



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

**Informe técnico Amazon Web Server**

Corporación Universitaria Remington.  
Facultad de ingeniería  
ingeniería de Sistemas

Daniel Lara Alvarez  
Juan Pablo Berrío López  
Opción de Trabajo de grado Seminario-Diplomado.  
2024

## Tabla de Contenidos

Tabla de figuras .....	3
Indice de tablas.....	4
Resumen .....	5
Marco conceptual y contextual .....	7
Computación en la nube.....	7
Amazon web services (AWS) .....	7
Máquina virtual .....	8
Instancia .....	8
base de datos.....	8
Servidor .....	9
WSL .....	9
Escritorio remoto.....	10
Desarrollo e implementación del aprendizaje .....	11
1. Virtualización y computación en la nube.....	11
1.1. Origen.....	11
1.2 Línea de tiempo de hitos de la computación en la nube.....	14
1.3 Comparación de servicios entre AWS, Azure y GCP.....	15
2. Contenedores de AWS, Servidor web y proxy .....	18
3. Identity and Access Management (IAM) y AWS CLI.....	28
4. Amazon S3(Simple Storage Service) y Amazon Relational Database Service (RDS).....	33
5. Balanceadores de carga y grupos de auto escalado.....	45
Conclusiones .....	62
Referencias .....	64

## Tabla de figuras

Figura 1. Línea de tiempo de hitos de la computación en la nube .....	14
Figura 2. Reglas de red de entrada .....	20
Figura 3. Archivo de hosts .....	22
Figura 4. Contenedores funcionando .....	23
Figura 5. Habilitación de módulos proxy.....	24
Figura 6. virtual host .....	25
Figura 7. Reinicio de servicio de Apache httpd .....	26
Figura 8. Páginas web comercio.com y tienda123456.com.....	27
Figura 9. Creación de organización en IAM.....	28
Figura 10. Creación de usuario en IAM.....	29
Figura 11. creación de grupo en IAM .....	30
Figura 12. Portal de acceso de usuario .....	31
Figura 13. Instalación de AWS CLI.....	32
Figura 14. Configuración de AWS CLI .....	33
Figura 15. creación de bucket en S3 .....	34
Figura 16. creación de Roles en IAM .....	35
Figura 17. Permisos de rol en IAM.....	36
Figura 18. política de bucket.....	37
Figura 19. configuración WSL.....	38
Figura 20. Subida de archivo a S3.....	39
Figura 21. Archivo en bucket.....	40
Figura 22. creación de base de datos MySQL.....	41
Figura 23. Configuración conectividad base de datos y EC2 .....	42
Figura 24. Configuración puertos.....	43
Figura 25. Instalación cliente MySQL .....	44
Figura 26. Acceso a base de datos.....	45
Figura 27. Creación de instancia de Windows.....	46
Figura 28. Instalación IIS .....	47
Figura 29. Edición de la página de inicio del IIS .....	48
Figura 30. Creación de snapshot .....	49
Figura 31. Creación de AMI .....	50
Figura 32. Creación de instancia a partir de AMI.....	51
Figura 33. Edición de página de inicio en server 2 .....	52
Figura 34. Creación de balanceador de cargas.....	53
Figura 35. Creación de Target Group.....	54
Figura 36. Acceso a servicios web a través de balanceador de carga .....	55
Figura 37. Creación de grupo de auto escalado .....	56
Figura 38. Configuración de grupo de auto escalado.....	57
Figura 39. Creación automática de instancias deseadas según auto escalado.....	58
Figura 40. Uso de Silverbench para forzar CPU.....	59
Figura 41. historial de actividad del grupo de auto escalado .....	60
Figura 42. Manejo de instancias del grupo de auto escalado .....	61

**Indice de tablas**

Tabla 1. Comparación de servicios entre AWS, Azure y GCP.....15

## Resumen

La computación en la nube ha revolucionado la forma en que las empresas gestionan, almacenan y acceden a los datos. Desde el nacimiento de la idea en la década de los 60, múltiples empresas han aportado al campo para hacer posible esta tecnología, y Amazon Web Services (AWS) ha sido un pilar fundamental en este avance. Desde sus inicios en 2006, AWS ha evolucionado hasta convertirse en el líder indiscutible en servicios de infraestructura en la nube, ofreciendo una amplia gama de servicios que permiten a las organizaciones adaptarse y escalar sus operaciones sin la necesidad de inversiones significativas en hardware.

Este enorme abanico de servicios incluye algunos que se han usado en ejercicios prácticos en este trabajo para demostrar su funcionalidad, tal es el caso del servicio de EC2, o Elastic Compute Cloud, el cual permite la creación de instancias de máquinas virtuales que pueden servir como servidores para aplicaciones de todo tipo, además de alojar contenedores que permitan ejecuciones de servicios usando un mínimo de recursos.

También se exploran servicios como S3(Simple Storage Service), el cual permite almacenar archivos en espacios reservados de la nube de Amazon. Otros servicios como RDS (Relational Database Service) ofrecen la posibilidad de crear bases de datos relacionales de manera sencilla y con un mantenimiento autogestionado por Amazon, simplificando lo que se requiere para mantener este tipo de base de datos.

Finalmente, las herramientas de balanceadores de carga y grupos de auto escalado garantizan que los servicios alojados en las instancias tengan alta disponibilidad, asegurando que en caso de que una instancia falle, una nueva instancia a modo de espejo pueda reemplazarla y mantener al máximo el acceso a la aplicación alojada.

La optimización de recursos, las herramientas para asegurar la disponibilidad, las robustas herramientas de seguridad y la escalabilidad convierten a AWS en un servicio que el campo de la computación en la nube, permitiendo a las empresas innovar rápidamente y adaptarse a las cambiantes necesidades del mundo moderno.

### **Palabras clave**

Computación en la nube, virtualización, Amazon web server, instancias, servidor

## **Marco conceptual y contextual**

Para poder ejecutar los ejercicios propuestos en este informe los cuales demuestran la funcionalidad de muchas de las herramientas ofrecidas por Amazon en su servicio de AWS, es necesario conocer y dominar algunas terminologías relacionadas con el tema, las cuales se exponen a continuación:

### **Computación en la nube**

La computación en la nube es la disponibilidad a pedido de los recursos de procesamiento como los servicios por Internet. Elimina la necesidad de que las empresas obtengan, configuren o administren recursos por su cuenta; de esta forma, solo paguen por lo que usan. Hay tres tipos de modelos de servicio de computación en la nube. La infraestructura como servicio, que ofrece servicios de procesamiento y almacenamiento, la plataforma como servicio, que ofrece un entorno de desarrollo y de implementación para compilar en la nube y el software como servicio, que proporciona apps como servicios (Google, 2024).

### **Amazon web services (AWS)**

Amazon Web Services (AWS) es la nube más adoptada y completa en el mundo, que ofrece más de 200 servicios integrales de centros de datos a nivel global. AWS cuenta con una cantidad de servicios y de características incluidas en ellos que supera la de cualquier otro proveedor de la nube, ofreciendo desde tecnologías de infraestructura como cómputo, almacenamiento y bases de datos hasta tecnologías emergentes como aprendizaje automático e inteligencia artificial, lagos de datos y análisis e internet de las cosas. Esto hace que llevar las aplicaciones existentes a la nube sea más rápido, fácil y rentable y permite crear casi cualquier cosa que se pueda imaginar (Amazon, 2024).

## **Máquina virtual**

Una máquina virtual (VM) es un entorno informático que funciona como un sistema aislado con su propia CPU, memoria, interfaz de red y almacenamiento, el cual se crea a partir de un conjunto de recursos de hardware. Mediante un sistema de software denominado hipervisor, se aíslan los recursos informáticos necesarios y se crean y gestionan las máquinas virtuales. Las máquinas virtuales permiten que se ejecuten varios sistemas operativos diferentes a la vez en una misma computadora, como cuando se ejecuta una distribución de Linux® en un sistema macOS o Windows. Cada sistema operativo se ejecuta tal como lo harían un sistema operativo o una aplicación en el hardware del host (Redhat, 2023).

## **Instancia**

Una instancia de computación en la nube es un recurso de servidor que brindan servicios en la nube de terceros. Si bien puede administrar y mantener los recursos del servidor físico en las instalaciones, hacerlo es costoso e ineficiente. Los proveedores de la nube mantienen el hardware en sus centros de datos y les dan acceso virtual a los recursos de computación en forma de instancia. Puede usar la instancia en la nube para ejecutar cargas de trabajo con uso intensivo de cómputos, como contenedores, bases de datos, microservicios y máquinas virtuales (Amazon, 2024).

## **base de datos**

Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático. Normalmente, una base de datos está controlada por un sistema de gestión de bases de datos (DBMS). En conjunto, los datos y el DBMS, junto con las aplicaciones asociadas a ellos, reciben el nombre de sistema de bases de datos, abreviado normalmente a simplemente base de datos. Los datos de los tipos

más comunes de bases de datos en funcionamiento actualmente se suelen utilizar como estructuras de filas y columnas en una serie de tablas para aumentar la eficacia del procesamiento y la consulta de datos. Así, se puede acceder, gestionar, modificar, actualizar, controlar y organizar fácilmente los datos. La mayoría de las bases de datos utilizan un lenguaje de consulta estructurada (SQL) para escribir y consultar datos (Oracle, 2023).

### **Servidor**

Es un aparato informático que almacena, distribuye y suministra información. Los servidores funcionan basándose en el modelo “cliente-servidor”. El cliente puede ser tanto un ordenador como una aplicación que requiere información del servidor para funcionar. Por tanto, un servidor ofrecerá la información demandada por el cliente siempre y cuando el cliente esté autorizado. Los servidores pueden ser físicos o virtuales. Los servicios que prestan los servidores son requeridos continuamente y, por tanto, la mayoría de los servidores nunca se apagan. Si un servidor dejara de funcionar, eso puede causar muchos problemas a los usuarios. Por tanto, los servidores suelen estar programados para ser tolerantes a fallos (Ticportal, 2023).

### **WSL**

WSL corresponde con las siglas de «Windows Subsystem Linux» y es un software que permite la instalación de sistemas operativos de la familia GNU/Linux en Windows. Si bien es cierto que antes de WSL podíamos virtualizar usando un programa como Virtual Box, la experiencia distaba un poco de la que necesitaban los desarrolladores. Envolturas como Vagrant consiguieron mejorar bastante la experiencia de trabajo, pero todavía había algunas cuestiones importantes que solucionar y sobre todo, Docker todavía no se adaptaba bien a esta solución. Por debajo WSL funciona con virtualización, la diferencia con otros sistemas como Virtual Box es

que la creación de las máquinas virtuales se realiza automáticamente y sin dificultades ni procesos complejos para el usuario (Garcia Fernan, 2023).

### **Escritorio remoto**

La tecnología de escritorio remoto puede ser definida como el medio por el cual uno o varios usuarios pueden acceder a distancia a un ordenador desde otro ordenador. Como su propio nombre indica, la conexión ha de ser remota, lo cual implica que ambos ordenadores estén situados en redes diferentes, cada uno con su propio IP (Protocolo de Internet). Esta tecnología es extremadamente útil tanto para empresas como para particulares que deseen hacer uso de un determinado ordenador se encuentren donde se encuentren, simplemente contando con una conexión a internet y permaneciendo ambos ordenadores encendidos de manera simultánea. Gracias a ello, el ahorro económico, técnico y de tiempo se hace patente al prescindir de desplazamientos innecesarios (pippol, 2017).

## Desarrollo e implementación del aprendizaje

### 1. Virtualización y computación en la nube

#### 1.1. Origen

la virtualización es una tecnología que permite emular de manera virtual un servidor, computador o sistema operativo, logrando que se ejecute un computador dentro de otro computador al mismo tiempo, en una instancia “aislada”. La creación de particiones en un disco duro físico también se considera virtualización. La capacidad de la máquina virtual que se ejecuta está limitada por la capacidad física del hardware del computador que la ejecuta.

Esta tecnología tan vital en la actualidad tiene sus orígenes en la década de 1960, con la creación de los sistemas operativos de multiprogramación de IBM, los cuales permitían que varios usuarios ejecutaran programas en un solo computador de tipo mainframe al mismo tiempo.

Con el avance de la tecnología, IBM lanzó el sistema operativo VM (virtual machine) en 1972, con el cual un solo servidor o máquina física podría ejecutar múltiples sistemas operativos a la vez usando máquinas virtuales. Esto permitió aprovechar al máximo los recursos de cada máquina y sería la base para la tecnología de virtualización actual.

Posteriormente la virtualización perdería algo de relevancia y popularidad en las décadas de 1980 y 1990 debido al surgimiento de los PC (computadores personales), los cuales eran más accesibles que los Mainframe de IBM. Sin embargo, para las grandes empresas, la virtualización seguía siendo muy importante, por lo cual empresas como VMware, desarrollarían software de virtualización para los PC, abriendo una nueva era para la virtualización y sentando las bases de lo que se convertiría luego en el cloud computing.

La computación en la nube o cloud computing, usa la virtualización como parte la tecnología que permite su funcionamiento, por lo cual ambas están estrechamente relacionadas.

ARPANET, cuyo origen se remonta a 1969, se considera como el predecesor de internet, se considera como una gran influencia para el surgimiento de la computación en la nube, ya que fue la primera red que permitía compartir archivos entre computadores que no estuvieran físicamente en el mismo sitio.

J.C.R Licklider quien estuvo detrás de este proyecto, imaginó un mundo donde todo el mundo estaría conectado. De esta visión, de esta idea, surge el desarrollo del cloud computing. De una idea en la que todos tendríamos la capacidad de acceder a programas y datos desde cualquier lugar del mundo (Fernández E., 2021).

El nacimiento del internet en 1990, creó la necesidad de servicios en línea más robustos, lo cual impulsó a que diferentes empresas en la década crearan sus propias redes privadas virtuales, también conocidas como VPN. A mediados de esta década, es que se empieza a hablar de cloud computing.

En 1999, Salesforce.com se convirtió en un ejemplo popular de usar la computación en la nube de manera exitosa. La usaron para ser pioneros en la idea de usar el internet para entregar programas de software a usuarios finales. El programa podía ser accesado y descargado por cualquiera con acceso a internet. Las empresas podían comprar software de una manera eficiente en costos y por demanda sin salir de la oficina (Foote K., 2021).

El verdadero auge de la computación en la nube ocurrió en la década de 2000. En 2002, Amazon Web Services (AWS) lanzó su plataforma inicial, proporcionando servicios básicos de almacenamiento y computación. Sin embargo, fue en 2006 cuando AWS introdujo Elastic

Compute Cloud (EC2), lo cual permitió a las empresas alquilar servidores virtuales según sus necesidades, marcando un punto significativo en la historia de la computación en la nube.

Google también jugó un papel crucial con el lanzamiento de Google App Engine en 2008, permitiendo a los desarrolladores construir y hospedar aplicaciones web en la infraestructura de Google. Microsoft entró en el mercado con Azure en 2010, ofreciendo una plataforma completa de servicios en la nube.

En 2011, IBM y Apple lanzaron sus versiones de computación en la nube con IBM Smartcloud y Icloud respectivamente, además de que Microsoft empezó a publicitar la nube en televisión, dándola a conocer al público en general como una solución para guardar fotos y videos de manera fácil.

Finalmente, Oracle lanzaría Oracle cloud en 2012, ofreciendo un paquete de 3 servicios lo que se convertirá en la norma de negocio, plataforma como servicio, infraestructura como servicio y software como servicio.

