



TRABAJO DE GRADO
Opción Seminario-Diplomado.

Implementación de infraestructura en la nube con AWS

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

Yhonayker Antonio Gil Colon.
Juan Pablo Berrio López (docente del seminario o diplomado).
Opción de Trabajo de grado Seminario-Diplomado.
2025.

Dedicatoria

Este trabajo está dedicado a mi padre, mi madre y mi abuela, quienes han sido mis pilares fundamentales en mi vida y educación. En especial, esta dedicatoria va en gran parte para mi padre, que con su apoyo incondicional ha sido clave para que pueda avanzar en mis estudios profesionales. Su esfuerzo, sacrificio y motivación son lo que me impulsa a superarme día a día cada reto y seguir a delante. Gracias, papá por creer en mí y brindarme estas oportunidades que han sido necesarias para crecer como persona y profesionalmente.

Tabla de Contenidos

Resumen.....	4
Marco conceptual y contextual	5
La virtualización.	5
Computación en la nube.....	5
Metodología.	6
Amazon Web Services (AWS)	6
Nginx.....	7
MobaXterm.....	8
Docker.....	8
HTTPD.....	8
Desarrollo e implementación del aprendizaje.....	8
creación de una instancia.	9
Creación de un Balanceador de Carga (Load Balancer).....	12
Creación del Escalado Automático (Auto Scaling).....	16
Creación de base de datos usando Aurora y RDS.....	19
Creando Buckets con s3.....	22
Creación del Clúster usando Docker.....	24
Figuras y tablas	¡Error! Marcador no definido.
Conclusiones.....	27
Referencias.....	29

Resumen

En este trabajo se realizará la implementación de una infraestructura en la nube haciendo uso de los servicios de Amazon Web Services (AWS), la cual nos permitirá crear y ejecutar servicios en la nube. En este trabajo se desplegaron Instancias EC2 la cual funcionaria como un ordenador remoto el cual estaría funcionando en este caso con Linux, pero también podría usarse con Windows.

Se le configuraron balanceadores de carga (Load Balancer) lo cual va a ayudar a distribuir cargas de trabajo haciendo uso de recursos informáticos, como, servidores virtuales y a su vez se le crearon grupos de escalado automático (Auto Scaling) lo que nos va a garantizar que la infraestructura sea adaptable y flexible siendo capaz de adaptarse a la demanda del tráfico.

Al mismo tiempo se le implemento grupos de seguridad (Security Groups) para que controlen las instancias y los accesos a estas, además se utilizan snapshots y volúmenes para la persistencia de datos. En términos almacenamiento se le fue necesario implementarle Amazon S3, lo cual nos servirá para el resguardo de archivos y futuras copias de seguridad.

En el ámbito de las bases de datos, fue configurado Amazon RDS la cual nos va a proporcionar una supervisión optimizada que también ayudaría a detectar problemas en el rendimiento rápidamente y Amazon Aurora, para una mejor gestión, escalabilidad y disponibilidad de la información almacenada.

En el transcurso del proyecto también se utilizaron herramientas para la administración y optimización para la correcta función de los servidores. Las herramientas usadas fueron MobaXterm para la gestión remota de instancias, Nginx y HTTPD como servidores web y Docker para la creación de contenedores de las aplicaciones, lo cual nos permitió que su despliegue sea más eficiente y modular.

Como resultado, logre construir una infraestructura fiable, segura y escalable, haciendo uso de las mejores prácticas en la administración de recursos en AWS. Este trabajo es la evidencia de como la correcta integración de diferentes servicios y herramientas nos ayudan a optimizar el rendimiento de una aplicación, reducir costos y una mejor disponibilidad del sistema.

Palabras clave

AWS, EC2, Auto Scaling, Load Balancer, S3, RDS, Docker, Seguridad, Infraestructura en la nube, HTTPD, Aurora, Servidores.

Marco conceptual y contextual

Para la realización de este trabajo de necesito realizar una respectiva investigación para saber cuáles fueron los orígenes de la virtualización y de la computación en la nube.

La virtualización.

comenzó en la década de 1960 con los mainframes de IBM, que introdujeron conceptos de tiempo comparativo y sistemas operativos que permitan ejecutar múltiples entornos en un solo hardware. Estos primeros sistemas (como CP-40 y CP/CMS) fueron pioneros en aprovechar de manera óptima los recursos y sentaron las bases para las técnicas de participación y emulación de sistemas.

Computación en la nube.

La idea de “computación en la nube” surge a partir de la necesidad de externalizar y escalar recursos informáticos de manera flexibles. El concepto tomó forma cuando se unieron avances en virtualización, redes y almacenamiento distribuido, permitiendo el desarrollo de plataformas que ofrecían infraestructura, plataformas y software a demanda.

Tabla 1. Comparación de diferentes servicios en la nube.

Categoría	AWS	Azure	GCP
computación	EC2, Lambda, ECS, EKS	Virtual Machines, Azure Function, AKS	Compute Engine, Cloud Function, GKE
Almacenamiento	S3, EBS, Glacier	Blob Storage, Disk Storage, Archive Storage	Cloud Storage, Persistent Disk
Base de datos	RDS, DynamoBD, Aurora	SQL Database, Cosmos DB, MySQL in Azure	Cloud SQL, Cloud Spanner, Firestore
Redes y CDN	VPC, CloudFront, Direct Connect	Virtual Network, Azure CDN, ExpressRoute	VPC, Cloud CDN, Interconnect
Contenedores	ECS, EKS, Fragate	AKS, Container Instances	GKE, Cloud run
Serverless	Lambda, Fragate	Azure Functions, Logic Apps	Cloud Functions, Cloud Run
IA y Machine Learning	SageMaker, Comprehend, Rekognition	Cognitive Services, Machine Learning Studio	AI Plataform, AutoML

Metodología.

En este trabajo se realizará la implementación de una infraestructura en la nube haciendo uso de los servicios de Amazon Web Services (AWS), la cual nos permitirá crear y ejecutar servicios en la nube. En este trabajo se desplegaron Instancias EC2 la cual funcionaria como un ordenador remoto el cual estaría funcionando en este caso con Linux, pero también podría usarse con Windows.

Se le configuraron balanceadores de carga (Load Balancer) lo cual va a ayudar a distribuir cargas de trabajo haciendo uso de recursos informáticos, como, servidores virtuales y a su vez se le crearon grupos de escalado automático (Auto Scaling) lo que nos va a garantizar que la infraestructura sea adaptable y flexible siendo capaz de adaptarse a la demanda del tráfico.

Al mismo tiempo se le implemento grupos de seguridad (Security Groups) para que controlen las instancias y los accesos a estas, además se utilizan snapshots y volúmenes para la persistencia de datos. En términos almacenamiento se le fue necesario implementarle Amazon S3, lo cual nos servirá para el resguardo de archivos y futuras copias de seguridad.

En el ámbito de las bases de datos, fue configurado Amazon RDS la cual nos va a proporcionar una supervisión optimizada que también ayudaría a detectar problemas en el rendimiento rápidamente y Amazon Aurora, para una mejor gestión, escalabilidad y disponibilidad de la información almacenada.

En el transcurso del proyecto también se utilizaron herramientas para la administración y optimización para la correcta función de los servidores. Las herramientas usadas fueron MobaXterm para la gestión remota de instancias, Nginx y HTTPD como servidores web y Docker para la creación de contenedores de las aplicaciones, lo cual nos permitió que su despliegue sea más eficiente y modular.

Amazon Web Services (AWS)

Amazon Web Services es una plataforma que ofrece una amplia colección de servicios que pueden ser usados a nivel infraestructural de formas diferentes, los servicios que fueron usados en este proyecto son:

- Amazon Elastic Compute Cloud (Amazon EC2): proporciona capacidad de computación escalable bajo demanda en Amazon Web Services (AWS) Cloud. El uso de Amazon EC2 reduce los costos de hardware para que pueda desarrollar e implementar aplicaciones más rápidas. Puede usar Amazon EC2 para iniciar tantos o tan pocos servidores virtuales como usted necesita, configure la seguridad y las redes, y administre el almacenamiento. Puede agregar capacidad (escala)

para manejar tareas pesadas en computación, como procesos mensuales o anuales, o picos en el sitio web tráfico. Cuando el uso disminuye, puede reducir la capacidad (escala hacia abajo) nuevamente.

