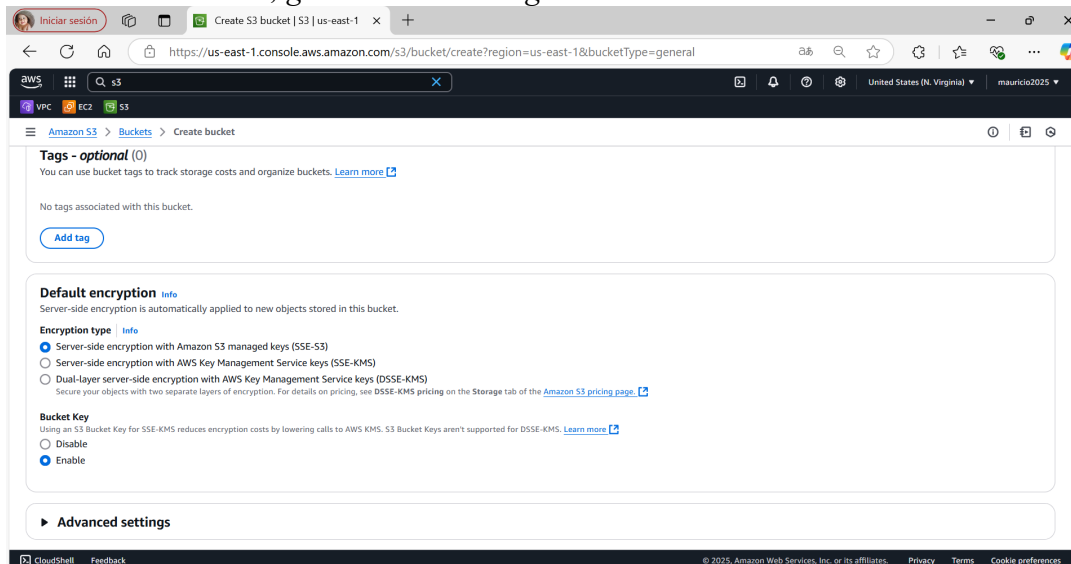
*Configuración para restringir el acceso público a los objetos almacenados en un bucket*



*Nota: Bloquear acceso público a los bucket, que por defectos son acceso privado. Activar El versionamiento de Bucket permitir almacenar archivos con el mismo nombre, pero con diferentes versiones. Es también utilizado para copias de seguridad. Adaptado de Seminario AWS, 2025.*

### **Ilustración 6**

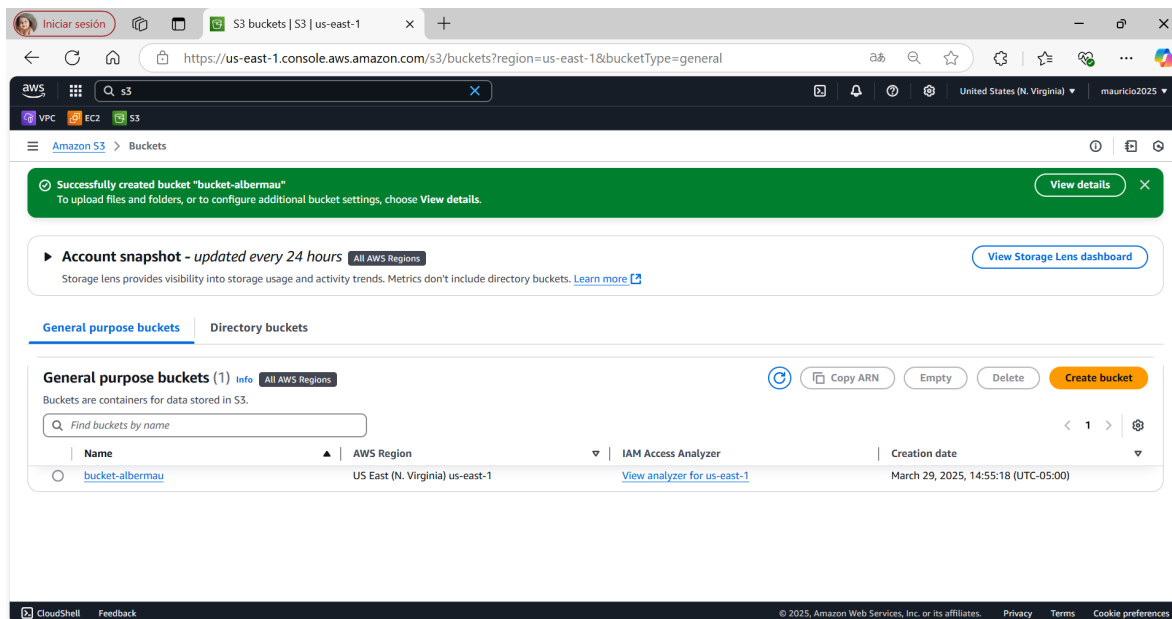
*Interfaz de Amazon S3 que permite agregar etiquetas al bucket y seleccionar el tipo de cifrado del lado del servidor, garantizando la seguridad de los datos almacenados.*



*Nota: Etiqueta de ambiente de desarrollo y pruebas. El mecanismo de encriptación es usado para encriptar los datos almacenado en carpeta, AWS se encarga en ocasiones manejar el cifrados de ellos...Adaptado del seminario AWS. Fuente propia 2025*

## Ilustración 7

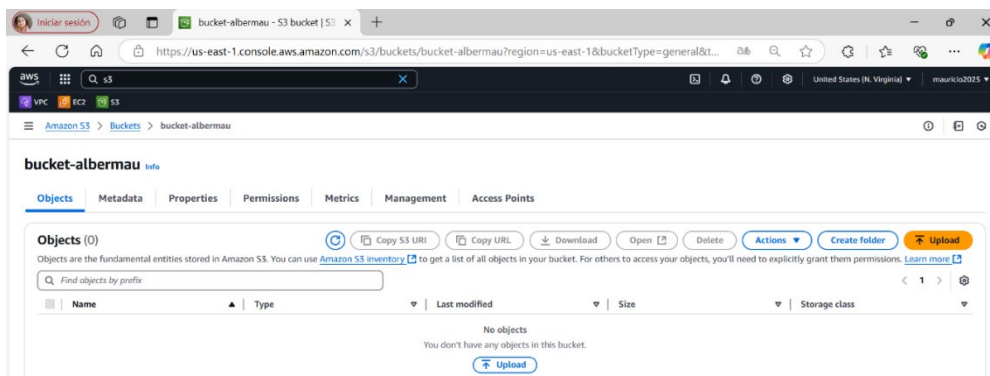
Visualización del mensaje de éxito tras la creación del bucket “bucket-albermau” en la región US East (N. Virginia), dentro de la consola de Amazon S3.



Nota: En este sitio creamos s3. Ingresamos al bucket para ver detalles de la creación. Adaptado del Seminario AWS, 2025

## Ilustración 8

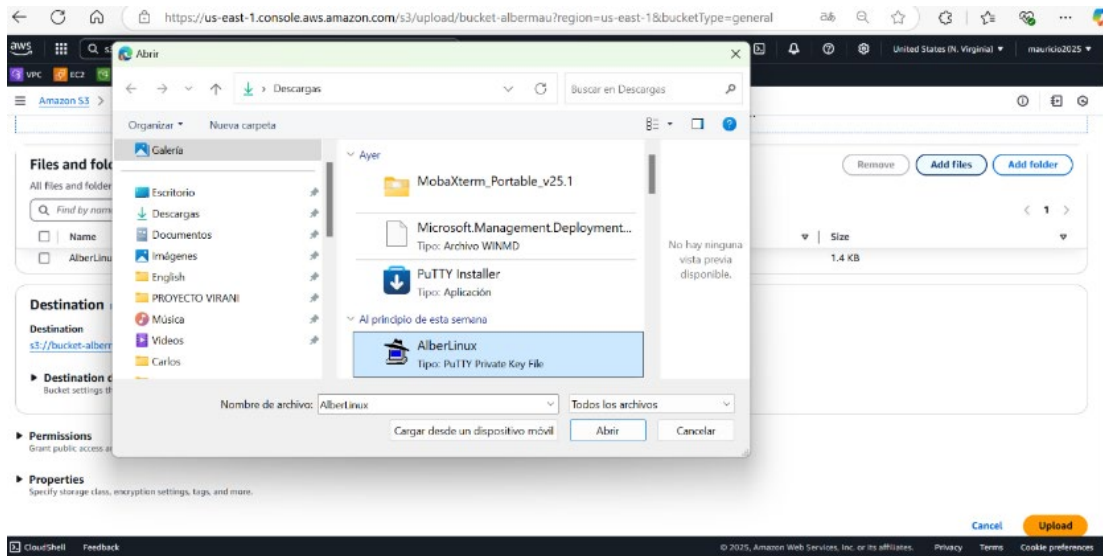
Interfaz de administración del bucket “bucket-albermau” dentro de Amazon S3, mostrando que aún no se han cargado objetos en su interior.



Nota: En este sitio de puede crear carpetas para adjuntar archivos de preferencia. Adaptado del Seminario AWS. Fuente propia, 2025.

### Ilustración 9

Selección del archivo “AlberLinux” desde el explorador de archivos de Windows para cargarlo al bucket “bucket-albermau”

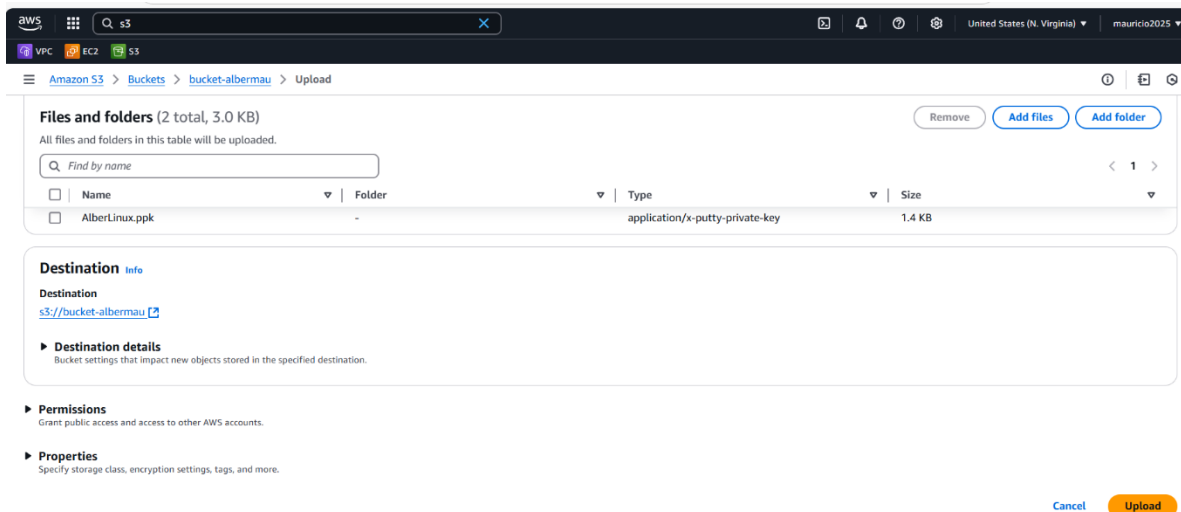


Nota: Captura de pantalla de la consola de Amazon Web Services (AWS).

Adaptado del Seminario AWS 2025.