## 1.2 Línea de tiempo de hitos de la computación en la nube



Figura 1. Línea de tiempo de hitos de la computación en la nube

Fuente: Elaboración propia

### 1.3 Comparación de servicios entre AWS, Azure y GCP

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

Servicio	AWS	Azure	GCP
<b>Máquina virtual:</b> permite la creación de máquinas virtuales que emulan hardware físico para sistemas operativos y aplicaciones	<b>EC2(elastic computer cloud):</b> múltiples instancias con opciones para diferentes necesidades de memoria, computación y almacenamiento	<b>Azure virtual Machines:</b> posee múltiples sistemas operativos e integración con servicios de azure	<b>Google compute Engine:</b> ofrece opciones de personalización, precios predefinidos e integración con los servicios de Google cloud
<b>Contenedores:</b> Servicios que facilitan la creación, despliegue y gestión de aplicaciones en contenedores, proporcionando una capa de abstracción sobre el hardware subyacente.	<b>Amazon ECS (elastic container service) y Amazon EKS (elastic kubernetes service):</b> servicios de orquestación de contenedores administrado que soporta Docker y proporciona kubernetes gestionados	<b>Azure kubernetes service y Azure container instances:</b> servicios de orquestación de contenedores basado kubernetes y permite ejecución de contenedores sin necesidad de gestionar servidores respectivamente	<b>Google Kubernetes Engine y Google cloud run:</b> servicio de Kubernetes totalmente gestionado y que permite ejecutar contenedores sin servidor respectivamente
<b>Almacenamiento de archivos:</b> Servicios de almacenamiento diseñados para compartir archivos entre múltiples instancias y usuarios, similares a los sistemas de archivos de red.	<b>Amazon EFS (Elastic File System):</b> proporciona un sistema de archivos escalable y elástico para uso con instancias de EC2.	<b>Azure Files:</b> ofrece un sistema de archivos basado en SMB (Server Message Block) accesible desde cualquier parte del mundo.	<b>Google Cloud Filestore:</b> proporciona almacenamiento de archivos de alto rendimiento, integrado con Compute Engine.
<b>Bases de datos Relacionales:</b> Servicios que ofrecen	<b>Amazon RDS (Relational Database Service):</b>	<b>Azure SQL Database:</b> es una base de datos	<b>Google Cloud SQL:</b> ofrece bases de datos MySQL, PostgreSQL

bases de datos relacionales gestionadas, soportando SQL y asegurando consistencia transaccional	soporta múltiples motores de bases de datos, como MySQL, PostgreSQL, SQL Server, y más.	relacional gestionada que proporciona escalabilidad automática y alto rendimiento.	y SQL Server gestionadas.
<b>Bases de Datos NoSQL:</b> Servicios que proporcionan bases de datos no relacionales, diseñadas para manejar grandes volúmenes de datos y escalabilidad horizontal.	<b>Amazon DynamoDB:</b> es una base de datos NoSQL totalmente gestionada, conocida por su baja latencia y alta disponibilidad.	<b>Azure Cosmos DB:</b> es un servicio de base de datos NoSQL distribuido globalmente, soportando múltiples modelos de datos.	<b>Google Cloud Firestore y Google Cloud Bigtable:</b> son bases de datos NoSQL optimizadas para diferentes tipos de cargas de trabajo.
<b>Redes virtuales:</b> Servicios que permiten crear y gestionar redes virtuales aisladas dentro de la nube, conectando instancias y servicios.	<b>Amazon VPC (Virtual Private Cloud):</b> permite la creación de redes virtuales personalizables y seguras.	<b>Azure Virtual Network:</b> ofrece la creación de redes virtuales, conexiones VPN, y otras funciones de red	<b>Google Virtual Private Cloud (VPC):</b> proporciona redes virtuales escalables con opciones avanzadas de configuración.
<b>Machine learning:</b> Servicios que proporcionan herramientas y entornos para construir, entrenar y desplegar modelos de machine learning.	<b>Amazon SageMaker:</b> facilita la construcción, entrenamiento y despliegue de modelos de machine learning a escala	<b>Azure Machine Learning:</b> ofrece un entorno basado en la nube para crear, entrenar y desplegar modelos de machine learning.	<b>Google AI Platform:</b> proporciona servicios y herramientas para desarrollar e implementar modelos de machine learning.
<b>Supervisión y monitoreo:</b> Servicios que permiten monitorear el rendimiento y estado de los recursos y	<b>Amazon CloudWatch:</b> proporciona monitoreo y gestión de recursos,	<b>Azure Monitor:</b> ofrece monitoreo de rendimiento y diagnósticos para aplicaciones y recursos de Azure.	<b>Google Cloud Monitoring (basado en Stackdriver):</b> proporciona monitoreo de aplicaciones y

aplicaciones en la nube.	aplicaciones y servicios de AWS.		servicios de Google Cloud.
<b>Streaming de datos:</b> Servicios que permiten la ingestión, procesamiento y análisis en tiempo real de flujos de datos.	<b>Amazon Kinesis:</b> facilita la ingestión y análisis de grandes flujos de datos en tiempo real.	<b>Azure Event Hubs:</b> permite la ingestión de eventos y datos en tiempo real a gran escala.	<b>Google Cloud Dataflow y Google Cloud Pub/Sub:</b> proporciona procesamiento en tiempo real y por lotes y permite la ingesta y distribución de flujos de datos respectivamente.
<b>Automatización y CI/CD:</b> Servicios que soportan la integración continua (CI) y la entrega continua (CD), facilitando la automatización de las pruebas y despliegues de aplicaciones.	<b>AWS CodePipeline y AWS CodeBuild:</b> soportan la automatización de CI/CD, integrándose con otros servicios de AWS.	<b>Azure DevOps y Azure Pipelines:</b> proporcionan un conjunto de herramientas para la gestión de proyectos y la automatización de CI/CD.	<b>Google Cloud Build:</b> permite la construcción y entrega continua de aplicaciones, integrándose con otros servicios de Google Cloud.
<b>Gestión de Identidades y Acceso:</b> Servicios que proporcionan gestión de identidades y control de acceso a recursos y servicios en la nube.	<b>AWS IAM (Identity and Access Management):</b> permite gestionar usuarios, roles y permisos para acceder a los servicios de AWS.	<b>Azure Active Directory:</b> ofrece gestión de identidades y control de acceso con integración para servicios de Microsoft.	<b>Google Cloud IAM:</b> proporciona gestión de identidades y políticas de acceso a recursos de Google Cloud.
<b>CDN (Content Delivery Network):</b> Servicios que distribuyen contenido a través de una red global de servidores, reduciendo la latencia	<b>Amazon CloudFront:</b> es un servicio de CDN que ofrece baja latencia y alta transferencia de datos a través de una red global.	<b>Azure CDN:</b> ofrece distribución de contenido con múltiples proveedores de CDN para optimizar la entrega.	<b>Google Cloud CDN:</b> utiliza la infraestructura global de Google para ofrecer contenido de manera eficiente y rápida.

y mejorando el rendimiento.			
<b>Bases de Datos Data Warehouse:</b> Servicios que proporcionan almacenamiento y análisis de datos a gran escala, optimizados para consultas analíticas complejas.	<b>Amazon Redshift:</b> es un servicio de data warehousing gestionado que permite el análisis de petabytes de datos.	<b>Azure Synapse Analytics (anteriormente SQL Data Warehouse):</b> es una solución de análisis que integra almacenamiento de datos y análisis de big data.	<b>Google BigQuery:</b> es un almacén de datos altamente escalable y sin servidor, optimizado para consultas rápidas.

fuente: Elaboración propia

## 2. Contenedores de AWS, Servidor web y proxy

En las instancias de Amazon web server se tiene la posibilidad de crear contenedores usando Docker, los cuales son micro instancias dentro de la instancia principal en las cuales se pueden alojar páginas web o aplicaciones usando un mínimo de recursos de la máquina.

Para ilustrar esto, se propone un ejercicio en el que se crearon contenedores en una instancia de Linux en los cuales se alojó una imagen de página web básica. Cada contenedor tiene un puerto de red específico desde la instancia principal (8080,8081,8082), aunque internamente todos tiene el mismo puerto (80). Las páginas web son accesibles de manera local a través de la IP de la instancia de Linux y apuntando al puerto, por ejemplo, usando en el navegador 18.191.244.225:8082 me debe llevar a la página alojada en el contenedor con ese puerto, pero no es una manera conveniente para acceder páginas web.

Para solucionar este problema y poder ingresar solo con la URL a las páginas comercio.com y tienda123456.com, las cuales están alojadas en los contenedores con puertos 8081 y 8082 respectivamente, se debe configurar un servidor web y un proxy inverso.

El web server usado para este ejercicio es Apache httpd, el cual permitirá acceder mediante http las páginas alojadas en los contenedores.

El servidor Apache de HTTP, coloquialmente conocido como Apache, es una aplicación de servidor web notable por jugar un rol clave en los inicios de la World Wide Web. Originalmente basado en el servidor NCSA HTTPd, el desarrollo de Apache empezó en los inicios de 1995 después de que el trabajo en el código de NCSA se estancó. Apache rápidamente sobrepasó a NCSA HTTPd como el servidor HTTP dominante, y ha permanecido como el servidor HTTP más popular en uso desde abril de 1996 (Docker, 2024).

Un proxy de reenvío, con frecuencia conocido como proxy, servidor proxy o proxy web, es un servidor que se sitúa delante de un grupo de máquinas cliente. Cuando esos ordenadores realizan solicitudes a sitios y servicios en Internet, el servidor proxy intercepta esas peticiones y luego se comunica con los servidores web en nombre de esos clientes, como un intermediario (Cloudflare, 2024).

Un proxy inverso es un servidor que se sitúa delante de uno o varios servidores web, e intercepta las solicitudes de los clientes. Esto es diferente de un proxy de reenvío, en el que el proxy se sitúa delante de los clientes. Con un proxy inverso, cuando los clientes envían solicitudes al servidor de origen de un sitio web, el servidor de proxy inverso intercepta esas solicitudes en el perímetro de la red. El servidor proxy inverso enviará entonces las solicitudes al servidor de origen y recibirá las respuestas del servidor de origen (Cloudflare, 2024).

Como requisito inicial debemos asegurarnos de que las reglas de entrada de la instancia de Linux en AWS permitan la comunicación con los puertos de cada contenedor (en este caso los puertos 8081 y 8082) y permitan también la entrada de solicitudes hacia el puerto http, el cual en este caso es el 80. Sin estas reglas configuradas de forma correcta será imposible interactuar con los contenedores y sus respectivas páginas web.

Cabe señalar que toda esta interacción se hace de forma local entre el PC del usuario y la instancia de Linux en la nube de AWS, ya que no se está trabajando con nombres de dominio de internet.

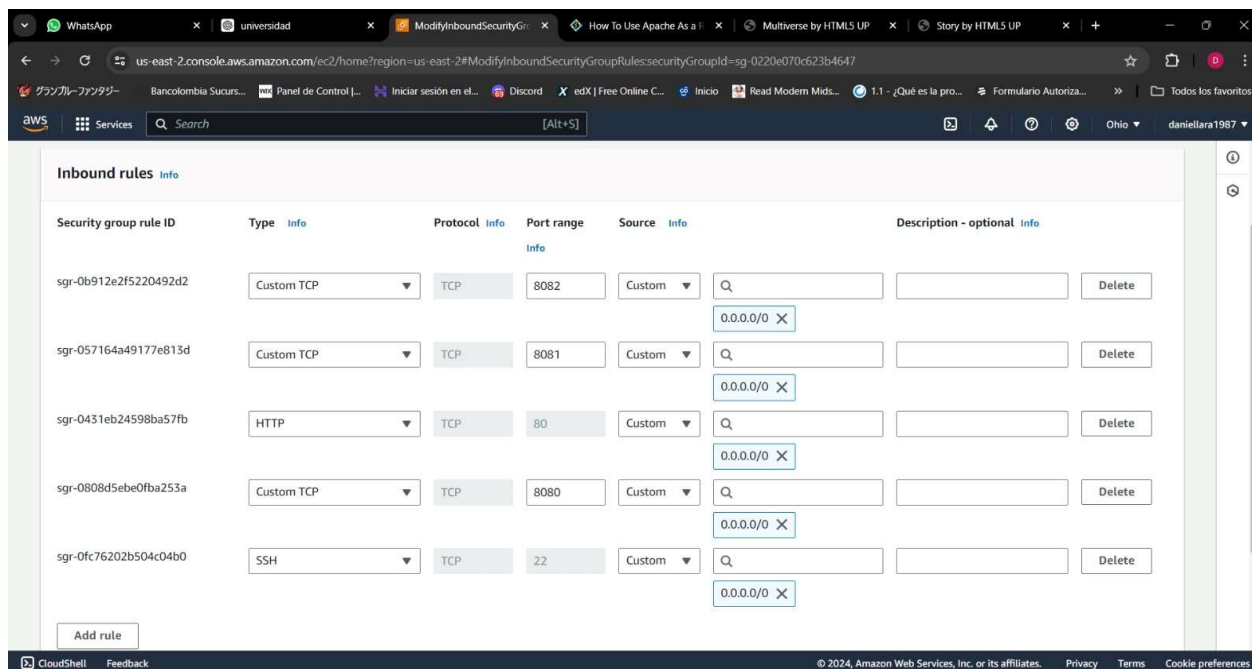


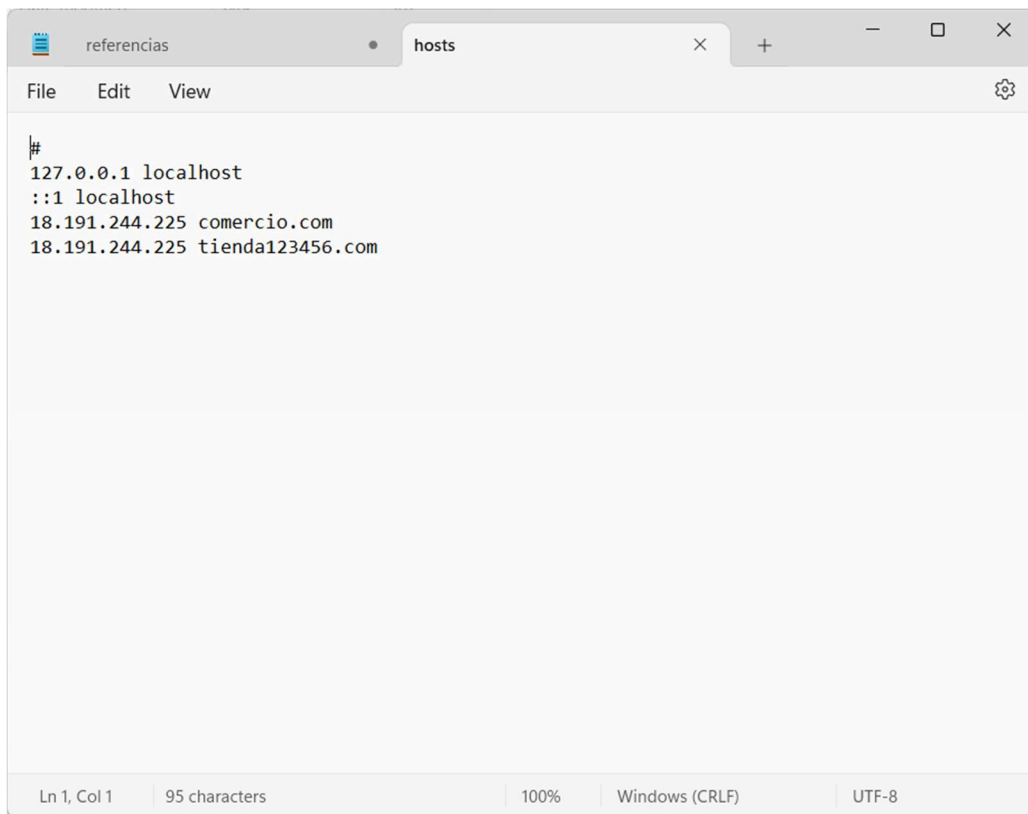
Figura 2. Reglas de red de entrada

Fuente: Elaboración propia

Otro requisito importante nacido de la interacción únicamente local es la edición del archivo de hosts que se encuentra en la carpeta “etc” de la ruta Windows/system32/drivers. Es necesario agregar la IP de la instancia de Linux a este archivo, seguido del nombre de la pagina web, esto con el fin de que, de manera local, los navegadores interpreten que al ingresar al sitio web con ese nombre, deben apuntar a la dirección IP dada (en este caso la dirección IP del servidor Linux es 18.191.244.225 y las páginas se llaman comercio.com y tienda123456.com respectivamente).

Dado que ambas paginas están alojadas en el mismo servidor, usan la misma IP, la diferenciación ocurre en los puertos que cada contenedor tiene, siendo una en el puerto 8081 y otra en el 8082.

Debido a que la página web tienda.com existe en internet, y esto ocasionaba errores en el navegador, se optó por usar el nombre de página tienda123456.com para evitar que los navegadores redirigieran a la página de internet publicada en la web.

A screenshot of a text editor window titled 'referencias' with a tab for 'hosts'. The editor shows the following content:

```
#  
127.0.0.1 localhost  
::1 localhost  
18.191.244.225 comercio.com  
18.191.244.225 tienda123456.com
```

The status bar at the bottom indicates 'Ln 1, Col 1', '95 characters', '100%', 'Windows (CRLF)', and 'UTF-8'.

*Figura 3. Archivo de hosts*

Fuente: Elaboración propia

Posteriormente, se debe conectar a la instancia de Linux mediante un cliente de SSH, en este caso se usó WSL, el cual nos permitirá interactuar con cada contenedor y hacer los ajustes necesarios para que todo funcione como se desea.