- Amazon Virtual Private Cloud (Amazon VPC): puede iniciar los recursos de AWS en un virtual aislada lógicamente red que has definido. Esta red virtual se parece mucho a una red tradicional que operaría en su propio centro de datos, con los beneficios de usar lo escalable infraestructura de AWS.S3.
- Amazon Simple Storage Service (Amazon S3): es un servicio de almacenamiento de objetos que ofrece escalabilidad líder en la industria disponibilidad, seguridad y rendimiento de los datos. Los clientes de todos los tamaños e industrias pueden utilizar Amazon S3 para almacenar y proteger cualquier cantidad de datos para una variedad de casos de uso, como lagos de datos sitios web, aplicaciones móviles, copia de seguridad y restauración, archivo, aplicaciones empresariales, IoT dispositivos y análisis de big data. Amazon S3 proporciona funciones de administración para que pueda optimizar organice y configure el acceso a sus datos para cumplir con su negocio específico, organización, etc y requisitos de cumplimiento.
- Amazon Relational Database Service (Amazon RDS): es un servicio web que facilita la configuración, operación y escala de un servicio base de datos relacional en la nube de AWS. Proporciona una capacidad rentable y redimensionable para un base de datos relacional estándar de la industria y gestiona tareas comunes de administración de bases de datos.
- Amazon Aurora (Aurora): es un motor de base de datos relacional totalmente administrado compatible con MySQL y PostgreSQL. Ya sabes cómo MySQL y PostgreSQL combinan la velocidad y fiabilidad de bases de datos comerciales de alta gama con la simplicidad y rentabilidad de bases de datos de código abierto. El código, las herramientas y las aplicaciones que usa hoy con su MySQL existente y las bases de datos PostgreSQL se pueden usar con Aurora. Con algunas cargas de trabajo, Aurora puede entregar hasta cinco veces el rendimiento de MySQL y hasta tres veces el rendimiento de PostgreSQL sin requerir cambios en la mayoría de sus aplicaciones existentes.

Nginx

Nginx es un software de código abierto que se utiliza para servir páginas web estáticas y dinámicas.

MobaXterm

MobaXterm es una caja de herramientas para la computación remota. En una sola aplicación de Windows, proporciona un montón de funciones que están diseñadas para programadores, webmasters, administradores de TI y casi todos los usuarios que necesitan manejar sus trabajos remotos de una manera más simple.

MobaXterm proporciona todas las herramientas de red remota importantes (SSH, RDP, X11, SFTP, FTP, Telnet, Rlogin, ...) para el escritorio de Windows, en un solo archivo exe portátil que funciona de forma inmediata. Algunos plugins se pueden utilizar para añadir funciones a MobaXterm como comandos Unix (bash, ls, cat, sed, grep, awk, rsync, ...).

Docker

Docker es una plataforma abierta para desarrollar, enviar y ejecutar aplicaciones. Docker le permite separar sus aplicaciones de su infraestructura para puede entregar software rápidamente. Con Docker, puedes administrar la infraestructura de la misma manera que administra sus aplicaciones. Aprovechando Docker metodologías para enviar, probar e implementar código, puede reducir significativamente el retraso entre escribir código y ejecutarlo en producción.

HTTPD

El Servidor HTTP Apache, coloquialmente llamado Apache, es una aplicación de servidor Web notable por desempeñar un papel clave en el crecimiento inicial de la World Wide Web. Originalmente basado en el servidor HTTPd de NCSA, el desarrollo de Apache comenzó a principios de 1995 después de que el trabajo en el código NCSA se estancara. Apache superó rápidamente a NCSA HTTPd como el servidor HTTP dominante, y ha seguido siendo el servidor HTTP más popular en uso desde abril de 1996.

Desarrollo e implementación del aprendizaje

En este apartado mostraré el paso a paso que hice para crear mi servidor en Amazon Web Services (AWS) siguiendo las instrucciones. Se va a crear un balanceador de carga con una aplicación que auto escale cuando un servidor llegue a cierto consumo ya pueda ser del 20 % de CPU o 80%.

Para empezar la creación de la instancia se deberá crear una cuenta en AWS para poder hacer uso de los servicios que iremos a ir usando. Ya una vez creada la cuenta o haber iniciado sesión, procederemos a buscar el primer servicio que vamos a usar que sería EC2 y ya estando ahí podríamos ya a iniciar a crear nuestra instancia.

Figura.1 Página de AWS.

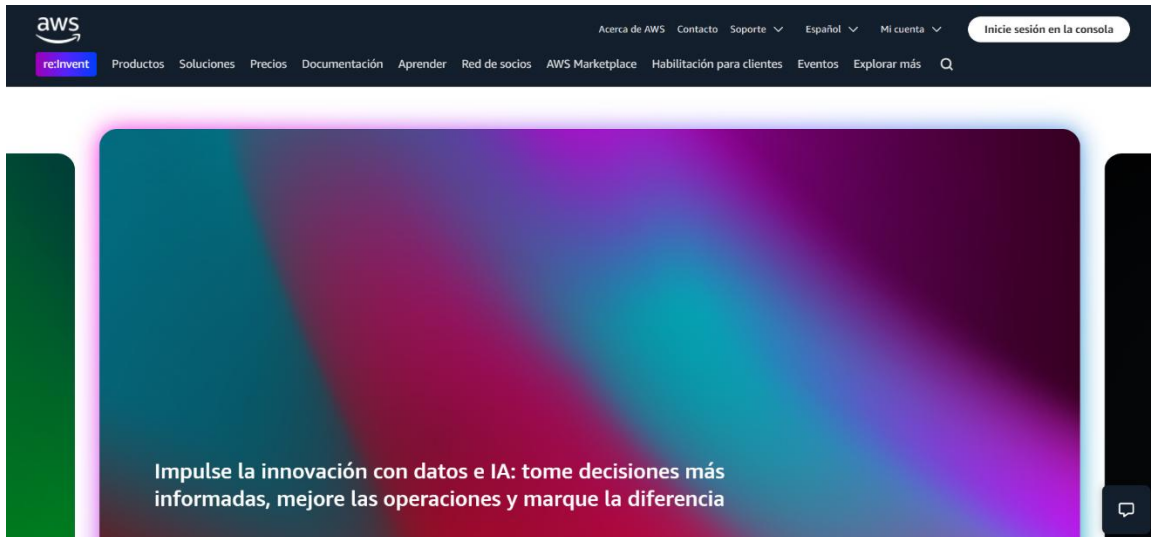
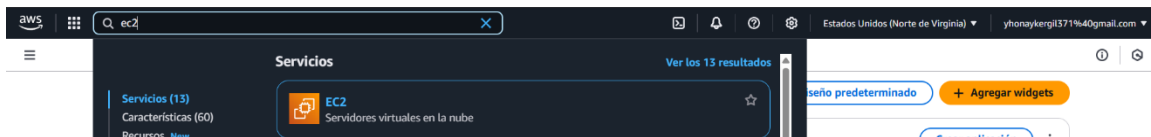


Figura.2 Como encontrar el servicio de EC2 en AWS.



creación de una instancia.

Ya una vez dentro de EC2 en el menú del lado derecho buscaremos el apartado que dice instancias la seleccionaremos y le daremos click a lanzar instancias. La cual va a redirigirnos a otra venta en la cual comenzaremos la configuración de esta. Empezaremos nombrando la instancia en mi caso “Linux-server-web” y luego se tendrá que escoger la imagen o sistema operativo a usar, en este caso usaremos para este proyecto se usar Linux.

Figura.3 Buscando la sección para la creación de la primera instancia.

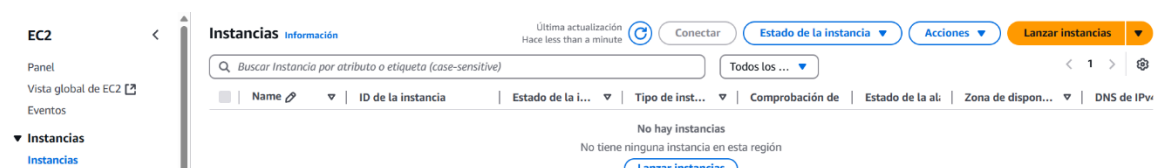


Figura.4 Configuración y creación de la instancia.

Lanzar una instancia [Información](#)

Amazon EC2 le permite crear máquinas virtuales, o instancias, que se ejecutan en la nube de AWS. Comience rápidamente siguiendo los sencillos pasos que se indican a continuación.

Nombre y etiquetas [Información](#)

Nombre
Linux-server-web [Agregar etiquetas adicionales](#)

▼ **Imágenes de aplicaciones y sistemas operativos (Imagen de máquina de Amazon)** [Información](#)

Una AMI es una plantilla que contiene la configuración de software (sistema operativo, servidor de aplicaciones y aplicaciones) necesaria para lanzar la instancia. Busque o examine las AMI si no ve lo que busca a continuación.

Q Busque en nuestro catálogo completo que incluye miles de imágenes de sistemas operativos y aplicaciones

Recientes Mis AMI **Inicio rápido**

Amazon Linux	macOS	Ubuntu	Windows	Red Hat	SUSE Linux	Debian

[Buscar más AMI](#)
Inclusión de AMI de AWS, Marketplace y la comunidad

Elegimos el tipo de instancia para el proyecto, se usará t2.micro de la capa gratuita que nos ofrece (AWS) ya que el proyecto no necesitará tantos recursos para poder correr en el servidor ya que con 1vCPU y 1GiB de memoria nos basta y sobra, ya si el proyecto fuese más grande si se debiera hacer uso de otro tipo de instancia lo que con llevaría a tener gastos en el servicio. Ya hecho eso crearemos una llave de tipo “.ppk” que nos servirá para iniciar sesión en el servidor haciendo uso de herramientas externas como MobaXterm.

Figura.5 Seleccionando el tipo de instancia y creando las llaves.