### Ilustración 10

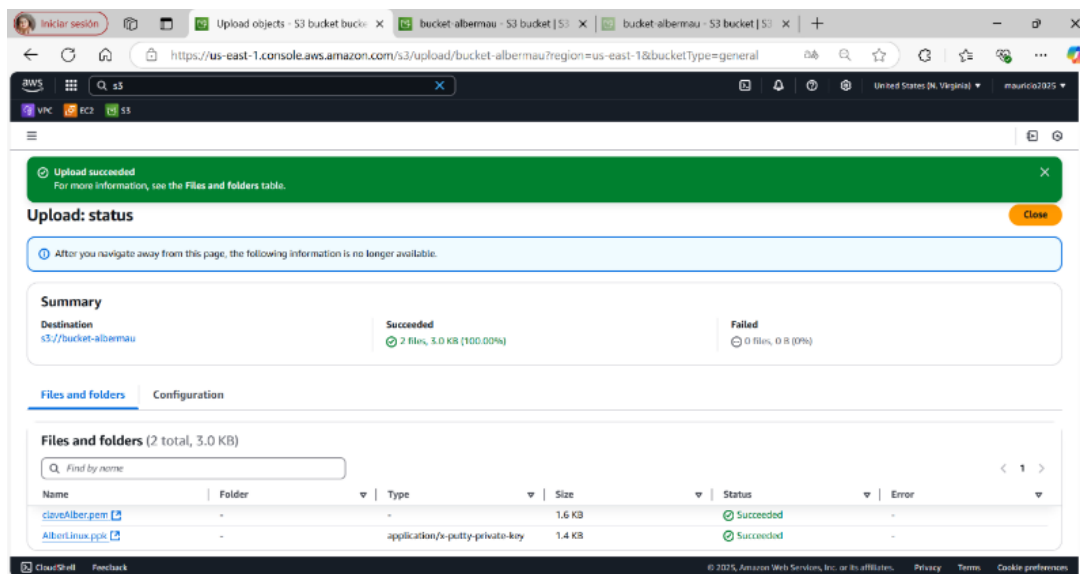
Selección del archivo “AlberLinux” desde el explorador de archivos de Windows para cargarlo al bucket



Nota: Captura de Upload para carga archivos. Fuente propia, 2025.

### Ilustración 11

Estado de carga de archivos clave en Amazon S3 utilizando la consola web de AWS.

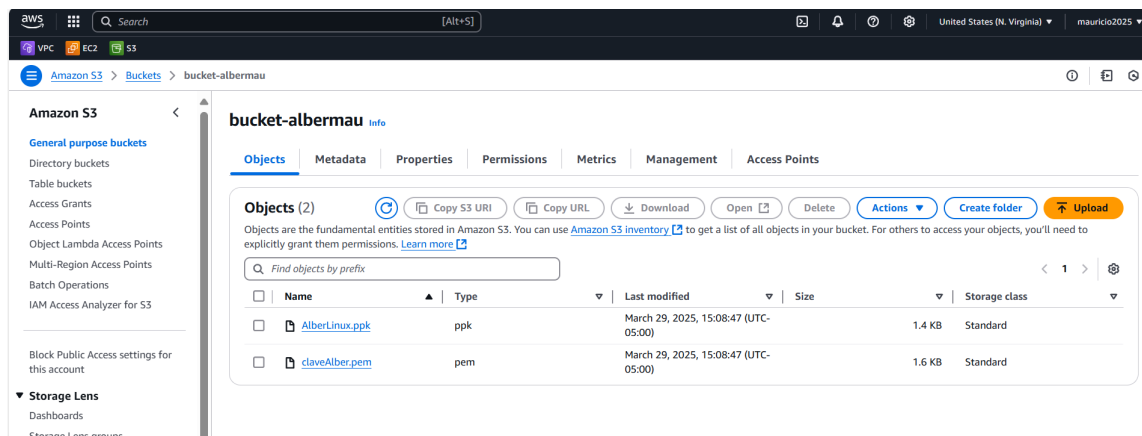


*Nota: Ingreso al bucket para ver los archivos cargados con su ultima fecha de*

*modificación. Adaptado del seminario AWS, 2025.*

### Ilustración 12

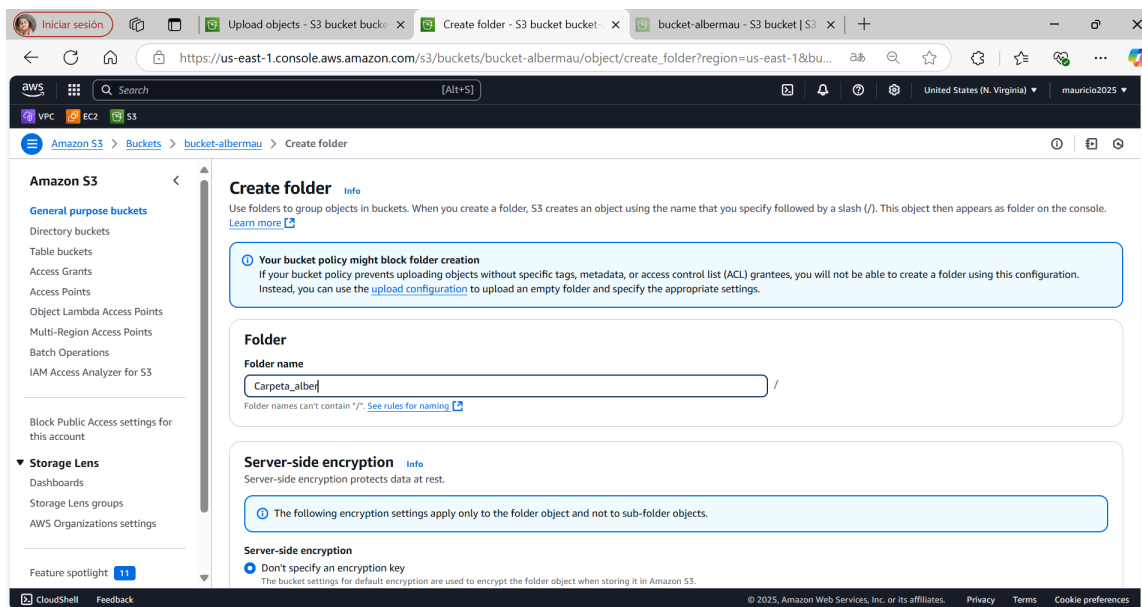
Objetos almacenados en un bucket de Amazon S3 a través de la consola web.



*Nota: En el interior del bucket se pudo crear carpetas, el nombre de la carpeta no necesariamente tiene que ser único debido a que pertenece al mismo bucket. Adaptado del Seminario AWS, 2025.*

### Ilustración 13

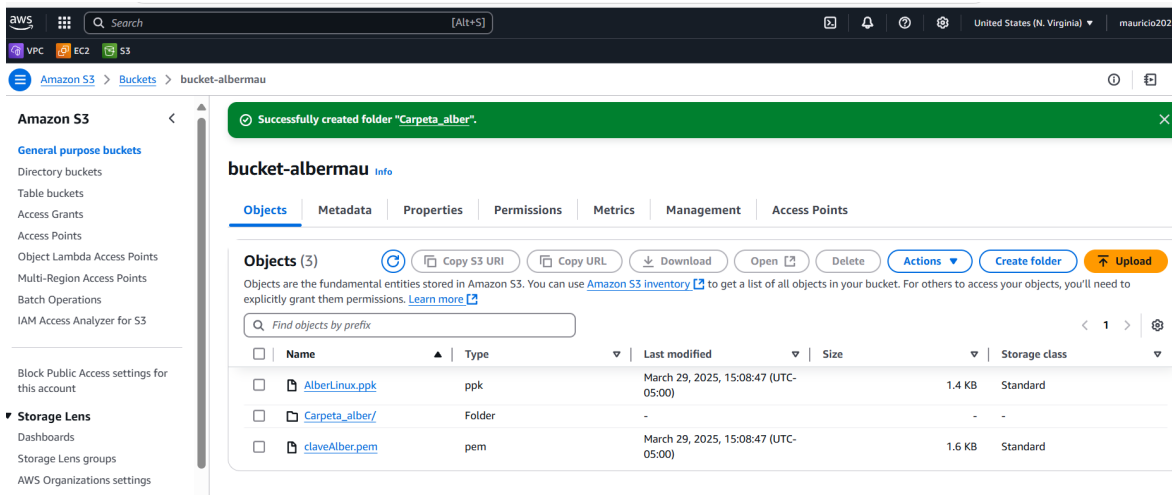
Creación de una carpeta en un bucket de Amazon S3 mediante la consola web.



*Nota: Muestra el interior del bucket se pueden crear carpetas, el nombre de la carpeta no necesariamente tiene que ser único debido a que pertenece al mismo bucket. Adaptado de Seminario AWS, Fuente propia 2025.*

### **Ilustración 14**

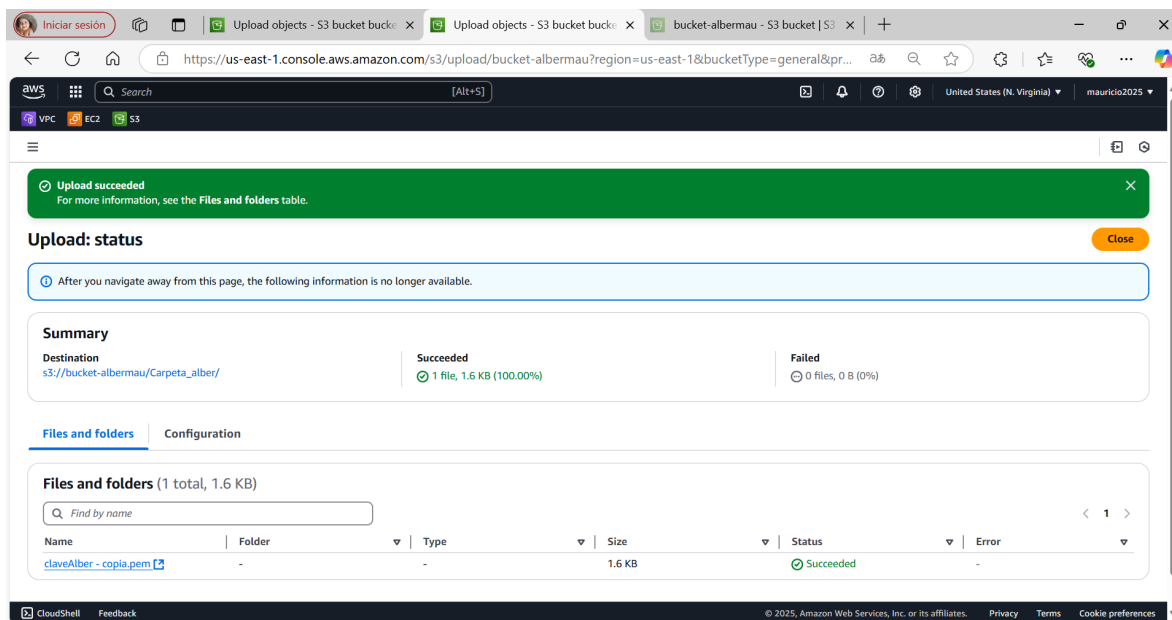
*Archivos y carpeta creada en el bucket de Amazon S3.*



*Nota: Se pueden subir archivos de preferencia, estos serán cobrados por ocupar un espacio dentro de ella. Fuente propia, 2025.*

