



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

**Informe técnico de Amazon Web Server**

Corporación Universitaria Remington.  
Facultad de ingeniería:  
Ingeniería de sistemas

Jhon Jaider Pérez Acosta, Eider Dubán Gómez Aristizábal, Anderson Durango Julio  
Juan Pablo Berrio López  
Opción de Trabajo de grado Seminario-Diplomado.  
2024

## Tabla de Contenidos

Tabla de figuras .....	4
Índice de tablas .....	6
Resumen .....	7
Marco conceptual y contextual .....	9
Amazon EC2 (Elastic Compute Cloud) .....	9
Amazon S3 (Simple Storage Service) .....	9
Amazon EBS (Elastic Block Store) .....	9
Amazon RDS (Relational Database Service) .....	9
Amazon DynamoDB .....	10
Amazon VPC (Virtual Private Cloud) .....	10
AWS Identity and Access Management (IAM) .....	10
Alojamiento Web y Aplicaciones .....	10
Desarrollo e implementación del aprendizaje.....	11
Historia de la virtualización y computación en la nube .....	11
Línea de tiempo.....	13
Comparación servicios AWS, Azure y GCP .....	13
Creación de la VPC .....	20
Creación de las instancias .....	20
Redes.....	22
Claves .....	23
Archivo Hosts.....	24
Configuración .....	24
Configuración virtual host.....	25
Comandos .....	25
Procedimiento .....	27
Resultados .....	28
Habilitación de IAM, implementación de S3 y RDS .....	31
Creación de usuario .....	32
Creación de grupo.....	32
Acceso de usuario .....	33
Instalación AWS CLI.....	34
Creación bucket .....	35
Creación de política para acceso al bucket.....	36
Creación del Rol .....	37
Archivo subido a S3 .....	37
Creación Rds .....	39
Modificación de las reglas para acceder a la base de datos .....	40
Instalación cliente MySQL .....	40
Base de datos de Rds.....	42

	3
Creación de balanceadores de carga y grupo de auto escalado .....	42
Conclusiones .....	53
Referencias .....	54

## Tabla de figuras

Figura 1	Línea de tiempo .....	13
Figura 2	VCP.....	20
Figura 3	Instancia de linux creada.....	21
Figura 4	detalles instancia.....	22
Figura 5	Reglas de conexión de red .....	23
Figura 6	Clave.....	23
Figura 7	Configuración Host local .....	24
Figura 8	configuración apache .....	24
Figura 9	configuración virtual host .....	25
Figura 10	WSL.....	27
Figura 11	comandos de instalación y el proceso de instalación del HTTP apache .....	28
Figura 12	página puerto 8080.....	28
Figura 13	página puerto 8082.....	29
Figura 14	página con nombre y puerto 8080 .....	29
Figura 15	página consultada con nombre .....	30
Figura 16	página con consulta asignada.....	30
Figura 17	nombre de página y puerto 8082 .....	31
Figura 18	Habilitación de IAM .....	31
Figura 19	Creación de usuario .....	32
Figura 20	creación de grupo .....	33
Figura 21	Acceso de usuario .....	33
Figura 22	Instalación CLI .....	34
Figura 23	creación bucket s3 .....	35
Figura 24	bucket creado .....	36
Figura 25	política de S3.....	36
Figura 26	creación de rol.....	37
Figura 27	subida de archivo a s3.....	38
Figura 28	archivo en bucket .....	38
Figura 29	creación de base de datos.....	39
Figura 30	configuración de puertos de la base de datos .....	40
Figura 31	instalación MySQL .....	41
Figura 32	conexión a base de datos.....	41
Figura 33	base de datos RDS.....	42
Figura 34	creación instancia de Windows.....	43
Figura 35	reglas de grupo de seguridad .....	43
Figura 36	instalación IIS .....	44
Figura 37	edición de página web .....	45
Figura 38	creación de imagen a partir de instantánea .....	45
Figura 39	creación de instancia a partir de imagen .....	46
Figura 40	edición de página web instancia 2 .....	47
Figura 41	creador de balanceador de cargas .....	47
Figura 42	configuración de balanceador .....	48
Figura 43	creación de grupo de destino.....	48
Figura 44	prueba de balanceador de carga .....	49

	5
Figura 45 creación de grupo de auto escalado .....	50
Figura 46 configuración de política de auto escalado .....	50
Figura 47 forzado de CPU a 100% .....	51
Figura 48 ejecución de política .....	52
Figura 49 instancias creadas automáticamente.....	52

**Índice de tablas**

Tabla 1 Comparación de servicios..... 14

## Resumen

La computación en la nube es un modelo de entrega de servicios de computación permitiendo el acceso a recursos tales como servidores, bases de datos, almacenamientos y software sin la necesidad de tener la infraestructura o sistema físico.

En este campo, AWS (Amazon Web Services) es la plataforma líder en servicios de computación en la nube la cual ofrece servicios los cuales incluyen computo, almacenamiento, base de datos, redes, inteligencia artificial y otras más permitiendo a los usuarios que la utilicen implementar aplicaciones de manera más rápida y confiable utilizando herramientas avanzadas para la gestión y análisis de datos.

En base a lo dicho anteriormente se mostrará la implementación de una estancia EC2 con dos contenedores en la nube implicando que se cree una VPC y las instancias necesarias, configurando subredes y asignando direcciones IP. Se detallarán los pasos para crear y configurar instancias EC2, seleccionando el sistema operativo, tipo de instancia, creación de claves y configuración de las reglas de seguridad.

Finalmente, se describirán los procesos de instalación y configuración de Apache HTTP Server en Linux, incluyendo la configuración de host virtuales y proxys explicándose los pasos para conectar una instancia AWS en Linux a un bucket S3 y una base de datos RDS MySQL, la creación de usuarios y roles en IAM, la instalación de AWS CLI y MySQL y creación de políticas y reglas de seguridad. Estos procesos lograrán una eficiente gestión de los recursos en la nube, demostrando flexibilidad y potencia de la virtualización y la computación en la nube.

También se demostrará la funcionalidad de los balanceadores de carga y grupos de auto escalado, los cuales permiten mantener aplicaciones montadas en instancia siempre disponibles,

monitoreando el estado de las instancias y reemplazándolas automáticamente si se detectan como no funcionales.

Todo esto reflejara las ventajas que ofrece AWS como servicio de computación en la nube, adaptándose a cada necesidad que tenga el usuario y ofreciendo una solución óptima para cada escenario.

### **Palabras clave**

- **Amazon web server**
- **Base de datos**
- **Cloud computing**
- **Bucket**
- **Servidor virtual**



## **Marco conceptual y contextual**

### **Amazon EC2 (Elastic Compute Cloud)**

Según Amazon (2024), Lo define como un servicio fundamental de computo en la nube ofrecido por Amazon Web Services (AWS). Permite a los usuarios alquilar maquinas virtuales llamadas instancias en la nube lo que proporciona capacidad informática escalable y flexible según las necesidades de los usuarios.

### **Amazon S3 (Simple Storage Service)**

Según Amazon (2024), se define como servicio de almacenamiento de objetos que ofrece Amazon Web Services (AWS) permitiendo a los usuarios almacenar y recuperar grandes cantidades de datos de forma segura y escalable a través del internet. Ofrece una durabilidad y escalabilidad lo que garantiza que los datos estén siempre accesibles cuando se los necesita.

### **Amazon EBS (Elastic Block Store)**

Según Amazon (2024), se refiere a almacenamiento en bloque persistente diseñado para proporcionar al espacio de datos persistentes para instancias de Amazon EC2. Ofrece diferentes tipos de volúmenes los cuales poseen características específicas de rendimiento y precio como volúmenes de propósito general.

### **Amazon RDS (Relational Database Service)**

Según Amazon (2024), se define servicio que simplifica el despliegue, operación y escalabilidad de base de datos relacionales en la nube permitiendo a los usuarios configurar fácilmente las instancias de base de datos, escalar la capacidad de almacenamiento y administrar tareas tales como mantenimiento, parches de software y copias de seguridad.

### **Amazon DynamoDB**

Según Amazon (2024), es definido como servicio de base de datos NoSQL que ofrece un rendimiento rápido y disponible siendo una opción popular para aplicaciones que necesiten un almacenamiento de datos de alto rendimiento y flexible el cual pueda manejar cargas de trabajo de cualquier tamaño desde aplicaciones pequeñas hasta sistemas distribuidos.

### **Amazon VPC (Virtual Private Cloud)**

Según Amazon (2024), Se define como un servicio de red que permite a los usuarios crear una red virtual aislada en la nube dando la oportunidad al usuario de definir su propia infraestructura de red lanzando recursos como lo son base de datos, instancias, etc. dentro de su propia red virtual

### **AWS Identity and Access Management (IAM)**

Según Amazon (2024), se trata de un servicio de gestión de identidades que permite a los usuarios controlar y asegurar el acceso a los recursos de (AWS) dando la posibilidad de crear y gestionar usuarios y grupos, así como identidades separadas para personas, aplicaciones o servicios.

### **Alojamiento Web y Aplicaciones**

Según Amazon (2023), Se define como el proceso de almacenar, gestionar y hacer accesible el contenido Web y aplicaciones a través de servidores conectados a internet. El alojamiento puede variar de servicios compartidos donde distintos sitios comparten el recurso de un mismo servidor hasta soluciones dedicadas que proporcionan servicios exclusivos para un solo cliente.

## **Desarrollo e implementación del aprendizaje**

### **Historia de la virtualización y computación en la nube**

El origen de la virtualización se da en los días de los PC llamados Mainframe en la década de los 60s cuando cada una de las piezas compuestas de hardware solo tenía la capacidad de realizar una acción a la vez o dicho de otra manera podían realizar un solo proceso a la vez. Pasado los años los clientes la exigencia de los clientes subió a tal nivel que pedían que se ampliaran los procesos y cantidad de usuarios que pudiera ese actual PC mainframe.

No sería hasta el final de la década de los 60s que IBM siendo la empresa reconocida en ese entonces desarrollara un sistema operativo CP-67 siendo un hipervisor inicial que introdujo la memoria virtual a la familia de servidores System 360 pero también habiendo otras soluciones para que múltiples usuarios pudieran trabajar en un solo servidor ocasionando que la tecnología perdiera intensidad durante varias décadas.

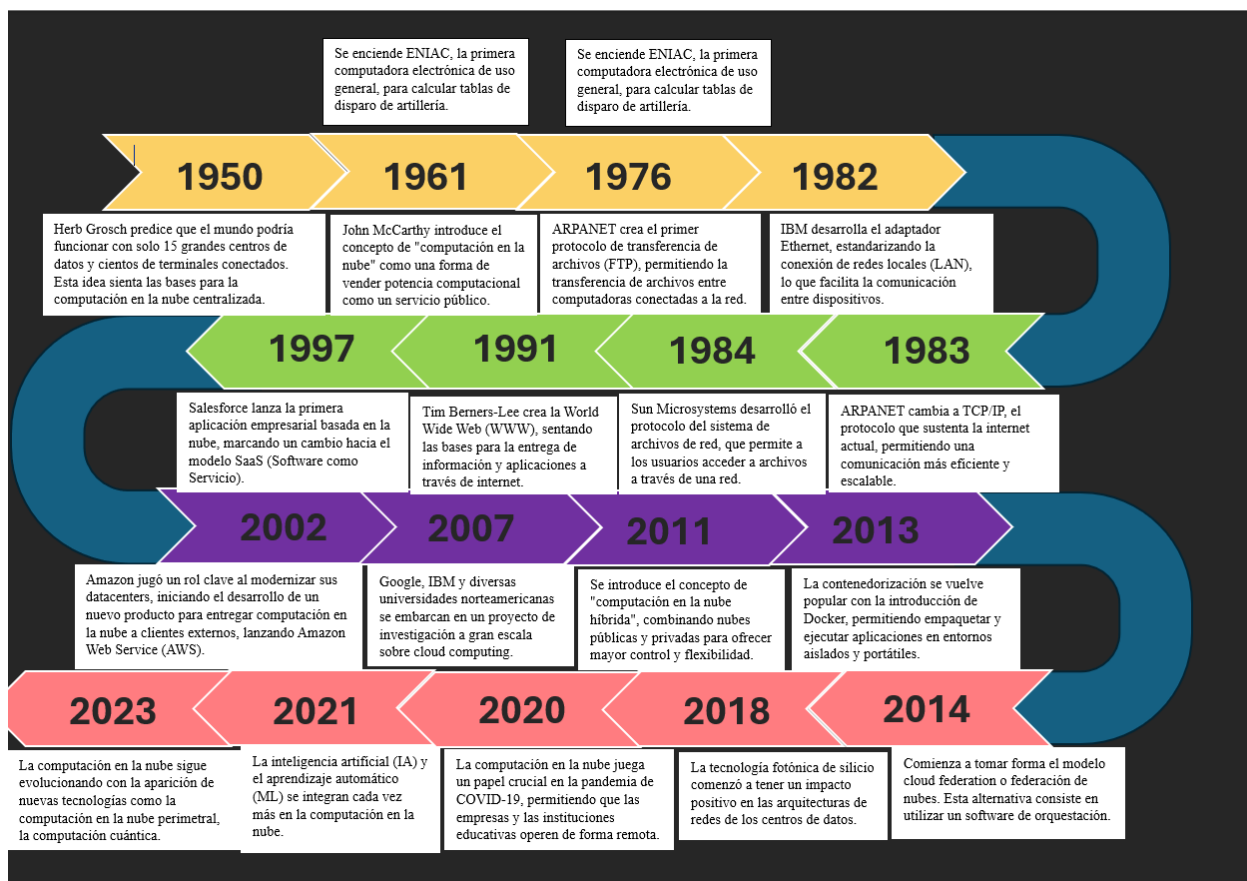
Pasaron los años y en la época de los 90s cuando muchas empresas tenían el desafío de mantener al día con pilas TI (tecnología de la información) de un proveedor y aplicaciones heredadas se dieron cuenta de la necesidad de usar sus recursos de medidores los cuales no estaban siendo utilizados frecuentemente. Al implementar la virtualización no solo dividieron su infraestructura de una mejor manera, sino que ejecutaron sus aplicaciones las cuales eran heredadas en tipos y versiones de OS (Sistema operativo) diferentes debido a su gran red compuesta por muchos tipos de PC en los cuales funcionaran distintos sistemas operativos y con ayuda del crecimiento del internet impulsar a adopción de la virtualización. A medida que la virtualización se comenzó a utilizar con mayor frecuencia lo cual redujo la dependencia del proveedor por los servidores y lo tomaron como base para el desarrollo de la computación en la nube.

Es importante mencionar que aquellos que estuvieron implicados en la creación de la computación en la nube serian primeramente el informático estadounidense Joseph Carl Robnett Licklider, junto con Bob Taylor siendo una de las figuras del desarrollo del internet y Larry Roberts e cual fue un científico estadounidense considerado el padre de la internet, dichos personajes presentaron el proyecto en 1969 dicho anteriormente a finales de los 60s, mientras pasaban los años se pudieron ver grandes avances tecnológicos tales como (IBM, primera máquina virtual en 1972, VPN Redes privadas virtualizadas en los 90s y ya hasta la actualidad con el cloud computing).

El cloud computing está creando oportunidades nuevas siendo importante para empresas o personas las cuales estén interesados en trabajar en la disciplina de la computación en la nube permitiendo una mayor flexibilidad en el trabajo ya sea en cualquier momento o lugar, acceder a datos que necesitemos y disponer y usar herramientas directamente desde la nube. Sin embargo, actualmente la demanda de profesionales que sean expertos en manejo de redes y procesos de la empresa lo que es un beneficio para los independientes que busquen empleos de este tipo, concluyendo de esta manera la historia de la virtualización y computación en la nube.

## Línea de tiempo

Figura 1 Línea de tiempo



Fuente: Autoría propia

## Comparación servicios AWS, Azure y GCP

En el panorama actual de la tecnología, Amazon Web Services (AWS), Microsoft Azure y Google Cloud Platform (GCP) se rigen como los tres principales proveedores de servicios en la nube. Cada uno ofrece una amplia gama de servicios y herramientas para cubrir las necesidades de empresas y organizaciones de diversos tamaños e industrias.