▼ **Tipo de instancia** [Información](#) | [Obtener asesoramiento](#)

Tipo de instancia

t2.micro Apto para la capa gratuita

Familia: t2 1 vCPU 1 GiB Memoria Generación actual: true
 Bajo demanda Ubuntu Pro base precios: 0.0134 USD per Hour
 Bajo demanda Linux base precios: 0.0116 USD per Hour
 Bajo demanda SUSE base precios: 0.0116 USD per Hour
 Bajo demanda Windows base precios: 0.0162 USD per Hour
 Bajo demanda RHEL base precios: 0.026 USD per Hour

Todas las generaciones

[Comparar tipos de instancias](#)

Se aplican costos adicionales a las AMI con software preinstalado

▼ **Par de claves (inicio de sesión)** [Información](#)

Puede utilizar un par de claves para conectarse de forma segura a la instancia. Asegúrese de que tiene acceso al par de claves seleccionado antes de lanzar la instancia.

Nombre del par de claves - obligatorio

Linux-Ix-Server [Crear un nuevo par de claves](#)

Para poder avanzar con la configuración deberemos crear una VPC para una mejor configuración en nuestro proyecto, la creación del VPC es algo sencilla ya que no hay mucho que modificar únicamente lo que hice fue cambiar el nombre y guardar. Ya teniendo nuestro VPC creado continuaremos con la configuración de red de nuestra instancia seleccionando nuestro VPC creado recientemente y en subred se tendrá que elegir “VPC-WEB-subnet-public2-us-east-2b”. Acá mismo se va a generar nuestro primer grupo de seguridad.

Figura.6 Creando el VPC que se usara en todo el proyecto.

Crear VPC [Información](#)

Una VPC es una parte aislada de la nube de AWS que contiene objetos de AWS, como instancias de Amazon EC2. Deslizar el ratón sobre un recurso para resaltar los recursos relacionados.

Configuración de la VPC

Recursos que se van a crear [Información](#)
Cree únicamente el recurso de VPC o la VPC y otros recursos de red.

Solo la VPC VPC y más

Generación automática de etiquetas de nombre [Información](#)
Ingrese un valor para la etiqueta Nombre. Este valor se utilizará para generar automáticamente etiquetas Nombre para todos los recursos de la VPC.

Generar automáticamente

VPC-web

Bloque de CIDR IPv4 [Información](#)
Determine la IP inicial y el tamaño de la VPC mediante la notación CIDR.

10.0.0.0/16 65,536 IPs

El tamaño del bloque CIDR debe estar entre /16 y /28.

Bloque de CIDR IPv6 [Información](#)

Sin bloque de CIDR IPv6
 Bloque de CIDR IPv6 proporcionado por Amazon

Vista previa

VPC [Mostrar detalles](#)
Su red virtual de AWS

VPC-web-vpc

Subredes (4)
Subredes dentro de esta VPC

- us-east-2a
 - VPC-web-subnet-public1-us-east-2a
 - VPC-web-subnet-private1-us-east-
- us-east-2b
 - VPC-web-subnet-public2-us-east-2b
 - VPC-web-subnet-private2-us-east-

Figura.7 Configuración de las redes que usara el VPC.

Configuraciones de red [Información](#)

VPC: obligatorio [Información](#)

vpc-072cf5eafe2a7ea19 (VPC-WEB-vpc)
10.0.0.0/16

Subred [Información](#)

subnet-0e7ac46ce39504c98 VPC-WEB-subnet-public2-us-east-2b
VPC: vpc-072cf5eafe2a7ea19 Propietario: 387149830728
Zona de disponibilidad: us-east-2b Tipo de zona: Zona de disponibilidad
Direcciones IP disponibles: 4091 CIDR: 10.0.16.0/20

Asignar automáticamente la IP pública [Información](#)

Desactivar

Firewall (grupos de seguridad) [Información](#)

Un grupo de seguridad es un conjunto de reglas de firewall que controlan el tráfico de la instancia. Agregue reglas para permitir que un tráfico específico llegue a la instancia.

Crear grupo de seguridad Seleccionar un grupo de seguridad existente

Nombre del grupo de seguridad - obligatorio

launch-wizard-1

Este grupo de seguridad se agregará a todas las interfaces de red. El nombre no se puede editar después de crear el grupo de seguridad. La longitud máxima es de 255 caracteres. Caracteres válidos: a-z, A-Z, 0-9, espacios y .-:/!#\$%&'()*+,@#[]+=&:{}|~`

Para finalizar la creación de nuestra primera instancia crearemos nuestro volumen (Almacenamiento) que tendrá el proyecto, en mi caso lo deje como venia por defecto. Ya hecho todos los pasos anteriores le daremos al botón de lanzar instancia y así ya tenemos creada nuestra primera instancia.

Figura.8 Configuración del almacenamiento o volumen de la instancia.

The screenshot shows the AWS console configuration for an EBS volume and the instance launch summary. On the left, under 'Almacenamiento (volúmenes)', the 'Volumen 1 (Raíz de AMI)' is configured with the following settings:

- Tipo de almacenamiento:** EBS
- Nombre del dispositivo - obligatorio:** /dev/xvda
- Instantánea:** snap-009f6275c7f008b15
- Tamaño (GiB):** 8
- Tipo de volumen:** gp3
- IOPS:** 3000
- Eliminar cuando termine:** Sí
- Cifrado:** No cifrado
- Clave de KMS:** Seleccionar
- Rendimiento:** 125

On the right, the 'Resumen' section shows the instance configuration:

- Número de instancias:** 1
- Imagen de software (AMI):** Amazon Linux 2023 AMI 2023.7.2...más información
- Tipo de servidor virtual (tipo de instancia):** t2.micro
- Firewall (grupo de seguridad):** Nuevo grupo de seguridad
- Almacenamiento (volúmenes):** Volúmenes: 1 (8 GiB)

Buttons for 'Cancelar', 'Lanzar instancia', and 'Código de versión preliminar' are visible at the bottom right.

Creación de un Balanceador de Carga (Load Balancer).

A todo esto, ya tenemos una instancia creada funcionando de manera correcta, iniciaremos la configuración de un Balanceador de carga lo que nos va a servir como un equilibrador de carga en respuesta a los cambios en el tráfico entrante en el proyecto. Al igual que como se hizo con la instancia buscaremos en el lado izquierdo el apartado de equilibrador de carga, dando click en balanceadores de carga para luego en el botón de crear balanceador de carga lo que no envía a una vista nueva en la que elegiremos el balanceador de carga de aplicaciones.

Figura.9 Buscando la sección para la creación de nuestro balanceador de carga.

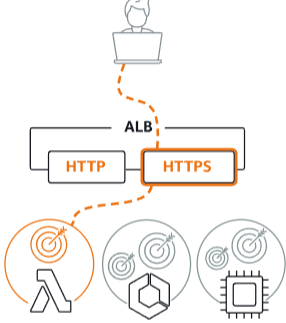
The screenshot shows the AWS console 'Balanceadores de carga' (Load Balancers) page. The left sidebar shows the navigation menu with 'Balanceadores de carga' selected under 'Equilibrio de carga'. The main content area shows the 'Balanceadores de carga' section with a search bar and a table of load balancers. The table is currently empty, and a message states: 'No hay ningún equilibrador de carga. No tiene ningún equilibrador de carga en us-east-1'. A 'Crear balanceador de carga' button is visible at the bottom of the table.

Figura.10 Seleccionando el tipo de balanceador de carga que se usará.

EC2 > Balanceadores de carga > Compare y seleccione el tipo de equilibrador de carga

Tipos de equilibradores de carga

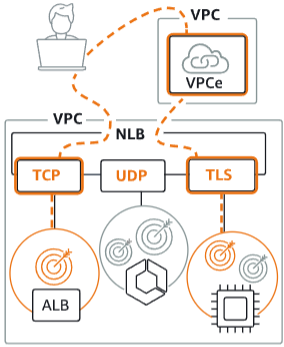
Balaceador de carga de aplicaciones Info



Elija un balanceador de carga de aplicaciones cuando necesite un conjunto de características flexibles para sus aplicaciones con tráfico HTTP y HTTPS. En el nivel de solicitud, los balanceadores de carga de aplicaciones proporcionan características avanzadas de enrutamiento y visibilidad dirigidas a arquitecturas de aplicación, incluidos microservicios y contenedores.

[Crear](#)


Balaceador de carga de red Info



Elija un equilibrador de carga de red cuando necesite un rendimiento ultraalto, descarga de TLS a gran escala, implementación centralizada de certificados, compatibilidad con UDP y direcciones IP estáticas para sus aplicaciones. En el nivel de conexión, los equilibradores de carga de red pueden controlar millones de solicitudes por segundo de forma segura a la vez que mantienen latencias ultrabajas.

[Crear](#)

Equilibrador de carga de gateway Info



Elija un equilibrador de carga de gateway cuando necesite implementar y administrar una flota de dispositivos virtuales de terceros compatibles con GENEVE. Estos dispositivos permiten mejorar los controles de las políticas, la seguridad y la conformidad.