Ya estando en consola se procede a revisar que los contenedores previamente creados estén en línea y funcionando, usando el comando “Docker ps”.



```

GNU nano 5.8 /etc/httpd/conf/httpd.conf
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
#
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so

#
# 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.
IncludeOptional conf.d/*.conf

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo     M-A Set Mark
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo     M-6 Copy

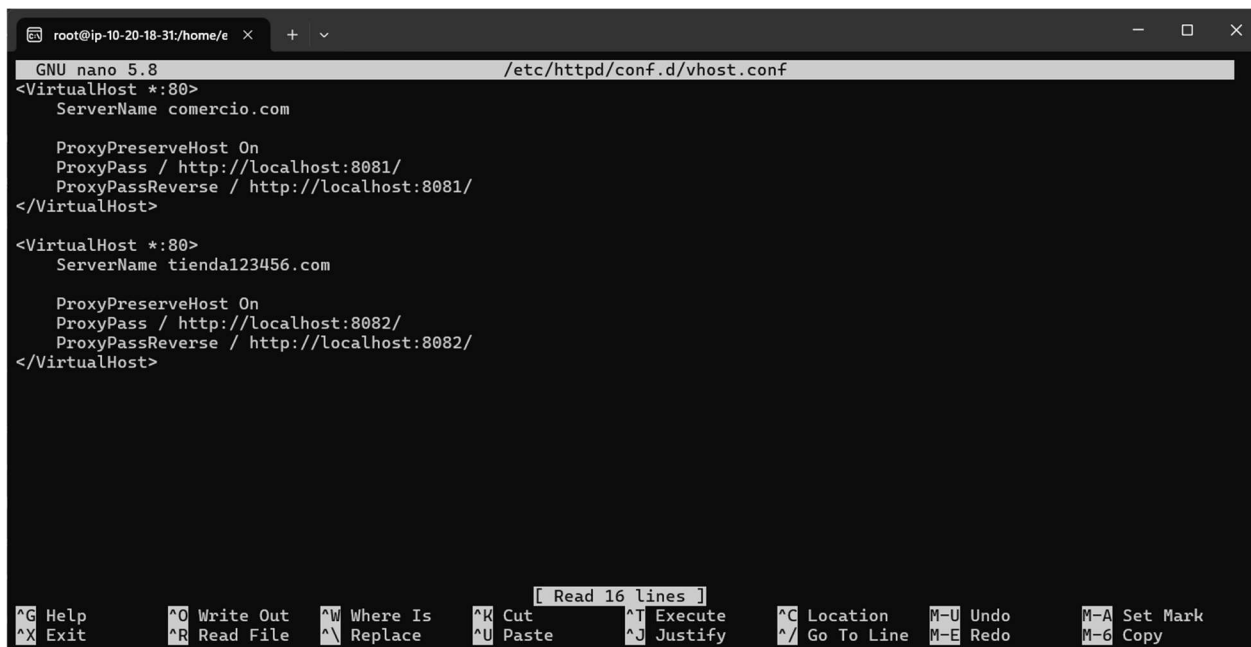
```

*Figura 5. Habilitación de módulos proxy*

Fuente: Elaboración propia

Luego se debe crear el archivo de virtual host usando el comando “sudo nano /etc/httpd/conf.d/vhost.conf”. Este nuevo archivo debe contener la información y los detalles de las redirecciones que se necesitan para cada página, las cuales acá aparecen en el apartado de ServerName.

En este archivo también se especifican las direcciones IP (en este caso la dirección del local host) y los puertos a los que debe redireccionarse cada solicitud hecha por los navegadores del cliente. Se puede evidenciar que el puerto principal por el que entran dichas solicitudes es el 80 y que dependiendo si el nombre de página es comercio.com o tienda123456.com, se redirigirá el navegador a la dirección del localhost con el puerto 8081 o 8082 correspondientemente.



```
root@ip-10-20-18-31:/home/e x + v
GNU nano 5.8 /etc/httpd/conf.d/vhost.conf
<VirtualHost *:80>
  ServerName comercio.com

  ProxyPreserveHost On
  ProxyPass / http://localhost:8081/
  ProxyPassReverse / http://localhost:8081/
</VirtualHost>

<VirtualHost *:80>
  ServerName tienda123456.com

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

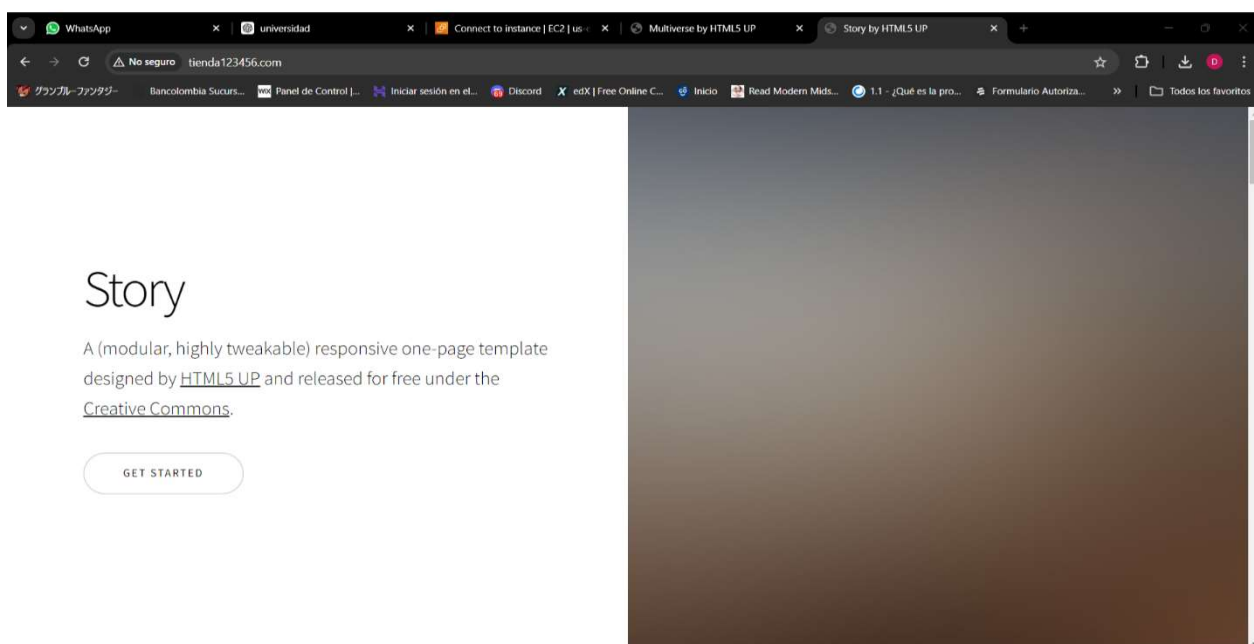
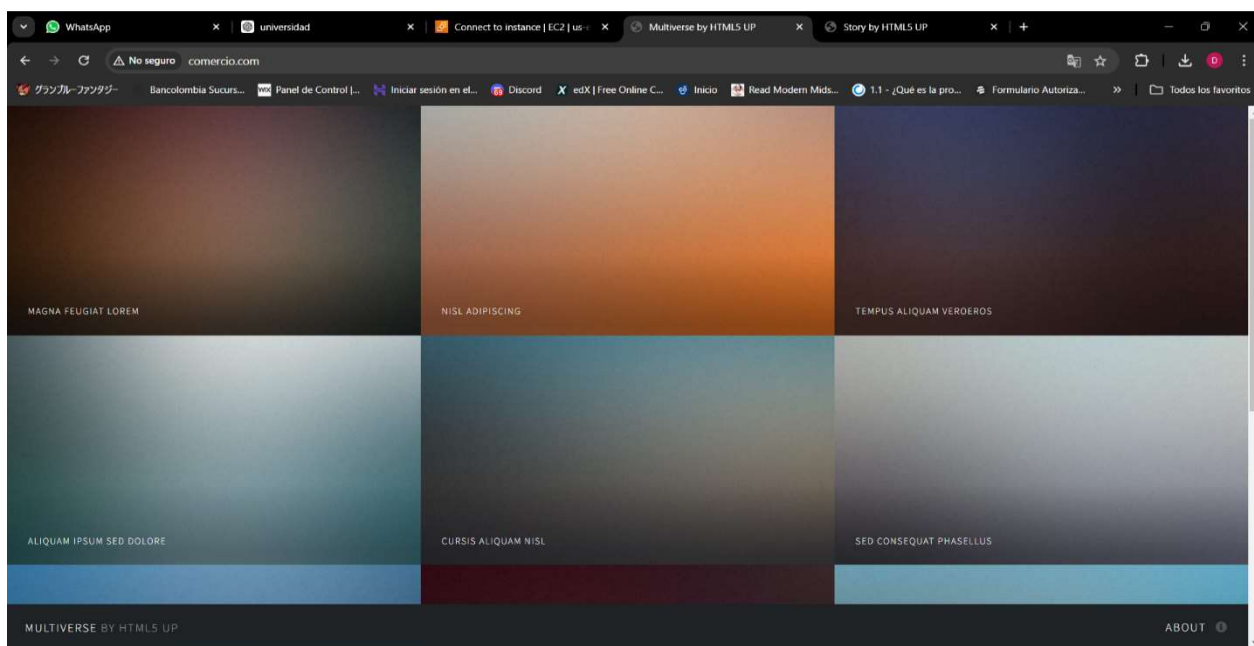
[ Read 16 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo      M-6 Copy
```

*Figura 6. virtual host*

Fuente: Elaboración propia

Esto culminaría las configuraciones necesarias para que el redireccionamiento generado por el proxy inverso se ejecute como es debido. Finalmente, se recomienda reiniciar los servicios de Apache httpd usando el comando “sudo systemctl restart httpd”, para asegurarnos que los servicios de proxy y web server respondan de manera correcta.





*Figura 8. Páginas web comercio.com y tienda123456.com*

Fuente: Elaboración propia

### 3. Identity and Access Management (IAM) y AWS CLI

AWS ofrece un servicio de identidades y acceso a los recursos que permite la creación de usuarios con permisos y roles específicos para mayor seguridad y administración de recursos.

Con AWS Identity and Access Management (IAM), puede especificar quién o qué puede acceder a los servicios y recursos en AWS, administrar de forma centralizada los permisos específicos y analizar el acceso para perfeccionar los permisos en todo AWS (Amazon, 2024).

Dentro del identity center la organización puede crear usuarios y grupos a los cuales asignarles recursos y limitar que pueden hacer con ellos.

También es posible crear roles que aplique específicamente a instancias o productos específicos de AWS.

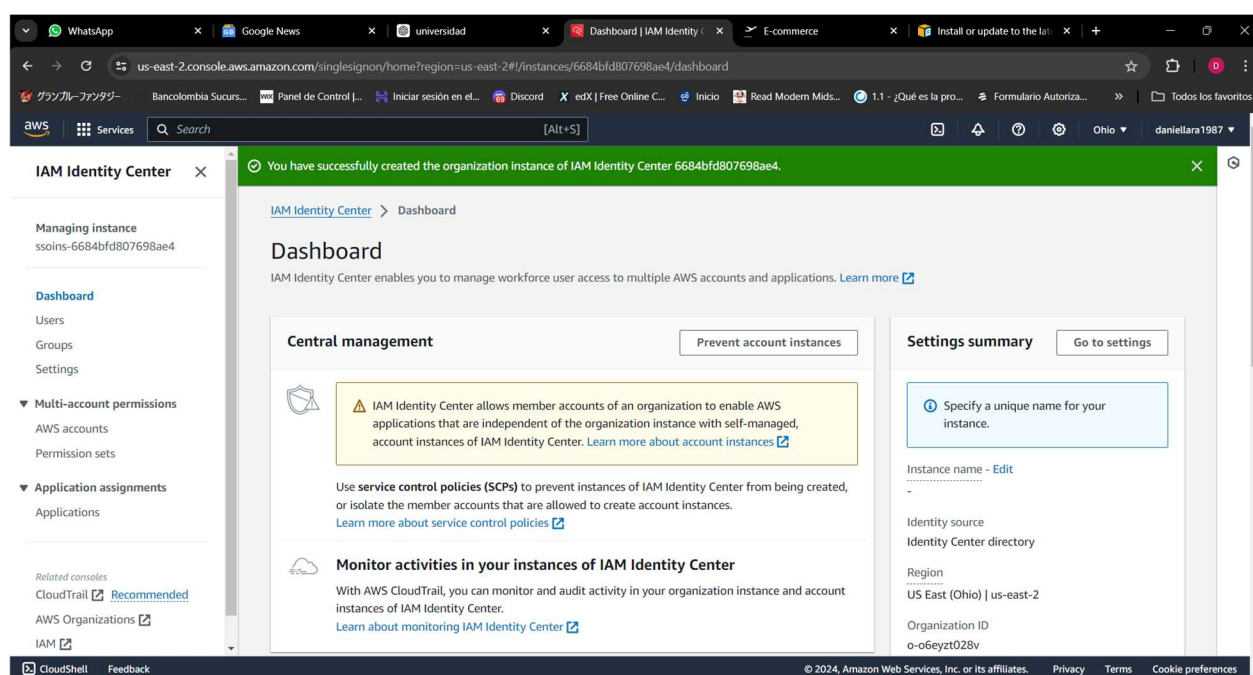


Figura 9. Creación de organización en IAM

Fuente: Elaboración propia

En este ejercicio se creará un usuario en el IAM para poder usarlo en la consola y acceder a los servicios de S3 a través de AWS CLI

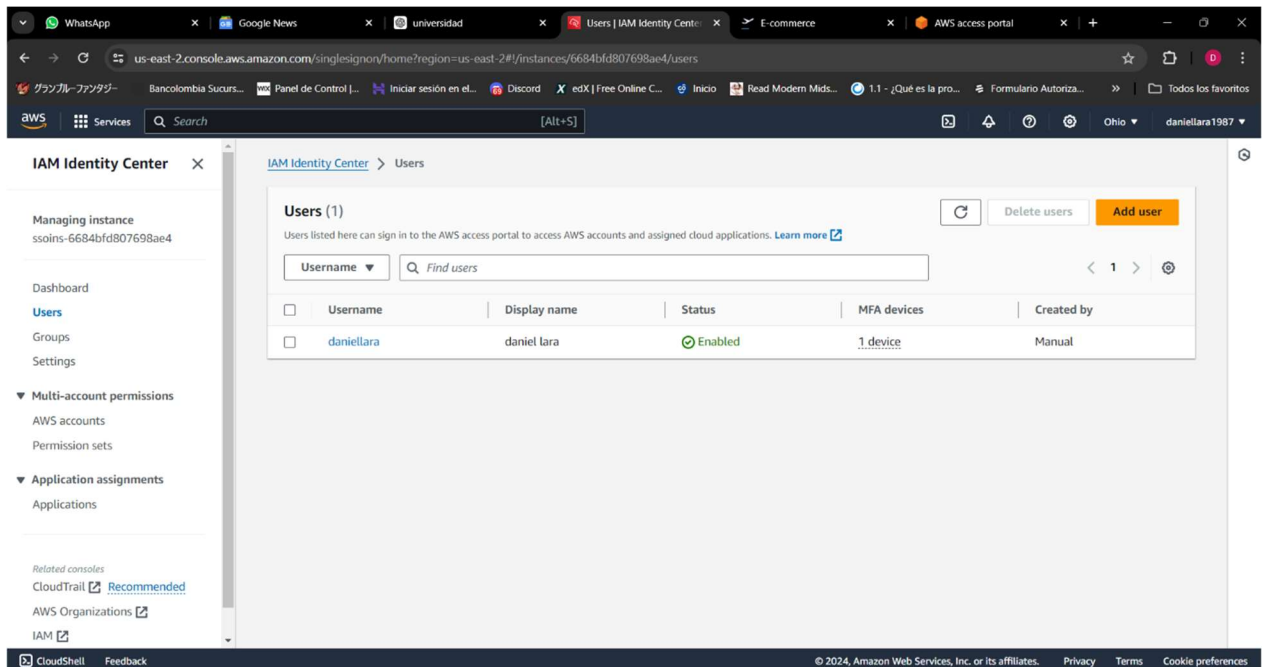
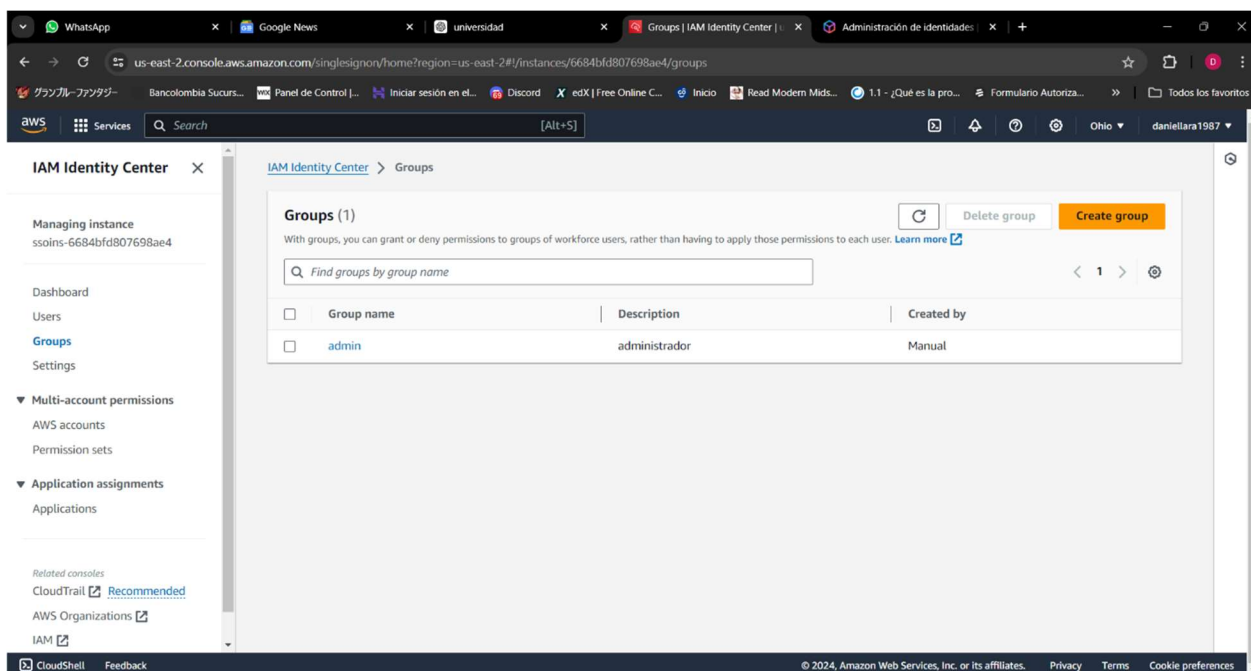