### **Ilustración 15**

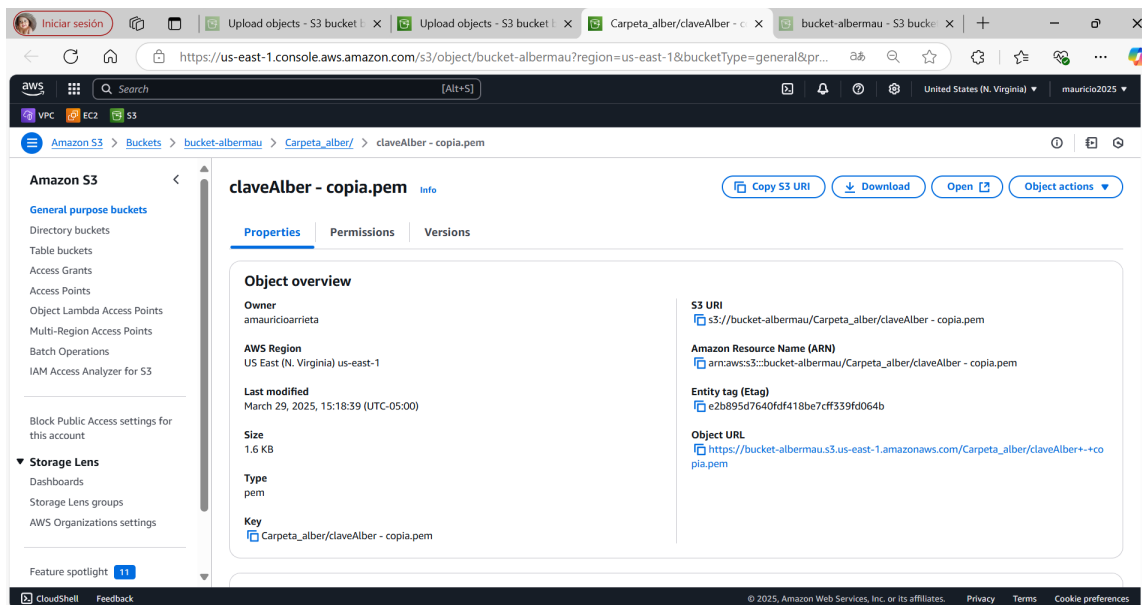
*Carga exitosa de archivo en carpeta de Amazon S3*



*Nota: Carga correcta del archivo. Fuente propia, 2025.*

## Ilustración 16

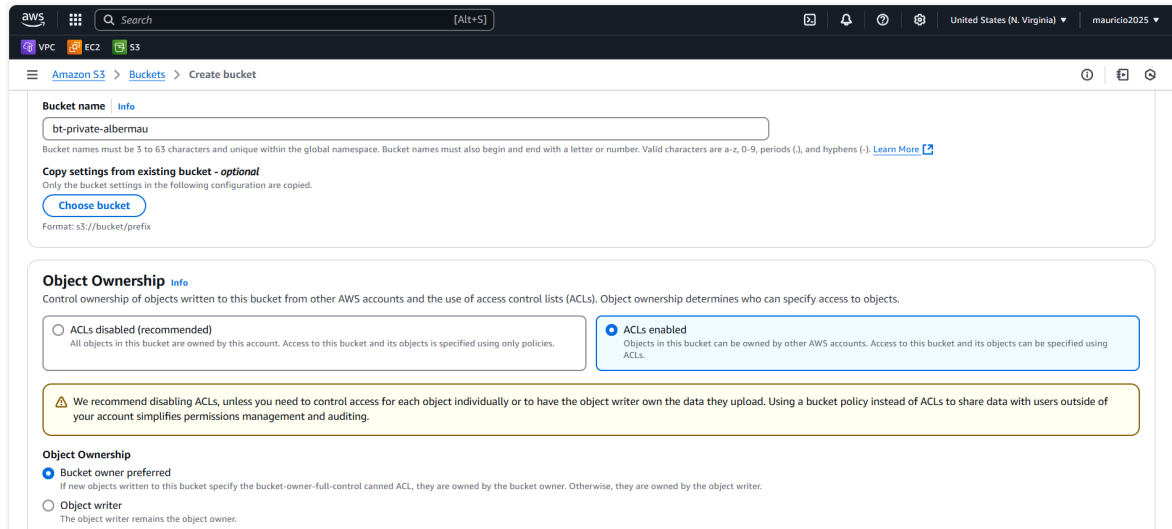
### Detalles del archivo cargado en la carpeta



*Nota: Los archivos cargados a esta carpeta obtuvieron una URL pública detallada por la ruta de la ubicación. Adaptado de Seminario AWS, 2025.*

## Ilustración 17

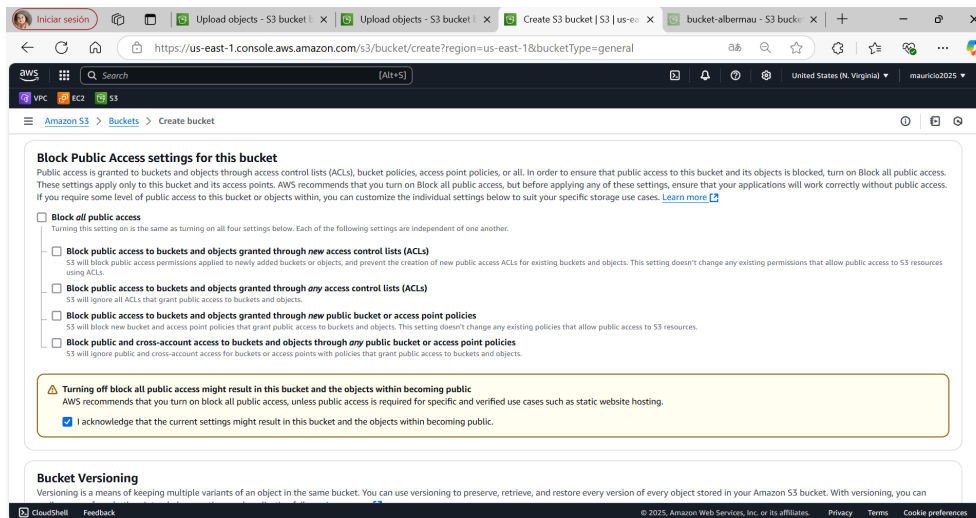
### Creación de un nuevo Bucket.



*Nota: creación de un nuevo bucket en Amazon S3, incluyendo la asignación del nombre. Adaptación de Seminario AWS, 2025.*

## Ilustración 18

### Acceso público bloqueado



*Nota: Deshabilitamos el bloqueo de acceso público, y habilitamos el versionamiento de bucket.*

## Ilustración 19

### Versionado, etiquetas y cifrado en la creación de un bucket

The screenshot displays the AWS console interface for creating a bucket. The 'Bucket Versioning' section is configured to 'Enable'. The 'Default encryption' section is set to 'Server-side encryption with Amazon S3 managed keys (SSE-S3)'. A green notification bar at the bottom states 'Successfully created bucket "bt-private-albermau"'. Below the notification, there is a table of 'General purpose buckets' with two entries: 'bt-private-albermau' and 'bucket-albermau'.

Name	AWS Region	IAM Access Analyzer	Creation date
<a href="#">bt-private-albermau</a>	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	March 29, 2025, 15:37:30 (UTC-05:00)
<a href="#">bucket-albermau</a>	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	March 29, 2025, 14:55:18 (UTC-05:00)

*Nota: Capture del cifrado en el bucket. Fuente propia, 2025.*

## Ilustración 20

### Creación de carpeta en el bucket privado

**bt-private-albermau** Info

Objects | Metadata | Properties | Permissions | Metrics | Management | Access Points

Objects (0) Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix   Show versions < 1 > ⚙️

Name	Type	Last modified	Size	Storage class
No objects You don't have any objects in this bucket.				

Upload

Abrir

« Cursos\_Universita... » SEMINARIO AWS

Organizar Nueva carpeta

Nombre	Fecha de	Contenido
1gulon	24/03/20	r de
aws-general	15/03/20	objetos
Contenedoras	29/03/20	almacenad
Crear AMI PERSONALIZADA opcion1	24/03/20	os en
Libro_TIC	20/03/20	Amazon S3
LINUX	24/03/20	Un bucket
	28/03/20	es un
		contenado
		r de
		objetos

Successfully created folder "Carpeta\_alber".

**bt-private-albermau** Info

Objects | Metadata | Properties | Permissions | Metrics | Management | Access Points

Objects (1) Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix   Show versions < 1 > ⚙️

Name	Type	Last modified	Size	Storage class
Carpeta_alber/	Folder	-	-	-

Nota:

Capture de pantalla. Fuente propia, 2025.

## Ilustración 21

### Permisos de control de acceso (ACL) del archivo Contenedor.

**Contenedoraws.txt** [Info](#) [Copy S3 URI](#) [Download](#) [Open](#) [Object actions](#)

Properties **Permissions** Versions

**Access control list (ACL)** [Edit](#)

Grant basic read/write permissions to AWS accounts. [Learn more](#)

Grantee	Object	Object ACL
<b>Object owner (your AWS account)</b> Canonical ID: <a href="#">99d2cdee4b9be0bd13aac85ceb052f7a34a5e0f263d5e576ed308a91e8223791</a>	Read	Read, Write
<b>Everyone (public access)</b> Group: <a href="#">http://acs.amazonaws.com/groups/global/AllUsers</a>	-	-
<b>Authenticated users group (anyone with an AWS account)</b> Group: <a href="#">http://acs.amazonaws.com/groups/global/AuthenticatedUsers</a>	-	-

**Edit access control list** [Info](#)

**Access control list (ACL)** [Learn more](#)

Grant basic read/write permissions to AWS accounts. [Learn more](#)

Grantee	Objects	Object ACL
<b>Object owner (your AWS account)</b> Canonical ID: <a href="#">99d2cdee4b9be0bd13aac85ceb052f7a34a5e0f263d5e576ed308a91e8223791</a>	<input checked="" type="checkbox"/> Read	<input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write
<b>Everyone (public access)</b> Group: <a href="#">http://acs.amazonaws.com/groups/global/AllUsers</a>	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Read	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Read <input type="checkbox"/> Write
<b>Authenticated users group (anyone with an AWS account)</b> Group: <a href="#">http://acs.amazonaws.com/groups/global/AuthenticatedUsers</a>	<input type="checkbox"/> Read	<input type="checkbox"/> Read <input type="checkbox"/> Write

**Warning:** When you grant access to the Everyone or Authenticated users group grantees, anyone in the world can access this object. [Learn more](#)

I understand the effects of these changes on this object.

**Access for other AWS accounts**  
No other AWS accounts associated with the resource

*Nota: Para publicar objetos se necesitaron dos condiciones; primero que el bucket sea público y segundo el archivo debe tener sus permisos de raíz o propios. Adaptación de Seminario AWS, 2025.*

## Ilustración 22

### Vista pública del archivo contenedor

Inicio sesión

Auto Scaling x Upload obj. x Upload obj. x Contenedo. x bt-private. x los bucket. x bt-private. x

<https://bt-private-albermau.s3.us-east-1.amazonaws.com/Contenedoraws.txt>

**Contenedor de objetos almacenados en Amazon S3**  
Un bucket es un contenedor de objetos almacenados en Amazon S3. Puede almacenar cualquier cantidad de objetos en un bucket y puede tener hasta 100 buckets en su cuenta. Los buckets son globales y deben tener un nombre único a través de toda la red de AWS.