[Crear](#)

► Classic Load Balancer - generación anterior

Ya dentro del balanceador de carga de aplicaciones realizaremos la configuración básica comenzando por el nombre que en mi caso le puse “LB-WebServer” el esquema que se usará expuesto a internet ya que usaremos direcciones IP públicas y a su vez el tipo de dirección será IPv4. En el mapeo de red usaremos la VPC que creamos y venimos usando y zonas de disponibilidad seleccionaremos las dos subredes públicas o las que estén creadas.

Figura.11 Nombrando el balanceado y configurando el esquema y tipo de dirección IPv4.

Configuración básica

Nombre del balanceador de carga
Debe ser nombre único dentro de su cuenta de AWS y no puede cambiarse después de crear el equilibrador de carga.

LB-WebServer

Se permite un máximo de 32 caracteres alfanuméricos, incluidos guiones, pero el nombre no puede comenzar ni terminar por un guión.

Esquema Info
El esquema no se puede cambiar después de crear el equilibrador de carga.

Expuesto a Internet

- Suministra el tráfico expuesto a Internet.
- Tiene direcciones IP públicas.
- El nombre DNS se puede resolver públicamente.
- Requiere una subred pública.

Interno

- Suministra el tráfico interno.
- Tiene direcciones IP privadas.
- El nombre DNS se puede resolver públicamente.
- Compatible con los tipos de direcciones IP IPv4 y Dualstack.

Tipo de dirección IP del equilibrador de carga Info
Seleccione el tipo de dirección IP de frontend que desea asignar al equilibrador de carga. La VPC y las subredes asignadas a este equilibrador de carga deben incluir los tipos de direcciones IP seleccionados. Las direcciones IPv4 públicas tienen un costo adicional.

IPv4
Incluye solo direcciones IPv4.

Dualstack
Incluye direcciones IPv4 e IPv6.

Dualstack sin IPv4 pública
Incluye una dirección IPv6 pública y direcciones IPv4 e IPv6 privadas. Compatible solo con equilibradores de carga expuestos a Internet.

Figura.12 Configurando el mapeo de red.

Mapeo de red info

El balanceador de carga dirige el tráfico a los destinos de las subredes seleccionadas y en función de la configuración de las direcciones IP.

VPC info

El equilibrador de carga existirá y escalará dentro de la VPC seleccionada. La VPC seleccionada también es el lugar donde se tienen que alojar los destinos del equilibrador de carga, a menos que se dirijan a destinos de Lambda o locales, o si se utiliza la interconexión de VPC. Para confirmar la VPC para sus objetivos, consulte [los grupos de destino](#). Para una VPC nueva, [crea una VPC](#).

VPC-WEB-vpc
vpc-072cf5eafe2a7ea19
CIDR de VPC IPv4: 10.0.0.0/16

Grupos de IP - nuevo info

Si lo desea, puede configurar un grupo de IPAM como la fuente preferida para las direcciones IP de sus equilibradores de carga. Cree o visualice los grupos en [la consola del administrador de direcciones IP de Amazon VPC](#).

Use el grupo de IPAM para direcciones IPv4 públicas

El grupo de IPAM que elija será la fuente preferida de direcciones IPv4 públicas. Si el grupo está agotado, AWS asignará las direcciones IPv4.

Zonas de disponibilidad y subredes info

Seleccione al menos dos zonas de disponibilidad y una subred para cada zona. Se colocará un nodo de equilibrador de carga en cada zona seleccionada y se escalará de forma automática en respuesta al tráfico. El equilibrador de cargas dirige el tráfico únicamente a los destinos de las zonas de disponibilidad seleccionadas.

us-east-2a (use2-az1)

Subred

Solo se utilizan los bloques CIDR correspondientes al tipo de dirección IP del equilibrador de cargas. Se necesitan al menos 8 direcciones IP disponibles para que el equilibrador de cargas escale de manera eficiente.

subnet-0639ebaebd31bb425
CIDR de subred IPv4: 10.0.0.0/20

VPC-WEB-subnet-public1-us-east-2a

us-east-2b (use2-az2)

Subred

Solo se utilizan los bloques CIDR correspondientes al tipo de dirección IP del equilibrador de cargas. Se necesitan al menos 8 direcciones IP disponibles para que el equilibrador de cargas escale de manera eficiente.

subnet-0e7ac46e39504c98
CIDR de subred IPv4: 10.0.16.0/20

VPC-WEB-subnet-public2-us-east-2b

En el apartado de grupos de seguridad lo dejaremos por defecto ya que este será únicamente para este balanceador de carga. En el direccionamiento deberemos configurar el protocolo HTTP con el puerto 80 y se le deberá seleccionar un grupo de destino (Target group) el cual tendrá una configuración básica seleccionando que su destino serán las instancias y darle un nombre por ejemplo “TG-instancias-web”, el puerto, protocolo dirección IP y la VPC serán las mismas que se vienen usando anteriormente no harban cambio en esas solamente en la versión de protocolo que vamos a usar será HTTP1 una vez hecho eso podremos usar el Target Group para poder terminar de crear el Balanceador de carga.

Figura.13 Creando un grupo de seguridad.

Grupos de seguridad info

Un grupo de seguridad consiste en un conjunto de reglas de firewall que controlan el tráfico hacia el equilibrador de carga. Seleccione un grupo de seguridad existente o [crea un nuevo grupo de seguridad](#).

Grupos de seguridad

Seleccione hasta 5 grupos de seguridad

default
sg-0a1bb0fb44662a75 VPC: vpc-072cf5eafe2a7ea19

Agentes de escucha y direccionamiento info

Un agente de escucha es un proceso que comprueba las solicitudes de conexión mediante el puerto y el protocolo que configure. Las reglas que defina para un agente de escucha determinan cómo el equilibrador de carga dirige las solicitudes a sus destinos registrados.

▼ Agente de escucha HTTP:80 Eliminar

Protocolo: HTTP : Puerto: 80
1-65535

Acción predeterminada info

Reenviar a: Seleccione un grupo de destino

Crear un grupo

En uso

TG-InstanceGroup	HTTP
Tipo de destino: Instancia, IPv4	

Etiquetas del agente de escucha - opcional

Considere la posibilidad de agregar etiquetas al agente de escucha. Las etiquetas permiten clasificar los recursos.

[Agregar etiqueta de agente de escucha](#)

Puede agregar hasta 50 etiquetas más.

Figura.14 Finalizando la configuración del balanceador de carga.

Paso 1
Especificar los detalles del grupo
 Paso 2
 Registrar destinos

Especificación de los detalles del grupo

El balanceador de carga dirige las solicitudes a los destinos de un grupo de destino y realiza comprobaciones de estado en los destinos.

Configuración básica

La configuración de esta sección no se puede cambiar después de crear el grupo de destino.

Elegir un tipo de destino

Instancias

- Admite el balanceo de carga en instancias dentro de una VPC específica.
- Facilita el uso de [Amazon EC2 Auto Scaling](#) para administrar y escalar la capacidad de EC2.

Direcciones IP

- Admite el balanceo de carga en recursos de VPC y en las instalaciones.
- Facilita el direccionamiento a varias direcciones IP e interfaces de red en la misma instancia.
- Ofrece flexibilidad con arquitecturas basadas en microservicios, lo que simplifica la comunicación entre aplicaciones.
- Admite destinos IPv6, lo que permite la comunicación IPv6 integral y NAT de IPv4 a IPv6.

Función Lambda

- Facilita el direccionamiento a una única función Lambda.
- Accesible solo para balanceadores de carga de aplicaciones.

Balanceador de carga de aplicaciones

- Ofrece la flexibilidad para que un balanceador de carga de red acepte y dirija solicitudes TCP dentro de una VPC específica.
- Facilita el uso de direcciones IP estáticas y PrivateLink con un balanceador de carga de aplicaciones.

Nombre del grupo de destino

TG-InstanciasWeb

Se permite un máximo de 32 caracteres alfanuméricos, incluidos guiones, pero el nombre no puede comenzar ni terminar por un guión.

Protocolo : Puerto

Elige un protocolo para su grupo de destino que corresponda al tipo de balanceador de carga que enviará el tráfico hacia él. Algunos protocolos ahora incluyen la detección de anomalías para los destinos y se pueden establecer opciones de mitigación una vez creado el grupo de destino. Esta dirección no se puede cambiar después de la creación

HTTP 1-65535

Tipo de dirección IP

Solo los destinos con el tipo de dirección IP indicado pueden registrarse en este grupo de destino.

IPv4

Cada instancia tiene una interfaz de red predeterminada (eth0) a la que se le asigna la dirección IPv4 privada principal. La dirección IPv4 privada principal de la instancia es la que se aplicará al destino.

IPv6

Cada instancia que registre debe tener asignada una dirección IPv6 principal. Esta se configura en la interfaz de red predeterminada de la instancia (eth0). [Obtenga más información](#)

VPC

Seleccione la VPC con las instancias que desea incluir en el grupo de destino. En esta lista solo están disponibles las VPC que admiten el tipo de dirección IP seleccionado anteriormente.

VPC-WEB-vpc
 vpc-073355a6ca7ca7e19
 CIDR de VPC IPv4: 10.0.0.0/16

Versión del protocolo

HTTP1

Envíe solicitudes a los destinos con HTTP/1.1. Compatible cuando el protocolo de solicitud es HTTP/1.1 o HTTP/2.