Tabla 1 Comparación de servicios

	<b>AWS</b>	<b>AZURE</b>	<b>GSP</b>
<b>PERSONA INFLUYENTE</b>	Andy Jassy, ex-CEO de AWS y actual CEO de Amazon, ha sido una figura clave en el desarrollo y crecimiento de AWS, impulsando su adopción global.	Scott Guthrie, vicepresidente Ejecutivo de la División Cloud + AI de Microsoft, ha sido un líder fundamental en la expansión de Azure y en la integración de tecnologías avanzadas como la inteligencia artificial y la computación en la nube.	Thomas Kurian, CEO de Google Cloud, ha jugado un papel crucial en la estrategia y expansión de GCP, enfocándose en la transformación digital y el crecimiento empresarial.
<b>APLICACIONES EMPRESARIALES</b>	<p><b>Netflix:</b> Utiliza AWS para la entrega de contenido global y el análisis de datos masivo.</p> <p><b>Airbnb:</b> Confía en AWS para escalar su infraestructura y ofrecer una experiencia de usuario consistente.</p> <p><b>General Electric:</b> Implementa AWS para mejorar sus operaciones industriales y análisis de datos.</p>	<p><b>Adobe:</b> Usa Azure para su suite Creative Cloud, mejorando la colaboración y la integración de servicios.</p> <p><b>HP:</b> Implementa Azure para transformar sus servicios y productos, integrando IoT y machine learning.</p> <p><b>Walmart:</b> Utiliza Azure para modernizar sus aplicaciones y mejorar la eficiencia operativa.</p>	<p><b>Twitter:</b> Aprovecha GCP para gestionar y analizar grandes volúmenes de datos en tiempo real.</p> <p><b>PayPal:</b> Utiliza GCP para mejorar la seguridad y la capacidad de procesamiento de transacciones.</p> <p><b>Spotify:</b> Confía en GCP para escalar su plataforma y ofrecer recomendaciones personalizadas a los usuarios.</p>

<b>VENTAJAS</b>	<p><b>Amplia gama de servicios</b></p> <p><b>Escalabilidad y confiabilidad</b></p> <p><b>Flexibilidad</b></p>	<p><b>Fuerte integración con Microsoft</b></p> <p><b>Herramientas de desarrollo familiares</b></p> <p><b>Precios competitivos</b></p>	<p><b>Innovación líder a la industria</b></p> <p><b>Precios Competitivos</b></p> <p><b>Cultura de código abierto</b></p> <p><b>Infraestructura global de Google</b></p>
<b>DESVENTAJAS</b>	<p><b>Complejidad</b></p> <p><b>Precios</b></p> <p><b>Falta de enfoque en la integración empresarial</b></p>	<p><b>Menor gama de servicios</b></p> <p><b>Ecosistema menos maduro</b></p> <p><b>Preocupaciones de dependencia de proveedores</b></p>	<p><b>Menor madurez del mercado</b></p> <p><b>Ecosistema de servicios limitados</b></p> <p><b>Curva de aprendizaje más pronunciada</b></p>
<b>COMPUTACION</b>	<p><b>AWS:</b> destaca por sus opciones flexibles y escalables.</p> <p><b>Amazon EC2:</b> permite lanzar instancias de servidores virtuales con configuraciones personalizables.</p> <p><b>AWS Lambda:</b> ofrece computación sin servidor, ejecutando código en</p>	<p><b>Azure Virtual Machines:</b> para ejecutar aplicaciones y cargas de trabajo en máquinas virtuales escalables.</p> <p><b>Azure Functions:</b> permite la computación sin servidor, ejecutando código sin preocuparse por la infraestructura subyacente.</p>	<p><b>Google Compute Engine:</b> que permite ejecutar instancias en la infraestructura de Google.</p> <p><b>Google Cloud Functions:</b> es la alternativa sin servidor, ideal para ejecutar funciones en respuesta a eventos.</p> <p><b>Google Kubernetes Engine (GKE):</b></p>

	<p>respuesta a eventos sin la necesidad de gestionar servidores.</p> <p><b>Amazon ECS:</b> facilita la gestión de aplicaciones en contenedores, soportando Docker y Kubernetes.</p>	<p><b>Azure Kubernetes Service (AKS):</b> es una solución gestionada para desplegar y gestionar aplicaciones en contenedores usando Kubernetes.</p>	<p>facilita la orquestación de aplicaciones en contenedores utilizando Kubernetes.</p>
<b>ALMACENAMIENTO</b>	<p><b>Amazon S3:</b> es conocido por su almacenamiento de objetos escalable y duradero.</p> <p><b>Amazon EBS:</b> proporciona almacenamiento de bloques de alto rendimiento para instancias de EC2. <b>Amazon Glacier:</b> es ideal para el archivo de datos a bajo costo para almacenamiento a largo plazo.</p>	<p><b>Azure Blob Storage:</b> permite almacenar grandes cantidades de datos no estructurados.</p> <p><b>Azure Disk Storage:</b> es una solución de almacenamiento de bloques de alto rendimiento para máquinas virtuales. <b>Azure Archive Storage:</b> ofrece almacenamiento de bajo costo para datos que se acceden raramente.</p>	<p><b>Google Cloud Storage:</b> es una solución de almacenamiento de objetos altamente escalable y segura. <b>Persistent Disk:</b> ofrece almacenamiento de bloques para instancias de Compute Engine. <b>Google Cloud Archive Storage:</b> es adecuado para el almacenamiento de datos a largo plazo y de bajo costo.</p>
	<p><b>Amazon RDS:</b> proporciona una plataforma gestionada para bases de datos relacionales como</p>	<p><b>Azure SQL Database:</b> es una base de datos SQL totalmente gestionada y escalable.</p>	<p><b>Cloud SQL:</b> ofrece bases de datos relacionales gestionadas, compatibles con MySQL,</p>



<b>BASES DE DATOS</b>	<p>MySQL, PostgreSQL y SQL Server.</p> <p><b>Amazon DynamoDB:</b> es una base de datos NoSQL gestionada con escalabilidad automática.</p> <p><b>Amazon Redshift:</b> es un almacén de datos en la nube optimizado para análisis de datos a gran escala.</p>	<p><b>Azure Cosmos DB:</b> es una base de datos NoSQL distribuida globalmente, que ofrece baja latencia y alta disponibilidad.</p> <p><b>Azure Synapse Analytics:</b> (anteriormente SQL Data Warehouse) es una solución de análisis y almacenamiento de datos.</p>	<p>PostgreSQL y SQL Server.</p> <p><b>Cloud Bigtable:</b> es una base de datos NoSQL para cargas de trabajo analíticas a gran escala.</p> <p><b>BigQuery:</b> es un almacén de datos gestionado que permite análisis de datos a gran escala con una arquitectura sin servidor.</p>
<b>REDES</b>	<p><b>Amazon VPC:</b> permite crear redes privadas virtuales.</p> <p><b>AWS Direct Connect:</b> ofrece una conexión de red dedicada desde las instalaciones del cliente a AWS, mejorando la latencia y el rendimiento.</p> <p><b>Amazon Route 53:</b> es un servicio de DNS y gestión de tráfico.</p>	<p><b>Azure Virtual Network:</b> permite la creación de redes virtuales, proporcionando aislamiento y segmentación de red.</p> <p><b>Azure ExpressRoute:</b> establece conexiones privadas entre Azure y la infraestructura local.</p> <p><b>Azure DNS:</b> es un servicio de DNS escalable y seguro.</p>	<p><b>Google Cloud VPC:</b> ofrece redes virtuales globales escalables.</p> <p><b>Cloud InterConnect:</b> proporciona una conexión de red dedicada y de alta capacidad.</p> <p><b>Cloud DNS:</b> es un servicio de DNS gestionado que ofrece infraestructura escalable y segura.</p>
	<p><b>Amazon SageMaker:</b> es una plataforma</p>	<p><b>Azure Machine Learning:</b> ofrece herramientas para</p>	<p><b>Google AI Platform:</b> ofrece herramientas para</p>

<p><b>INTELIGENCIA ARTIFICIAL Y MACHING LEARNING</b></p>	<p>completa para construir, entrenar y desplegar modelos de machine learning. <b>AWS Rekognition:</b> utiliza aprendizaje profundo para análisis de imágenes y videos. <b>AWS Lex:</b> permite construir interfaces de conversación, como las utilizadas por Amazon Alexa.</p>	<p>desarrollar, entrenar y desplegar modelos de machine learning. <b>Azure Cognitive Services:</b> proporciona API y servicios que agregan capacidades de IA como visión, habla y lenguaje. <b>Azure Bot Service:</b> es una plataforma para construir y gestionar bots inteligentes.</p>	<p>desarrollar, entrenar y desplegar modelos de machine learning. <b>Cloud Visión API:</b> proporciona análisis de imágenes basado en machine learning. <b>Dialogflow:</b> es una plataforma para construir interfaces conversacionales y chatbots.</p>
<p><b>HERRAMIENTAS DE DESARROLLO</b></p>	<p><b>AWS CodeBuild:</b> es un servicio de integración continua. <b>AWS CodePipeline:</b> automatiza las fases de construcción, prueba y despliegue de aplicaciones. <b>AWS Cloud9:</b> es un entorno de desarrollo integrado basado en la nube.</p>	<p><b>Azure DevOps:</b> es un conjunto de servicios de desarrollo colaborativo. <b>Azure App Service:</b> proporciona una plataforma para construir y hospedar aplicaciones web y móviles. <b>Visual Studio Online:</b> (ahora Visual Studio Codespaces) es un entorno de</p>	<p><b>Cloud Build:</b> es un servicio de CI/CD que compila, prueba y despliega software. <b>Google App Engine:</b> permite construir y desplegar aplicaciones web y móviles sin gestionar la infraestructura. <b>Cloud Code:</b> ofrece herramientas para desarrollar, depurar y</p>

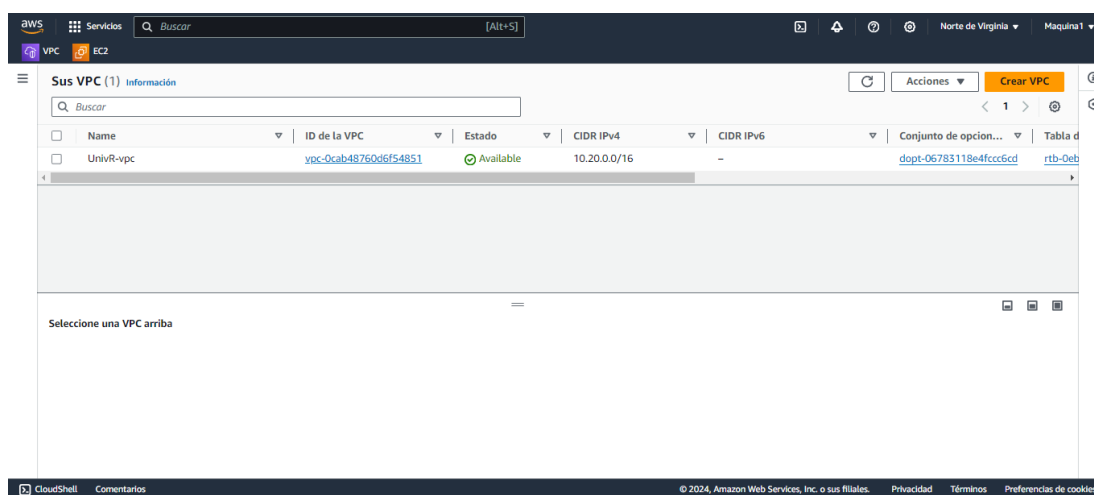
		desarrollo basado en la nube.	desplegar aplicaciones Kubernetes.
<b>SERVICIOS ESPECIFICOS DE APLICACIONES</b>	<p><b>Amazon Chime:</b> es un servicio de comunicación unificada.</p> <p><b>Amazon WorkSpaces:</b> ofrece escritorios como servicio.</p> <p><b>AWS Amplify:</b> proporciona herramientas para construir aplicaciones móviles y web.</p>	<p><b>Microsoft Teams:</b> es una plataforma de colaboración y comunicación.</p> <p><b>Windows Virtual Desktop:</b> permite desplegar y escalar escritorios virtuales. <b>Azure App Service:</b> es una plataforma para aplicaciones web, móviles y API.</p>	<p><b>Google Meet:</b> es una plataforma de videoconferencias y comunicación.</p> <p><b>Google Workspace:</b> (anteriormente G Suite) es un conjunto de herramientas de productividad y colaboración.</p> <p><b>Firebase:</b> proporciona servicios para construir aplicaciones móviles y web.</p>

Fuente: Autoría propia

## Creación de la VPC

Para crear el VPC elegimos una región, dentro de esa región se creó el componente VPC en el que se crearan mis recursos que va a tener un direccionamiento IP en el que se crearan las subredes y sobre las subredes se crearan los recursos que tendrá un direccionamiento 10.20.0.0/16 ipv4 las cuales tendrá 16 bits reservadas para la dirección de red y las otras 16 estarán reservadas para diferentes recursos que serán creados.

*Figura 2 VCP*



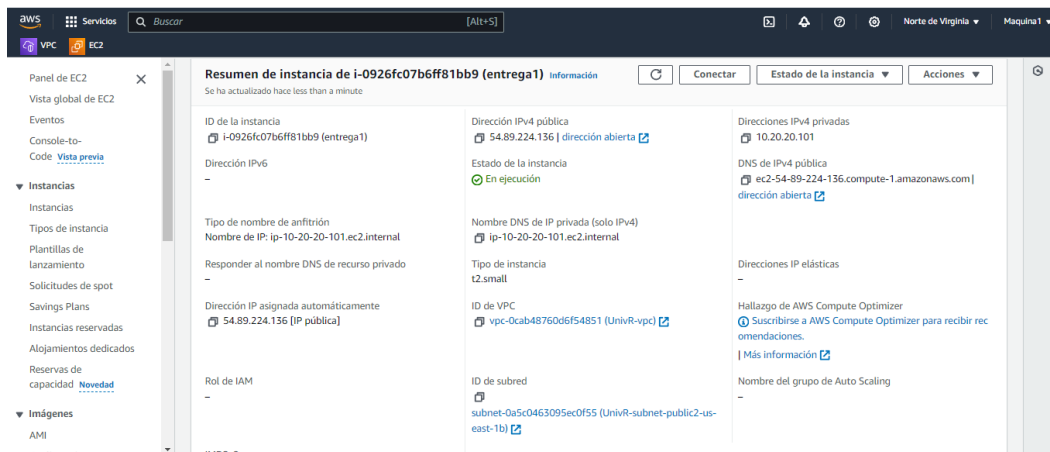
Fuente: Autoría Propia

## Creación de las instancias

La creación de las instancias se realizar escribiendo en el buscador de AWS las siglas EC2 e inmediatamente dando en la opción de instancias, y dentro de esta opción, se da en la opción de la esquina superior derecha llamada “Lanzar instancias” teniendo como primera opción el nombre, seguido de los sistemas operativos que se van a utilizar teniendo como opción varios sistemas operativos y escogiendo AWS Linux.

En la siguiente opción se escoge el tipo de instancia que es lo que se debe tener claro lo que se va a utilizar, se escoge el tipo de instancia gratuita llamada **t2.small** y se procede también a crear la **key pair** , grupo de seguridad, el tráfico sin requisitos y el espacio de disco duro.

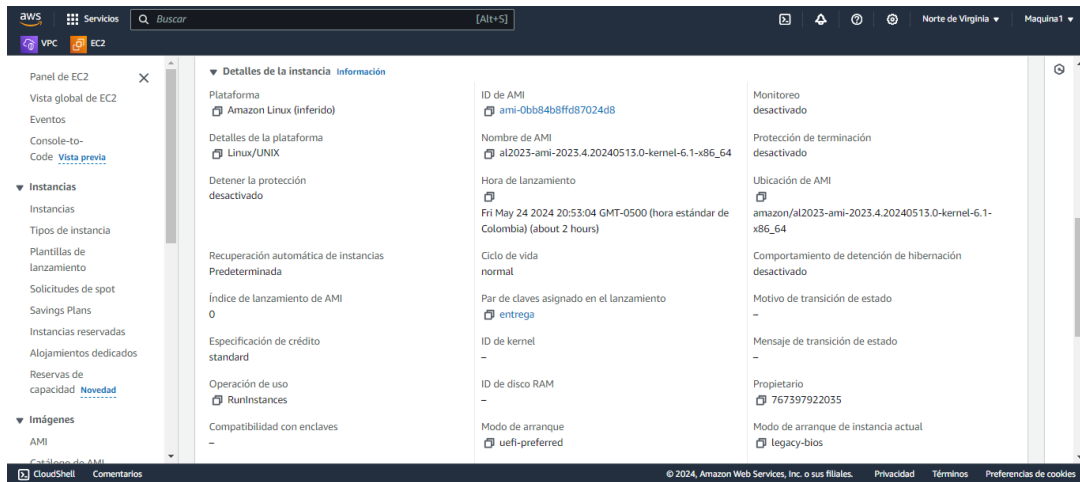
*Figura 3 Instancia de linux creada*



Fuente: Autoría propia

En la siguiente opción se escoge el tipo de instancia que es lo que se debe tener claro lo que se va a utilizar, se escoge el tipo de instancia gratuita llamada **t2.small** y se procede también a crear la **key pair** , grupo de seguridad, el tráfico sin requisitos y el espacio de disco duro.