*Nota: De esta manera se puede visualizar los archivos del bucket. Fuente propia, 2025.*

## Ilustración 23

### Objetos almacenados en el bucket

The screenshot shows the Amazon S3 console interface. On the left, there's a navigation menu with options like 'General purpose buckets', 'Directory buckets', 'Table buckets', etc. The main area is divided into two sections. The top section, 'General purpose buckets (2)', shows a table of buckets:

Name	AWS Region	IAM Access Analyzer	Creation date
<a href="#">bt-private-albermau</a>	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	March 29, 2025, 15:37:30 (UTC-05:00)
<a href="#">bucket-albermau</a>	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	March 29, 2025, 14:55:18 (UTC-05:00)

The bottom section, 'bucket-albermau', shows the 'Objects' tab with a table of objects:

Name	Type	Last modified	Size	Storage class
<a href="#">AlberLinux.ppk</a>	ppk	March 29, 2025, 15:08:47 (UTC-05:00)	1.4 KB	Standard
<a href="#">Carpeta_alber/</a>	Folder	-	-	-
<a href="#">claveAlber.pem</a>	pem	March 29, 2025, 15:08:47 (UTC-05:00)	1.6 KB	Standard

*Nota: Capas de Amazon s3 – store class. Adaptación del Seminario AWS, 2025.*

## Ilustración 24

### Administración de objetos en el bucket

bucket-albermau [Info](#)

The screenshot shows the 'Objects' tab for the 'bucket-albermau' bucket. The 'Carpeta\_alber/' folder is selected, and the context menu is open, showing various actions. The 'Edit storage class' option is highlighted, indicating that the storage class can be changed for this object.

*Nota: Se destaca la posibilidad de "Edit storage class" (Editar clase de almacenamiento), lo que indica que se puede cambiar la clase de almacenamiento de este objeto a otra, como Standard. Adaptación del Seminario AWS, 2025.*

## Ilustración 25

### Edición de la clase de almacenamiento de un objeto en el bucket

#### Edit storage class [Info](#)

This action creates a copy of the object with updated settings and a new last-modified date. You can change the storage class without making a new copy of the object using a [lifecycle rule](#). [View copy restrictions and limitations](#)

Storage class	Designed for	Availability Zones	Min storage duration	Min billable object size	Monitoring and auto-tiering fees	Retrieval fees
<input type="radio"/> Standard	Frequently accessed data (more than once a month) with milliseconds access	≥ 3	-	-	-	-
<input checked="" type="radio"/> Intelligent-Tiering	Data with changing or unknown access patterns	≥ 3	-	-	Per-object fees apply for objects >= 128 KB	-
<input type="radio"/> One Zone-IA	Infrequently accessed data (once a month)	1	30 days	128 KB	-	Per-GB fees

*Nota: Intelligent-Tiering, Standard-IA, One Zone-IA o Glacier, dependiendo de la necesidad.*

## Ilustración 26

### Creación de bucket para sitio público

#### Create bucket [Info](#)

Buckets are containers for data stored in S3.

#### General configuration

**AWS Region**  
US East (N. Virginia) us-east-1

**Bucket type** [Info](#)

**General purpose**  
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

**Directory**  
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

**Bucket name** [Info](#)  
web-site-alber

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). [Learn More](#)

**Copy settings from existing bucket - optional**  
Only the bucket settings in the following configuration are copied.

Format: s3://bucket/prefix

#### Object Ownership [Info](#)

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

**ACLs disabled (recommended)**  
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

**ACLs enabled**  
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

**⚠ We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.**

**Object Ownership**

**Bucket owner preferred**  
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

**Object writer**  
The object writer remains the object owner.

**ℹ If you want to enforce object ownership for new objects only, your bucket policy must specify that the bucket-owner-full-control canned ACL is required for object uploads. [Learn more](#)**

*Nota: Capture de pantalla. Fuente propia, 2025.*

## Ilustración 27

### Bloqueo de acceso público en el bucket

**Block Public Access settings for this bucket**

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

**Block all public access**  
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through new access control lists (ACLs)**  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**  
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

**⚠ Turning off block all public access might result in this bucket and the objects within becoming public**  
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

*Nota: Se **Bloquea todo el acceso público**: No está activado. Si se activa, se aplicarán todas las restricciones de acceso público. Adaptación del Seminario AWS, 2025.*

## Ilustración 28

### Disposición de Versionado y Etiquetas en el Bucket

**Bucket Versioning**

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore easily recover from both unintended user actions and application failures. [Learn more](#)

**Bucket Versioning**

Disable

Enable

---

**Tags - optional (0)**

You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.

[Add tag](#)

*Nota: **Bucket Versioning (Versionado de Bucket)**. **Habilitado (Enabled)**: Esto significa que se guardarán múltiples versiones de los objetos dentro del bucket. Adaptación del Seminario AWS, 2025.*

## Ilustración 29

### Configuración de Static Website Hosting en el Bucket

The screenshot shows the AWS S3 console interface for the bucket 'web-site-alber'. The left sidebar contains navigation options like 'General purpose buckets', 'Directory buckets', 'Table buckets', 'Access Grants', 'Access Points', 'Object Lambda Access Points', 'Multi-Region Access Points', 'Batch Operations', 'IAM Access Analyzer for S3', 'Block Public Access settings for this account', 'Storage Lens', 'Dashboards', 'Storage Lens groups', and 'AWS Organizations settings'. The main content area is divided into several sections:

- Bucket overview:** Shows 'AWS Region' as 'US East (N. Virginia) us-east-1', 'Amazon Resource Name (ARN)' as 'arn:aws:s3::web-site-alber', and 'Creation date' as 'March 29, 2025, 18:28:31 (UTC-05:00)'.
- Static website hosting:** Includes an 'Edit' button and a recommendation to use 'AWS Amplify Hosting for static website hosting'. A 'Create Amplify app' button is also present.
- S3 static website hosting:** Currently set to 'Disabled'.

*Nota: Para alojar un sitio estático en S3, se debió editar esta configuración y habilitar la opción de "Static website hosting". Adaptación del Seminario AWS, 2025.*

## Ilustración 30

### Carga de Archivos y Carpetas - Control de Acceso (ACL) al Cargar archivos en el bucket.

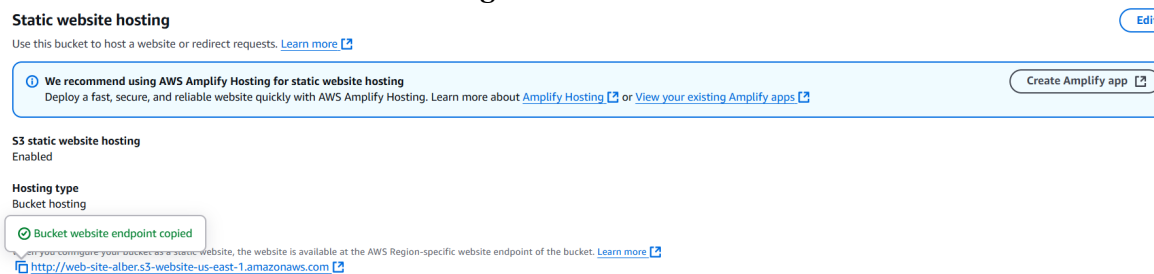
The screenshot shows the 'Upload' screen in the AWS S3 console. It includes the following elements:

- Upload Info:** Instructions on how to upload files and folders, with a 'Learn more' link.
- Files and folders (77 total, 3.7 MB):** A table listing files and folders to be uploaded. The table has columns for Name, Folder, Type, and Size.
- Access control list (ACL):** A section for configuring permissions. It includes a warning: 'AWS recommends using S3 bucket policies or IAM policies for access control.' Below this, there are options to 'Choose from predefined ACLs' or 'Specify individual ACL permissions'. The 'Predefined ACLs' section shows 'Private (recommended)' and 'Grant public-read access' (which is selected).
- Warning:** A yellow box with a warning icon stating 'Granting public-read access is not recommended' and a checkbox 'I understand the risk of granting public-read access to the specified objects.' which is checked.
- Properties:** A section for specifying storage class, encryption settings, tags, and more.
- Buttons:** 'Cancel' and 'Upload' buttons at the bottom right.

*Nota: Después de configurado el sitio web estático. Cargamos los archivos en formato html y editamos el acceso público. Adaptación del Seminario AWS, 2025.*

### Ilustración 31

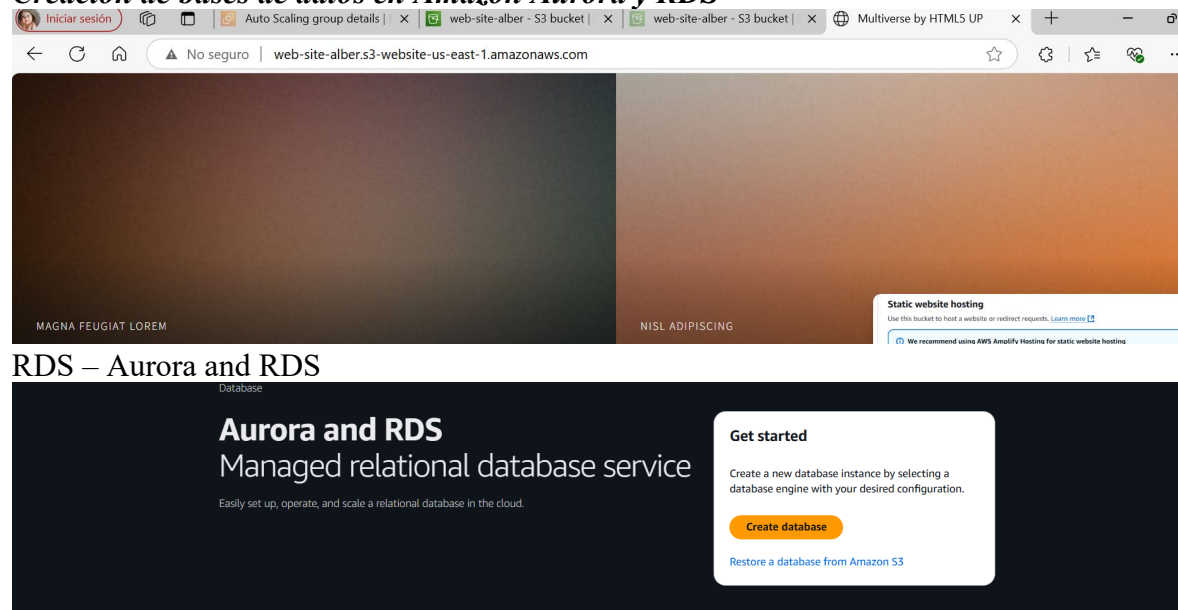
#### Habilitación de Static Website Hosting en Amazon



*Nota: Se muestra la configuración de **hosting de sitios web estáticos en Amazon S3**. Estado actual: **Habilitado**. El Tipo de hosting: **Bucket hosting** (el sitio web se sirve directamente desde el bucket de S3). El Endpoint del sitio web, se ha generado un enlace para acceder al sitio web alojado en S3. Adaptación del Seminario AWS, 2025.*