HTTP2

Envíe solicitudes a los destinos con HTTP/2. Compatible cuando el protocolo de solicitud es HTTP/2 o gRPC, pero las características específicas gRPC no están disponibles.

gRPC

Envíe solicitudes a los destinos con gRPC. Compatible cuando el protocolo de solicitud es gRPC.

Creación del Escalado Automático (Auto Scaling).

Ya una vez hayamos hecho todos los pasos anteriores podremos hacer el auto escalado, pero necesitamos crear una plantilla de lanzamiento la cual nos permitirá crear una configuración de instancia guardada que se podrá reutilizar para lanzar más adelante.

Para agilizar la creación de la plantilla crearemos una AMI la cual será la que va a contener la configuración de software (por ejemplo, sistemas operativos, servidores o aplicaciones). Si queremos crear la AMI lo primero que tenemos que hacer es ir a nuestra instancia creada y seleccionarla, dirigirnos al botón donde dice acciones lo que nos desplegará otras opciones, pero la que nos interesa es la que dice “Imágenes y plantillas” la cual seleccionaremos y le daremos al que dice “Crear imagen”.

Figura.15 Como dirigirse a la creación de imagen.



La configuración de la AMI es muy sencilla ya que lo que se hará, va a ser un tipo de copia por lo cual solo tendríamos que nombrarla y ya la tendríamos creada. Ahora continuando con la creación del escalador automático, ya teniendo nuestra plantilla que vamos a usar, podremos seguir con la configuración como ya se viene haciendo en pasos anteriores se va a nombrar (Por ejemplo “ImagenLinux”). Continuando toca realizar la elección de lanzamiento de instancia la cual será el entorno de red de VPC en el que se lanzan las instancias luego en el apartado de redes seleccionaremos el VPC que ya venimos usando y en las subredes elegiremos las 2 subredes públicas que en mi caso tienen el nombre de “(VPC-WEB-subnet-public1-us-east-2a) y (VPC-WEB-subnet-public2-us-east-2b)”. Para finalizar el escalador automático tenemos que definir la capacidad deseada del grupo para satisfacer la demanda, ya sea en forma manual o mediante el escalamiento automático y los límites de escalamiento sobre cuánto puede aumentarse o disminuirse la capacidad deseada.

Figura.16 Creación y configuración de la AMI.

ID de la instancia
i-Deo0994f8ab65739a (Linux)

Nombre de la imagen
ImagenLinux

Descripción de la imagen (opcional)
Descripción de la imagen

Reservar instancia
Cuando se selecciona esta opción, Amazon EC2 reserva la instancia para mantener los datos en reposo cuando se terminan las instancias de los volúmenes asociados. Esto garantiza la coherencia de datos.

Volúmenes de instancia

Tipo de almacenamiento	Dispositivo	Instantánea	Tamaño	Tipo de volumen	IOPS	Rendimiento	Eliminar cuando termine	Cifrado
EBS	/dev/xvck...	Crear una nueva instantánea...	8	SSD de uso general de EBS...	3000		<input checked="" type="checkbox"/> Habilitar	<input type="checkbox"/> Habilitar

Etiquetas (opcional)
Las etiquetas son marcas que se asignan a un recurso de AWS. Cada etiqueta consta de una clave y un valor opcional. Puede utilizarlas para buscar entre los recursos y filtrarlos, o para hacer un seguimiento de sus costos en AWS.

Etiquetar imágenes e instancias juntas
Etiquete la imagen y las instancias con la misma etiqueta.

Etiquetar imagen e instancias por separado
Etiquete la imagen y las instancias con diferentes etiquetas.

No hay etiquetas asociadas a este recurso.

Puede agregar hasta 50 etiquetas más.

Figura.17 Configuración del escalado automático sección de nombre y selección de plantilla.

Nombre

Nombre del grupo de Auto Scaling
Escriba un nombre para identificar al grupo.

AS-WEB

Debe ser único para esta cuenta en la región actual y no puede superar los 255 caracteres.

Plantilla de lanzamiento info

Para las cuentas creadas después del 31 de mayo de 2023, la consola de EC2 solo admite la creación de grupos de escalado automático con plantillas de lanzamiento. No se recomienda crear grupos de escalado automático con configuraciones de lanzamiento, pero aún se podrá hacer a través de la CLI y la API hasta el 31 de diciembre de 2023.

Plantilla de lanzamiento
Elija una configuración de lanzamiento que contenga la configuración de nivel de instancia, como la imagen de máquina de Amazon (AMI), el tipo de instancia, el par de claves y los grupos de seguridad.

PlantillaWeb

Crear una configuración de lanzamiento

Versión
Default (1)

Crear una versión de plantilla de lanzamiento

Descripción -	Plantilla de lanzamiento PlantillaWeb lt-05185b98095015062	Tipo de instancia t2.micro
AMI ID ami-0157c5f0fc56bd77c	Grupos de seguridad -	Solicitar instancias de spot No
Nombre del par de claves Linux	ID de grupos de seguridad sg-07917c6ded6670a62	

Detalles adicionales

Almacenamiento (volúmenes) -	Fecha de creación Sat Mar 29 2025 15:47:59 GMT-0500 (hora estándar de Colombia)
--	---

Figura.18 Elección de lanzamiento de instancia y configuración de redes.

Elegir las opciones de lanzamiento de instancias Info

Elija el entorno de red de VPC en el que se lanzan las instancias y personalice los tipos de instancias y las opciones de compra.

Requisitos de tipo de instancias Info [Anular la plantilla de lanzamiento](#)

Puede mantener los mismos atributos o tipos de instancias de la plantilla de lanzamiento, o bien puede optar por anular la plantilla de lanzamiento al especificar atributos de instancia diferentes o al agregar los tipos de instancias de forma manual.

Plantilla de lanzamiento	Versión	Descripción
PlantillaWeb [?] it-03185b98095015062	Default	-

Tipo de instancia
t2.micro

Red Info

Para la mayoría de las aplicaciones, puede utilizar varias zonas de disponibilidad y dejar que EC2 Auto Scaling equilibre sus instancias entre las zonas. La VPC predeterminada y las subredes predeterminadas son adecuadas para comenzar rápidamente.

VPC
Elija la VPC que define la red virtual para el grupo de Auto Scaling.

vpc-072cf5eafe2a7ea19 (VPC-WEB-vpc)
10.0.0.0/16

[Crear una VPC](#) [\[?\]](#)

Zonas de disponibilidad y subredes
Defina qué zonas de disponibilidad y subredes puede utilizar el grupo de Auto Scaling en la VPC elegida.

Seleccionar zonas de disponibilidad y subredes

us-east-2a | subnet-0639eabed31bb425 (VPC-WEB-subnet-public1-us-east-2a) [\[x\]](#)
10.0.0.0/20

us-east-2b | subnet-0e7ac46ce39504c98 (VPC-WEB-subnet-public2-us-east-2b) [\[x\]](#)
10.0.16.0/20

[Crear una subred](#) [\[?\]](#)

Distribución de zonas de disponibilidad - nueva
El escalamiento automático equilibra automáticamente las instancias en todas las zonas de disponibilidad. Si se producen errores de lanzamiento en una zona, seleccione una estrategia.

Mejor esfuerzo equilibrado
Si los lanzamientos fallan en una zona de disponibilidad, el escalamiento automático intentará lanzarse en otra zona de disponibilidad en buen estado.

Solo equilibrado
Si los lanzamientos fallan en una zona de disponibilidad, el escalamiento automático seguirá intentando lanzarse en la zona de disponibilidad en mal estado para mantener una distribución equilibrada.

[Cancelar](#) [Omitir para revisar](#) [Anterior](#) [Siguiente](#)

Figura.19 Configuración del escalado y del tamaño de grupo.

Configurar escalamiento y tamaño de grupo - *opcional* Info

Defina la capacidad deseada de su grupo y los límites de escalamiento. Si lo desea, puede añadir un escalamiento automático para ajustar el tamaño de su grupo.

Tamaño del grupo Info

Defina el tamaño inicial del grupo de escalamiento automático. Después de crear el grupo, puede cambiar su tamaño para satisfacer la demanda, ya sea en forma manual o mediante el escalamiento automático.

Tipo de capacidad deseado
Elija la unidad de medida para el valor de capacidad deseado. Las vCPU y la memoria (GiB) solo son compatibles con grupos de instancias mixtos configurados con un conjunto de atributos de instancia.

Unidades (número de instancias) [\[v\]](#)

Capacidad deseada
Especifique el tamaño de su grupo.

Escalado Info

Puede cambiar el tamaño de su grupo de escalamiento automático de forma manual o automática para cumplir con los cambios en la demanda.

Límites de escalamiento
Establezca límites sobre cuánto puede aumentarse o disminuirse la capacidad deseada.

Capacidad deseada mínima **Capacidad deseada máxima**

Capacidad igual o inferior a la deseada Capacidad igual o superior a la deseada

Escalamiento automático - opcional
Elija si desea utilizar una política de seguimiento de destino Info

Puede configurar otras políticas de escalado basadas en métricas y un escalado programado después de crear su grupo de escalamiento automático.