Figura 4 detalles instancia

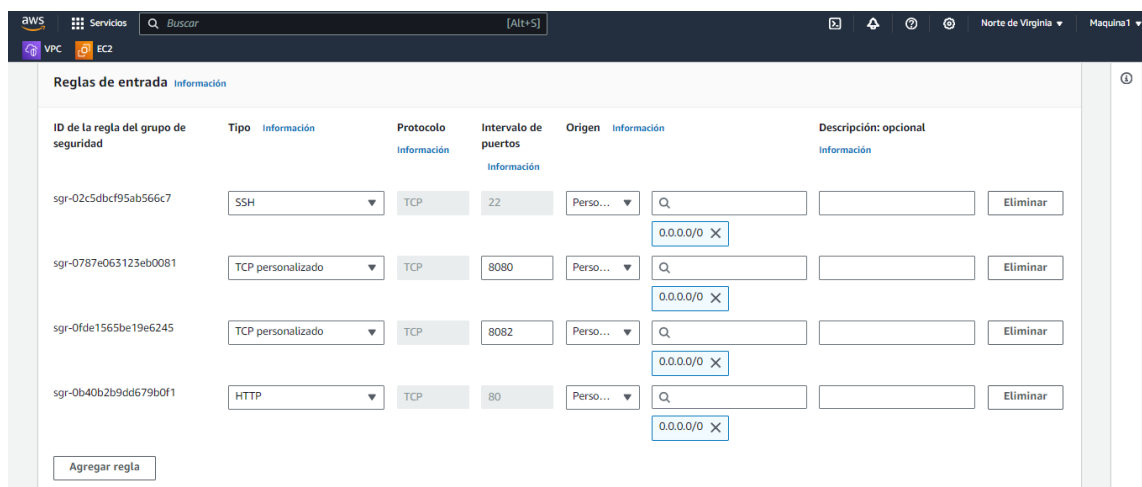


Fuente: Autoría propia

## Redes

Se Agregaron 3 reglas que permiten que se puedan acceder a los contenedores por los puertos correspondientes. Se tiene la de SSH nos permite conectar y las demás están en 0.0.0.0/0 para que así cualquier usuario pueda entrar, el custom TCP para personalizar este mismo y el HTTP que permite conectar al Apache.

Figura 5 Reglas de conexión de red

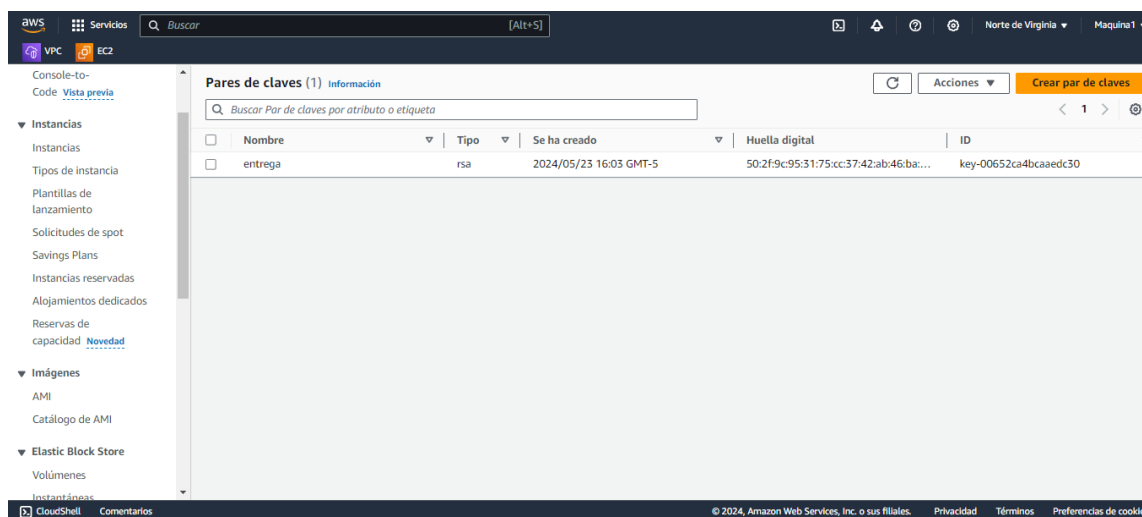


Fuente: Autoría propia

## Claves

Creamos una clave para la instancia que dará un certificado de seguridad que permite autenticarme a la instancia, el cual es creado cuando se establece la máquina virtual que se usa para acceder a la instancia de Linux mediante un cliente SSH como WSL.

Figura 6 Clave



Fuente: Autoría propia

## Archivo Hosts

Se modifica el archivo hosts para que pueda acceder a la aplicación sin la dirección IP y sea dirigido con el nombre que se le asigna a la dirección IP.

*Figura 7 Configuración Host local*

```

# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# 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.
#
#:::
#
54.89.224.136 tienda1.com
54.89.224.136 comercio.com

```

Fuente: autoría propia

## Configuración

La configuración que se le hace al APACHE antes de iniciarlo.

*Figura 8 configuración apache*

```

GNU nano 5.8 /etc/httpd/conf/httpd.conf Modified
#
# Customizable error responses come in three flavors:
# 1) plain text 2) local redirects 3) external redirects
#
# Some examples:
#ErrorDocument 500 "The server made a boo boo."
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
#
#
# EnableMMAP and EnableSendfile: On systems that support it,
# memory-mapping or the sendfile syscall may be used to deliver
# files. This usually improves server performance, but must
# be turned off when serving from networked-mounted
# filesystems or if support for these functions is otherwise
# broken on your system.
# Defaults if commented: EnableMMAP On, EnableSendfile Off
#
#EnableMMAP off
#EnableSendfile on
#
# Supplemental configuration
#
# Load config files in the "/etc/httpd/conf.d" directory, if any.
#CONFIGURATION
IncludeOptional conf.d/*.conf
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so

```


Fuente: autoría propia



## Configuración virtual host

Siendo la configuración que se realiza al virtual host con los pasos para su configuración.

*Figura 9 configuración virtual host*



```

GNU nano 5.8 /etc/httpd/conf.d/vhost.conf
<VirtualHost *:80>
  ServerName comercio.com

  ProxyPreserveHost On
  ProxyPass / http://localhost:8080/
  ProxyPassReverse / http://localhost:8080/
</VirtualHost>

<VirtualHost *:80>
  ServerName tienda01.com

  ProxyPreserveHost On
  ProxyPass / http://localhost:8082/
  ProxyPassReverse / http://localhost:8082/
</VirtualHost>
  
```

Fuente: autoría propia

## Comandos

### Paso 1 comando para instalar Apache HTTP Server

Sudo yum install httpd -y

La función de este comando es instalar el servidor Apache en nuestro sistema de manera automática

### Paso 2 comando para configurar e iniciar Apache

sudo systemctl enable httpd

Este comando asegura que el servicio de Apache HTTP inicie de manera automática en cada arranque del sistema

```
sudo systemctl start httpd
```

Mientras que este comando inicia el servicio Apache HTTP, lo que significa que el servidor WEB se ejecutará y podrá estar disponible para manejar la solicitud

### **Paso 3 Abrir el archivo de configuración principal**

```
sudo nano /etc/httpd/conf/httpd.conf
```

Este comando abre el archivo de editor de texto de Apache

### **Paso 4 cargar módulos proxys en Apache**

```
LoadModule proxy_module modules/mod_proxy.so
```

El cual carga el módulo proxy permitiendo que apache actúe como proxy

```
LoadModule proxy_http_module modules/mod_proxy_http.so
```

Y este comando carga el módulo proxy permitiendo configuraciones al proxy

### **Paso 5 Cargar y poder editar configuraciones de virtual host y de proxy**

```
sudo nano /etc/httpd/conf.d/vhost.conf
```

Crea el archivo VHOST en el directorio de apache para así definir los servidores virtuales.

### **Paso 6 Configurar virtual host y proxy**

```
<VirtualHost *:80>
```

El cual define que las configuraciones sean aplicadas a las solicitudes en el Puerto 80

```
    ServerName comercio.com
```

Especificando el nombre del servidor virtual

```
    ProxyPreserveHost On
```

Este comando indica al apache que preserve el HOST original en las solicitudes

ProxyPass / <http://localhost:8082/>

Configura el apache para así poder enviar todas las solicitudes a la URL especificada

ProxyPassReverse / <http://localhost:8082/>

Este comando reescribe los encabezados de respuesta del backend para que los enlaces y direccionamientos apunten directamente con el cliente </VirtualHost>

### Paso 7 reiniciar el apache

Este comando reinicia el servicio de apache para aplicar los cambios realizados en la configuración

Estos son los pasos realizados para la instalación y configuración del apache HTTP server con soporte en proxy en un Sistema de Linux. Esta configuración permite que el apache actúe como intermediario entre clientes y servidores redirigiendo y escribiendo solicitudes HTTP según sea lo necesario.

### Procedimiento

WLS donde instalamos el Apache HTTP, se inicia y se configura

Figura 10 WSL

```
root@LAPTOP-E703H3VE:~# ssh -i "entrega_psa" ec2-user@ec2-54-89-224-136.compute-1.amazonaws.com
Amazon Linux 2023
Last login: Sat May 25 02:04:56 2024 from 181.78.20.146
[ec2-user@ip-18-20-20-101 ~]$ yum install httpd -y
Error: This command has to be run with superuser privileges (under the root user on most systems).
[ec2-user@ip-18-20-20-101 ~]$ sudo su
[root@ip-18-20-20-101 ec2-user]# yum install httpd -y
Last metadata expiration check: 1:28:58 ago on Sat May 25 01:53:56 2024.
Dependencies resolved.
Package               Architecture Version      Repository    Size
-----
Installing:
httpd                 x86_64      2.4.59-2.amzn2023.0.1  amazonLinux  47 k
Installing dependencies:
apr                   x86_64      1.7.2-2.amzn2023.0.2  amazonLinux  129 k
apr-util              x86_64      1.6.3-1.amzn2023.0.1  amazonLinux  98 k
apr-util-devel        noarch      1.6.3-1.amzn2023.0.1  amazonLinux  19 k
apr-util-libs         x86_64      2.4.59-2.amzn2023.0.3  amazonLinux  1.4 M
httpd-core            noarch      2.4.59-2.amzn2023.0.1  amazonLinux  14 k
httpd-devel           x86_64      2.4.59-2.amzn2023.0.1  amazonLinux  81 k
httpd-libs            x86_64      1.0.9-4.amzn2023.0.2  amazonLinux  315 k
mailcap               noarch      2.1.49-3.amzn2023.0.3  amazonLinux  33 k
Installing weak dependencies:
apr-util-openssl     x86_64      1.6.3-1.amzn2023.0.1  amazonLinux  17 k
mod_http2            x86_64      2.0.27-1.amzn2023.0.2  amazonLinux  166 k
```

Fuente: autoría propia

Figura 11 comandos de instalación y el proceso de instalación del HTTP apache

```

root@ip-10-20-20-101/homej x +
Verifying      : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch          4/12
Verifying      : httpd-2.4.59-2.amzn2023.x86_64                          5/12
Verifying      : httpd-core-2.4.59-2.amzn2023.x86_64                    6/12
Verifying      : httpd-filesystem-2.4.59-2.amzn2023.noarch              7/12
Verifying      : httpd-tools-2.4.59-2.amzn2023.x86_64                  8/12
Verifying      : libbrotli-1.0.9-4.amzn2023.0.2.x86_64                 9/12
Verifying      : mailcap-2.1.49-3.amzn2023.0.3.noarch                  10/12
Verifying      : mod_httpd-2.0.27-1.amzn2023.0.2.x86_64                11/12
Verifying      : mod_lua-2.4.59-2.amzn2023.x86_64                      12/12

Installed:
apr-1.7.2-2.amzn2023.0.2.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-2.4.59-2.amzn2023.x86_64          httpd-core-2.4.59-2.amzn2023.x86_64
httpd-filesystem-2.4.59-2.amzn2023.noarch  httpd-tools-2.4.59-2.amzn2023.x86_64
libbrotli-1.0.9-4.amzn2023.0.2.x86_64    mailcap-2.1.49-3.amzn2023.0.3.noarch
mod_httpd-2.0.27-1.amzn2023.0.2.x86_64    mod_lua-2.4.59-2.amzn2023.x86_64

Complete!
[root@ip-10-20-20-101 ec2-user]# sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-10-20-20-101 ec2-user]# sudo systemctl star httpd
Unknown command verb star.
[root@ip-10-20-20-101 ec2-user]# sudo systemctl start httpd
[root@ip-10-20-20-101 ec2-user]# sudo nano /etc/httpd/conf/httpd.conf
[root@ip-10-20-20-101 ec2-user]# sudo nano /etc/httpd/conf.d/vhost.conf
[root@ip-10-20-20-101 ec2-user]# sudo systemctl restart httpd
[root@ip-10-20-20-101 ec2-user]# sudo nano /etc/httpd/conf.d/vhost.conf
[root@ip-10-20-20-101 ec2-user]# sudo systemctl restart httpd
[root@ip-10-20-20-101 ec2-user]# sudo nano /etc/httpd/conf/httpd.conf
[root@ip-10-20-20-101 ec2-user]# sudo nano /etc/httpd/conf.d/vhost.conf
[root@ip-10-20-20-101 ec2-user]#