### Ilustración 32

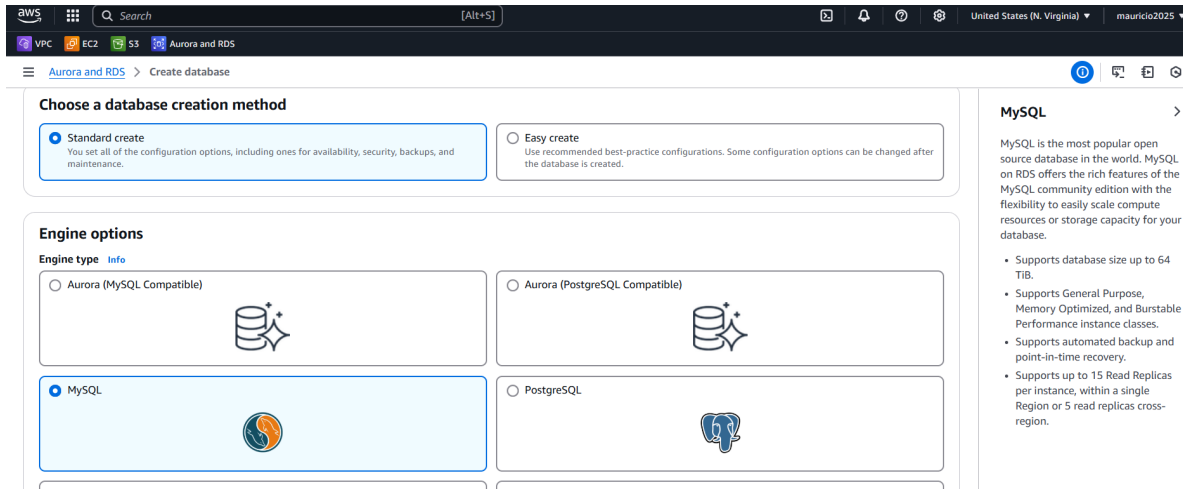
#### Creación de bases de datos en Amazon Aurora y RDS



*Nota: Aquí se muestra la configuración de **creación de una base de datos en Amazon RDS con AWS Aurora (PostgreSQL Compatible)** seleccionado como motor de base de datos. Adaptación del Seminario AWS, 2025.*

### Ilustración 33

#### Selección del motor de base de datos en el bucket.



*Nota: "Standard create", permite personalizar opciones como disponibilidad, seguridad, backups y mantenimiento. Adaptación del Seminario AWS, 2025.*

## Ilustración 34

### Opciones de disponibilidad y durabilidad en la creación de bases de datos

Choose a sample template to meet your use case.

**Production**  
Use defaults for high availability and fast, consistent performance.

**Dev/Test**  
This instance is intended for development use outside of a production environment.

**Free tier**  
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. [Info](#)

**MySQL** >

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

**Availability and durability**

**Deployment options** [Info](#)

Choose the deployment option that provides the availability and durability needed for your use case. AWS is committed to a certain level of uptime depending on the deployment option you choose. Learn more in the [Amazon RDS service level agreement \(SLA\)](#).

**Multi-AZ DB cluster deployment (3 instances)**  
Creates a primary DB instance with two readable standbys in separate Availability Zones. This setup provides:
 

- 99.95% uptime
- Redundancy across Availability Zones
- Increased read capacity
- Reduced write latency

**Multi-AZ DB instance deployment (2 instances)**  
Creates a primary DB instance with a non-readable standby instance in a separate Availability Zone. This setup provides:
 

- 99.95% uptime
- Redundancy across Availability Zones

**Single-AZ DB instance deployment (1 instance)**  
Creates a single DB instance without standby instances. This setup provides:
 

- 99.5% uptime
- No data redundancy

*Nota: La configuración de disponibilidad y durabilidad en Amazon RDS, con diferentes opciones de despliegue para la base de datos. Se ha elegido “Free Tier”, porque permite usar RDS gratuitamente con ciertas limitaciones. Adaptación del Seminario AWS, 2025.*

## Ilustración 35

### Opciones de conectividad para una base de datos

**Connectivity** [Info](#) ⓘ

**Compute resource**

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

**Don't connect to an EC2 compute resource**  
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

**Connect to an EC2 compute resource**  
Set up a connection to an EC2 compute resource for this database.

**Virtual private cloud (VPC)** [Info](#)

Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

VPC-WEB-Alber-vpc (vpc-07bca697f1b5b5322)  
4 Subnets, 2 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

ⓘ After a database is created, you can't change its VPC.

**DB subnet group** [Info](#)

Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

Create new DB Subnet Group

**Public access** [Info](#)

**Yes**  
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

**No**  
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which

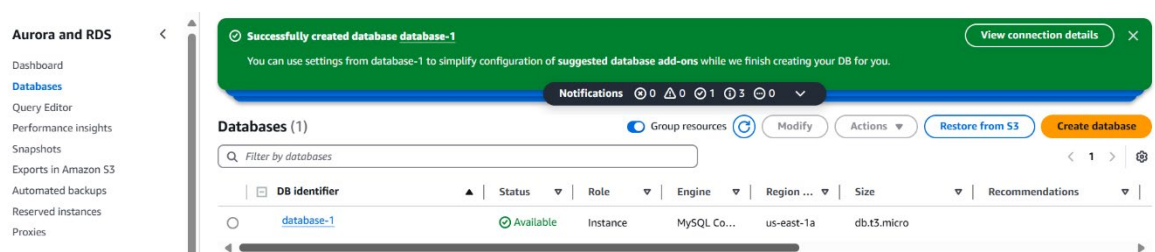
ⓘ You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

Cancel Create database

**Nota:** En connectivity, se ha seleccionado "**Don't connect to an EC2 compute resource**", lo que significa que la base de datos no estará vinculada directamente a una instancia EC2. Se puede conectarse manualmente más adelante si se requiere. Adaptación del Seminario AWS, 2025.

### Ilustración 36

#### Validación correcta



**Nota:** Instancia de base de datos en Amazon **RDS**. **Publicly accessible:** Yes, La base de datos tiene acceso público, lo que significa que puede ser accesible desde fuera de la VPC). Adaptación del Seminario AWS, 2025.

### Ilustración 37

#### Infraestructura en AWS

**Aurora and RDS**

- Dashboard
- Databases
- Query Editor
- Performance insights
- Snapshots
- Exports in Amazon S3
- Automated backups
- Reserved instances
- Proxies
- Subnet groups
- Parameter groups
- Option groups
- Custom engine versions
- Zero-ETL integrations [New](#)
- Events

**Connectivity & security**

**Endpoint & port**

Endpoint: [database-1.c0xgm0s0e4d4gus-east-1.rds.amazonaws.com](#)

Port: 3306

**Networking**

Availability Zone: us-east-1a

VPC: [VPC-WEB-Alber-vpc \(vpc-07bca697f1b5b5322\)](#)

Subnet group: [default-vpc-07bca697f1b5b5322](#)

Subnets: [subnet-0af5ca416c0e1dadd](#), [subnet-0a4af0cf108f803e2](#), [subnet-07c131a7bdcc978a6](#), [subnet-04ca1c5f4ecc20c4](#)

Network type: IPv4

**Security**

VPC security groups: [default \(sg-0929855eb0ec6af61\)](#) [Active](#)

Publicly accessible: Yes

Certificate authority: [Info](#) [rds-ca-rsa2048-g1](#)

Certificate authority date: May 25, 2061, 18:34 (UTC-05:00)

DB instance certificate expiration date: March 29, 2026, 19:49 (UTC-05:00)

**Inbound rules (2)**

Search:

<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	Port range
<input type="checkbox"/>	-	sgr-06f6507010c245782	IPv4	HTTP	TCP	80
<input type="checkbox"/>	-	sgr-0294a9ae66be33d1d	-	All traffic	All	All

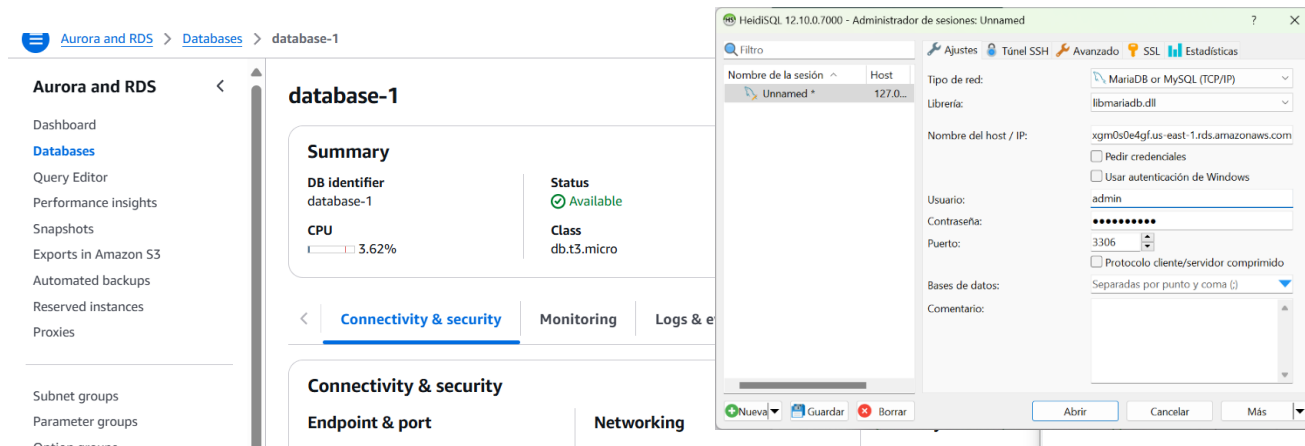
Custom TCP | TCP | 3306 | Anyw... |  |

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

*Nota: Se editaron las reglas de entrada (Inbound rules) del grupo de seguridad asociado a tu infraestructura en AWS, específicamente agregando el puerto 3306. Adaptación del Seminario AWS, 2025.*

### Ilustración 38

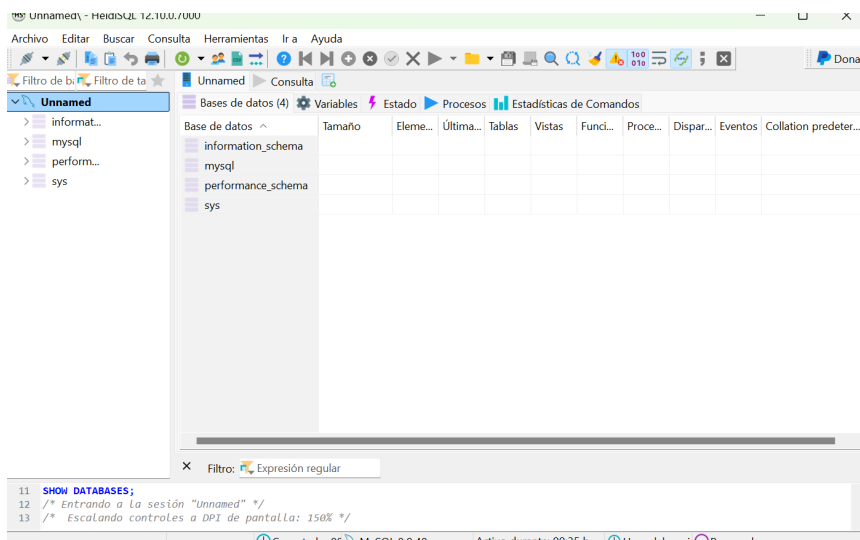
#### Administrador de sesiones HeidiSQL



*Nota: Esta imagen muestra HeidiSQL conectado a una base de datos MySQL. Adaptación del Seminario AWS, 2025.*