Sin políticas de escalamiento
Su grupo de escalamiento automático mantendrá su tamaño inicial y no se redimensionará de forma dinámica para satisfacer la demanda.

Política de escalado de seguimiento de destino
Elija una métrica y un valor objetivo de CloudWatch y deje que la política de escalamiento ajuste la capacidad deseada en proporción al valor de la métrica.

Creación de base de datos usando Aurora y RDS.

Para la creación y configuración de nuestra primera base de datos deberemos buscar “Aurora and RDS” para poder iniciar a configurarla, ya una vez dentro de la configuración de la base de datos tendremos dos opciones, crearla de manera estándar o crearla de manera sencilla, nosotros usaremos la manera estándar ya que nos ofrece mas opciones para hacerla a nuestro gusto. Siguiendo se nos pedirá que elijamos uno de los ocho motores de bases de datos que nos ofrece AWS, este proyecto usara MySQL ya que es una de las mas populares y sencillas de usar. Una vez elegido el motor de bases de datos toca seleccionar una plantilla para adaptarla a lo que vamos a usar, en este caso se usara la plantilla de la capa gratuita ya que este proyecto no se necesitara una base de datos más avanzada. Ahora toca nombrar la base de datos (por ejemplo “database-2”), le daremos un nombre de usuario maestro y una contraseña que nos servirá para conectarnos a la base de datos sin necesidad de acceder en Aurora RDS ya habiendo hecho eso tendríamos completa la base de datos.

Figura.20 Búsqueda de Aurora RDS.

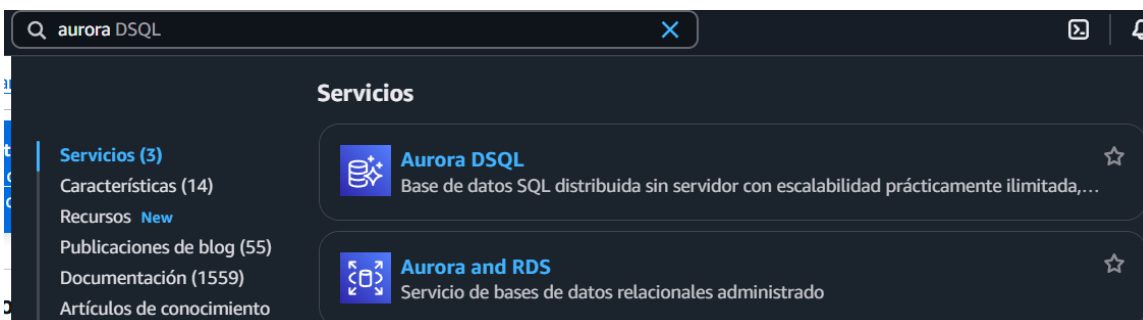


Figura.21 paso 1 para la creación de la base de datos.

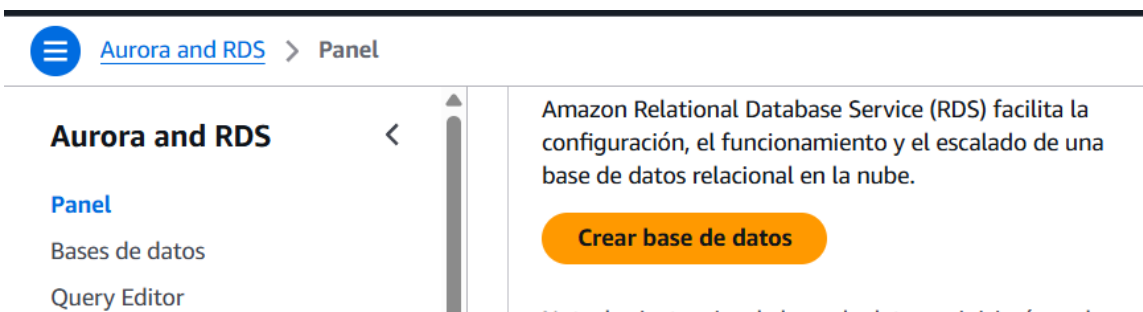


Figura.21 Creación y selección de motor de bases de datos.

Crear base de datos [Información](#)


Elegir un método de creación de base de datos


Creación estándar
Puede definir todas las opciones de configuración, incluidas las de disponibilidad, seguridad, copias de seguridad y mantenimiento.


Creación sencilla
Utilice las configuraciones recomendadas. Algunas opciones de configuración se pueden cambiar después de crear la base de datos.


Opciones del motor


Tipo de motor [Información](#)


Aurora (MySQL Compatible) 


Aurora (PostgreSQL Compatible) 


MySQL 

PostgreSQL 

MariaDB 

Oracle 

Microsoft SQL Server 

IBM Db2 

Edición

Comunidad de MySQL

Versión del motor [Información](#)
Vea las versiones de motor que admiten las siguientes características de base de datos.

▼ **Ocultar filtros**

Mostrar solo las versiones compatibles con el clúster de base de datos multi-AZ [Información](#)
Crea un clúster de base de datos multi-AZ con una instancia de base de datos principal y dos instancias de base de datos en espera que se pueden leer. Los clústeres de base de datos multi-AZ ofrecen una latencia de confirmación de transacciones hasta dos veces más rápida y conmutación por error automática en menos de 35 segundos.

Mostrar solo versiones compatibles con las escrituras optimizadas de Amazon RDS [Información](#)
Las escrituras optimizadas de Amazon RDS mejoran el rendimiento de escritura hasta 2 veces sin costo adicional.

Versión del motor

MySQL 8.0.40

Activar el soporte extendido de RDS [Información](#)
El soporte extendido de Amazon RDS es un [oferta pagada](#). Al seleccionar esta opción, acepta que se le cobre por esta oferta si utiliza la versión principal de la base de datos después de la fecha de finalización del soporte estándar de RDS para esa versión. Consulte la fecha de finalización del soporte estándar de su versión principal en el [Documentación de RDS para MySQL](#).

Figura.22 Selección de plantillas.

Plantillas

Elija una plantilla de ejemplo para adaptarla a su caso de uso.

Producción
Utilice los valores predeterminados para disfrutar de una alta disponibilidad y de un rendimiento rápido y constante.

Desarrollo y pruebas
Esta instancia se ha diseñado para su uso en desarrollo, fuera de un entorno de producción.

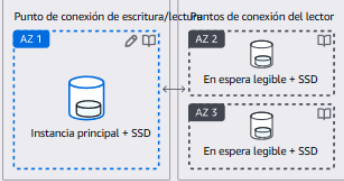
Capa gratuita
Utilice el nivel gratuito de RDS para desarrollar nuevas aplicaciones, probar aplicaciones existentes o adquirir experiencia práctica con Amazon RDS. [Información](#)

Disponibilidad y durabilidad

Opciones de implementación [Información](#)
Elija la opción de implementación que proporcione la disponibilidad y durabilidad necesarias en función del caso de uso. AWS se compromete a un determinado nivel de tiempo de actividad según la opción de implementación que elija. Obtenga más información en el [Acuerdo de nivel de servicios \(SLA\) de Amazon RDS](#).

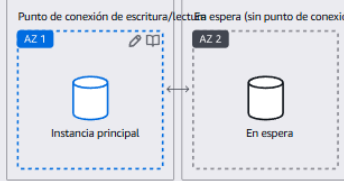
Implementación de clúster de base de datos multi-AZ (3 instancias)
Crea una instancia de base de datos principal con dos en espera legibles en zonas de disponibilidad separadas. Esta configuración proporciona:

- Tiempo de actividad del 99,5 %
- Redundancia entre zonas de disponibilidad
- Mayor capacidad de lectura
- Menor latencia de escritura



Implementación de instancias de base de datos multi-AZ (2 instancias)
Crea una instancia de base de datos principal con una instancia en espera no legible en una zona de disponibilidad independiente. Esta configuración proporciona:

- Tiempo de actividad del 99,5 %
- Redundancia entre zonas de disponibilidad



Implementación de una instancia de base de datos de zona de disponibilidad única (1 instancia)
Crea una única instancia de base de datos sin instancias en espera. Esta configuración proporciona:

- 99,5% uptime
- Sin redundancia de datos

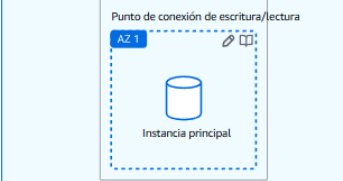


Figura.23 Configuración de la base de datos.

Configuración

Identificador de instancias de bases de datos [Información](#)

Escriba un nombre para la instancia de base de datos. El nombre debe ser único en relación con todas las instancias de base de datos pertenecientes a su cuenta de AWS en la región de AWS actual.

El identificador de la instancia de base de datos no distingue entre mayúsculas y minúsculas, pero se almacena con todas las letras en minúsculas (como en "mydbinstance"). Restricciones: de 1 a 63 caracteres alfanuméricos o guiones. El primer carácter debe ser una letra. No puede contener dos guiones consecutivos. No puede terminar con un guion.