Figura 10. Creación de usuario en IAM

Fuente: Elaboración propia

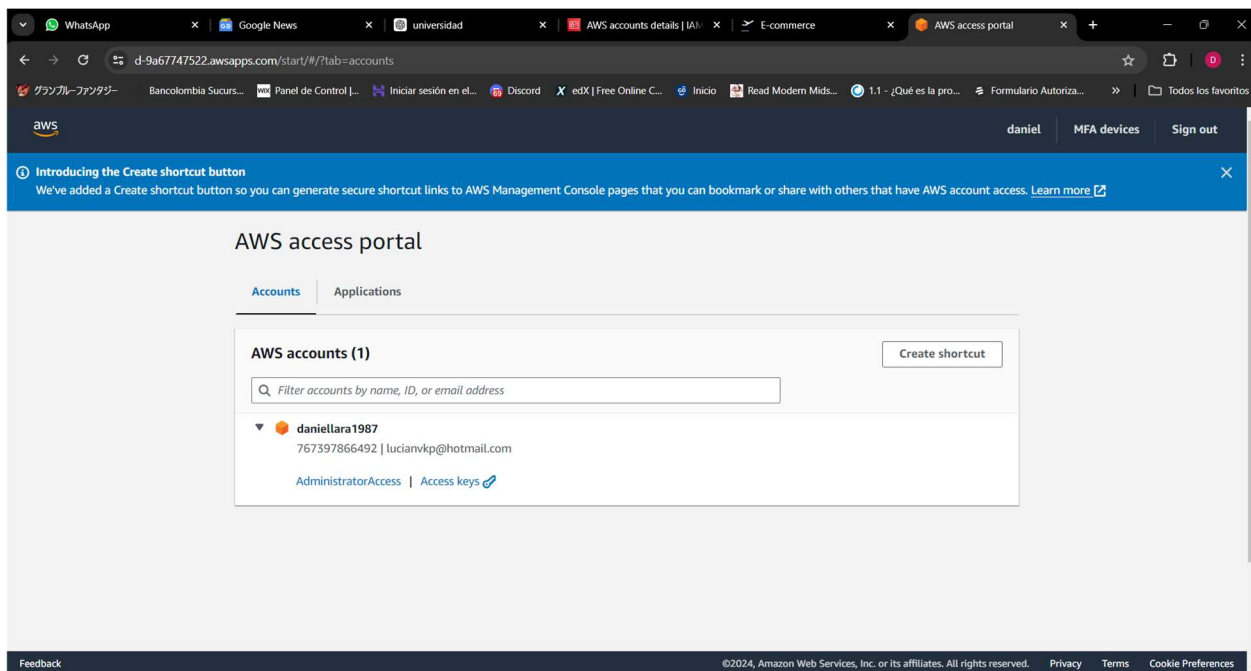
Para que este usuario pueda acceder a los servicios de AWS y la instancia ya creada, se le asignará a un grupo al cual se le han asignado permisos de administrador



*Figura 11. creación de grupo en IAM*

Fuente: Elaboración propia

Después de configurar el usuario nuevo, este recibirá un correo electrónico con el acceso a la plataforma, mediante el cual podrá asignar una contraseña, activar la verificación con autenticador para mayor seguridad y finalmente acceder a los recursos que el administrador le ha asignado



*Figura 12. Portal de acceso de usuario*

Fuente: Elaboración propia

A través de este portal, se obtienen las llaves de acceso que se usaran para instalar el CLI, el cual se debe instalar de manera local en la consola de WSL usando el comando “sudo ./aws/install” después de haberlo descargado y descomprimido.

```

root@ip-10-20-18-31/home/e x + v
inflating: aws/dist/docutils/writers/latex2e/titlingpage.tex
inflating: aws/dist/docutils/writers/latex2e/xelatex.tex
inflating: aws/dist/docutils/writers/odf_odt/styles.odt
creating: aws/dist/docutils/parsers/rst/
creating: aws/dist/docutils/parsers/rst/include/
inflating: aws/dist/docutils/parsers/rst/include/isogrkl.txt
inflating: aws/dist/docutils/parsers/rst/include/isopub.txt
inflating: aws/dist/docutils/parsers/rst/include/isoamsb.txt
inflating: aws/dist/docutils/parsers/rst/include/isocyr1.txt
inflating: aws/dist/docutils/parsers/rst/include/isogrkl2.txt
inflating: aws/dist/docutils/parsers/rst/include/isoamsn.txt
inflating: aws/dist/docutils/parsers/rst/include/mmlalias.txt
inflating: aws/dist/docutils/parsers/rst/include/isogrkl4-wide.txt
inflating: aws/dist/docutils/parsers/rst/include/README.txt
inflating: aws/dist/docutils/parsers/rst/include/mmlextra-wide.txt
inflating: aws/dist/docutils/parsers/rst/include/isoamsc.txt
inflating: aws/dist/docutils/parsers/rst/include/isodia.txt
inflating: aws/dist/docutils/parsers/rst/include/isocyr2.txt
inflating: aws/dist/docutils/parsers/rst/include/isoamsr.txt
inflating: aws/dist/docutils/parsers/rst/include/mmlextra.txt
inflating: aws/dist/docutils/parsers/rst/include/isomopf.txt
inflating: aws/dist/docutils/parsers/rst/include/isolat1.txt
inflating: aws/dist/docutils/parsers/rst/include/xhtml1-special.txt
inflating: aws/dist/docutils/parsers/rst/include/isotech.txt
inflating: aws/dist/docutils/parsers/rst/include/isolat2.txt
inflating: aws/dist/docutils/parsers/rst/include/isomopf-wide.txt
inflating: aws/dist/docutils/parsers/rst/include/isogrkl3.txt
inflating: aws/dist/docutils/parsers/rst/include/isoams0.txt
inflating: aws/dist/docutils/parsers/rst/include/isonum.txt
inflating: aws/dist/docutils/parsers/rst/include/isoamsr.txt
inflating: aws/dist/docutils/parsers/rst/include/xhtml1-symbol.txt
inflating: aws/dist/docutils/parsers/rst/include/isoamsa.txt
inflating: aws/dist/docutils/parsers/rst/include/s5defs.txt
inflating: aws/dist/docutils/parsers/rst/include/xhtml1-lat1.txt
inflating: aws/dist/docutils/parsers/rst/include/isomfkl.txt
inflating: aws/dist/docutils/parsers/rst/include/isoamsr-wide.txt
inflating: aws/dist/docutils/parsers/rst/include/isomfkl-wide.txt
inflating: aws/dist/docutils/parsers/rst/include/isobox.txt
inflating: aws/dist/docutils/parsers/rst/include/isogrkl4.txt
You can now run: /usr/local/bin/aws --version
[root@ip-10-20-18-31 ec2-user]#

```

*Figura 13. Instalación de AWS CLI*

Fuente: Elaboración propia

La Interfaz de la línea de comandos de AWS (AWS CLI) es una herramienta unificada para administrar los servicios de AWS. Solo tendrá que descargar y configurar una única herramienta para poder controlar varios servicios de AWS desde la línea de comandos y automatizarlos mediante scripts (Amazon, 2024).

Luego de instalado, el CLI debe ser configurado para que responda a las credenciales de acceso del usuario que creamos, esto lo hacemos con el comando “aws configure”, mediante el cual debemos agregar las líneas que corresponden a la llave de acceso y la llave secreta, las cuales obtenemos en el portal de acceso del usuario. Posteriormente se configura también la zona en la que está creado el usuario y el formato de respuesta.

```

root@ip-10-20-18-31/home/e x + v
inflating: aws/dist/docutils/parsers/rst/include/isoamsb.txt
inflating: aws/dist/docutils/parsers/rst/include/isoeynl.txt
inflating: aws/dist/docutils/parsers/rst/include/isogrK2.txt
inflating: aws/dist/docutils/parsers/rst/include/isoamsn.txt
inflating: aws/dist/docutils/parsers/rst/include/mmlalias.txt
inflating: aws/dist/docutils/parsers/rst/include/isogrK4-wide.txt
inflating: aws/dist/docutils/parsers/rst/include/README.txt
inflating: aws/dist/docutils/parsers/rst/include/mmlextra-wide.txt
inflating: aws/dist/docutils/parsers/rst/include/isoamscl.txt
inflating: aws/dist/docutils/parsers/rst/include/isodia.txt
inflating: aws/dist/docutils/parsers/rst/include/isocyr2.txt
inflating: aws/dist/docutils/parsers/rst/include/isoamsr.txt
inflating: aws/dist/docutils/parsers/rst/include/mmlextra.txt
inflating: aws/dist/docutils/parsers/rst/include/isomopf.txt
inflating: aws/dist/docutils/parsers/rst/include/isolat1.txt
inflating: aws/dist/docutils/parsers/rst/include/xhtml1-special.txt
inflating: aws/dist/docutils/parsers/rst/include/isotech.txt
inflating: aws/dist/docutils/parsers/rst/include/isolat2.txt
inflating: aws/dist/docutils/parsers/rst/include/isomopf-wide.txt
inflating: aws/dist/docutils/parsers/rst/include/isogrK3.txt
inflating: aws/dist/docutils/parsers/rst/include/isoams0.txt
inflating: aws/dist/docutils/parsers/rst/include/isonum.txt
inflating: aws/dist/docutils/parsers/rst/include/isoamscl.txt
inflating: aws/dist/docutils/parsers/rst/include/xhtml1-symbol.txt
inflating: aws/dist/docutils/parsers/rst/include/isoamsa.txt
inflating: aws/dist/docutils/parsers/rst/include/s5defs.txt
inflating: aws/dist/docutils/parsers/rst/include/xhtml1-lat1.txt
inflating: aws/dist/docutils/parsers/rst/include/isoamfrk.txt
inflating: aws/dist/docutils/parsers/rst/include/isoamscl-wide.txt
inflating: aws/dist/docutils/parsers/rst/include/isoamfrk-wide.txt
inflating: aws/dist/docutils/parsers/rst/include/isobox.txt
inflating: aws/dist/docutils/parsers/rst/include/isogrK4.txt
root@INFORMATICA-PC:/home/daniel# sudo ./aws/install
You can now run: /usr/local/bin/aws --version
root@INFORMATICA-PC:/home/daniel# aws s3 cp "/mnt/d/Personal/ejemplo.txt" s3://bucketseminario/
upload failed: ./././mnt/d/Personal/ejemplo.txt to s3://bucketseminario/ejemplo.txt Unable to locate credentials
root@INFORMATICA-PC:/home/daniel# aws configure
AWS Access Key ID [None]: ASIA3FLD2FP6L47VKGQS
AWS Secret Access Key [None]: FJSxPEC7r93GshVg9CmHdg3sovhuUJufGjTnJ7SA
Default region name [None]: us-east-2
Default output format [None]: json

```

Figura 14. Configuración de AWS CLI

Fuente: Elaboración propia

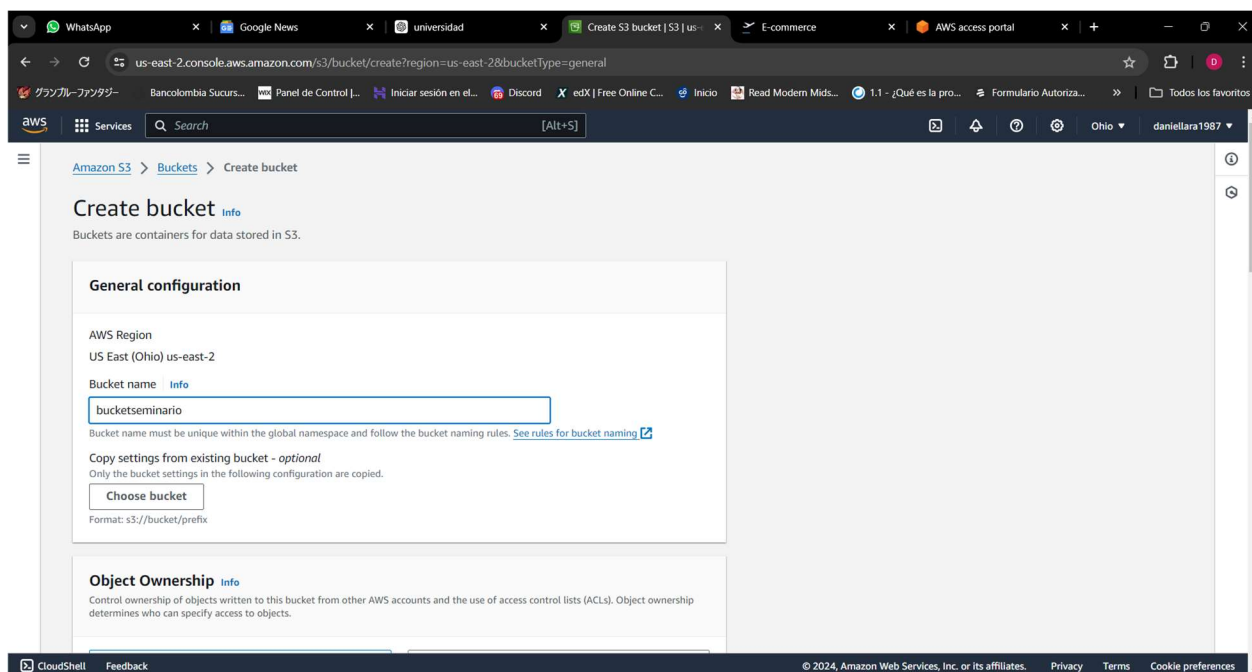
#### 4. Amazon S3(Simple Storage Service) y Amazon Relational Database Service (RDS)

El servicio de S3 nos permite guardar archivos en la nube de Amazon y acceder a ellos mediante la plataforma de AWS o la consola de SSH.

Amazon Simple Storage Service (Amazon S3) es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento líderes en el sector. Clientes de todos los tamaños y sectores pueden almacenar y proteger cualquier cantidad de datos para prácticamente cualquier caso de uso, como los lagos de datos, las aplicaciones nativas en la nube y las aplicaciones móviles. Gracias a las clases de almacenamiento rentables y a las características de administración fáciles de usar, es posible optimizar los costos, organizar

los datos y configurar controles de acceso detallados para cumplir con requisitos empresariales, organizacionales y de conformidad específicos (Amazon, 2024).

Para continua con el ejercicio, se procederá a crear un “bucket” de S3 al cual subiremos un archivo de texto como ejemplo.