### Ilustración 39

#### Creación del Docker



*Nota: Detenemos la aplicación httpd. Adaptación del Seminario AWS, 2025.*

## Ilustración 40

### Inicio de Sesión en la aplicación mobaXterm.

The screenshot shows the MobaXterm interface with a terminal window. The terminal output is as follows:

```

Last login: Mon Mar 24 14:22:13 2025 from 181.119.64.122
[ec2-user@ip-10-0-8-12 ~]$ sudo su
[root@ip-10-0-8-12 ec2-user]# dnf remove httpd
Dependencies resolved.

Package Architecture Version Repository Size
-----
Removing:
httpd x86_64 2.4.62-1.amzn2023 @amazonlinux 61 k
Removing unused dependencies:
apr x86_64 1.7.5-1.amzn2023.0.4 @amazonlinux 299 k
apr-util x86_64 1.6.3-1.amzn2023.0.1 @amazonlinux 217 k
apr-util-openssl x86_64 1.6.3-1.amzn2023.0.1 @amazonlinux 24 k
generic-logos-httpd noarch 18.0.0-12.amzn2023.0.3 @amazonlinux 21 k
httpd-core x86_64 2.4.62-1.amzn2023 @amazonlinux 4.7 M
httpd-filessystem noarch 2.4.62-1.amzn2023 @amazonlinux 464
httpd-tools x86_64 2.4.62-1.amzn2023 @amazonlinux 201 k
libbrotli x86_64 1.0.9-4.amzn2023.0.2 @amazonlinux 771 k
mailcap noarch 2.1.49-3.amzn2023.0.3 @amazonlinux 78 k
mod_http2 x86_64 2.0.27-1.amzn2023.0.3 @amazonlinux 444 k
mod_lua x86_64 2.4.62-1.amzn2023 @amazonlinux 143 k

Transaction Summary
-----
Remove 12 Packages

Freed space: 6.9 M
Is this ok [y/N]:
  
```

Below this, the terminal shows the removal of the packages:

```

Removed:
apr-1.7.5-1.amzn2023.0.4.x86_64          apr-util-1.6.3-1.amzn2023.0.1.x86_64
apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64  generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
httpd-2.4.62-1.amzn2023.x86_64          httpd-core-2.4.62-1.amzn2023.x86_64
httpd-filessystem-2.4.62-1.amzn2023.noarch  httpd-tools-2.4.62-1.amzn2023.x86_64
libbrotli-1.0.9-4.amzn2023.0.2.x86_64    mailcap-2.1.49-3.amzn2023.0.3.noarch
mod_http2-2.0.27-1.amzn2023.0.3.x86_64  mod_lua-2.4.62-1.amzn2023.x86_64

Complete!
[root@ip-10-0-8-12 ec2-user]#
  
```

Next, the terminal shows the installation of Docker:

```

Complete!
[root@ip-10-0-8-12 ec2-user]# yum install docker
Last metadata expiration check: 16:46:32 ago on Sat Mar 29 21:52:23 2025.
Dependencies resolved.

Package Architecture Version Repository Size
-----
Installing:
docker x86_64 25.0.8-1.amzn2023.0.1 amazonlinux 44
Installing dependencies:
containerd x86_64 1.7.25-1.amzn2023.0.1 amazonlinux 36
iptables-libs x86_64 1.8.8-3.amzn2023.0.2 amazonlinux 401
iptables-nft x86_64 1.8.8-3.amzn2023.0.2 amazonlinux 183
libcgroup x86_64 3.0-1.amzn2023.0.1 amazonlinux 75
libnetfilter_conntrack x86_64 1.0.8-2.amzn2023.0.2 amazonlinux 58
libnftnl x86_64 1.0.1-19.amzn2023.0.2 amazonlinux 30
libnftnl x86_64 1.2.2-2.amzn2023.0.2 amazonlinux 84
pigz x86_64 2.5-1.amzn2023.0.3 amazonlinux 83
runc x86_64 1.2.4-1.amzn2023.0.1 amazonlinux 3.4

Transaction Summary
-----
Install 10 Packages

Total download size: 84 M
Installed size: 319 M
Is this ok [y/N]:
  
```

Nota:

Instalación el Docker implementando el comando “yum install docker + enter”. Adaptado Seminario AWS 2025.

## Ilustración 41

### Instalación de paquetes y contenedores

```

[ec2-34-230-8-115.compute-1.amazonaws.com (ec2-user)]
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

Quick connect...
/home/ec2-user/
Name Size (KB)
. 1
.ssh 1
.bash_history 1
.bash_logout 1
.bash_profile 1
.bashrc 1
alberaws 1
alberawsCopia 1
file1 0
file2 0

Remote monitoring
Follow terminal folder

[ec2-34-230-8-115.compute-1.amazonaws.com]
[ec2-34-230-8-115.compute-1.amazonaws.com]# yum install docker.io
Last metadata expiration check: 16:51:29 ago on Sat Mar 29 21:52:23 2025.
No match for argument: docker.io
Error: Unable to find a match: docker.io
[ec2-34-230-8-115.compute-1.amazonaws.com]# yum update
Last metadata expiration check: 16:52:29 ago on Sat Mar 29 21:52:23 2025.

=====
WARNING:
A newer release of "Amazon Linux" is available.

Available Versions:

Version 2023.6.20250317:
Run the following command to upgrade to 2023.6.20250317:

dnf upgrade --releasever=2023.6.20250317

Release notes:
https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.6.20250317.html
=====

Dependencies resolved.
Nothing to do.
Complete!
[ec2-34-230-8-115.compute-1.amazonaws.com]# systemctl status docker
○ docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; preset: disabled)
   Active: inactive (dead)
   TriggeredBy: ○ docker.socket
   Docs: https://docs.docker.com
[ec2-34-230-8-115.compute-1.amazonaws.com]# ^C
[ec2-34-230-8-115.compute-1.amazonaws.com]# systemctl status docker

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

```

[ec2-34-230-8-115.compute-1.amazonaws.com]# systemctl status docker
○ docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; preset: disabled)
   Active: active (running) since Sun 2025-03-30 14:48:20 UTC; 35s ago
   TriggeredBy: ○ docker.socket
   Docs: https://docs.docker.com
   Process: 487422 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
   Process: 487423 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0)
   Main PID: 487424 (dockerd)
   Tasks: 8
   Memory: 41.9M
   CPU: 283ms
   CGroup: /system.slice/docker.service
           └─487424 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-

Mar 30 14:48:19 ip-10-0-8-12.ec2.internal systemd[1]: Starting docker.service - Docker Application Container Engine:
Mar 30 14:48:19 ip-10-0-8-12.ec2.internal dockerd[487424]: time="2025-03-30T14:48:19.995202263Z" level=info msg="Starting
Mar 30 14:48:20 ip-10-0-8-12.ec2.internal dockerd[487424]: time="2025-03-30T14:48:20.122484926Z" level=info msg="API server
Mar 30 14:48:20 ip-10-0-8-12.ec2.internal dockerd[487424]: time="2025-03-30T14:48:20.571247940Z" level=info msg="Listening
Mar 30 14:48:20 ip-10-0-8-12.ec2.internal dockerd[487424]: time="2025-03-30T14:48:20.595063363Z" level=info msg="Daemon
Mar 30 14:48:20 ip-10-0-8-12.ec2.internal dockerd[487424]: time="2025-03-30T14:48:20.595276824Z" level=info msg="Daemon
Mar 30 14:48:20 ip-10-0-8-12.ec2.internal dockerd[487424]: time="2025-03-30T14:48:20.647696354Z" level=info msg="Daemon
Mar 30 14:48:20 ip-10-0-8-12.ec2.internal systemd[1]: Started docker.service - Docker Application Container Engine:
Lines 1-22/22 (END)

```

Nota:

```

[ec2-34-230-8-115.compute-1.amazonaws.com]# systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[ec2-34-230-8-115.compute-1.amazonaws.com]#

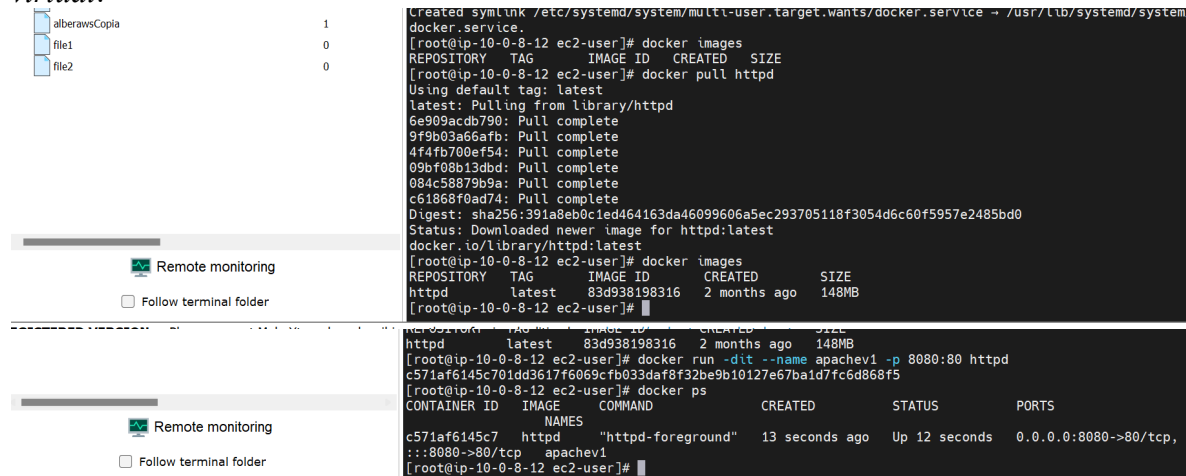
```

Esta imagen muestra la instalación de un segundo paquete mediante el comando `– yum install docker.io`, si está instalado por defecto, se verifica sus estados con la siguiente línea `systemctl status docker`, y lo iniciamos con el comando `systemctl start docker`. Al final nos muestra que la aplicación o el motor se está ejecutando.

Digitamos el comando `-systemctl enable docker`, para que el motor inicie de manera automática

## Ilustración 42

Utilización del comando `-docker images-` para visualizar las imágenes que contiene la maquina virtual.

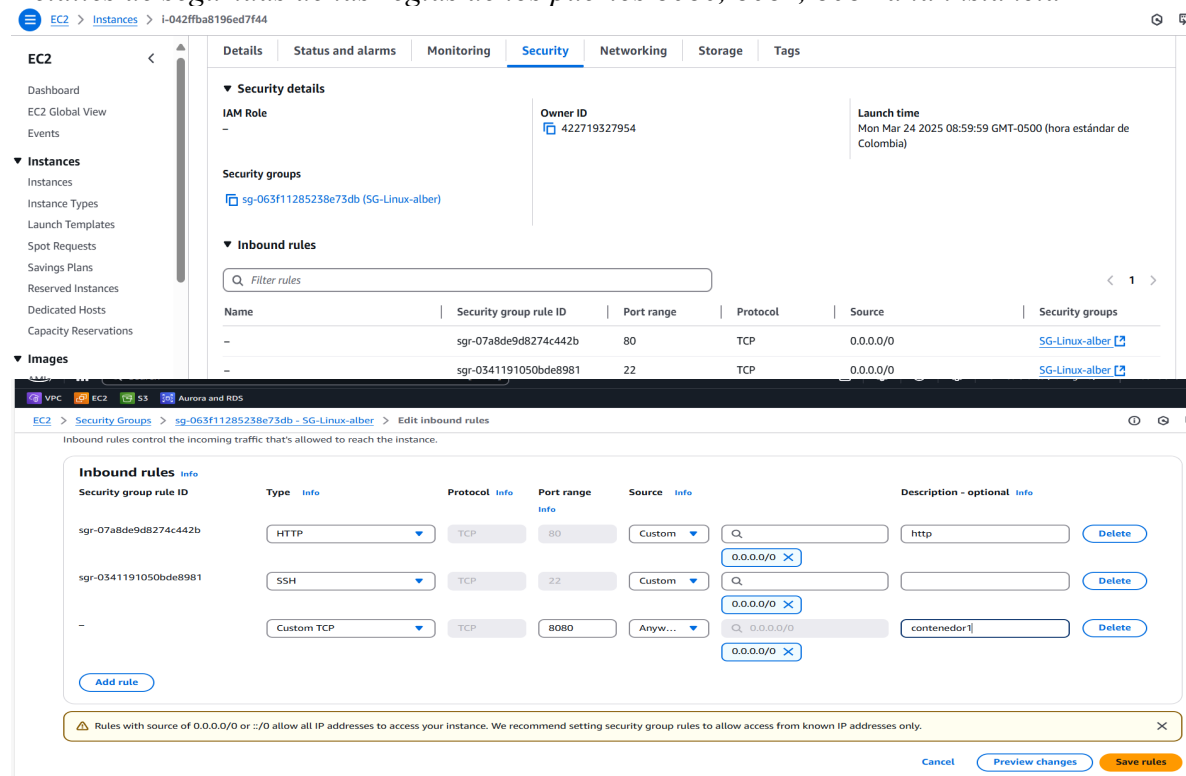