```

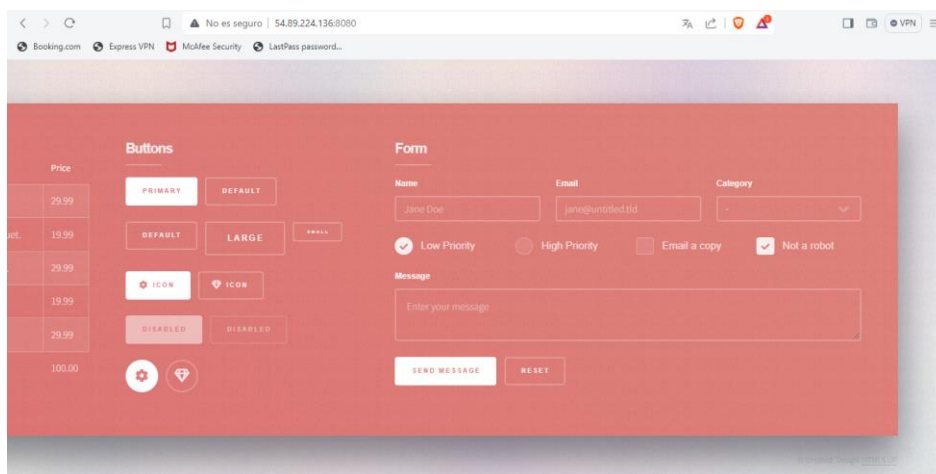
Fuente: autoría propia

## Resultados

### Puerto 8080

Consulta con puerto 8080 y dirección IP

Figura 12 página puerto 8080

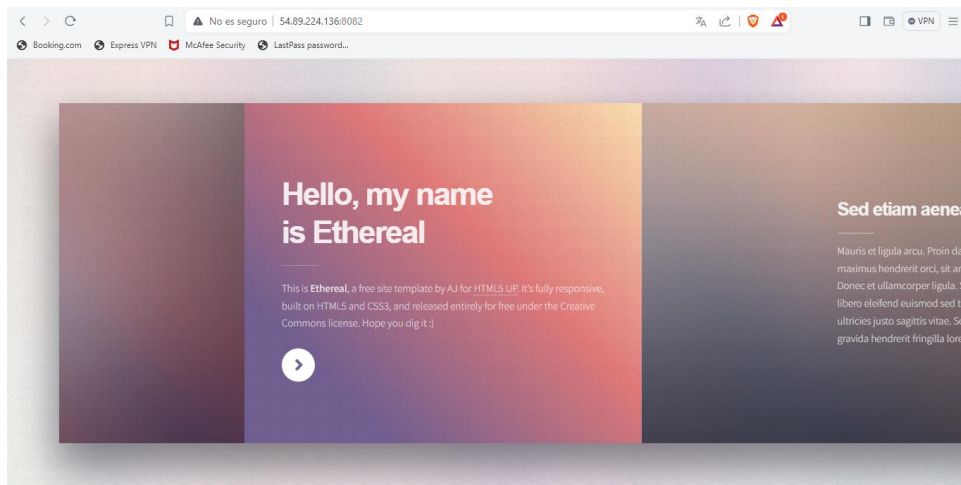


Fuente: autoría propia

## Puerto 8082

Consulta con puerto 8082 y dirección IP

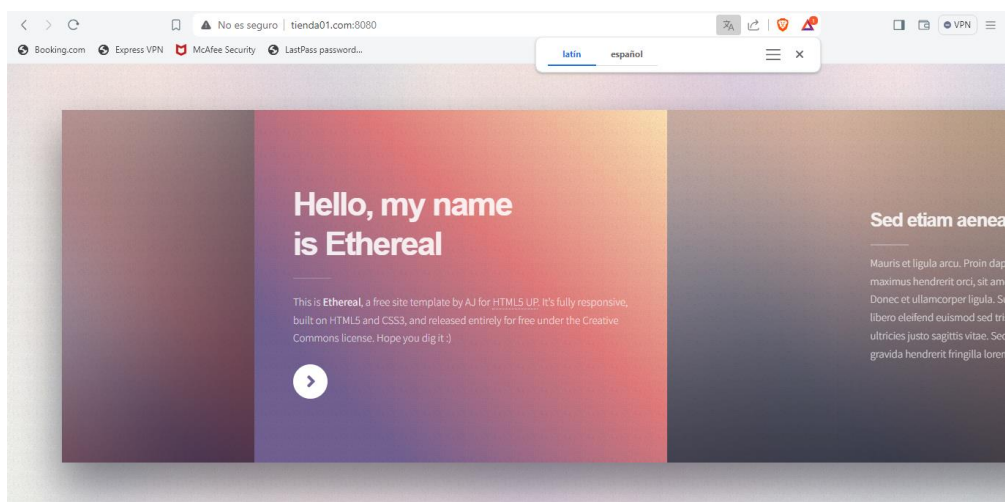
*Figura 13 página puerto 8082*



Fuente: autoría propia

## Consulta con Nombre de la página y puerto 8080

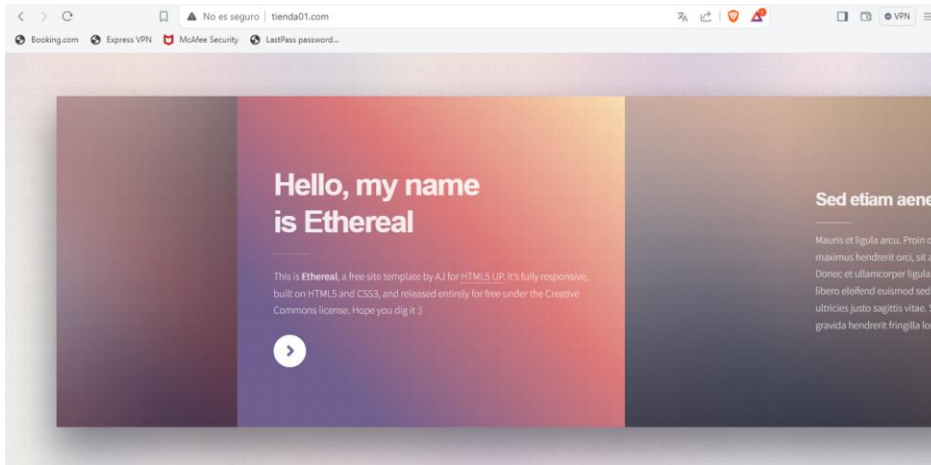
*Figura 14 página con nombre y puerto 8080*



Fuente: autoría propia

## Consulta con nombre de la página asignada en el archivo hosts

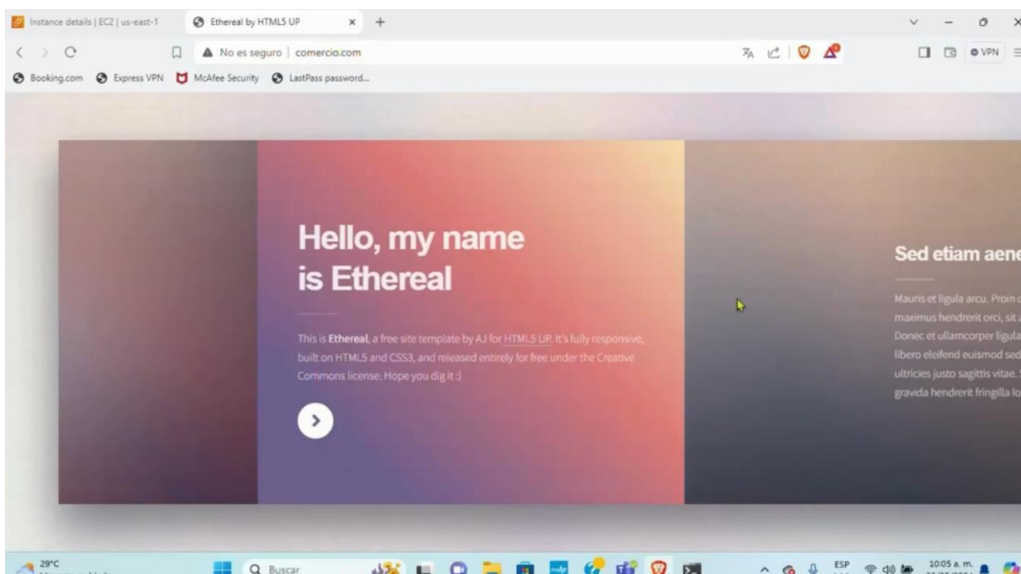
*Figura 15 página consultada con nombre*



Fuente: autoría propia

## Consulta con Nombre de la página asignada

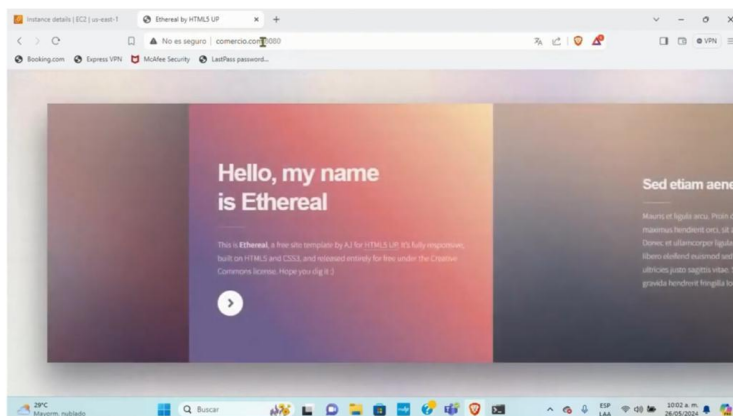
*Figura 16 página con consulta asignada*



Fuente: autoría propia

## Consulta con Nombre de la página y puerto 8082

Figura 17 nombre de página y puerto 8082

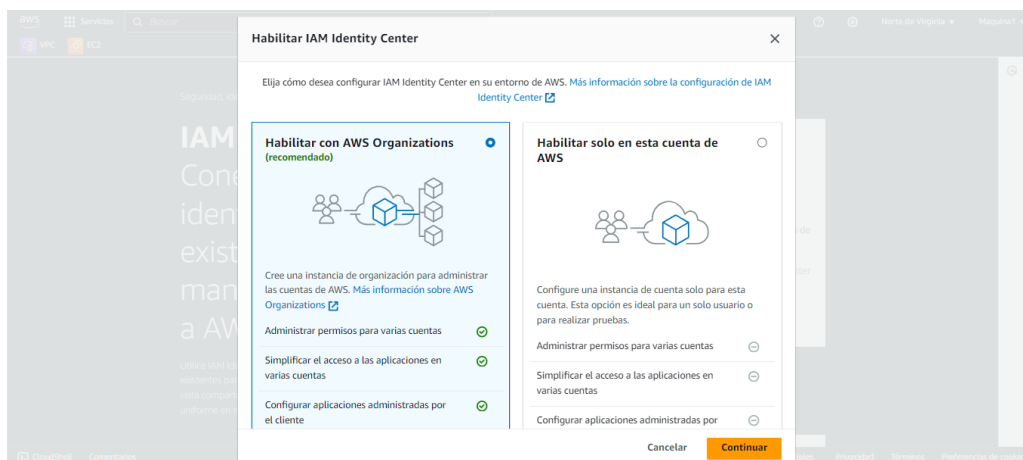


Fuente: autoría propia

## Habilitación de IAM, implementación de S3 y RDS

Para realizar este ejercicio, lo primero que se debe hacer es habilitar el IAM identity center

Figura 18 Habilitación de IAM



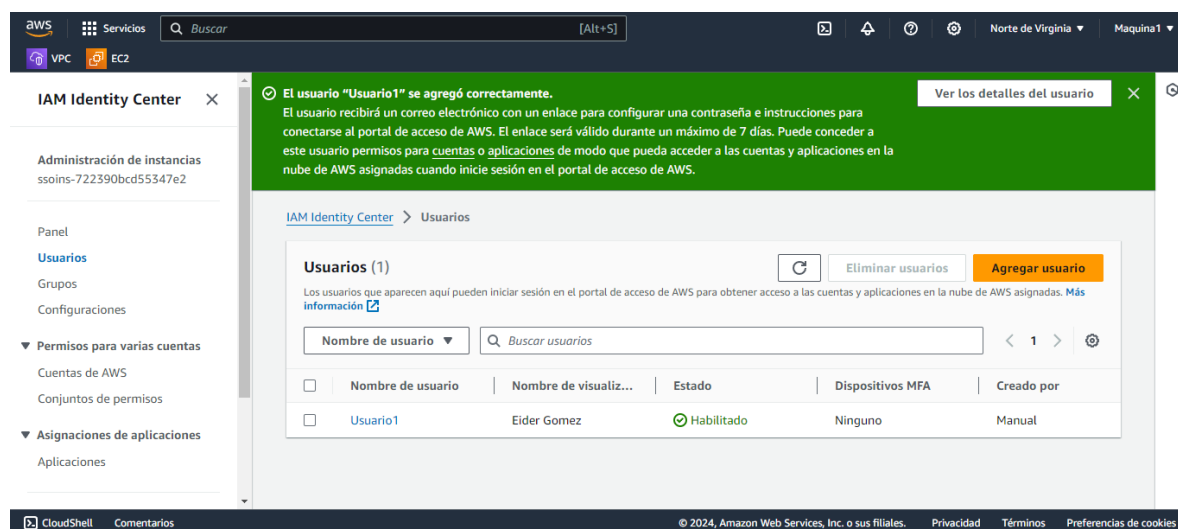
Fuente: Autoría propia

AWS Identity and Access Management (IAM) es un servicio web que lo ayuda a controlar de forma segura el acceso a los recursos de AWS. Con IAM, puede administrar de forma centralizada los permisos que controlan a qué recursos de AWS pueden acceder los usuarios. Utilice IAM para controlar quién está autenticado (ha iniciado sesión) y autorizado (tiene permisos) para utilizar recursos (Amazon, 2024).

### Creación de usuario

Luego de activar la versión de organización del IAM, procedemos a crear el usuario, el cual será el que nos permita conectarnos al S3.

*Figura 19 Creación de usuario*



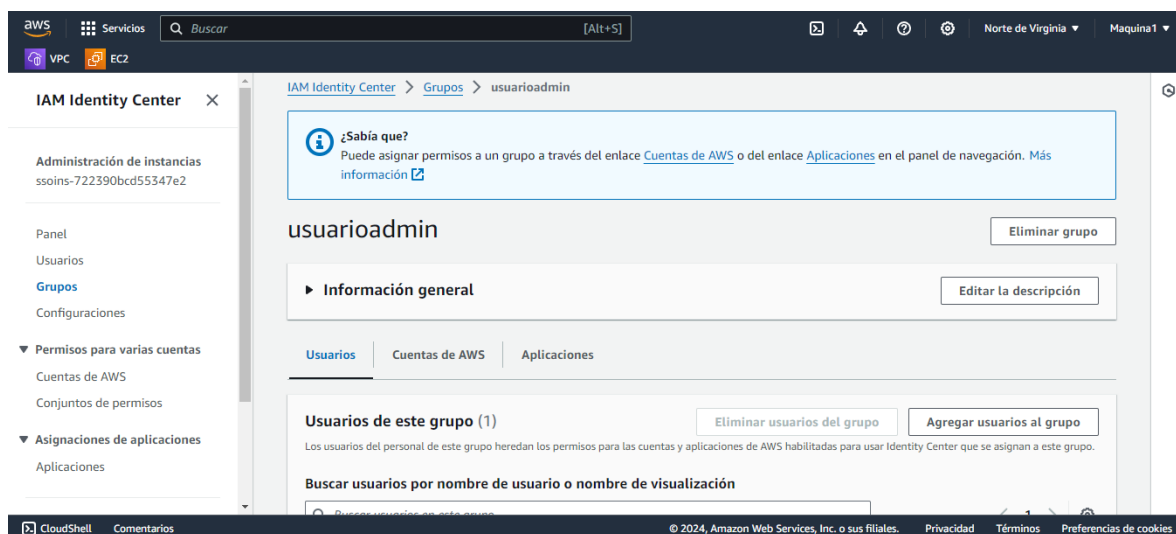
Fuente: Autoría propia

### Creación de grupo

Posteriormente, debemos crear un grupo al cual le asignaremos permisos de administrador para que acceda a todos los recursos necesarios. A este grupo asignaremos el usuario anteriormente creado



Figura 20 creación de grupo

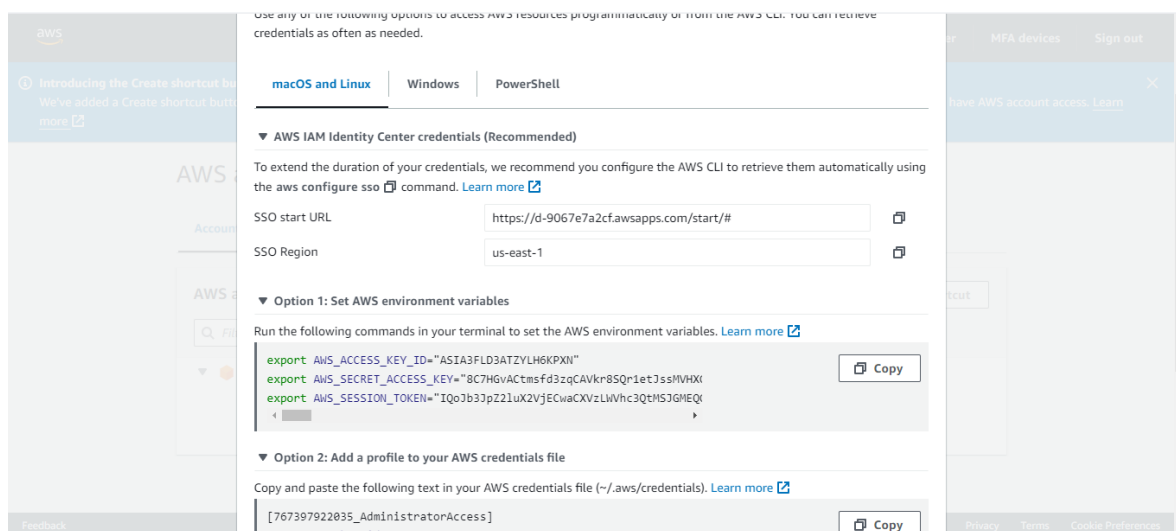


Fuente: Autoría propia

## Acceso de usuario

Después de verificar el usuario, procedemos a logearlo y así obtener las llaves de acceso que permitirán conectar la instancia de S3 y cargar el archivo desde una carpeta local

Figura 21 Acceso de usuario



Fuente: Autoría propi

## Instalación AWS CLI

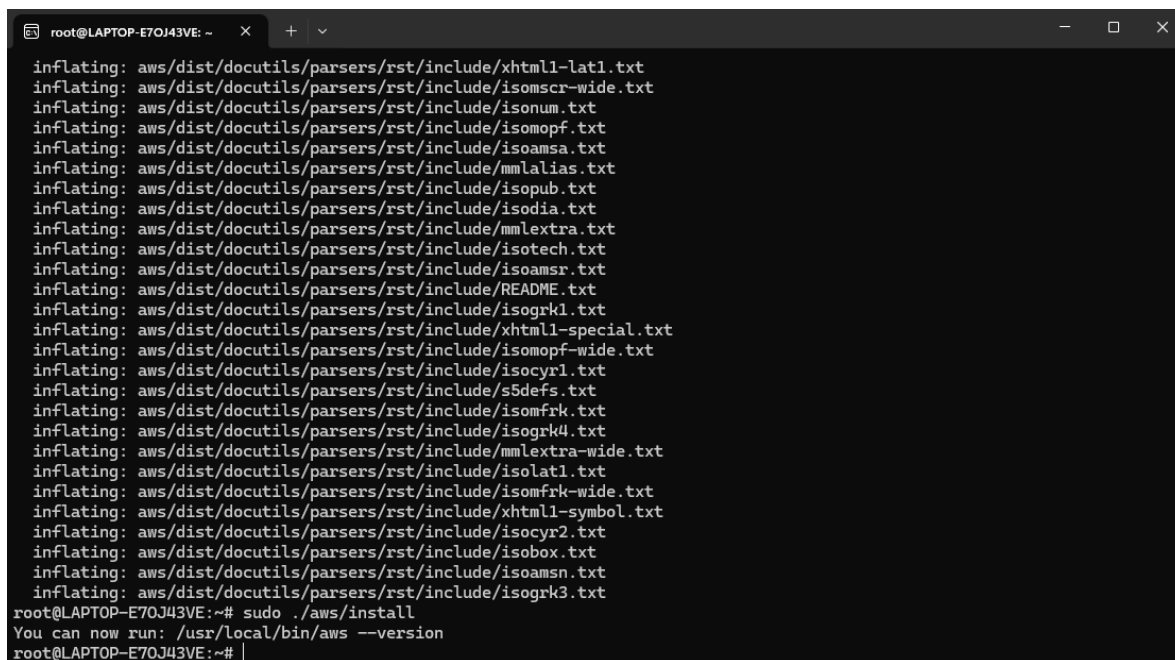
Ahora se debe instalar en la consola de WSL el AWS CLI, el cual actuara como puente para conectarnos al bucket del S3 y subir el archivo.