*Figura 15. creación de bucket en S3*

Fuente: Elaboración propia

Posteriormente se deben crear roles en la consola de IAM para que la instancia de EC2 y los usuarios puedan conectarse a estos nuevos recursos

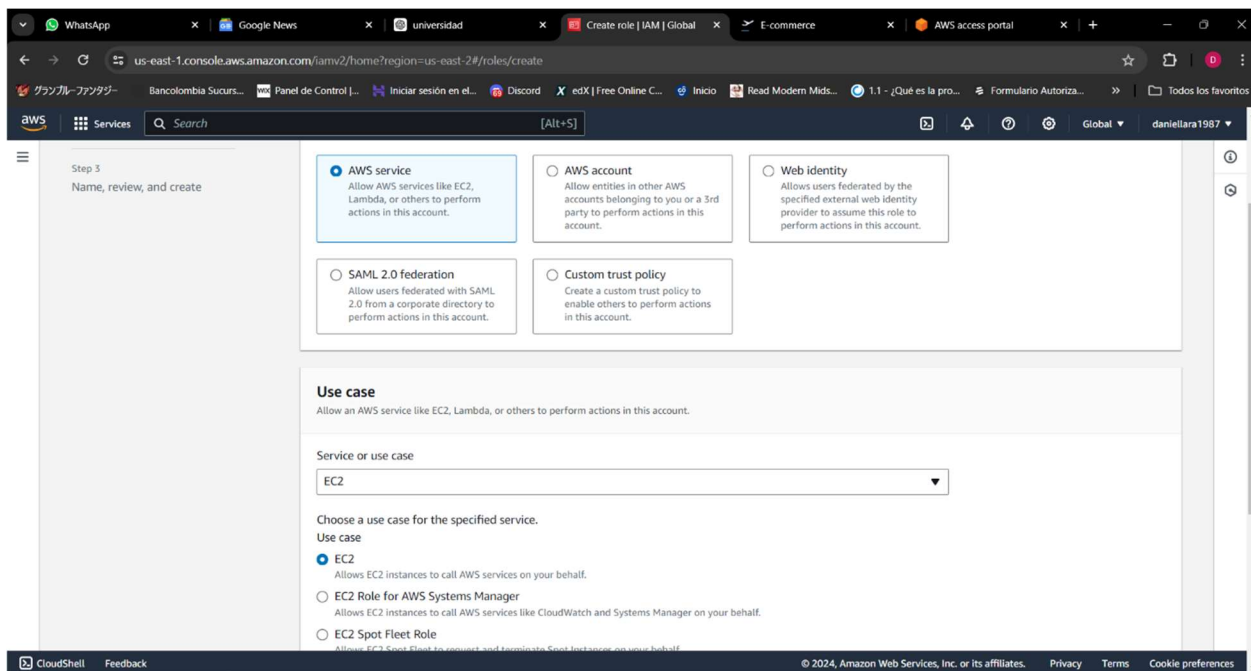
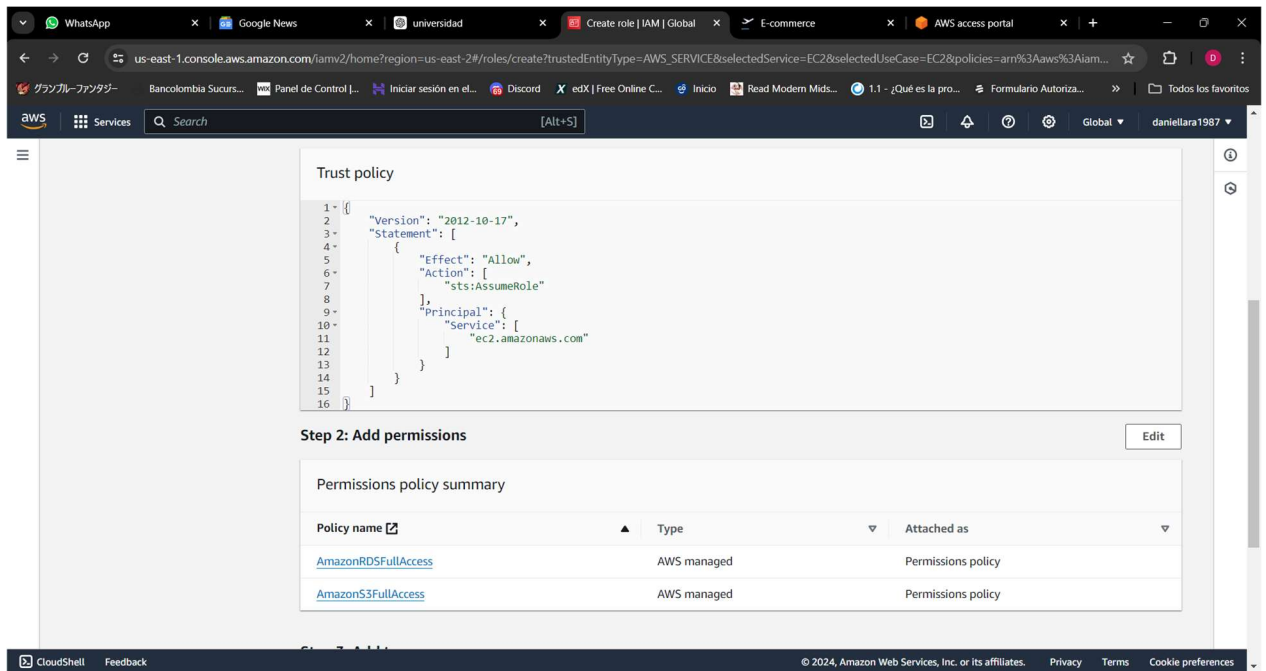


Figura 16. creación de Roles en IAM

Fuente: Elaboración propia

Seleccionando inicialmente el tipo de servicio y el caso de uso, para luego agregar las políticas específicas que activaran los servicios de S3 y RDS. Esta política debe ser cargada a la instancia que se creó anteriormente para que tome efecto.



The screenshot displays the AWS IAM console interface. At the top, the browser address bar shows the URL: `us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-2#/roles/create?trustedEntityType=AWS_SERVICE&selectedService=EC2&selectedUseCase=EC2&policies=arn%3Aaws%3Aiam...`. The main content area is titled "Trust policy" and shows a JSON configuration for a trust policy. Below this, there is a section titled "Step 2: Add permissions" with an "Edit" button. Underneath, a "Permissions policy summary" table lists two policies: "AmazonRDSFullAccess" and "AmazonS3FullAccess", both of which are "AWS managed" and attached as "Permissions policy".

```
1- {
2-   "Version": "2012-10-17",
3-   "Statement": [
4-     {
5-       "Effect": "Allow",
6-       "Action": [
7-         "sts:AssumeRole"
8-       ],
9-       "Principal": {
10-        "Service": [
11-          "ec2.amazonaws.com"
12-        ]
13-      }
14-    }
15-  ]
16- }
```

Policy name	Type	Attached as
<a href="#">AmazonRDSFullAccess</a>	AWS managed	Permissions policy
<a href="#">AmazonS3FullAccess</a>	AWS managed	Permissions policy

Figura 17. Permisos de rol en IAM

Fuente: Elaboración propia

A este espacio de almacenamiento o “bucket”, también se le deben asignar unas políticas para permitir el acceso desde la consola de WSL, la cual se puede crear en el apartado de seguridad del bucket

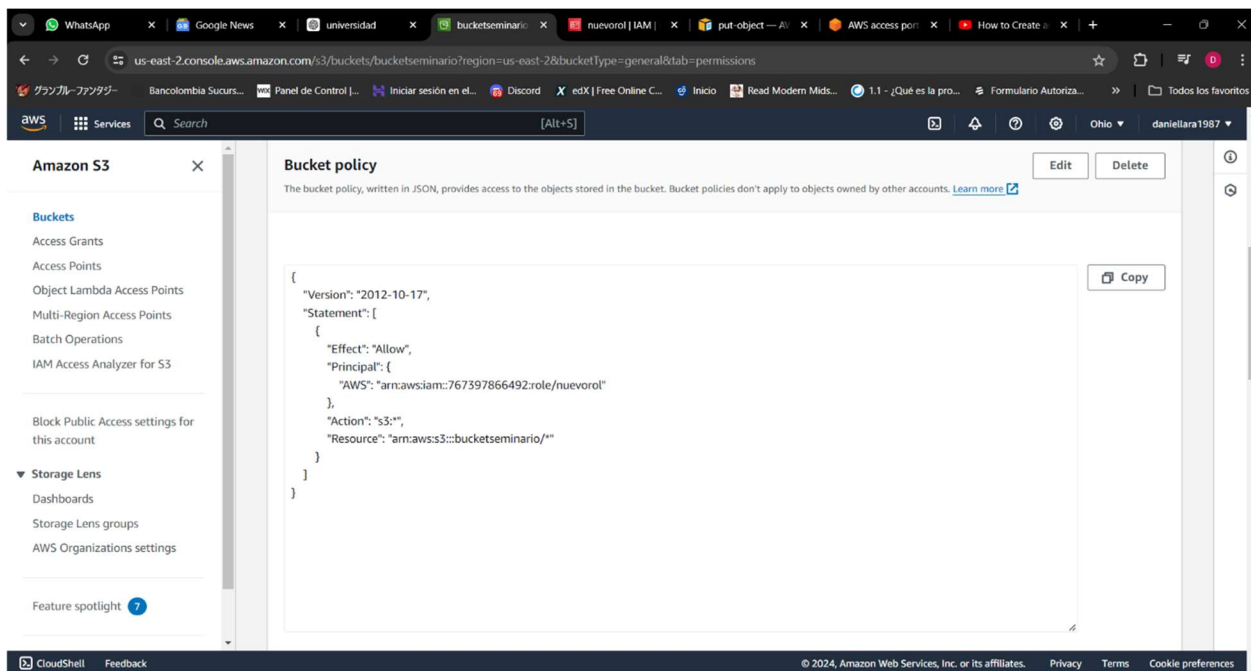


Figura 18. política de bucket

Fuente: Elaboración propia

En esta política se configura que el rol de acceso definido pueda acceder al recurso del S3 creado.

El siguiente paso es asegurarnos que WSL este correctamente configurado para que encuentre el archivo del equipo local que se desea subir a S3, lo cual se puede hacer usando el comando “sudo nano /etc/wsl.conf”. Esto permitirá que WSL pueda acceder a las rutas de carpeta de Windows del equipo local.



```
root@ip-10-20-18-31:/home/e
GNU nano 5.8 /etc/wsl.conf Modified
[automount]
enabled = true
root = /mnt/
options = "metadata"
```

*Figura 19. configuración WSL*

Fuente: Elaboración propia

Para continuar con el ejercicio propuesto, el archivo de texto que se subirá a S3 se encuentra en la ruta `D/personal/ejemplo.txt`, la cual se llamará, junto con el nombre del bucket creado, en esta línea de comando `“aws s3api put-object --bucket bucketseminario --key ejemplo.txt --body "/mnt/d/Personal/ejemplo.txt"”`, la cual subirá el archivo al espacio de almacenamiento deseado

```

root@INFORMATICA-PC: /hor x + v
An error occurred (InvalidAccessKeyId) when calling the PutObject operation: The AWS Access Key Id you provided does not exist in our records.
root@INFORMATICA-PC:/home/daniel# aws s3api put-object --bucket bucketseminario --key ejemplo.txt --body "/mnt/d/PersonaL/ejemplo.txt"

An error occurred (InvalidAccessKeyId) when calling the PutObject operation: The AWS Access Key Id you provided does not exist in our records.
root@INFORMATICA-PC:/home/daniel# aws s3api put-object --bucket bucketseminario --key ejemplo.txt --body "/mnt/d/PersonaL/ejemplo.txt"

An error occurred (InvalidAccessKeyId) when calling the PutObject operation: The AWS Access Key Id you provided does not exist in our records.
root@INFORMATICA-PC:/home/daniel# export AWS_ACCESS_KEY_ID="ASIA3FLD2FP6MFOZNI3K"
export AWS_SECRET_ACCESS_KEY="Y+u0Dv8Ebue3RZvx9fTFpb3/LC8n7npgFLX+rd02"
export AWS_SESSION_TOKEN="IQoJb3JpZ2luX2VjEAAoCXVzLWVhc3QtMiJlMEYCIQD4ADfh8pvqC2rqCSSZVwHhEocxKbBayqdSYtFJhi/iIgIhAKwQmjfZpJJ6Dutn07kTvgWS8JqCkC+eWnHgB3gxYEpKvsCCJP//////////wEQABoMnzY3Mzk30DY2NDkyIgzd5gTmLXhwzh8buN0qzLwB2MdpqdxoGFZVwy0T+hDJmMdjB8Z2MSe++eabVjdJ+ixusEWtq9+JudxPZqqqaSjSPz6pd+m0sSOJ082ggFeUJDy5R06QpWZjK/GZUYHaXDdy48YWQNEGsuBrwfQgQDYXImPWEBYL+Qls1ce0dyfIquoA+dz7nqAD+SaS1HvbEtTX7qo4IZP/dq1UQm43Nh26sWkZKeN343//8U6D00qxrMJEAQv7dbo/Y0yoGReAvn7ks8Vh2QUaMnDkRBQOnaGyU0aCw9IwNlp6ekL2xwQxkZncbZ7VpmsNTvc5eEZxz1Jbza+M0wvHKmbScsGnKq/v3sZTVMejSKNTysHggGdJ+Yyyf0dZT+rxUW/WvKXI1HqptzILYmouOydIwLHPHTU0gBSbaDLNq0Co+mnp4k6gpy3cVPMpKqNi0exdcWsXNypaB3fw+8R5fr5CjCMgviyBjqmAZv1/TNx9ryKg71TJ+2//vZfG1I60y0G70L6MXF1/Lfb5zoaELVTt8JMogEzPKwicQxkYDJMxn6H/6X9KBmR55rGdwfh0LMDC8exAPXMa6Zxf80M5P7fhLDyqygSnu7AacdF009d4UcLhng9d+jqIqwtYejq2Qdb2k8qAZwsIwZntYxmZggue61zlsUNgcJZ1ZF1MjIHEp25quLY0+sfaUc0CGTzk0="
root@INFORMATICA-PC:/home/daniel# aws s3api put-object --bucket bucketseminario --key ejemplo.txt --body "/mnt/d/PersonaL/ejemplo.txt"
{
  "ETag": "\"3a39d3b7c5046d3736fcdc51058ecacf\"",
  "ServerSideEncryption": "AES256"
}
root@INFORMATICA-PC:/home/daniel#

```

Figura 20. Subida de archivo a S3

Fuente: Elaboración propia

A continuación, se puede apreciar el archivo ya subido en la plataforma S3 de AWS, el cual ya se encuentra en el bucket creado, el cual tiene por nombre bucketseminario.

Cabe señalar que también es posible subir archivos a través de la plataforma de AWS mediante el botón de Upload, como se puede apreciar en la imagen

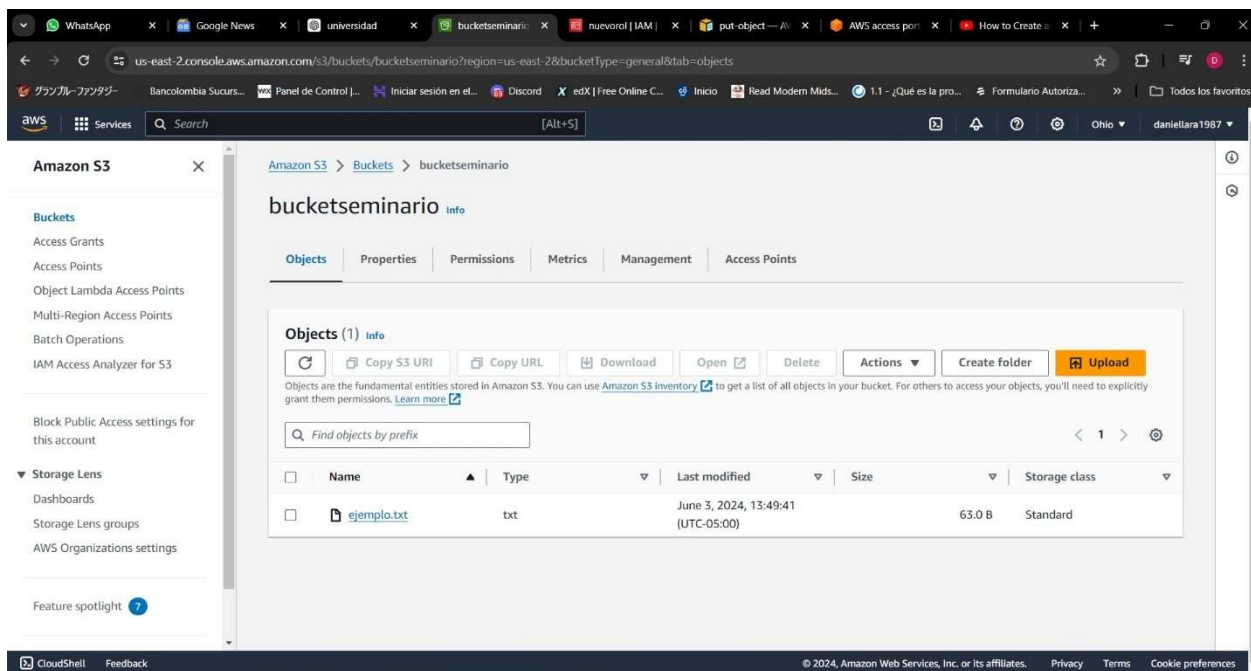


Figura 21. Archivo en bucket

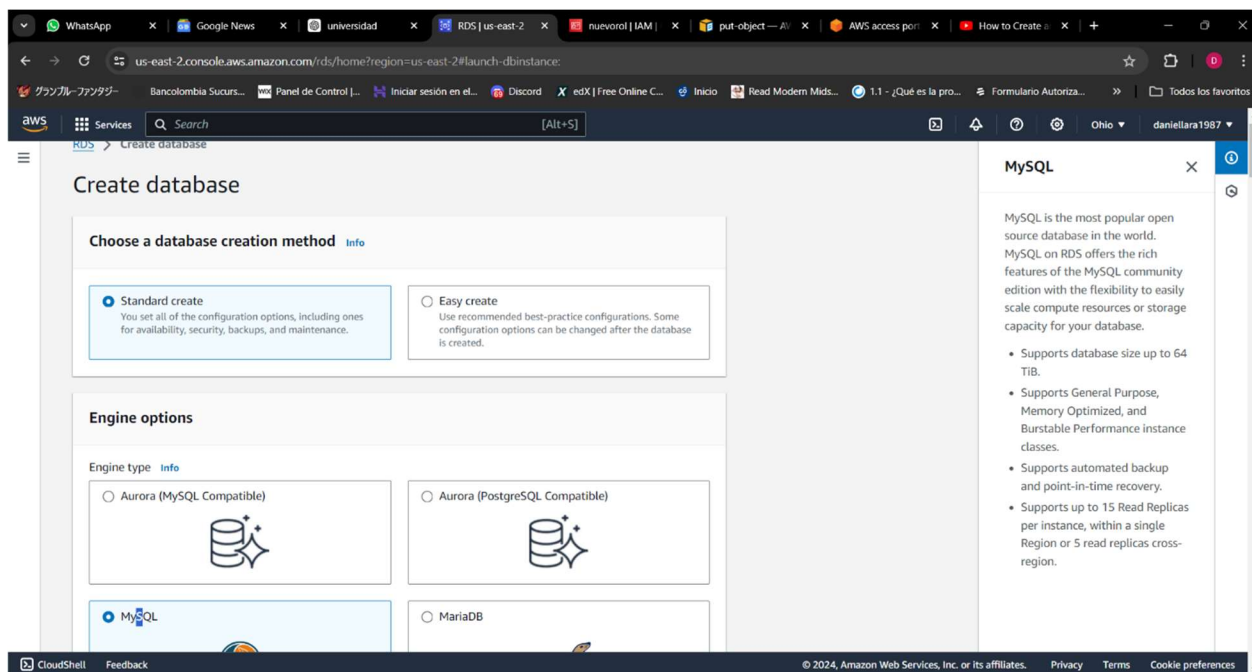
Fuente: Elaboración propia

El siguiente servicio de AWS a explorar es el de RDS, el cual nos permitirá crear bases de datos en la nube de Amazon y conectarnos a ellas mediante la instancia de EC2 ya creada.