```
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[root@ip-10-0-8-12 ec2-user]# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
[root@ip-10-0-8-12 ec2-user]# docker pull httpd
Using default tag: latest
latest: Pulling from library/httpd
6e909acdb790: Pull complete
9f9b03a66afb: Pull complete
4f4fb700ef54: Pull complete
09bf08b13dbd: Pull complete
084c58879b9a: Pull complete
c61868f0ad74: Pull complete
Digest: sha256:391a8eb0c1ed464163da46099606a5ec293705118f3054d6c60f5957e2485bd0
Status: Downloaded newer image for httpd:latest
docker.io/library/httpd:latest
[root@ip-10-0-8-12 ec2-user]# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
httpd latest 83d938198316 2 months ago 148MB
[root@ip-10-0-8-12 ec2-user]# docker run --dit --name apachev1 -p 8080:80 httpd
c571af6145c701dd3617f6069c fb033daf8f32be9b10127e67ba1d7fc6d868f5
[root@ip-10-0-8-12 ec2-user]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
c571af6145c7 httpd "httpd-foreground" 13 seconds ago Up 12 seconds 0.0.0.0:8080->80/tcp,
:::8080->80/tcp apachev1
[root@ip-10-0-8-12 ec2-user]#
```

Nota: En esta se Inserta el comando `-docker run --dit --name apachev1 -p 8080:80 httpd-` para que el docker funcione en segundo plano. También el comando `-docker ps-` para ver si el docker está funcionando.

## Ilustración 43

Detalles de seguridad de las reglas de los puertos 8080, 8081, 8082 a la instancia



The screenshot shows the AWS Management Console interface for an EC2 instance. The 'Security' tab is selected, and the 'Inbound rules' section is expanded. The following table represents the data shown in the 'Inbound rules' table:

Name	Security group rule ID	Port range	Protocol	Source	Security groups
-	sgr-07a8de9d8274c442b	80	TCP	0.0.0.0/0	SG-Linux-alber
-	sgr-0341191050bde8981	22	TCP	0.0.0.0/0	SG-Linux-alber
-	-	8080	Custom TCP	0.0.0.0/0	contenedor1

Below the table, the 'Edit inbound rules' form is visible, showing the configuration for the custom rule on port 8080:

- Security group rule ID:** sgr-07a8de9d8274c442b
- Type:** HTTP
- Protocol:** TCP
- Port range:** 80
- Source:** Custom (0.0.0.0/0)
- Description - optional:** http

At the bottom, there is a warning message: "Rules with source of 0.0.0.0/0 or :::0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only."

**Ilustración 44**

Creación y funcionamiento de los contenedores.

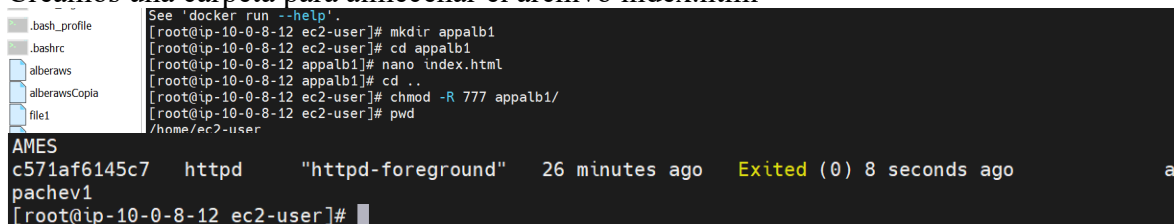
Los



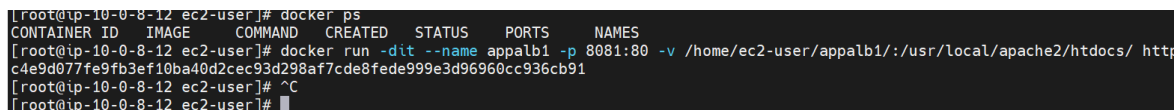
**It works!**

comando como: `-docker stop + ID del contenedor` nos sirve para detener el contenedor. El `-docker ps -a` nos muestra los contenedores creados y estado de este.

Creamos una carpeta para almacenar el archivo `index.html`



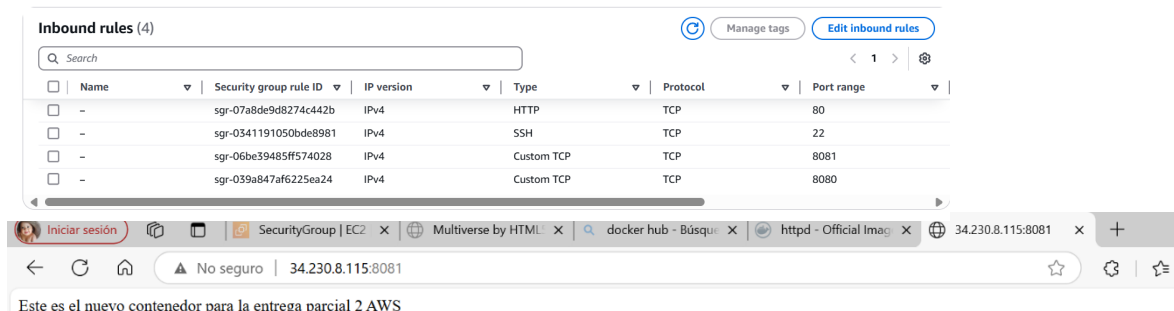
Creamos el contenedor con el comando `docker run -dit --name appalb1 -p 8081:80 -v /home/ec2-user/appalb1:/usr/local/apache2/htdocs/ httpd`. Esto hará que Apache sirva el contenido de `index.html` en `http://<IP-EC2>:8081`



*Nota: Capture de pantalla sobre creación de contenedores docker mediante comandos y la aplicación.*

**Ilustración 45**

Añadimos la regla del puerto 8081 en la instancia, editamos seguridad



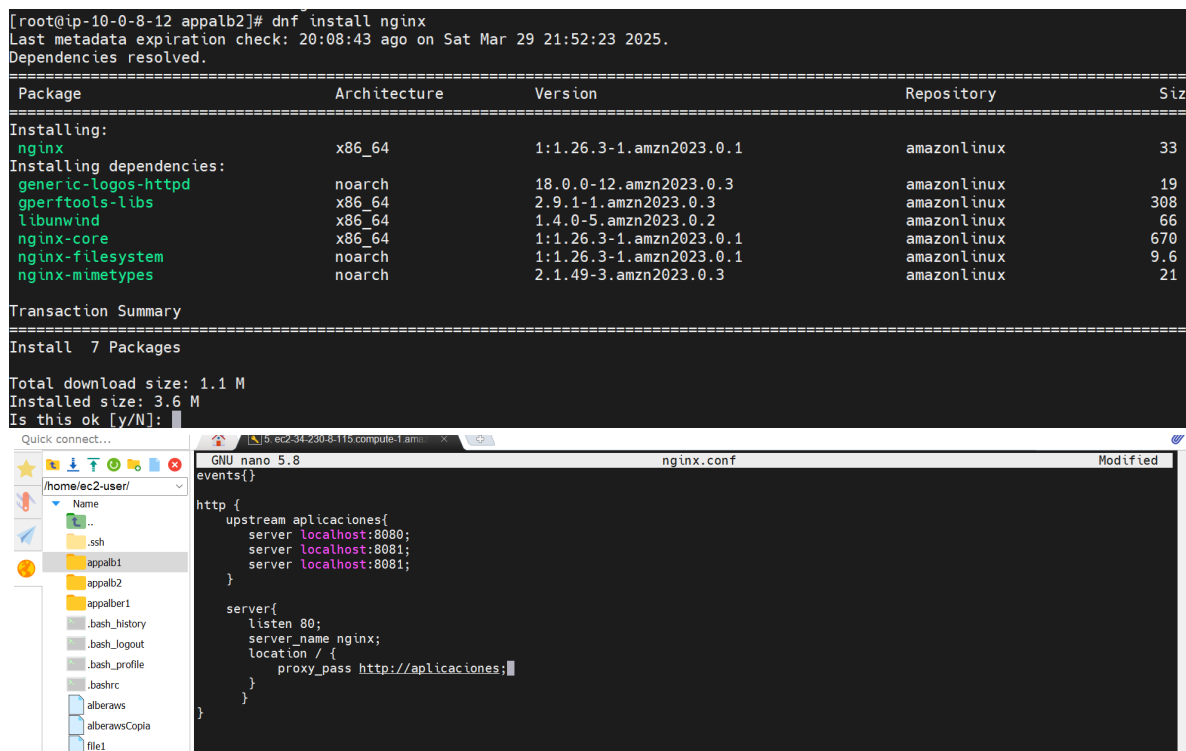
*Nota: Capture de pantalla de resultados del seminario AWS 2025. Fuente propia.*

## Ilustración 46

Instalacion de la aplicación nginx en la instancia AWS ECS con mobaXterm

```
[root@ip-10-0-8-12 appalb2]# dnf install nginx
Last metadata expiration check: 20:08:43 ago on Sat Mar 29 21:52:23 2025.
Dependencies resolved.
=====
Package                Architecture      Version           Repository        Size
=====
Installing:
nginx                  x86_64           1:1.26.3-1.amzn2023.0.1  amazonlinux      33
Installing dependencies:
generic-logos-httpd   noarch           18.0.0-12.amzn2023.0.3  amazonlinux      19
gperftools-libs       x86_64           2.9.1-1.amzn2023.0.3    amazonlinux      308
libunwind              x86_64           1.4.0-5.amzn2023.0.2    amazonlinux      66
nginx-core             x86_64           1:1.26.3-1.amzn2023.0.1  amazonlinux      670
nginx-filesystem      noarch           1:1.26.3-1.amzn2023.0.1  amazonlinux      9.6
nginx-mimetypes       noarch           2.1.49-3.amzn2023.0.3    amazonlinux      21
=====
Transaction Summary
=====
Install 7 Packages