La Command Line Interface o AWS CLI se refiere a una herramienta unificada y de código abierto que se implementa con el objetivo de gestionar los productos de la plataforma de Amazon Web Service. La AWS CLI también permite que el usuario pueda interactuar con los servicios del sistema a través de comandos en la shell de línea de comandos.

Esta herramienta también se caracteriza por contribuir a la automatización de los servicios del sistema y de labores tediosas de la infraestructura (keepcoding, 2024).

El comando que se usara para instalar el CLI después de descargarlo y descomprimirlo es “./aws/install”

Figura 22 Instalación CLI



```
root@LAPTOP-E7OJ43VE: ~  
inflating: aws/dist/docutils/parsers/rst/include/xhtml1-lat1.txt  
inflating: aws/dist/docutils/parsers/rst/include/isomscr-wide.txt  
inflating: aws/dist/docutils/parsers/rst/include/isonum.txt  
inflating: aws/dist/docutils/parsers/rst/include/isomopf.txt  
inflating: aws/dist/docutils/parsers/rst/include/isoamsa.txt  
inflating: aws/dist/docutils/parsers/rst/include/mmlalias.txt  
inflating: aws/dist/docutils/parsers/rst/include/isopub.txt  
inflating: aws/dist/docutils/parsers/rst/include/isodia.txt  
inflating: aws/dist/docutils/parsers/rst/include/mmlextra.txt  
inflating: aws/dist/docutils/parsers/rst/include/isotech.txt  
inflating: aws/dist/docutils/parsers/rst/include/isoamsr.txt  
inflating: aws/dist/docutils/parsers/rst/include/README.txt  
inflating: aws/dist/docutils/parsers/rst/include/isogrkl.txt  
inflating: aws/dist/docutils/parsers/rst/include/xhtml1-special.txt  
inflating: aws/dist/docutils/parsers/rst/include/isomopf-wide.txt  
inflating: aws/dist/docutils/parsers/rst/include/isocyr1.txt  
inflating: aws/dist/docutils/parsers/rst/include/s5defs.txt  
inflating: aws/dist/docutils/parsers/rst/include/isomfrk.txt  
inflating: aws/dist/docutils/parsers/rst/include/isogrkl4.txt  
inflating: aws/dist/docutils/parsers/rst/include/mmlextra-wide.txt  
inflating: aws/dist/docutils/parsers/rst/include/isolat1.txt  
inflating: aws/dist/docutils/parsers/rst/include/isomfrk-wide.txt  
inflating: aws/dist/docutils/parsers/rst/include/xhtml1-symbol.txt  
inflating: aws/dist/docutils/parsers/rst/include/isocyr2.txt  
inflating: aws/dist/docutils/parsers/rst/include/isobox.txt  
inflating: aws/dist/docutils/parsers/rst/include/isoamsn.txt  
inflating: aws/dist/docutils/parsers/rst/include/isogrkl3.txt  
root@LAPTOP-E7OJ43VE:~# sudo ./aws/install  
You can now run: /usr/local/bin/aws --version  
root@LAPTOP-E7OJ43VE:~#
```

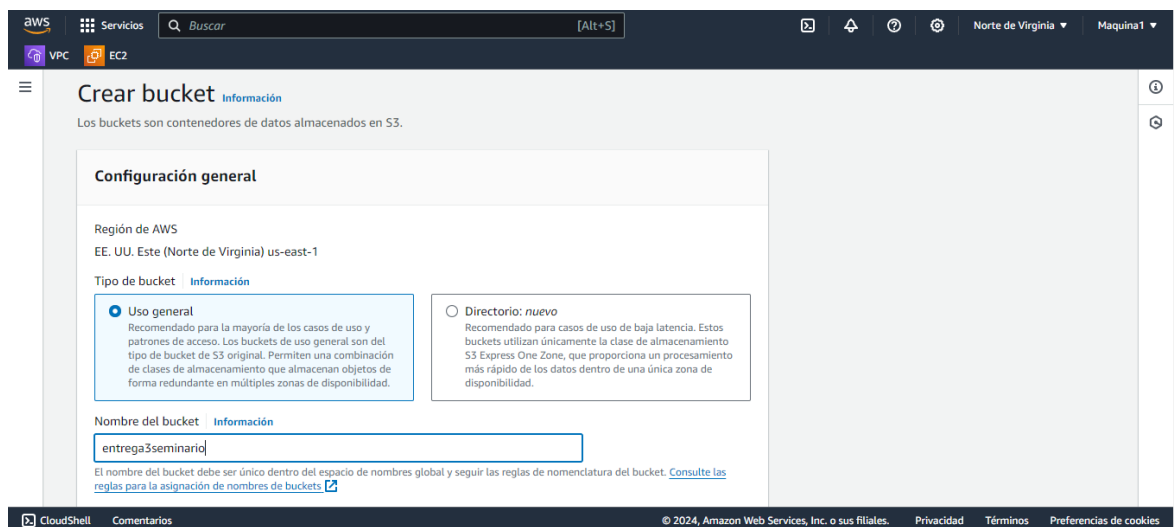
Fuente: Autoría propia

## Creación bucket

El siguiente paso es la creación del bucket de S3, en el cual alojaremos nuestro archivo.

Amazon Simple Storage Service (Amazon S3) es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento líderes del sector. Los clientes de todos los tamaños y sectores pueden utilizar Amazon S3 para almacenar y proteger cualquier cantidad de datos para diversos casos de uso, tales como lagos de datos, sitios web, aplicaciones móviles, copia de seguridad y restauración, archivado, aplicaciones empresariales, dispositivos IoT y análisis de big data. Amazon S3 proporciona funciones de gestión para que pueda optimizar, organizar y configurar el acceso a sus datos para satisfacer sus requisitos empresariales, organizativos y de conformidad específicos (Amazon, 2024).

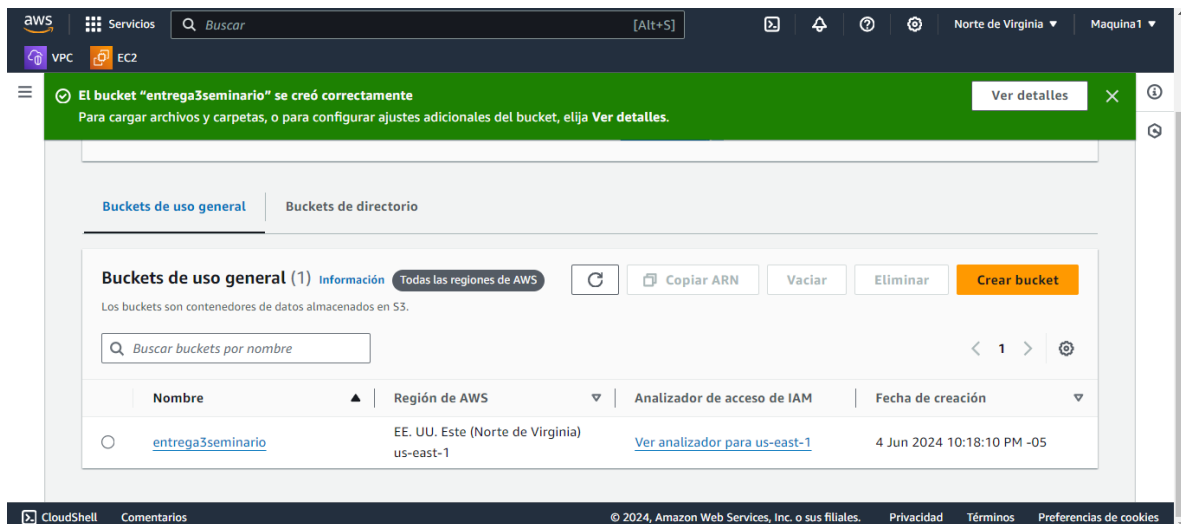
*Figura 23 creación bucket s3*



Fuente: Autoría propia

Después de crear el bucket, debemos configurar algunas otras reglas para permitir que el usuario de IAM tenga acceso a este.

Figura 24 bucket creado

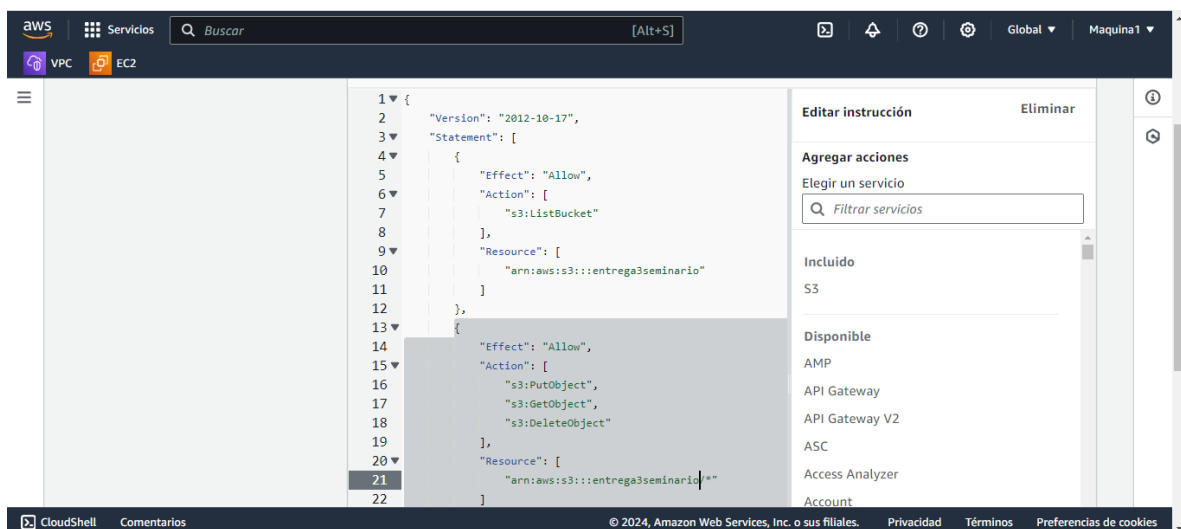


Fuente: Autoría propio

### Creación de política para acceso al bucket

al bucket se le deben asignar unas políticas para permitir el acceso y la interacción con la instancia, usando la opción de políticas de IAM.

Figura 25 politica de S3

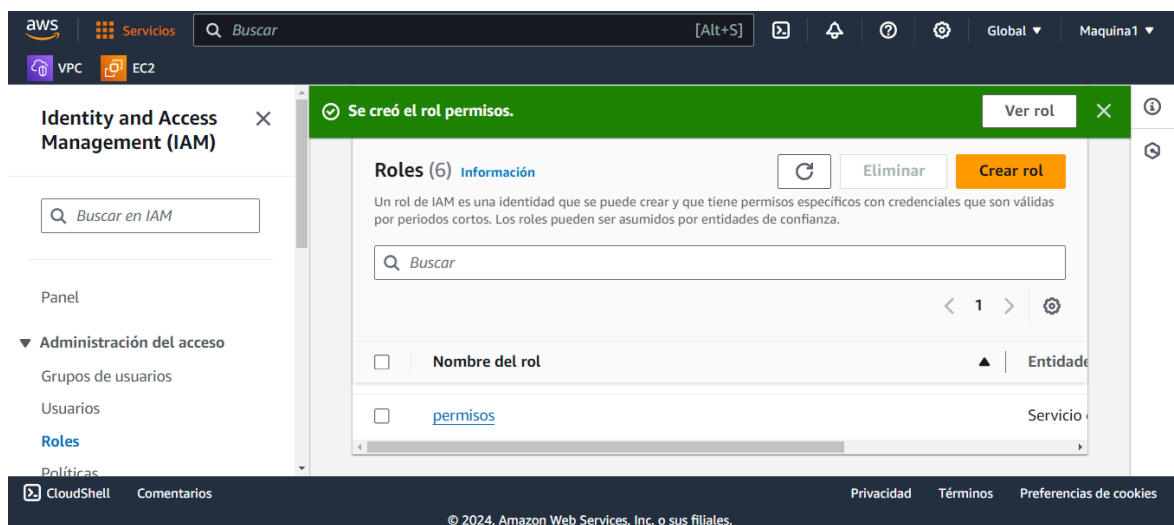


Fuente: Autoría propia

## Creación del Rol

Posteriormente debemos asignar estas políticas a un rol que podemos crear para este caso. Este rol luego se les asignara a las reglas de seguridad de la instancia de EC2 con la que va a interactuar

*Figura 26 creación de rol*

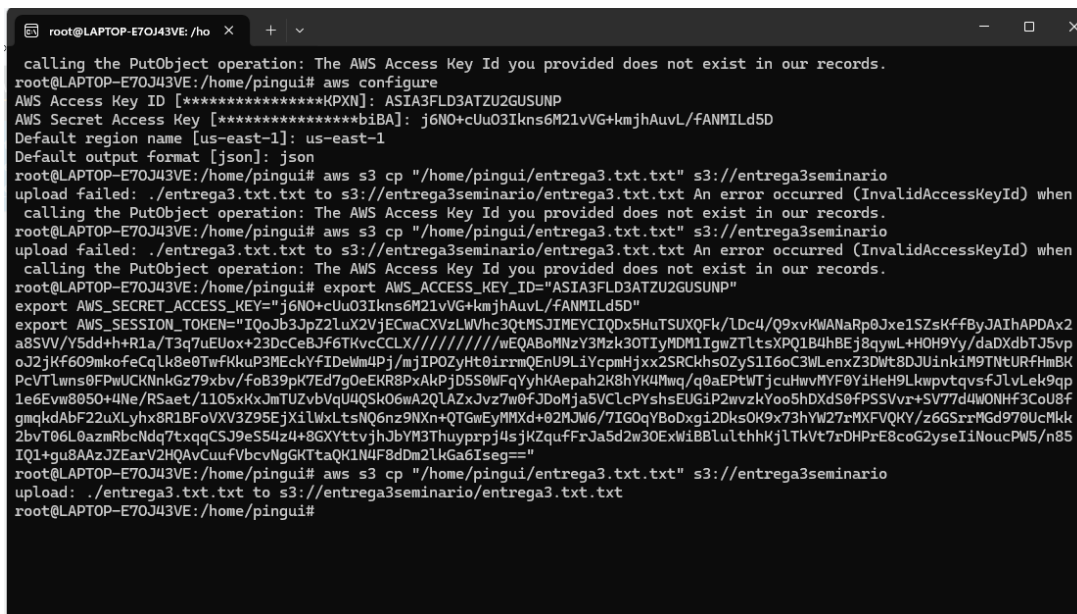


Fuente: Autoría propia

## Archivo subido a S3

Para terminar, se usan las llaves de acceso del usuario creado, se configura el CLI con la región del bucket y se usa el comando “aws s3 cp ruta del archivo a subir s3:// nombre del bucket”