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. Los clientes pueden optimizar el rendimiento con características, como Multi-AZ con dos modos de espera legibles, escrituras y

lecturas optimizadas e instancias basadas en AWS Graviton3, y elegir entre varias opciones de precios para administrar los costos de manera eficaz (Amazon, 2024).

Para lograr esta conexión lo primero que se debe hacer es crear la base de datos en el servicio de RDS de la plataforma de AWS, la cual en esta ocasión se montara en MySQL



*Figura 22. creación de base de datos MySQL*

Fuente: Elaboración propia

Durante la creación se puede configurar de manera fácil que la base de datos pueda conectarse con una instancia de EC2 ya existente, de esta manera la base de datos tomara la VCP que la instancia tiene configurada y permitirá la interacción mediante los protocolos de red correspondiente.

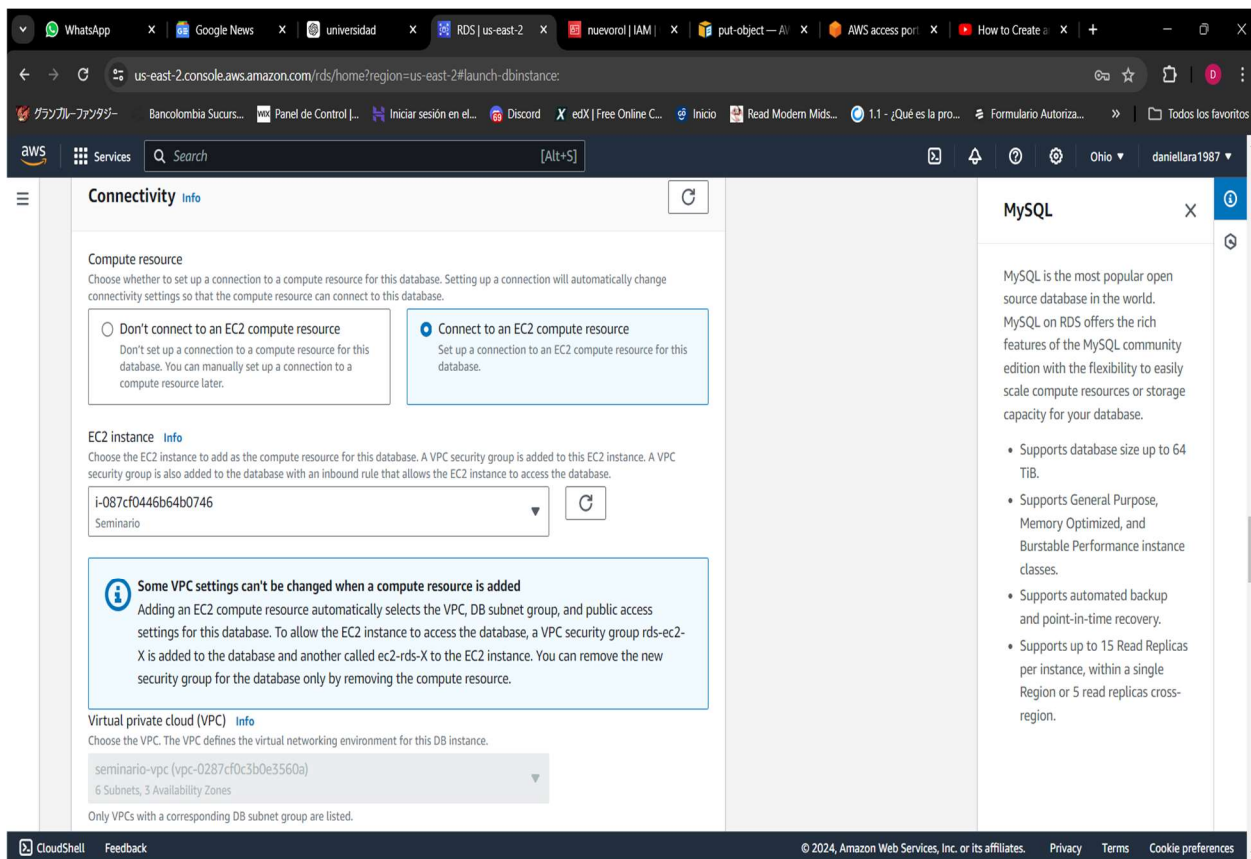


Figura 23. Configuración conectividad base de datos y EC2

Fuente: Elaboración propia

También se le pueden asignar las mismas reglas de seguridad que tiene la instancia para que los puertos necesarios funcionen correctamente, en este caso, es necesario que se habilite el puerto TCP 3306, el cual es el puerto de comunicación por defecto de MySQL.

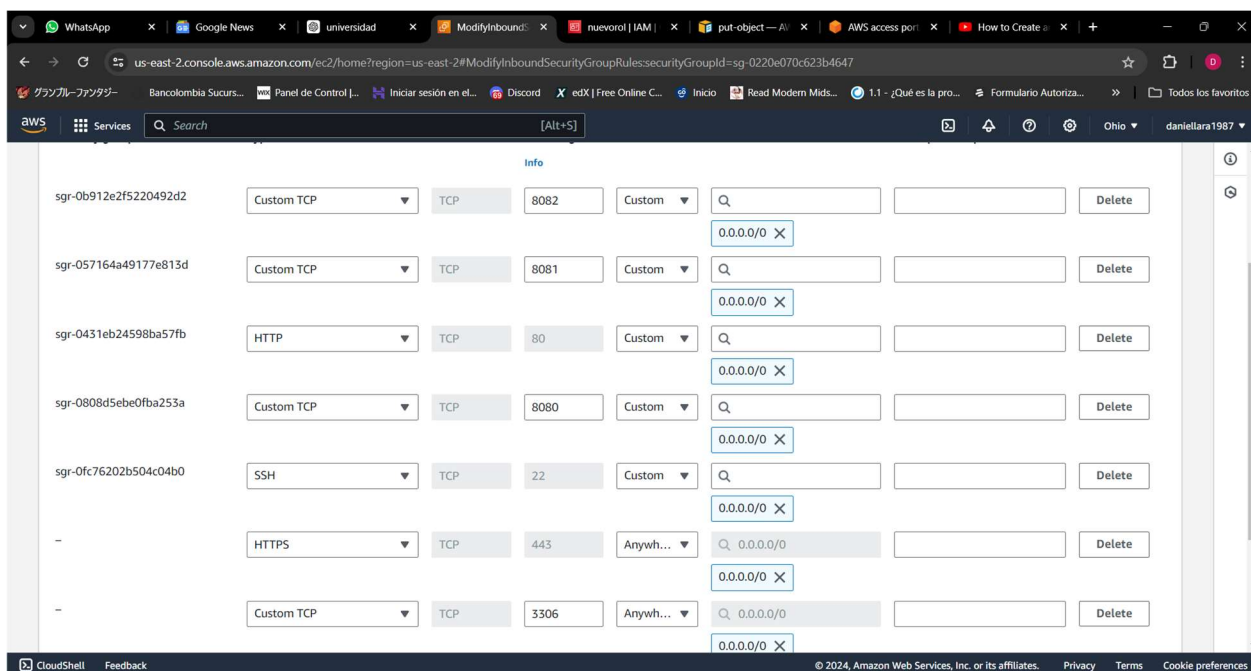


Figura 24. Configuración puertos

Fuente: Elaboración propia

El siguiente paso es instalar el cliente de MySQL en la instancia de EC2 mediante la consola de WSL usando los comandos “sudo yum install -y <https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm>”, “sudo yum install -y mysql-community-client”, “sudo dnf update -y” y “sudo dnf install -y mariadb105-server”. Esto permitirá que, mediante la consola de la instancia, se pueda acceder a las bases de datos de MySQL creadas en RDS.

```

ec2-user@ip-10-20-18-31:~
Last metadata expiration check: 4:13:23 ago on Mon Jun 3 15:01:57 2024.
mysql57-community-release-el7-11.noarch.rpm                               116 kB/s | 25 kB   00:00
Dependencies resolved.
=====
Package                        Architecture      Version          Repository        Size
=====
Installing:
mysql57-community-release      noarch           el7-11          @commandline     25 k
Transaction Summary
=====
Install 1 Package

Total size: 25 k
Installed size: 31 k
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                                1/1
  Installing    : mysql57-community-release-el7-11.noarch 1/1
  Verifying     : mysql57-community-release-el7-11.noarch 1/1
=====
WARNING:
  A newer release of "Amazon Linux" is available.

Available Versions:

```

*Figura 25. Instalación cliente MySQL*

Fuente: Elaboración propia

Finalmente, para tener acceso a la base de datos, necesitamos usar el comando “mysql -h seminario-1.c7ugq0iueu1a.us-east-2.rds.amazonaws.com -P 3306 -u admin -p”, el cual tiene llama el endpoint de la base de datos, el puerto correspondiente y el usuario, para posteriormente solicitar la contraseña. A seguir, se usa el comando “show databases” para mostrar las bases de datos que se encuentran en el servicio de RDS creado y el comando “status” para mostrar sus detalles.

```
root@ip-10-20-18-31:/home/e 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:          50
Current database:
Current user:           admin@10.20.18.31
SSL:                    Not in use
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server:                 MySQL
Server version:         8.0.35 Source distribution
Protocol version:       10
Connection:             seminar10-1.c7ugq@iueula.us-east-2.rds.amazonaws.com via TCP/IP
Server characteraset:   utf8mb4
Db characteraset:       utf8mb4
Client characteraset:   utf8mb3
Conn. characteraset:    utf8mb3
TCP port:               3306
Uptime:                 1 hour 57 min 9 sec

Threads: 3  Questions: 17030  Slow queries: 0  Opens: 353  Flush tables: 3  Open tables: 233  Queries per second avg: 2.422
-----

MySQL [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.001 sec)

MySQL [(none)]>
MySQL [(none)]> |
```

*Figura 26. Acceso a base de datos*

Fuente: Elaboración propia

## 5. Balanceadores de carga y grupos de auto escalado

El último ejercicio propone usar las funcionalidades de balanceadores de carga y grupos de auto escalado para asegurar que las instancias creadas en las que se aloja un servicio web simple, siempre esté disponible y en caso de que alguna de las instancias falle, otra más tome su lugar.

Lo primero que se debe hacer es crear una instancia EC2 de Windows, a la cual nos conectaremos mediante escritorio remoto usando la IP pública de la instancia para realizar las configuraciones necesarias para montar el servicio web.

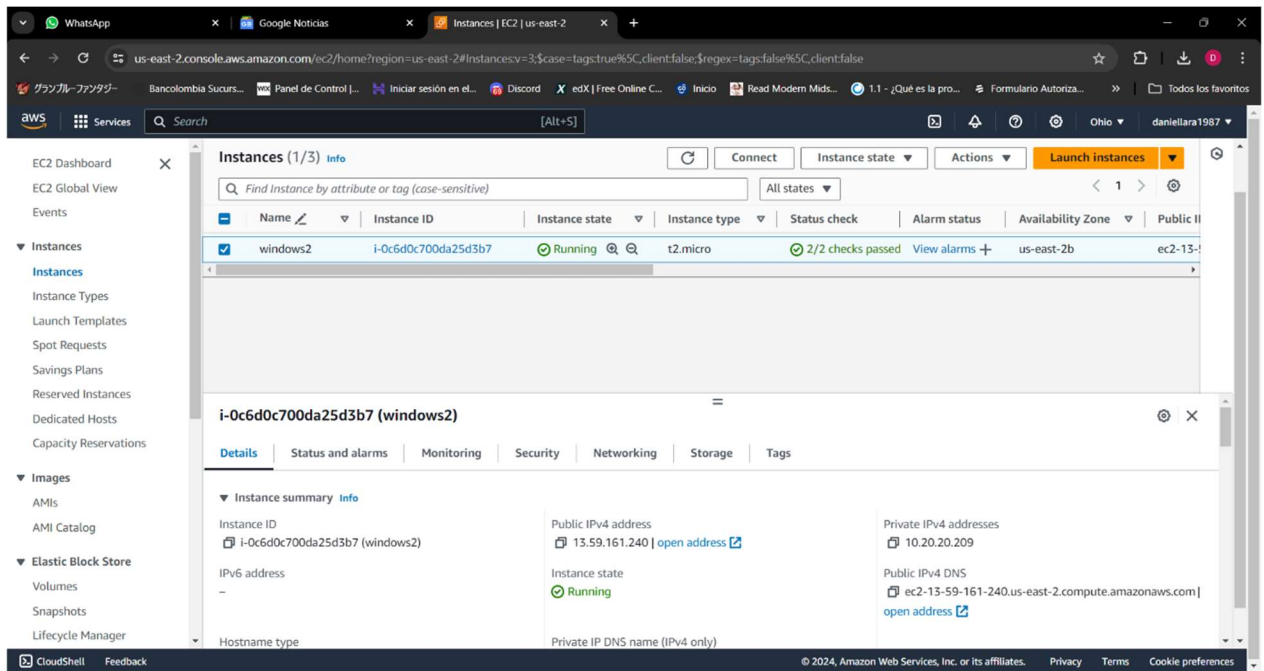
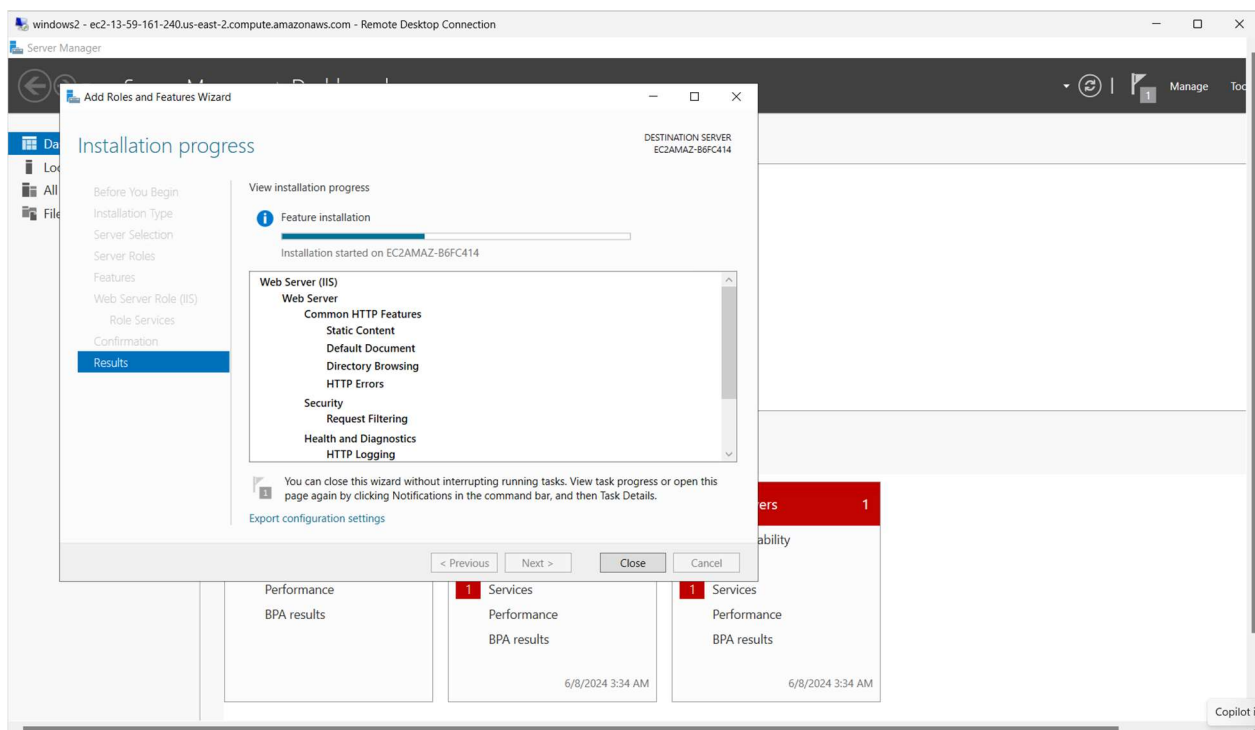