▼ **Configuración de credenciales**

Nombre de usuario maestro [Información](#)

Escriba un ID de inicio de sesión para el usuario maestro de la instancia de base de datos.

1 a 16 caracteres alfanuméricos. El primer carácter debe ser una letra.

Administración de credenciales

Puede usar AWS Secrets Manager o administrar sus credenciales de usuario maestro.

Administrado en AWS Secrets Manager - más seguro
RDS genera una contraseña y la administra durante todo su ciclo de vida mediante AWS Secrets Manager.

Autoadministrado
Cree su propia contraseña o pida a RDS que cree una contraseña para que pueda administrarla.

Generar contraseña automáticamente
Amazon RDS puede generar una contraseña en su nombre, o bien puede especificar su propia contraseña.

Contraseña maestra | [Información](#)

Seguridad de la contraseña Neutro

Restricciones mínimas: al menos 8 caracteres ASCII imprimibles. No puede contener ninguno de los siguientes símbolos: / * @

Confirmar la contraseña maestra | [Información](#)

Configuración de la instancia

Las opciones de configuración de la instancia de base de datos que aparecen a continuación están limitadas a las que admite el motor que ha seleccionado anteriormente.

Clase de instancia de base de datos | [Información](#)

▼ **Ocultar filtros**

Mostrar las clases de instancia que admiten las escrituras optimizadas de Amazon RDS [Información](#)
Las escrituras optimizadas de Amazon RDS mejoran el rendimiento de escritura hasta 2 veces sin costo adicional.

Incluir clases de generación anterior

Clases estándar (incluye clases m)
 Clases optimizadas para memoria (incluye clases r y x)
 Clases ampliables (incluye clases t)

2 vCPUs 1 GiB RAM Red: hasta 2,085 Mbps

Figura.24 Finalización de la configuración de la base de datos.

Configuración de la instancia

Las opciones de configuración de la instancia de base de datos que aparecen a continuación están limitadas a las que admite el motor que ha seleccionado anteriormente.

Clase de instancia de base de datos | [Información](#)

▼ **Ocultar filtros**

Mostrar las clases de instancia que admiten las escrituras optimizadas de Amazon RDS [Información](#)
Las escrituras optimizadas de Amazon RDS mejoran el rendimiento de escritura hasta 2 veces sin costo adicional.

Incluir clases de generación anterior

Clases estándar (incluye clases m)
 Clases optimizadas para memoria (incluye clases r y x)
 Clases ampliables (incluye clases t)

2 vCPUs 1 GiB RAM Red: hasta 2,085 Mbps

Almacenamiento

Tipo de almacenamiento [Información](#)

Los volúmenes de almacenamiento SSD de IOPS aprovisionadas (io2) ya están disponibles.

Rendimiento de referencia determinado por el tamaño del volumen

Almacenamiento asignado [Información](#)

GiB

El valor de almacenamiento asignado debe ser de 20 GiB a 6,144 GiB

► **Configuración de almacenamiento adicional**

Creando Buckets con s3

Para la creación y configuración de nuestro Bucket deberemos buscarla como S3 en el buscador, al seleccionarla nos redirigirá a su página donde vamos a tener la opción de “crear bucket”. La configuración un bucket es una tarea sencilla ya que únicamente habría que darle un nombre ya que las demás opciones no las vamos a modificar estas se quedaran como vienen por defecto y eso sería todo con la configuración de esta.

Figura.25 Búsqueda de S3.

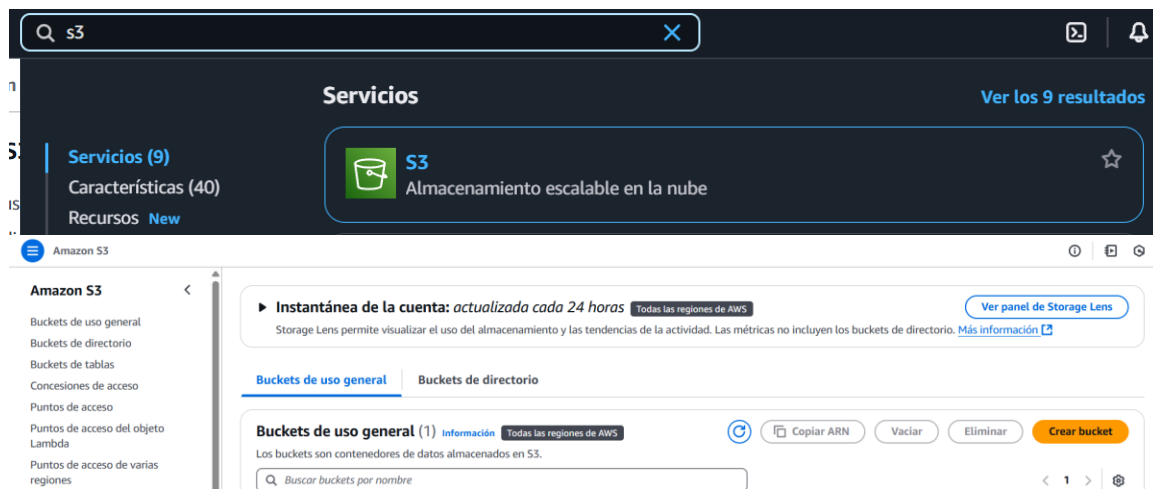


Figura.26 Creación y configuración del bucket.

Crear bucket [Información](#)
 Los buckets son contenedores de datos almacenados en S3.

Configuración general

Región de AWS
 EE.UU. Este (Ohio) us-east-2

Tipo de bucket [Información](#)

Uso general
 Recomendado para la mayoría de los casos de uso y patrones de acceso. Los buckets de uso general son del tipo de bucket de S3 original. Permite una combinación de clases de almacenamiento que almacenan objetos de forma redundante en múltiples zonas de disponibilidad.

Directorio
 Recomendado para casos de uso de baja latencia. Estos buckets utilizan únicamente la clase de almacenamiento S3 Express One Zone, que proporciona un procesamiento más rápido de los datos dentro de una única zona de disponibilidad.

Nombre del bucket [Información](#)

BK-web

Los nombres de los buckets deben tener entre 3 y 63 caracteres y ser únicos dentro del espacio de nombres global. Los nombres de los buckets también deben empezar y terminar con una letra o un número. Los caracteres válidos son a-z, 0-9, puntos (.) y guiones (-). [Más información](#)

Copiar la configuración del bucket existente: opcional
 Solo se copia la configuración del bucket en los siguientes ajustes.

[Elegir el bucket](#)

Formato: s3://bucket/prefijo

Propiedad de objetos [Información](#)

Controle la propiedad de los objetos escritos en este bucket desde otras cuentas de AWS y el uso de listas de control de acceso (ACL). La propiedad de los objetos determina quién puede especificar el acceso a los objetos.

ACL deshabilitadas (recomendada)
 Todos los objetos de este bucket son propiedad de esta cuenta. El acceso a este bucket y sus objetos se especifica solo mediante políticas.

ACL habilitadas
 Los objetos de este bucket pueden ser propiedad de otras cuentas de AWS. El acceso a este bucket y sus objetos se puede especificar mediante ACL.

Propiedad del objeto
 Aplicada al propietario del bucket

Configuración de bloqueo de acceso público para este bucket

Se concede acceso público a los buckets y objetos a través de listas de control de acceso (ACL), políticas de bucket, políticas de puntos de acceso o todas las anteriores. A fin de garantizar que se bloquee el acceso público a todos sus buckets y objetos, active Bloquear todo el acceso público. Esta configuración se aplica exclusivamente a este bucket y a sus puntos de acceso. AWS recomienda activar Bloquear todo el acceso público, pero, antes de aplicar cualquiera de estos ajustes, asegúrese de que las aplicaciones funcionarán correctamente sin acceso público. Si necesita cierto nivel de acceso público a los buckets u objetos, puede personalizar la configuración individual a continuación para adaptarla a sus casos de uso de almacenamiento específicos. [Más información](#)