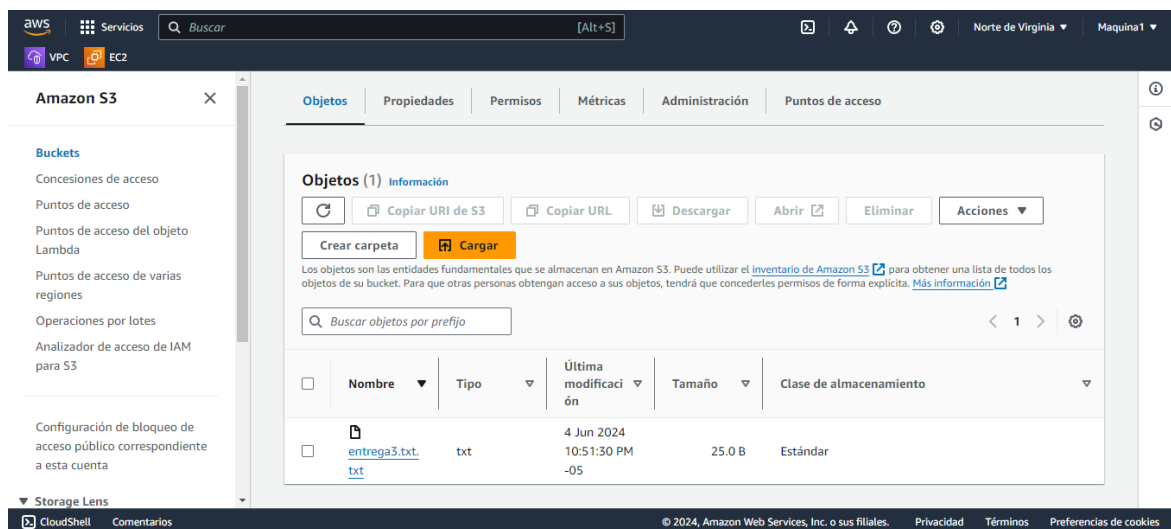
Figura 27 subida de archivo a s3



Fuente: Autoría propia

Acá en la plataforma de AWS, ya se puede apreciar que el archivo se encuentra en el bucket de s3 correctamente

Figura 28 archivo en bucket



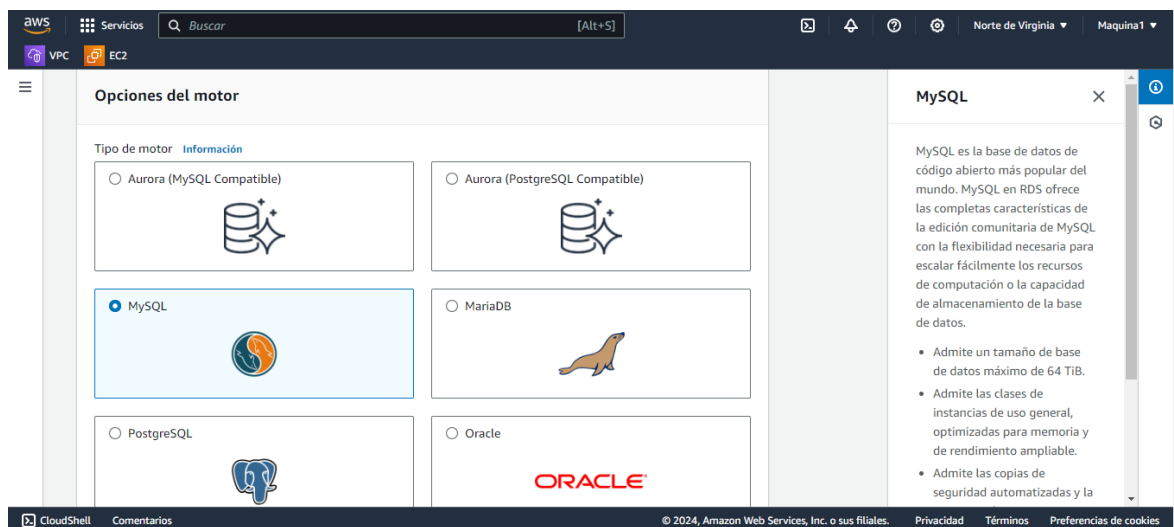
Fuente: Autoría propia

## Creación Rds

Para este paso, lo primero será crear en el servicio de RDS, la base de datos con la que vamos a interactuar, en este caso será una base de datos de MySQL.

Amazon RDS es un servicio de bases de datos relacionales fácil de administrar, optimizado para el costo total de propiedad. Es fácil de configurar, operar y escalar según la demanda. Amazon RDS automatiza las tareas indiferenciadas de administración de bases de datos, como el aprovisionamiento, la configuración, las copias de seguridad y la aplicación de revisiones. Amazon RDS permite a los clientes crear una nueva base de datos en cuestión de minutos y ofrece flexibilidad para personalizar las bases de datos a fin de satisfacer sus necesidades en 8 motores y 2 opciones de implementación (Amazon, 2024).

*Figura 29 creación de base de datos*

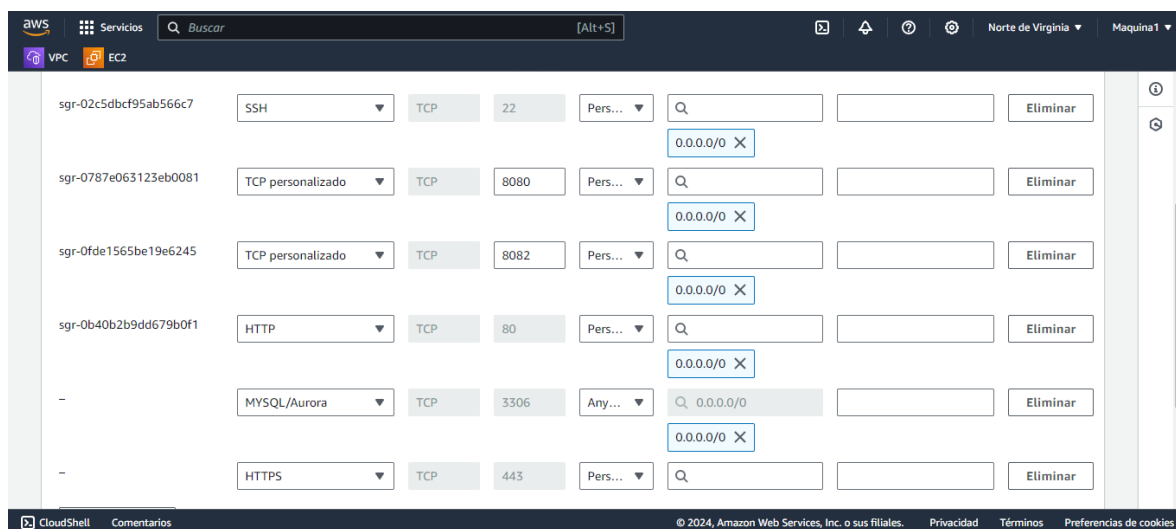


Fuente: Autoría propia

## Modificación de las reglas para acceder a la base de datos

Después de crear la base de datos, debemos asegurarnos de que las reglas de seguridad permitan la interacción con el puerto 3306 y los puertos 80 y 443, para que podamos acceder a la base de datos desde la instancia de EC2

Figura 30 configuración de puertos de la base de datos



Fuente: Autoría propia

## Instalación cliente MySQL

Ahora se debe instalar en la instancia, el servicio de MySQL, con el cual nos conectaremos a través de consola a la base de datos



Figura 31 instalación MySQL

```

root@ip-10-20-20-101:/home/ x + v
WARNING:
A newer release of "Amazon Linux" is available.

Available Versions:

Version 2023.4.20240528:
Run the following command to upgrade to 2023.4.20240528:

dnf upgrade --releasever=2023.4.20240528

Release notes:
https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.4.20240528.html

-----
Installed:
mariadb-connector-c-3.1.13-1.amzn2023.0.3.x86_64 mariadb-connector-c-config-3.1.13-1.amzn2023.0.3.noarch
mariadb105-3:10.5.23-1.amzn2023.0.1.x86_64 mariadb105-backup-3:10.5.23-1.amzn2023.0.1.x86_64
mariadb105-common-3:10.5.23-1.amzn2023.0.1.x86_64 mariadb105-cracklib-password-check-3:10.5.23-1.amzn2023.0.1.x86_64
mariadb105-errmsg-3:10.5.23-1.amzn2023.0.1.x86_64 mariadb105-gssapi-server-3:10.5.23-1.amzn2023.0.1.x86_64
mariadb105-server-3:10.5.23-1.amzn2023.0.1.x86_64 mariadb105-server-utils-3:10.5.23-1.amzn2023.0.1.x86_64
mysql-selinux-1.0.4-2.amzn2023.0.3.noarch perl-B-1.80-477.amzn2023.0.6.x86_64
perl-DBD-MariaDB-1.22-1.amzn2023.0.4.x86_64 perl-DBI-1.643-7.amzn2023.0.3.x86_64
perl-Data-Dumper-2.174-469.amzn2023.0.2.x86_64 perl-File-Copy-2.34-477.amzn2023.0.6.noarch
perl-FileHandle-2.03-477.amzn2023.0.6.noarch perl-Math-BigInt-1.9998.39-2.amzn2023.0.2.noarch
perl-Math-BigRat-0.2614-458.amzn2023.0.2.noarch perl-Math-Complex-1.59-477.amzn2023.0.6.noarch
perl-Sys-Hostname-1.23-477.amzn2023.0.6.x86_64 perl-base-2.27-477.amzn2023.0.6.noarch

Complete!
[root@ip-10-20-20-101 ec2-user]#

```

Fuente: Autoría propia

## Conexión a Rds

finalmente usamos el comando “mysql -h nombre de la base de datos.us-east-

2.rds.amazonaws.com -P 3306 -u admin -p” para conectarnos a la base de datos que creamos anteriormente

Figura 32 conexión a base de datos

```

root@ip-10-20-20-101:/home/ x + v
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> status

mysql Ver 15.1 Distrib 10.5.23-MariaDB, for Linux (x86_64) using EditLine wrapper

Connection id:          25
Current database:
Current user:           admin@10.20.20.101
SSL:                    Not in use
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server:                 MySQL
Server version:         10.5.23 Source distribution
Protocol version:       10
Connection:             database-1.cja@geo061sx.us-east-1.rds.amazonaws.com via TCP/IP
Server characteraset:   utf8mb4
Db characteraset:       utf8mb4
Client characteraset:   utf8mb3
Conn. characteraset:    utf8mb3
TCP port:               3306
Uptime:                 4 min 59 sec

Threads: 3  Questions: 9719  Slow queries: 0  Opens: 371  Flush tables: 3  Open tables: 251  Queries per second avg: 32.505

MySQL [(none)]> |

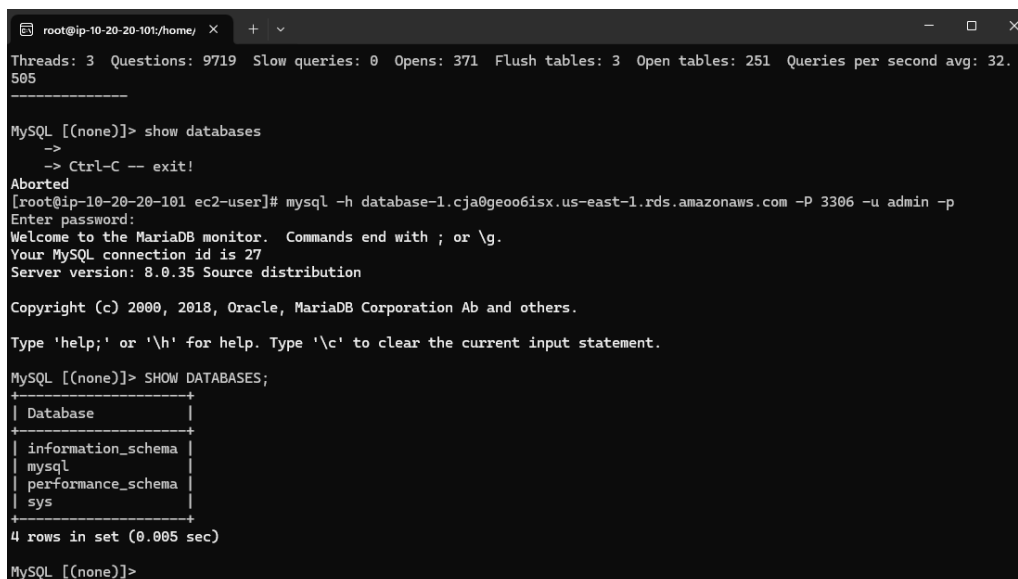
```

Fuente: Autoría propia

## Base de datos de Rds

Usando el comando SHOW DATABASES, podemos lograr mostrar lo que existe dentro de la base de datos a la que nos hemos conectado

Figura 33 base de datos RDS



```
root@ip-10-20-20-101:/home/ x + v
Threads: 3 Questions: 9719 Slow queries: 0 Opens: 371 Flush tables: 3 Open tables: 251 Queries per second avg: 32.505
-----
MySQL [(none)]> show databases
->
-> Ctrl-C -- exit!
Aborted
[root@ip-10-20-20-101 ec2-user]# mysql -h database-1.cja0geoo6isx.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 27
Server version: 8.0.35 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.005 sec)

MySQL [(none)]>
```

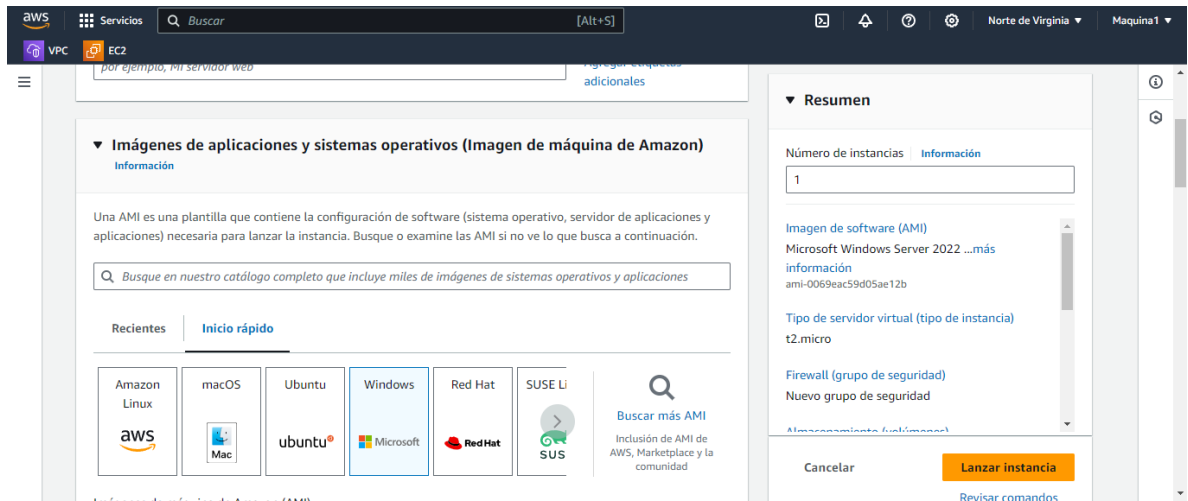
Fuente: Autoría propia

## Creación de balanceadores de carga y grupo de auto escalado

En este ejercicio final se utilizarán los servicios de balanceadores de carga y grupos de auto escalado, con los cuales se puede automatizar la creación de instancias para mantenerlas siempre disponibles.

El primer paso es crear una instancia de Windows que servirá de base.