Figura 27. Creación de instancia de Windows

Fuente: Elaboración propia

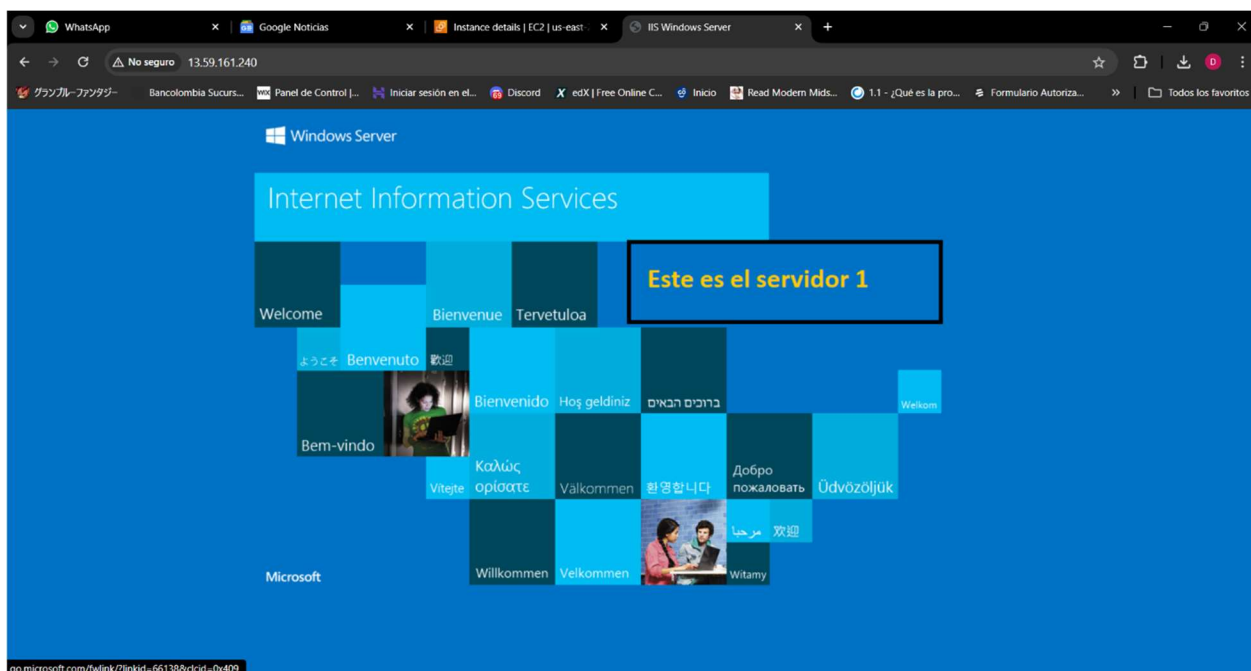
Luego de conectar a la instancia, se procede a abrir el server manager para instalar el servicio de IIS (Internet Information Service), el cual funcionara como servidor web simple para ser accedido desde la IP publica de la instancia.



*Figura 28. Instalación IIS*

*Fuente: Elaboración propia*

Para identificar cada servidor web se ha optado por modificar la pagina de inicio que se muestra al acceder al servicio, agregando un texto que indica si es el server 1 o el server 2 que será creado posteriormente. Esto se hace de manera sencilla al acceder el archivo de la pagina web, que se encuentra en un directorio del IIS y editarlo usando Paint.



*Figura 29. Edición de la página de inicio del IIS*

*Fuente:* Elaboración propia

Como siguiente paso, se tomará esta instancia y se creará un snapshot a partir de ella, para crear fácilmente otras instancias a futuro con la misma configuración, clave de acceso, e incluso con el IIS ya configurado y la página web editada.

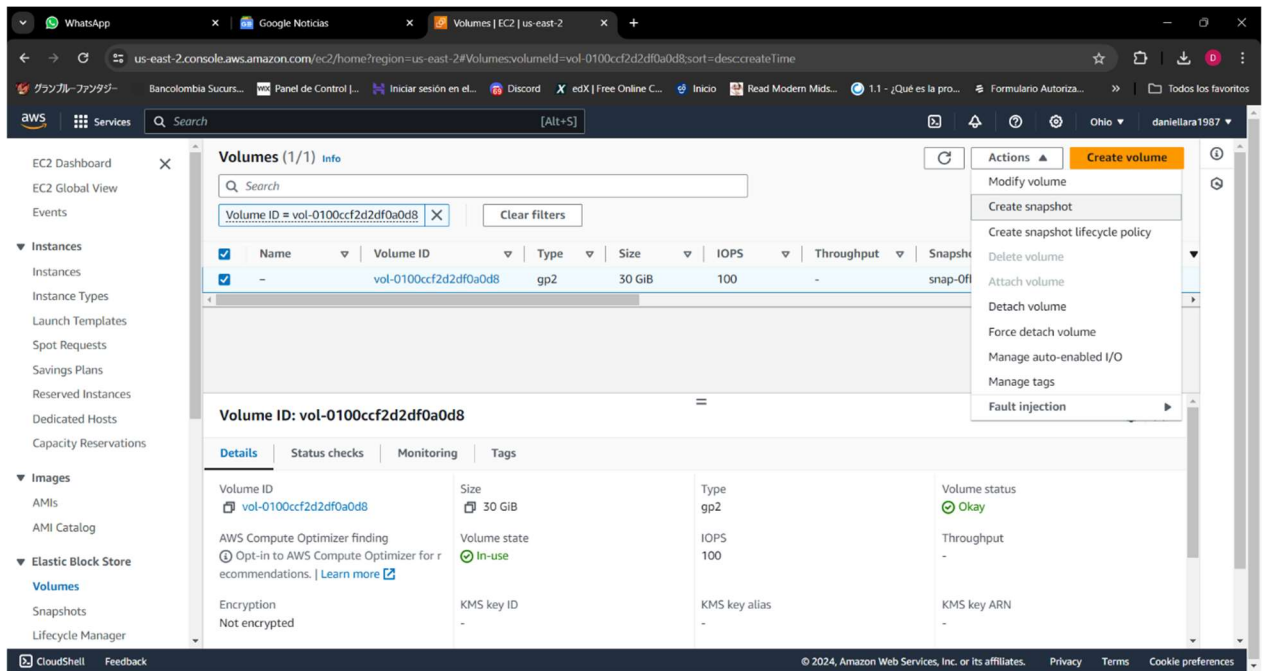
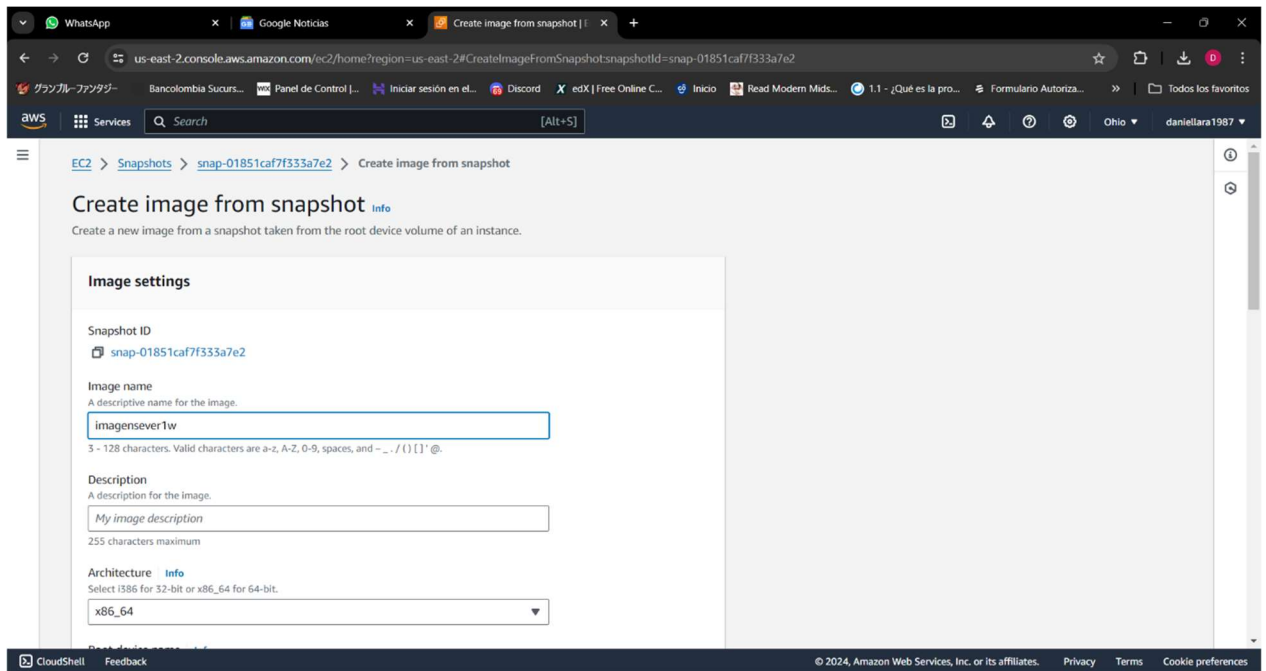


Figura 30. Creación de snapshot

Fuente: Elaboración propia

Este snapshot luego se convertirá en una imagen o AMI (Amazon machine image), que luego podremos usar como plantilla al crear otras instancias con la misma configuración.



*Figura 31. Creación de AMI*

*Fuente:* Elaboración propia

Usando esta AMI que hemos creado, lanzaremos una nueva instancia con la misma configuración. Todo esto se hace para probar la funcionalidad del balanceador de cargas, el cual requiere un mínimo de 2 instancias funcionales a la vez.

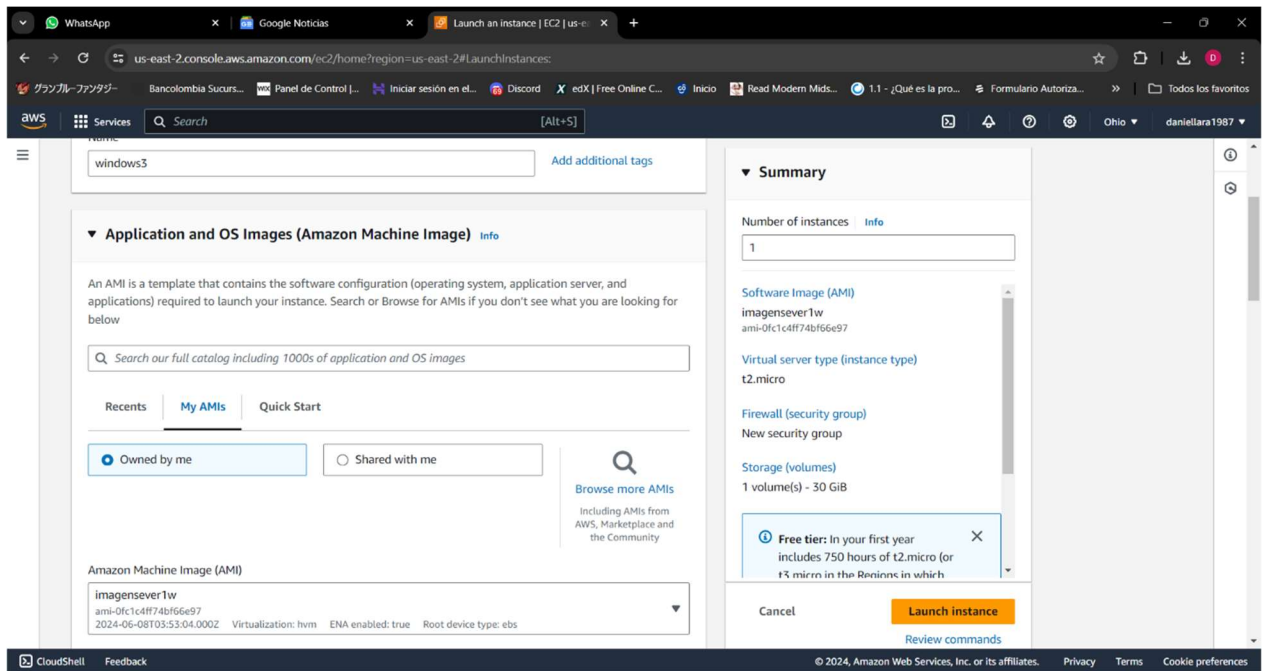


Figura 32. Creación de instancia a partir de AMI

Fuente: Elaboración propia

Lo siguiente es editar la pagina web de inicio para diferenciarla de la que se encuentra en la otra instancia. Por ello se le ha escrito un texto simple que dice server 2 en un color distintivo.

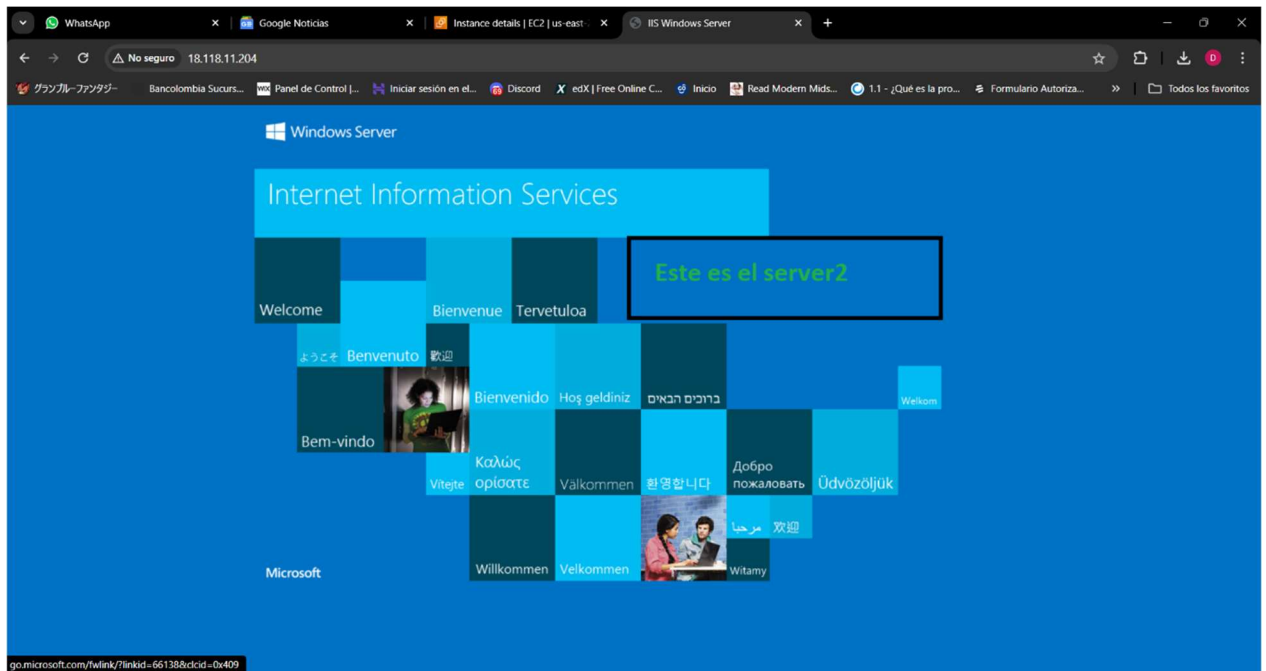
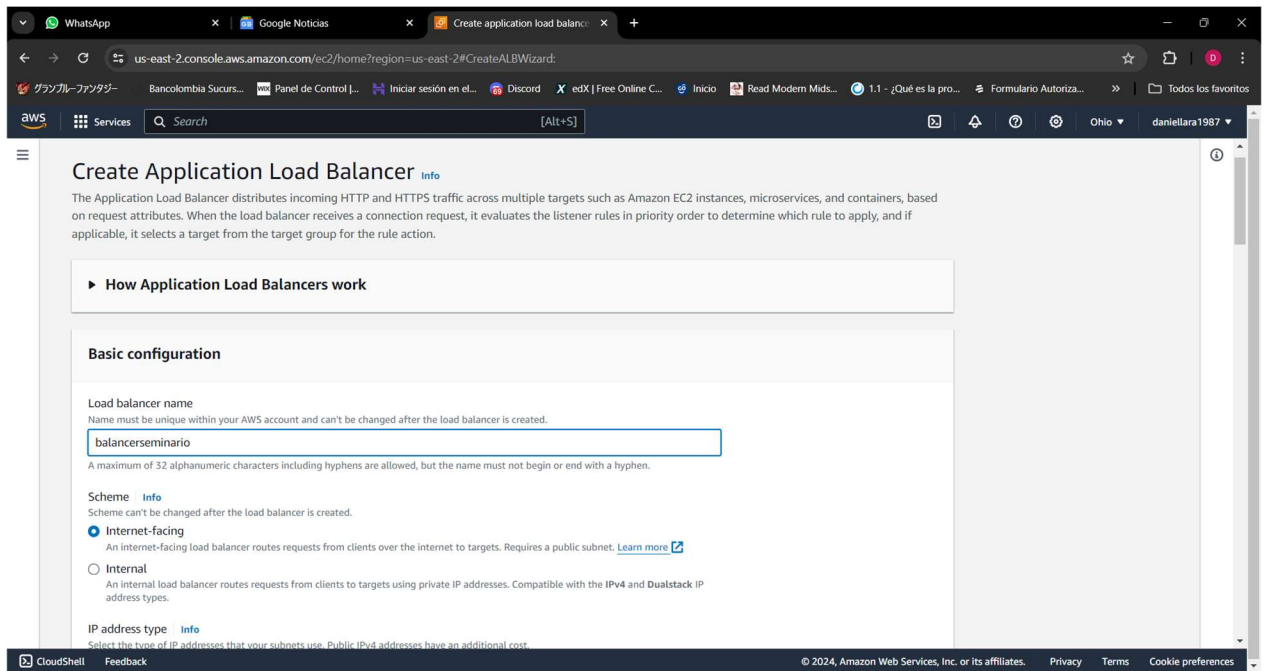