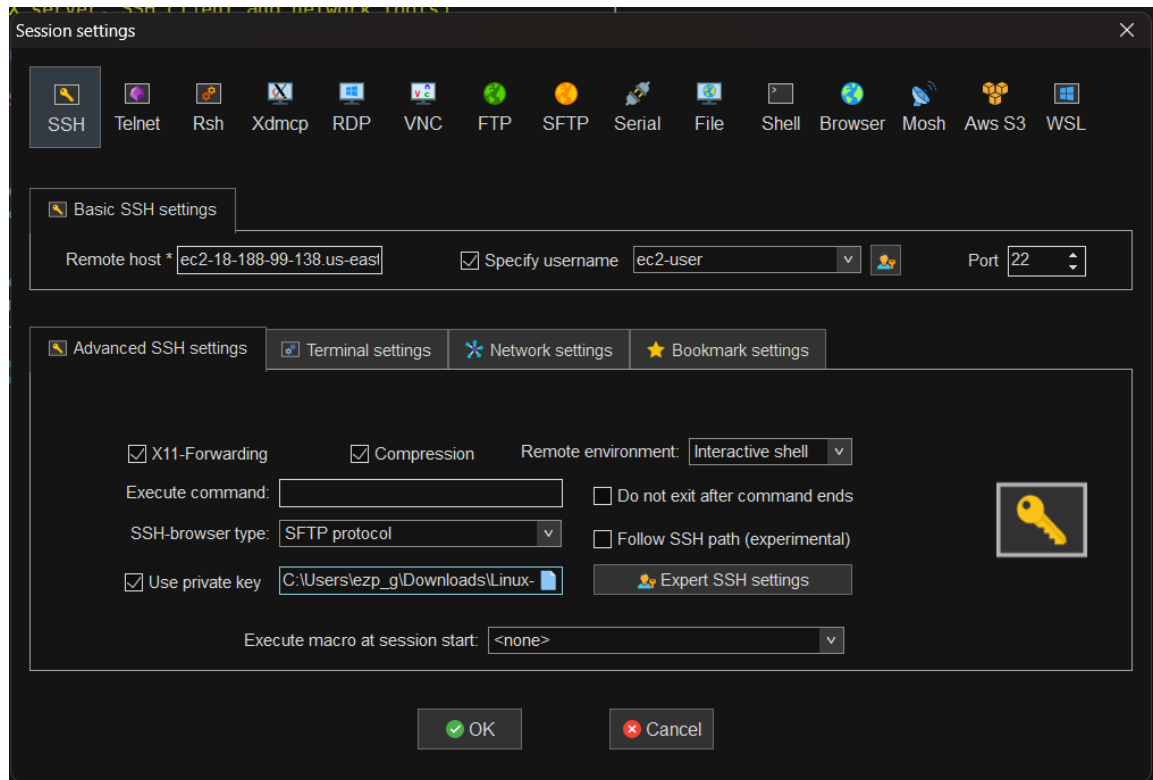
Bloquear todo el acceso público
 Activar esta configuración equivale a activar las cuatro opciones que aparecen a continuación. Cada uno de los siguientes ajustes son independientes entre sí.

- Bloquear el acceso público a buckets y objetos concedido a través de nuevas listas de control de acceso (ACL)**
 S3 bloqueará los permisos de acceso público aplicados a objetos o buckets agregados recientemente, y evitará la creación de nuevas ACL de acceso público para buckets y objetos existentes. Esta configuración no cambia los permisos existentes que permiten acceso público a los recursos de S3 mediante ACL.
- Bloquear el acceso público a buckets y objetos concedido a través de cualquier lista de control de acceso (ACL)**
 S3 ignorará todas las ACL que conceden acceso público a buckets y objetos.
- Bloquear el acceso público a buckets y objetos concedido a través de políticas de bucket y puntos de acceso públicas nuevas**
 S3 bloqueará las nuevas políticas de bucket y puntos de acceso que concedan acceso público a buckets y objetos. Esta configuración no afecta a las políticas ya existentes que permiten acceso público a los recursos de S3.
- Bloquear el acceso público y entre cuentas a buckets y objetos concedido a través de cualquier política de bucket y puntos de acceso pública**
 S3 ignorará el acceso público y entre cuentas en el caso de buckets o puntos de acceso que tengan políticas que concedan acceso público a buckets y objetos.

Creación del Clúster usando Docker.

Interfaz de MobaXterm para iniciar sesión y realizar la correcta configuración del clúster.

Figura.27 Creación y configuración del bucket.



Estos comandos se usan para instalar y habilitar el uso de Docker en el proyecto.

Figura.28 Instalación de Docker.

```
[root@ip-10-0-9-199 ec2-user]# yum install docker
```

Figura.29 Habilitación de Docker.

```
[root@ip-10-0-9-199 ec2-user]# system enable docker
```

Este comando va a descargar imágenes de los contenedores de la web de Docker, en este caso descargara la imagen de httpd.

Figura.30 Descarga de imagen httpd.

```
[root@ip-10-0-9-199 ec2-user]# docker pull httpd
```

Con este comando ya estaríamos creando nuestro primer contenedor.

Figura.31 Comando para crear un contenedor

```
[root@ip-10-0-9-199 ec2-user]# docker run -dit --name apachev1 -p 8080:80 httpd
```

Este comando es para instalar Nginx en el proyecto.

Figura.32 Comando para instalar nginx

```
[root@ip-10-0-9-199 ec2-user]# dnf install nginx
```

Estos comandos se usan para iniciar y habilitar el uso de nginx en el proyecto.

Figura.33 Comando para iniciar nginx.

```
[root@ip-10-0-9-199 ec2-user]# systemctl start nginx
```

Figura34. Comando para habilitar nginx.

```
[root@ip-10-0-9-199 ec2-user]# systemctl enable nginx
```

Este comando se usa para abrir el archivo en el que se va a configurar el balanceador de carga en Nginx.

Figura.35 Apertura de archivo nginx.conf.

```
[root@ip-10-0-9-199 ec2-user]# nano nginx.conf
```

Este bloque de configuración Nginx sirve como si fuera hacer un balanceo de carga entre varias aplicaciones que están ejecutándose en el mismo servidor, pero en diferentes puertos.

Figura.36 Balanceo de carga en Nginx.

```
events {}|
http {
    upstream aplicaciones{
        server localhost:8080;
        server localhost:8081;
        server localhost:8082;
    }|
server{
    listen 80;
    server_name nginx;
    location /{
        proxy_pass http://aplicaciones;
    }
}
}|
```

En este archivo de Nginx de deberá de tener esta configuración de dns para que al momento de buscar estas direcciones de manera local nos muestren alguna de las instancia que ya creamos.

Figura.37 Configuración de sitios locales

```
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com       # source server
#       38.25.63.10      x.acme.com          # x client host
#
# localhost name resolution is handled within DNS itself.
#   127.0.0.1             localhost
#   ::1                   localhost
# Added by Docker Desktop
192.168.1.206 host.docker.internal
192.168.1.206 gateway.docker.internal
# To allow the same kube context to work on the host and the container:
127.0.0.1 kubernetes.docker.internal
# End of section
3.95.214.60 www.exito.com
3.95.214.60 www.dl.com
3.95.214.60 www.tiendal.com
```

Figura.38 Corroboración de que todo lo que hemos venido haciendo funciona de manera correcta



Conclusiones

La implementación de una infraestructura en la nube usando los servicios gratuitos que ofrece Amazon Web Services (AWS) se logró demostrar que es una solución eficiente, escalable y Segura para la gestión de recursos informáticos. En el transcurso del Proyecto se logró desplegar de manera exitosa una instancia EC2, realizar y configurar balanceadores de carga (Load Balancer) y escaladores automáticos (Auto Scaling), lo cual nos permitió adaptar la infraestructura al demandante tráfico de manera dinámica. De igual forma fue posible realizar la integración de otros servicios como Amazon S3, Amazon Aurora y RDS para la base de datos y almacenamiento de información ayudándonos de herramientas externas como Docker, Nginx y MobaXterm lo que nos facilitó realizar una correcta administración y configuración del Sistema.

En este trabajo puse en práctica la combinación de varias tecnologías y servicios lo que demostró que puede mejorar notablemente el rendimiento, la disponibilidad y seguridad de una aplicación, solamente haciendo uso de la capa gratuita que ofrece Amazon Web Service. La experiencia que obtuve en la creación del Proyecto me ayudó a ver que lo importante de seguir las mejores prácticas en lo que viene ser la gestión de recursos en AWS y el gran potencial que tienen los servicios en la nube. Para finalizar, el proyecto no solo cumplió con su objetivo, sino que también me sirvió como aprendizaje y tener un valioso conocimiento en el uso de las nuevas tecnologías y lo importante que son tener y poder crear infraestructuras escalables y confiables

Referencias

- Amazon Web Services. (2025). *Guía para desarrolladores: Amazon Elastic Computed Cloud* <https://docs.aws.amazon.com/pdfs/AWSEC2/latest/UserGuide/ec2-ug.pdf#concepts>
- Amazon Web Services. (2025). *Guía para desarrolladores: Amazon Virtual Private Cloud* <https://docs.aws.amazon.com/pdfs/vpc/latest/userguide/vpc-ug.pdf#what-is-amazon-vpc>
- Amazon Web Services. (2025). *Guía para desarrolladores: Amazon Simple Storage Service* <https://docs.aws.amazon.com/pdfs/AmazonS3/latest/userguide/s3-userguide.pdf#Welcome>
- Amazon Web Services. (2025). *Guía para desarrolladores: Amazon Relational Database Service* <https://docs.aws.amazon.com/pdfs/AmazonRDS/latest/UserGuide/rds-ug.pdf#Welcome>
- Amazon Web Services. (2025). *Guía para desarrolladores: Amazon Aurora* https://docs.aws.amazon.com/pdfs/AmazonRDS/latest/AuroraUserGuide/aurora-ug.pdf#CHAP_AuroraOverview
- MobaXterm (2025). *Documentación de MobaXterm: MobaExterm* <https://mobaxterm.mobatek.net/documentation.html#1>
- Docker (2025). *Documentación de Docker: Docker* <https://docs.docker.com/get-started/docker-overview/>
- Docker (2025). *Docker official images:* https://hub.docker.com/_/httpd