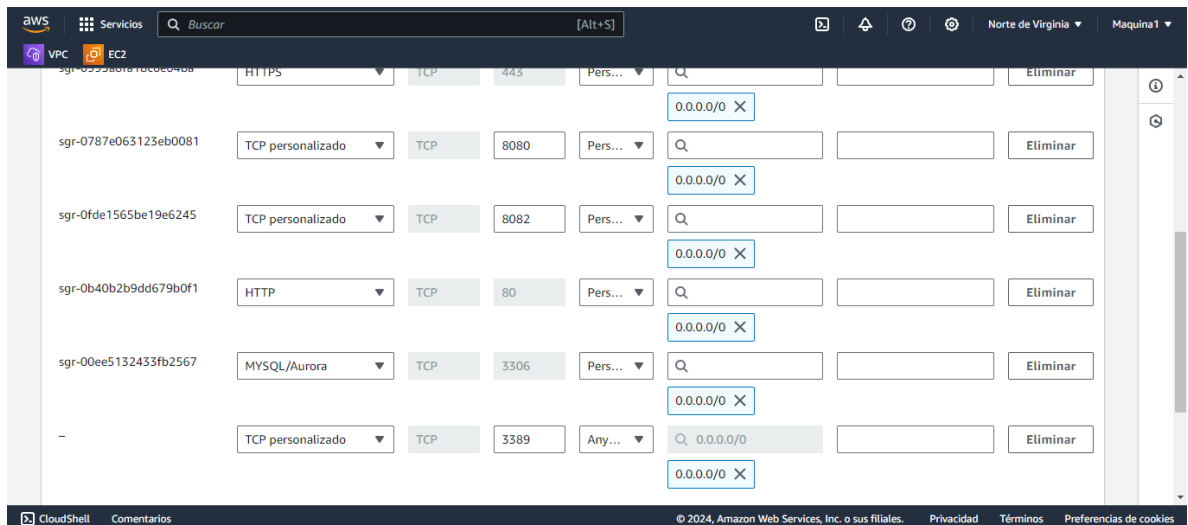
Figura 34 creación instancia de Windows



Fuente: Autoría propia

Se debe asegurar que el security group tenga abierto los puertos 80 y 3306, para poder conectarnos a la instancia a través de escritorio remoto y que responda a peticiones de HTTP

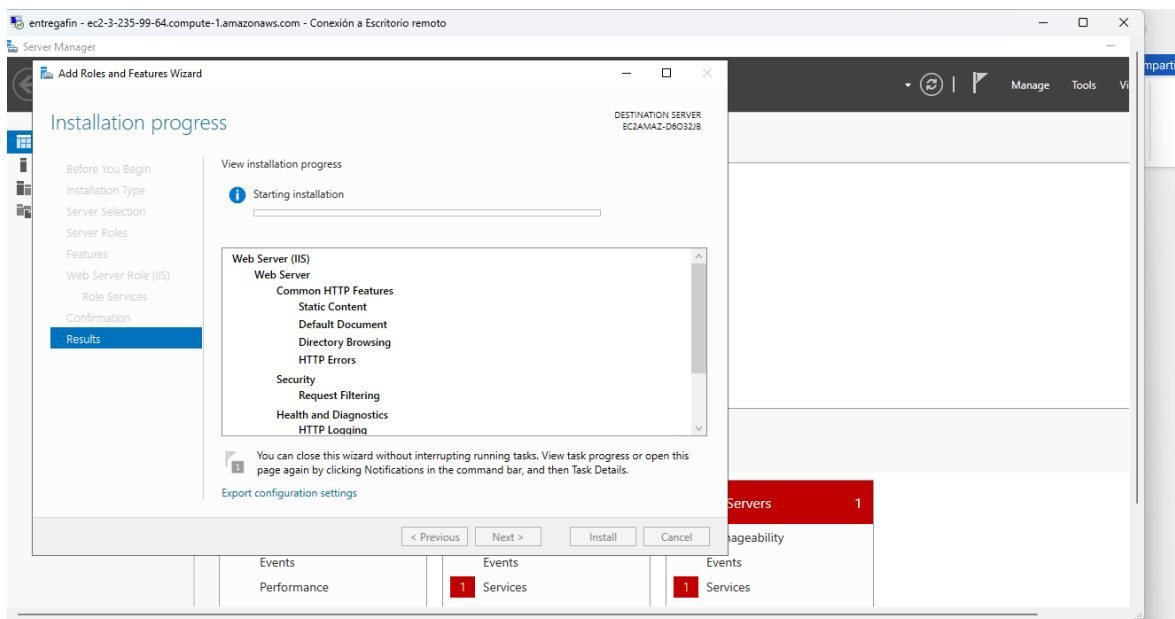
Figura 35 reglas de grupo de seguridad



Fuente: Autoría propia

Luego de creada la instancia, nos conectamos a ella, cambiamos su contraseña para ahorrar tiempo y se instala el servicio de internet information service (IIS) para crear una aplicación web básica

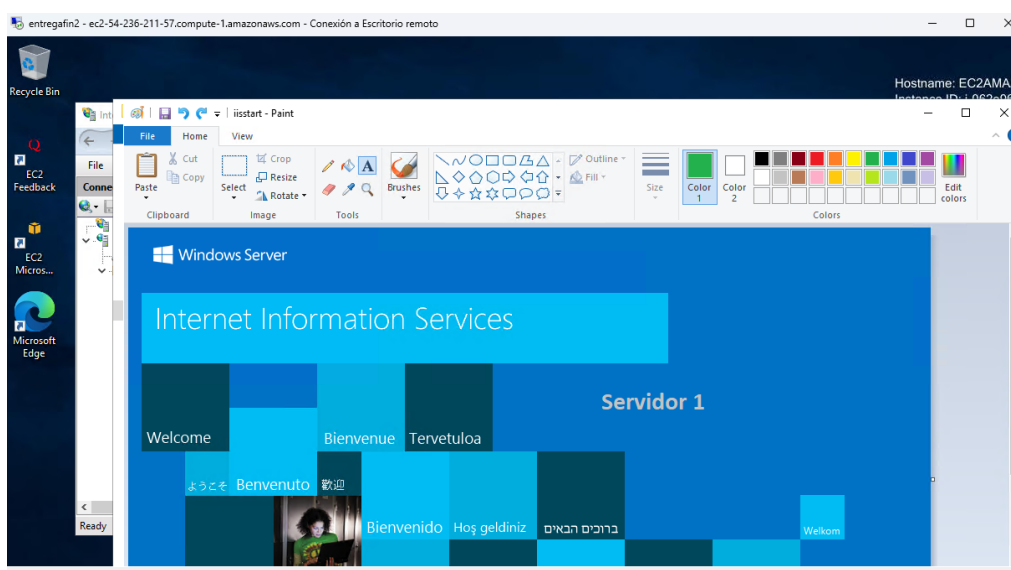
*Figura 36 instalación IIS*



Fuente: Autoría propia

El siguiente paso es editar la página web del ISS para poder identificarla, para eso le pondremos el texto servidor 1

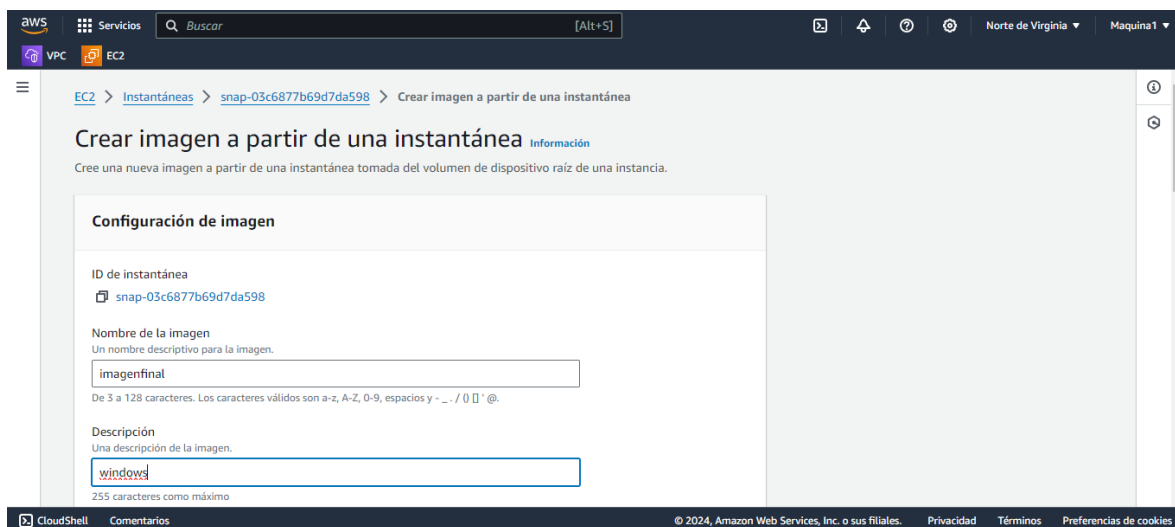
Figura 37 edición de página web



Fuente: Autoría propia

Luego se creará una instantánea de la instancia y a partir de esa instantánea se creará una imagen (AMI) con la cual se establecerá la plantilla que se usara para crear el resto de las instancias, ya con el servicio de IIS configurado y una contraseña simple que no necesitara la llave de acceso

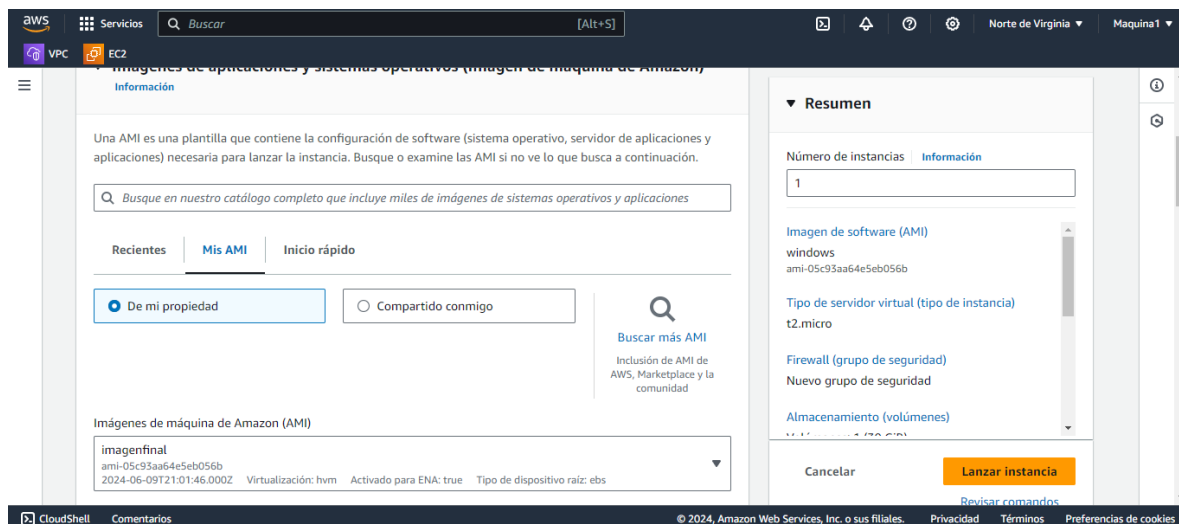
Figura 38 creación de imagen a partir de instantánea



Fuente: Autoría propia

Ahora creamos una nueva instancia usando esta imagen que se ha creado, la cual se encuentra en la pestaña de mis AMI, con esto se ahorra la instalación y configuración que ya se ha realizado en la instancia anterior

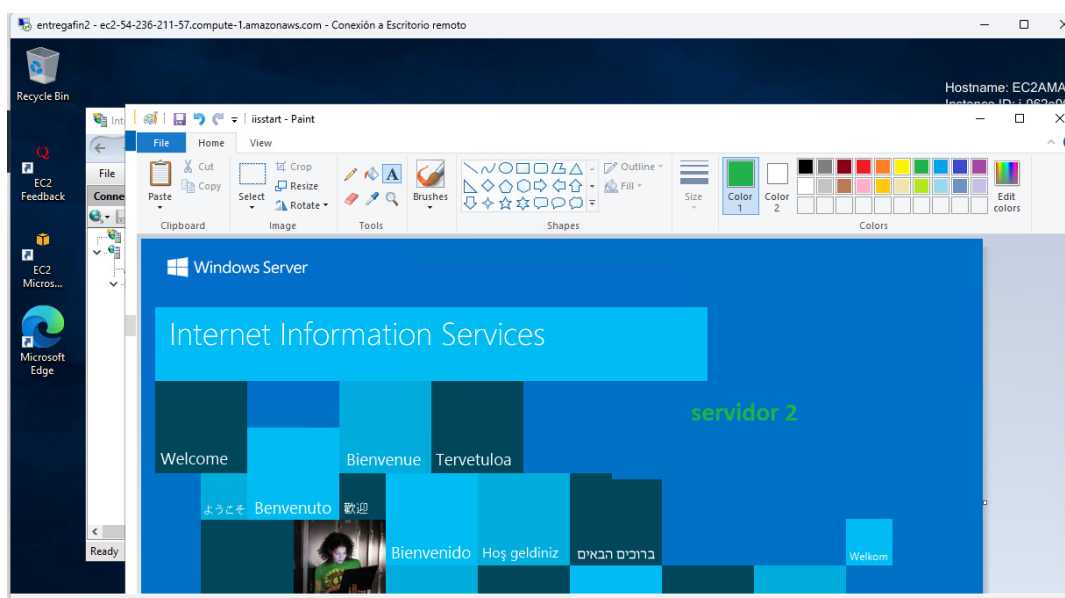
*Figura 39 creación de instancia a partir de imagen*



Fuente: Autoría propia

Para identificar esta nueva instancia se editará nuevamente la página web agregándole el texto de server 2

Figura 40 edición de página web instancia 2



Fuente: Autoría propia

Después de tener ambas instancias identificadas y funcionando, los siguiente es crear el balanceador de cargas, que en este caso es un balanceador de carga de aplicaciones

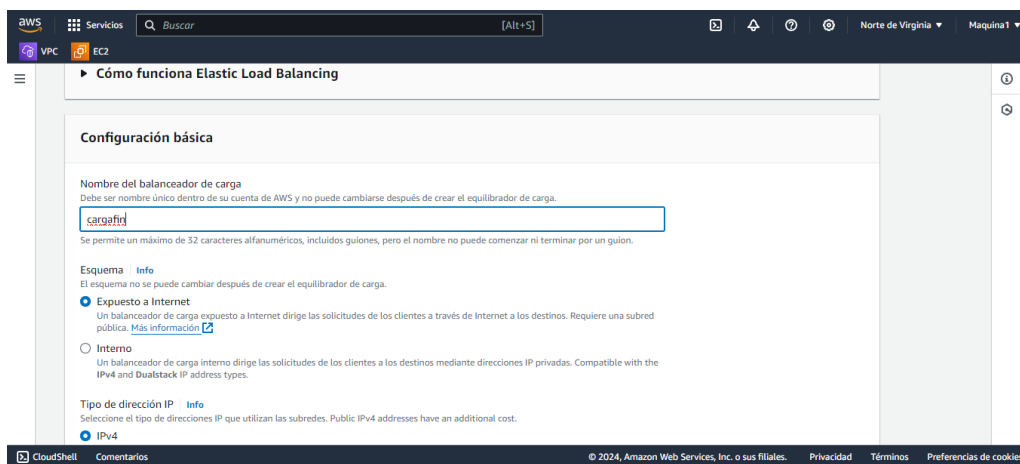
Figura 41 creador de balanceador de cargas



Fuente: Autoría propia

El balanceador se creará expuesto a internet, para que reciba las peticiones de los usuarios y defina a que instancia ingresa cada uno y así evitar que se congestionen y se caigan

*Figura 42 configuración de balanceador*



Fuente: Autoría propia

En la configuración del balanceador se debe definir el grupo objetivo de instancias que se van a asociar y sobre cuales el balanceador actuara