Total download size: 1.1 M
Installed size: 3.6 M
Is this ok [y/N]:
```



```
GNU nano 5.8 nginx.conf Modified
events{}

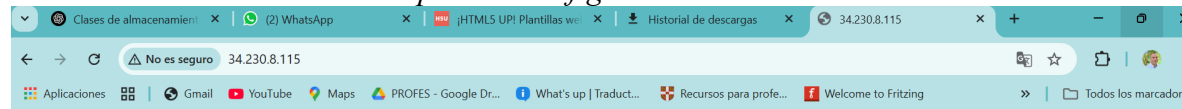
http {
    upstream aplicaciones{
        server localhost:8080;
        server localhost:8081;
        server localhost:8081;
    }

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

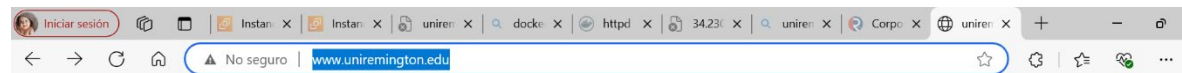
*Nota: Luego editamos eun archivo de configuración de Nginx, que se usa como un balanceador de carga para distribuir las solicitudes HTTP entre varios servidores backend o aplicaciones.*

### Ilustración 47

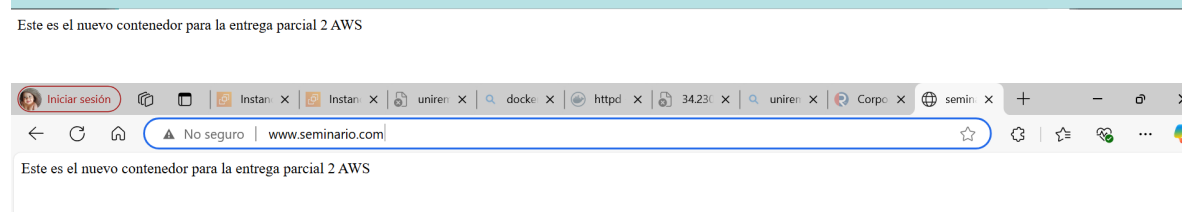
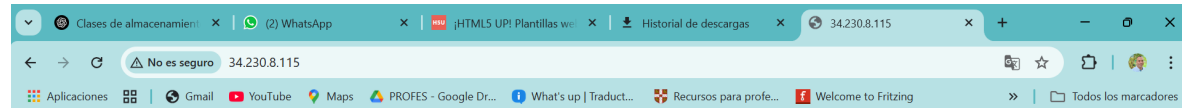
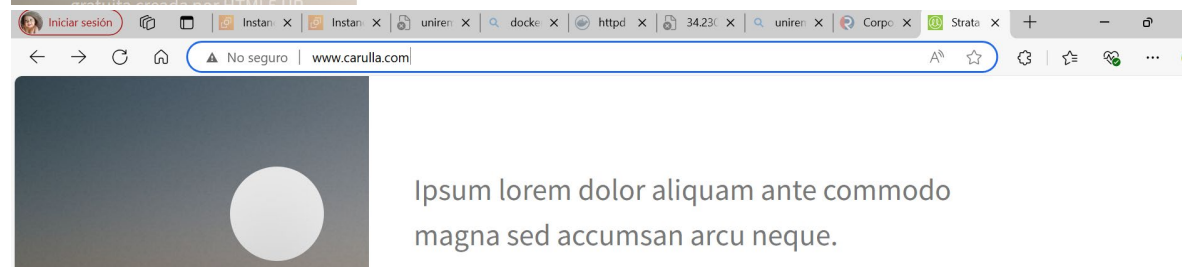
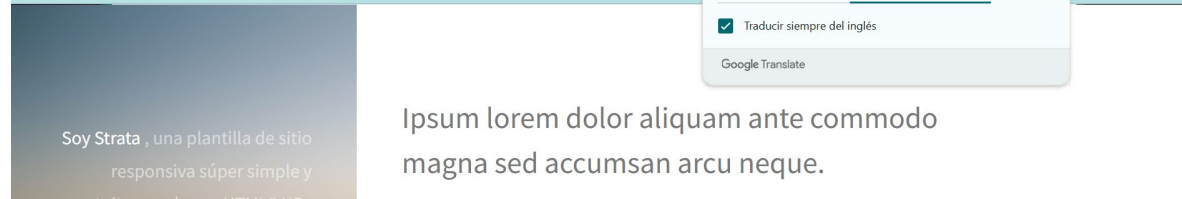
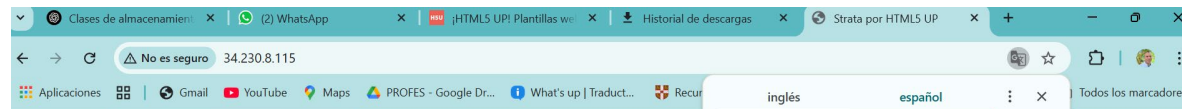
#### Resultados del balanceador después de configurar los archivos local hosts



¡Funciona!



It works!



Nota: Capture de pantalla sobre los resultados. Adaptado del seminario AWS 2025.

Fuente propia

## ***Conclusiones***

En este trabajo de grado hemos podido obtener distintos conocimientos para el desarrollo e implementación de servicios en la nube de AWS.

A través de la teoría y la práctica realizada en la plataforma de AWS obtuvimos la posibilidad de hacer uso de varios servicios de nube, entre los cuales encontramos VPC, EC2, S3, auto scaling group, Docker entre otros, todos ellos nos ayudaron a obtener una visión más amplia y clara en el uso de los servicios así como la importancia que estos ofrecen hoy en día para el desarrollo de aplicaciones y la infraestructura tecnológica con la que se cuenta actualmente en esta era, lo cual pudimos comprobar y observar al momento de experimentar con estas herramientas ofrecidas por los servicios, siendo claves para la creación de entornos escalables y de disponibilidad, aumentando así la eficiencia en los sistemas. Todo este proceso también nos ayudó a comprender mejor las buenas prácticas en la configuración de sus recursos, así como la estructura que los compone, entendiendo de esa manera el funcionamiento de cada servicio y como se integran entre sí para responder a necesidades reales. brindándonos una buena base para seguir explorando, conociendo y trabajando en los servicios de nube en AWS...

¿Cómo ayuda el uso de máquinas virtuales EC2 de Amazon Web Services (AWS) a disminuir los costos operativos y de mantenimiento en el uso de aplicaciones?

La utilización de herramientas que nos brinda la plataforma de AWS, ha avanzado de manera paulatina, evolucionando consigo los servicios de la nube, y mejorando eficazmente el uso tecnológico en las aplicaciones, obteniendo mayor beneficio para el sector empresarial, ya que ofrece una amplia infraestructura de cómputo, adaptable y modulable. La implementación de la herramienta EC2 nos permite crear, administrar, diseñar y elaborar máquinas virtuales que operan por medio de las instancias designadas en la región, sin necesidad de un hardware físico de manera que reducimos los gastos de manera significativa en la operación y mantenimiento de los recursos implementados. Además de lo mencionado anteriormente, tenemos la posibilidad de modificar los recursos en función de la oferta y demanda, beneficio que logramos a través de la herramienta de auto Scaling.

El servicio de EC2 también nos ofrece otras herramientas para crear aplicaciones de forma segura, ágil y eficaz, tales como los contenedores, muy apreciados por su portabilidad y su capacidad para adaptarse a diversos contextos, de igual modo los Dokers, que trabajan paralelamente en la implementación de aplicaciones en la nube. Sin embargo, gracias al progreso favorable de AWS, contamos con herramientas como el equilibrador de carga y las AMIS, que están encargadas de la distribución en el tráfico de datos de las estancias, estas dos herramientas nos brindan con efectividad amplias aplicaciones, robustas, seguras y eficientes.

En conclusión, podemos afirmar que la utilización de las herramientas Amazon EC2 nos proporcionan un ambiente sumamente beneficioso y provechoso para los que utilizamos y aprovechamos dichos recursos. Gracias a su versión gratuita, muchos estudiantes pueden aprender y ejecutar sistemas operativos. Además de ser práctica, hemos comprobado su elevada eficiencia y rentabilidad para el desarrollo en la nube. En mi opinión, las compañías deberían implementar más estas herramientas.

## Referencias

AWS. (2025). Referencia general de Amazon Web Services (AWS) . Recuperado de:

[https://docs.aws.amazon.com/es\\_es/general/latest/gr/Welcome.html](https://docs.aws.amazon.com/es_es/general/latest/gr/Welcome.html)

Google Cloud. (2025). Productos de Google Cloud. Recuperado de:

[https://cloud.google.com/products?\\_gl=1\\*tol0ej\\*\\_up\\*MQ..&gclid=CjwKCAjwnPS-BhBxEiwAZjMF0v5iwHHEtMSzkwDCr2WVZ2FyLAfgGpBopjmZbZOgAGDbyHzQmtUEBoCTB8QAvD\\_BwE&gclsrc=aw.ds&hl=es\\_419#featured-products](https://cloud.google.com/products?_gl=1*tol0ej*_up*MQ..&gclid=CjwKCAjwnPS-BhBxEiwAZjMF0v5iwHHEtMSzkwDCr2WVZ2FyLAfgGpBopjmZbZOgAGDbyHzQmtUEBoCTB8QAvD_BwE&gclsrc=aw.ds&hl=es_419#featured-products)

Cloudwards.net. (2025). AWS vs Azure vs Google Cloud: *Comparison of the Big Three*

*Cloud Providers*. Recuperado de: <https://www.cloudwards.net/aws-vs-azure-vsgoogle/>

Datacamp. (2025). AWS, Azure and GCP Service Comparison for Data Science & AI.

Recuperado de: <https://www.datacamp.com/cheat-sheet/aws-azure-and-gcp-servicecomparison-for-data-science-and-ai>

Datacamp. (2025). AWS vs. Azure vs. Google Cloud: *Una comparación completa*.

Recuperado de: <https://www.datacamp.com/es/blog/aws-vs-azure-vs-gcp>

Bastidas, L.R. (2007). *El inicio del siglo XXI*. Planeta. Sitio web:

<http://www.rbastidasl.com/libro-inicio-del-sigloxxi>.

AWS. (2025). Referencia general de Amazon Web Services (AWS). Recuperado de:

[https://docs.aws.amazon.com/es\\_es/general/latest/gr/Welcome.html](https://docs.aws.amazon.com/es_es/general/latest/gr/Welcome.html)

Cloudwards.net. (2025). AWS vs Azure vs Google Cloud: Comparison of the Big Three

Cloud Providers. Recuperado de: <https://www.cloudwards.net/aws-vs-azure-vsgoogle/>

Google Cloud. (2025). Compara los servicios de AWS y Azure con Google Cloud.

Recuperado de: <https://cloud.google.com/docs/get-started/aws-azure-gcp-servicecomparison?form=MG0AV3&hl=es-419>

*Amazon Virtual Private Cloud*. (s. f.). *¿Qué es Amazon VPC?* Recuperado de de:

[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 Elastic Compute Cloud*. (s. f.). *¿Qué es Amazon EC2?* Recuperado de:

[https://docs.aws.amazon.com/es\\_es/AWSEC2/latest/UserGuide/concepts.html](https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/concepts.html)

Google Cloud. (s. f.). *¿Qué son los contenedores?* Google Cloud. Recuperado de:  
<https://cloud.google.com/learn/what-are-containers?hl=es>

«*What is Docker?* » (2024, 10 septiembre). Docker Documentation. Recuperado de:  
<https://docs.docker.com/get-started/docker-overview/>