*Figura 43 creación de grupo de destino*

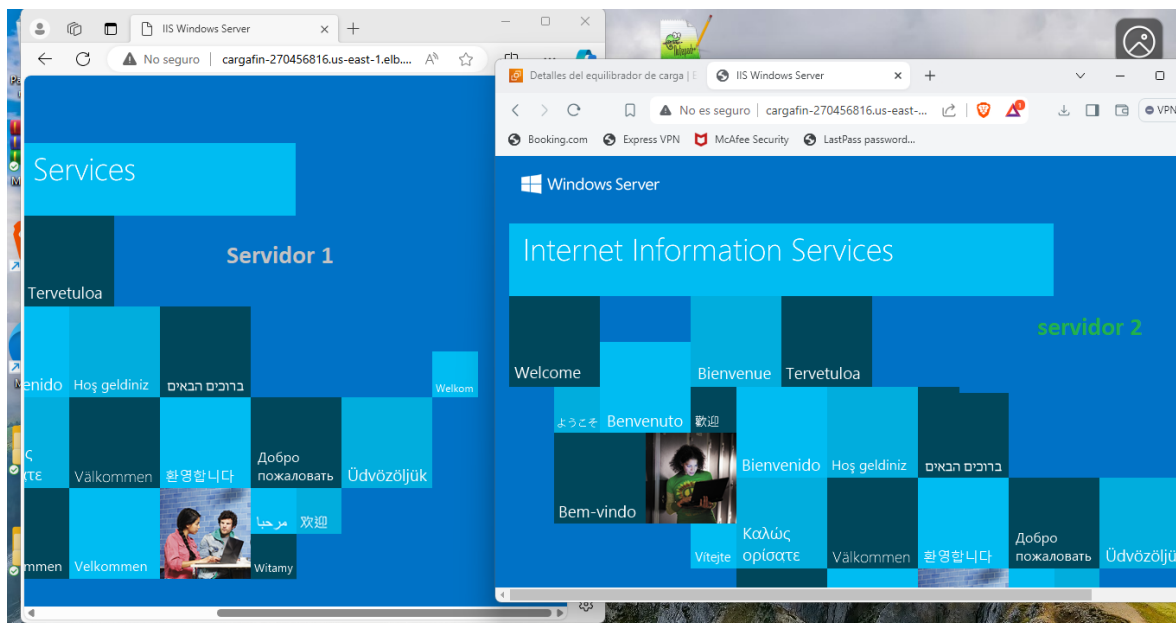


Fuente: Autoría propia



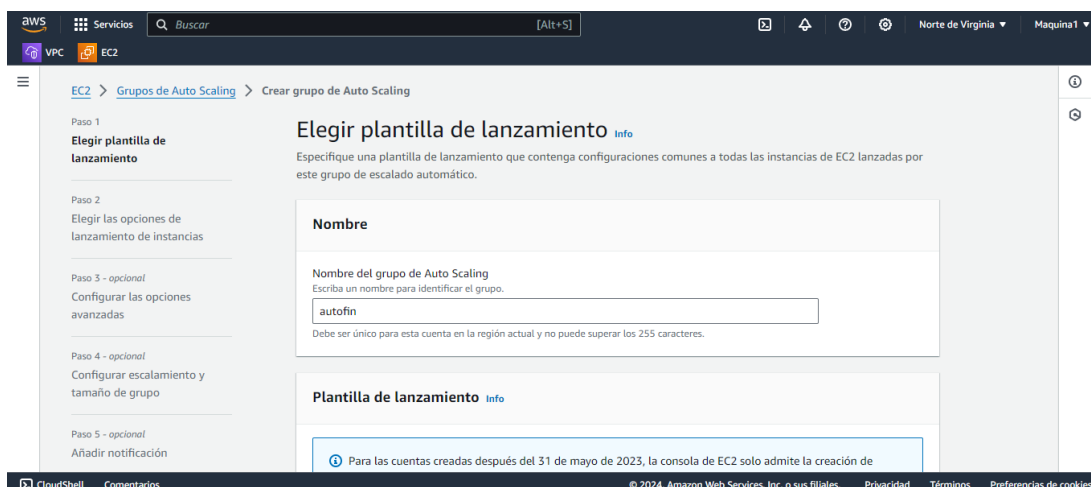
Después de tener todo esto listo, lo podemos probar al ingresar con el link que se encuentra en el balanceador de carga, el que siendo 1 solo, igual abrirá la página web de cada instancia cuando se hace desde navegadores distintos

*Figura 44 prueba de balanceador de carga*



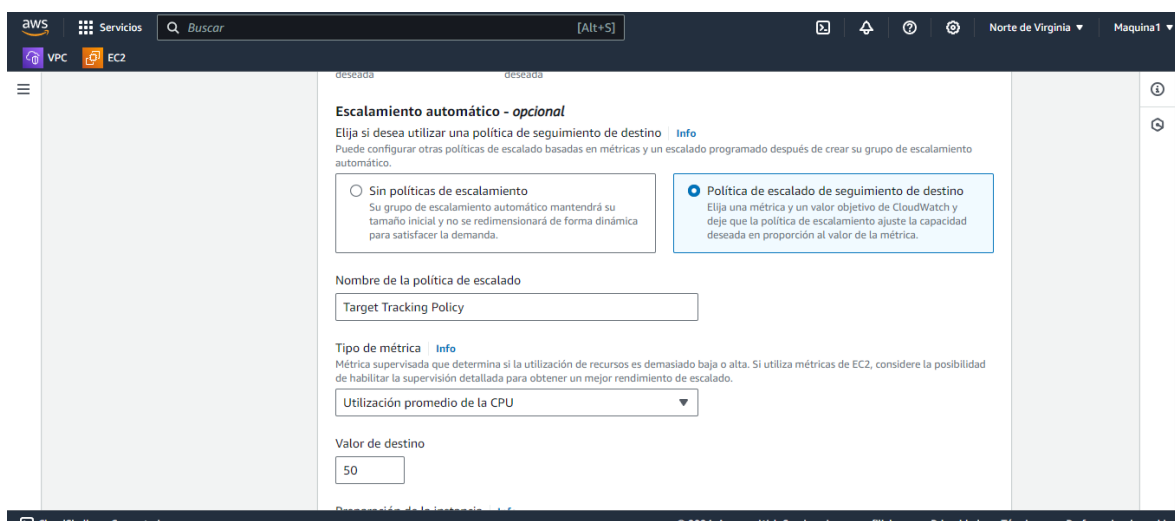
Fuente: Autoría propia

La siguiente parte será la creación del grupo de auto escalado, el cual usara el balanceador de carga que creamos y a partir de la AMI anterior, va a crear nuevas instancias de manera automática según la necesidad.

*Figura 45 creación de grupo de auto escalado*

Fuente: Autoría propia

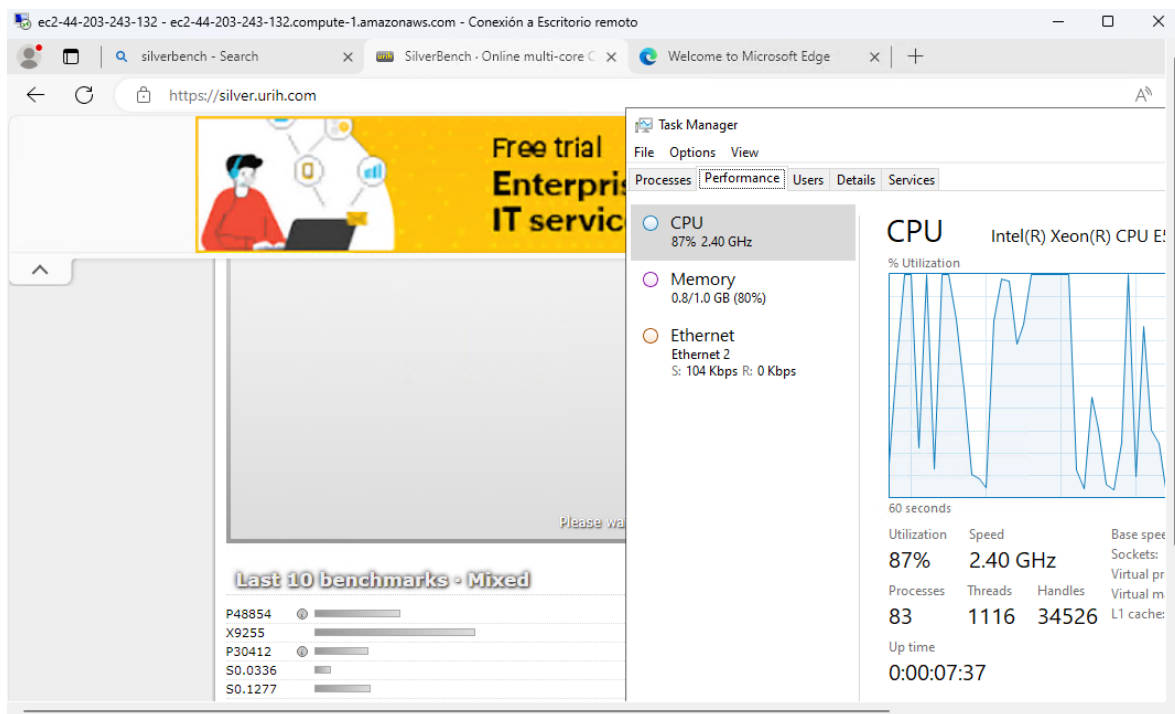
El grupo de auto escalado tiene una opción de escalamiento automático, la cual configuraremos para que cree nuevas instancias cuando el uso de CPU sea mayor al 50%. La cantidad de instancias que creara las definimos anteriormente en la configuración, poniendo una cantidad deseada, una cantidad mínima y una máxima.

*Figura 46 configuración de política de auto escalado*

Fuente: Autoría propia

Para probar que esta política funcione bien, nos conectaremos a una de las instancias creadas automáticamente, luego usaremos la página silverbench.com para forzar el uso de CPU al 100% y hacer que la política cree una nueva instancia para reemplazar esta

*Figura 47 forzado de CPU a 100%*



Fuente: Autoría propia

La activación de la política la podemos ver en la sección de historial de actividad del grupo de auto escalado.

Figura 48 ejecución de política

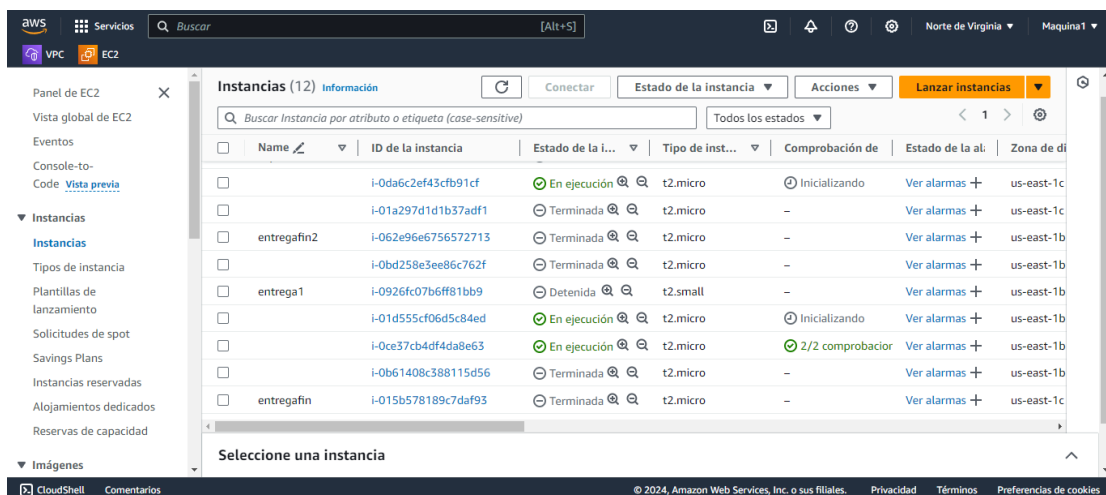


Fuente: Autoría propia

Finalmente se pueden ver las instancias creadas por la política de manera automática.

La política también se asegurará que el número de instancias no supere el máximo que definimos y luego de que se estabilizan eliminara las instancias extra creadas para mantener el mínimo y ahorra costos

Figura 49 instancias creadas automáticamente



Fuente: Autoría propia

## Conclusiones

En este trabajo se han explorado algunas de las funcionalidades que convierten a AWS en uno de los servicios líderes en lo que a computación en la nube se refiere.

La implementación de una instancia EC2 con dos contenedores en la nube demuestra cómo la computación en la nube revoluciona la manera en que las empresas gestionan sus recursos informáticos. A través de la creación de una VPC, la configuración de subredes y la asignación de direcciones IP, se ha destacado la facilidad y flexibilidad que ofrece AWS para desplegar y administrar aplicaciones de manera eficiente y confiable.

La configuración detallada de instancias EC2, la instalación y configuración de Apache HTTP Server en Linux, junto con la conexión a servicios como S3 y RDS MySQL, comprueba la capacidad de AWS para satisfacer las necesidades de cualquier proyecto, desde la implementación hasta la gestión de la infraestructura y los datos. Además, la funcionalidad de los balanceadores de carga y los grupos de auto escalado demuestra cómo AWS garantiza la disponibilidad y escalabilidad de las aplicaciones, adaptándose dinámicamente a la demanda del usuario.

A pesar de que solo se han usado una mínima parte de todas las soluciones que tiene AWS, estas demuestran las ventajas significativas que ofrece Amazon como proveedor líder de servicios de computación en la nube, proporcionando soluciones óptimas para una amplia gama de escenarios empresariales.

## Referencias

Amazon (2023). Alojamiento de aplicaciones. Obtenido de:

<https://aws.amazon.com/es/application-hosting/>

Amazon (2023). Computación en la nube con AWS. Obtenido de:

<https://aws.amazon.com/what-is-aws/>

Amazon (2023). Conozca nuestros productos. Obtenido de: <https://aws.amazon.com/es/>

Amazon (2023). Historias de éxito de los clientes. Obtenido de:

<https://aws.amazon.com/solutions/case-studies/>

Amazon (2024). ¿Qué es Amazon S3? Obtenido de:

[https://docs.aws.amazon.com/es\\_es/AmazonS3/latest/userguide/Welcome.html](https://docs.aws.amazon.com/es_es/AmazonS3/latest/userguide/Welcome.html)

Amazon (2024). Amazon Relational Database Service. Obtenido de:

<https://aws.amazon.com/es/rds/>

Amazon (2024). Bienvenido a la documentación de AWS. Obtenido de:

<https://docs.aws.amazon.com/>

Ceste (2024). AWS vs Azure vs Google Cloud: Comparativa de las 3 plataformas cloud líderes. Obtenido de: <https://www.cestes.es/noticias/aws-vs-azure-vs-google-cloud-comparativa-de-las-3-plataformas-cloud-lideres/>

Ceste (2024). Conoce las diferencias entre los principales servicios de Cloud Computing: AWS, Azure y GCP. Obtenido de: <https://www.cestes.es/areas-de-negocio/tecnologia/diferencias-aws-azure-gcp/>

Educaopen (2023). Historia de la nube: desde los años 50 hasta nuestros días. Obtenido de: <https://www.educaopen.com/digital-lab/blog/software/historia-de-la-nube>.

Fernandez E. (2024). Historia de la computación en la nube. Obtenido de:

<https://www.tokioschool.com/noticias/historia-computacion-nube/>

Fiz J. (2019). AWS vs Azure vs GCP: todos los servicios cloud frente a frente. Obtenido de: <https://www.paradigmadigital.com/dev/comparativa-servicios-cloud-aws-azure-gcp/>

Flexa (2024). 10 sitios Que Usan AWS (Amazon Web Services) Y No Sabías. Obtenido de: <https://flexa.cloud/es/10-sitios-web-que-usan-aws-amazon-web-services-y-no-sab%C3%ADas/>

Google (2023). Compara los servicios de AWS y Azure con Google Cloud. Obtenido de: <https://cloud.google.com/docs/get-started/aws-azure-gcp-service-comparison?hl=es-419>

Hewlett Packard Enterprise (2024). ¿Qué es la virtualización? Obtenido de: <https://www.hpe.com/lamerica/es/what-is/virtualization.html#:~:text=El%20origen%20de%20la%20virtualizaci%C3%B3n,un%20proceso%20a%20la%20vez.>

Jelvez H. (2023, 25 de mayo). Comparativa de Servicios en la Nube: Azure, AWS y GCP – Encuentra la Mejor Opción para tu Negocio. Obtenido de: <https://funcionit.com/comparativa-de-servicios-en-la-nube-azure-aws-y-gcp-encuentra-la-mejor-opcion-para-tu-negocio/>

Kcpdynamics (2023, 10 de mayo). Microsoft Azure App Service. Obtenido de: <https://www.kcpdynamics.com/blog-microsoft-dynamics/microsoft-azure-app-service/>

Keepcoding (2022, 13 de junio). ¿Qué es AWS CLI? Obtenido de: <https://keepcoding.io/blog/que-es-aws-cli/>

Microsoft (2024). Comparación de AWS con los servicios de Azure. Obtenido de: <https://learn.microsoft.com/es-es/azure/architecture/aws-professional/services>

Microsoft (2024). Encontrar los servicios de Azure para tus aplicaciones web. Obtenido de: <https://azure.microsoft.com/es-es/products/category/web>

Verdict (2020, 18 de junio). Cloud Computing: Timeline. Obtenido de: <https://www.verdict.co.uk/cloud-computing-timeline/>

Wikipedia (2024). Amazon Web Services. Obtenido de: [https://es.wikipedia.org/wiki/Amazon\\_Web\\_Services](https://es.wikipedia.org/wiki/Amazon_Web_Services)

Wikipedia (2024). Andy Jassy. Obtenido de: [https://es.wikipedia.org/wiki/Andy\\_Jassy](https://es.wikipedia.org/wiki/Andy_Jassy)

Wikipedia (2024). Google cloud. Obtenido de: [https://es.wikipedia.org/wiki/Google\\_Cloud](https://es.wikipedia.org/wiki/Google_Cloud)

Wikipedia (2024). Microsoft Azure. Obtenido de: [https://es.wikipedia.org/wiki/Microsoft\\_Azure](https://es.wikipedia.org/wiki/Microsoft_Azure)

Wikipedia (2024). Scott Guthrie. Obtenido de: [https://es.wikipedia.org/wiki/Scott\\_Guthrie](https://es.wikipedia.org/wiki/Scott_Guthrie)

Wikipedia (2024). Thoms Kurian. Obtenido de: [https://en.wikipedia.org/wiki/Thomas\\_Kurian](https://en.wikipedia.org/wiki/Thomas_Kurian)