Figura 33. Edición de página de inicio en server 2

Fuente: Elaboración propia

Finalmente se crea el balanceador de cargas, el cual recibirá las peticiones de frente al internet y distribuirá las peticiones entre las 2 instancias asociadas para garantizar que siempre haya una alta disponibilidad.

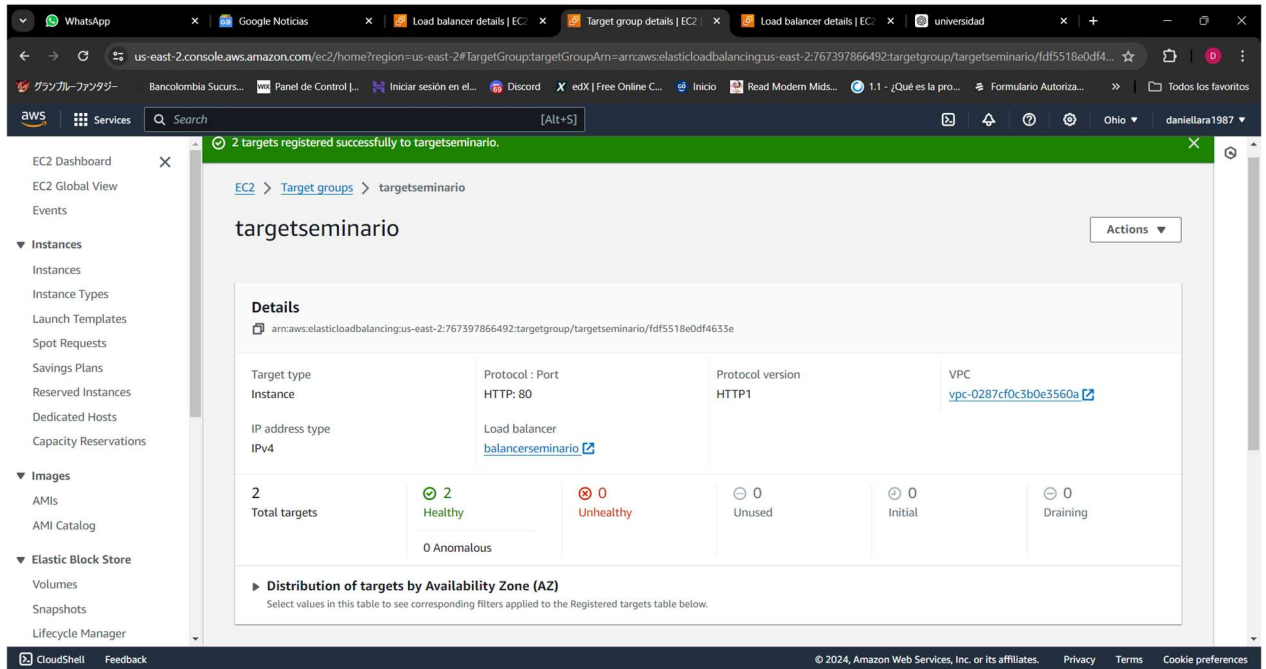
Elastic Load Balancing distribuye automáticamente el tráfico entrante entre varios destinos, por ejemplo, instancias EC2, contenedores y direcciones IP en una o varias zonas de disponibilidad. Monitorea el estado de los destinos registrados y enruta el tráfico solamente a destinos en buen estado. Elastic Load Balancing escala el equilibrador de carga a medida que el tráfico entrante va cambiando con el tiempo. Puede escalarse automáticamente para adaptarse a la mayoría de las cargas de trabajo (Amazon, 2024).



*Figura 34. Creación de balanceador de cargas*

*Fuente: Elaboración propia*

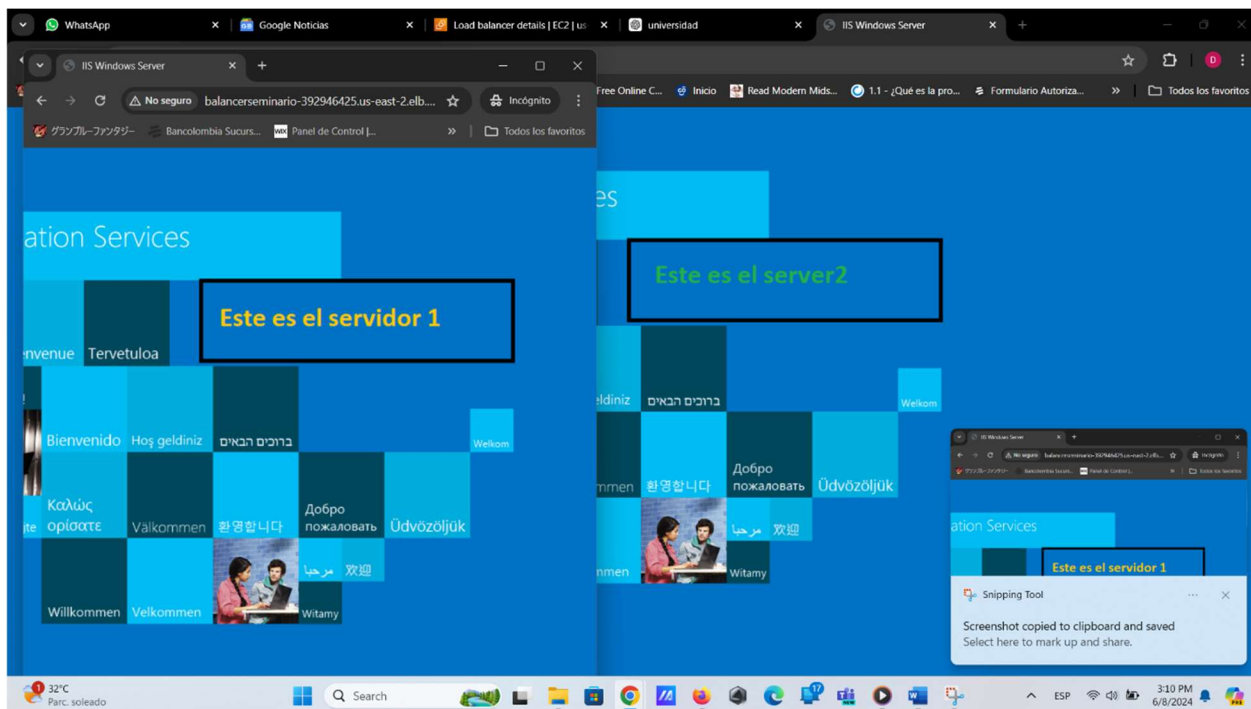
Para que el balanceador funcione correctamente en el ejemplo propuesto de usar servicios web básicos, se debe configurar en modo de balanceador de aplicaciones. También debe crearse un grupo objetivo (target group), el cual contendrá las instancias a las que se asociará el balanceador y las cuales debe monitorear



*Figura 35. Creación de Target Group*

*Fuente:* Elaboración propia

Al terminar de configurar el balanceador de cargas, podemos usar la URL que proporciona para acceder a las paginas de las instancias. Estas peticiones de acceso serán enviadas por el balanceador hacia las respectivas instancias para tratar de mantener el servicio siempre arriba. Usando distintos navegadores o sesiones en incognito es posible demostrar el acceso a ambas páginas web editadas simultáneamente.



*Figura 36. Acceso a servicios web a través de balanceador de carga*

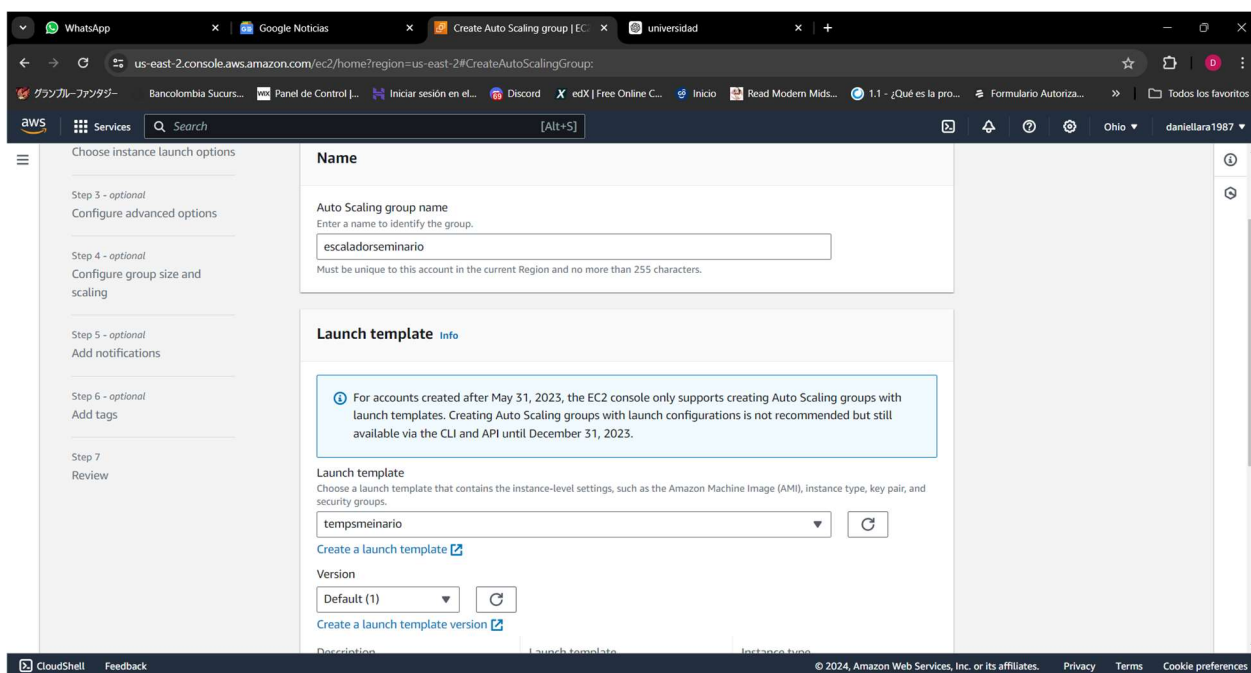
*Fuente:* Elaboración propia

El balanceador de cargas tiene un servicio más que permite la creación automática de instancias a partir de las instancias registradas en él u otros target groups.

Este servicio se conoce como grupo de auto escalado.

Un Auto Scaling group (grupo de escalado automático) contiene una colección de instancias de Amazon EC2 que se tratan como una agrupación lógica a efectos de escalado automático y administración. Un grupo de escalado automático también le permite utilizar características de Amazon EC2 Auto Scaling, como sustituciones de comprobaciones de estado y políticas de escalado. Tanto el mantenimiento del número de instancias en un grupo de Auto Scaling como el escalado automático son las funcionalidades básicas del servicio de Amazon

EC2 Auto Scaling. Un grupo de Auto Scaling comienza lanzando suficientes instancias para satisfacer su capacidad deseada. Mantiene este número de instancias realizando comprobaciones de estado periódicas en las instancias del grupo. El grupo de Auto Scaling sigue manteniendo un número fijo de instancias, aunque una de ellas deje de estar en buen estado. Si una instancia pasa a tener un estado incorrecto, el grupo termina la instancia en mal estado y lanza otra instancia para sustituirla (Amazon, 2024).



*Figura 37. Creación de grupo de auto escalado*

*Fuente:* Elaboración propia

El grupo de auto escalado se puede configurar para que dependiendo si la instancia tiene un consumo de CPU determinado, esto se considere como una razón para lanzar otra instancia para reemplazarla. Esta configuración puede ser en base a uso de CPU, consumo de red (salida y entrada) y definido por el balanceador de cargas. En este ejemplo se opta por usar una

configuración de mínimo de 2 instancias, 3 deseadas y un máximo de 5, con una política de escalamiento basada en uso promedio de CPU de 50% y un agregado en métricas de 60 segundos.

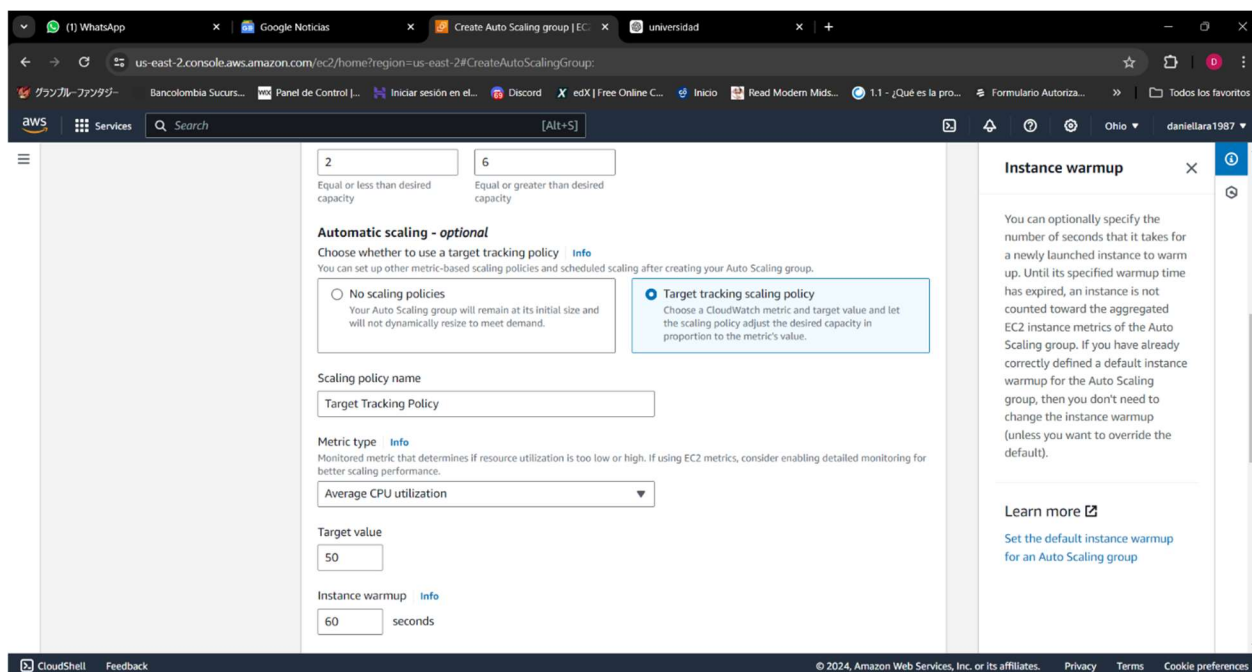


Figura 38. Configuración de grupo de auto escalado

Fuente: Elaboración propia

Inmediatamente después de creada la política de auto escalado, se procede automáticamente a crear las instancias definidas en la configuración, un numero deseado, un mínimo y un máximo.

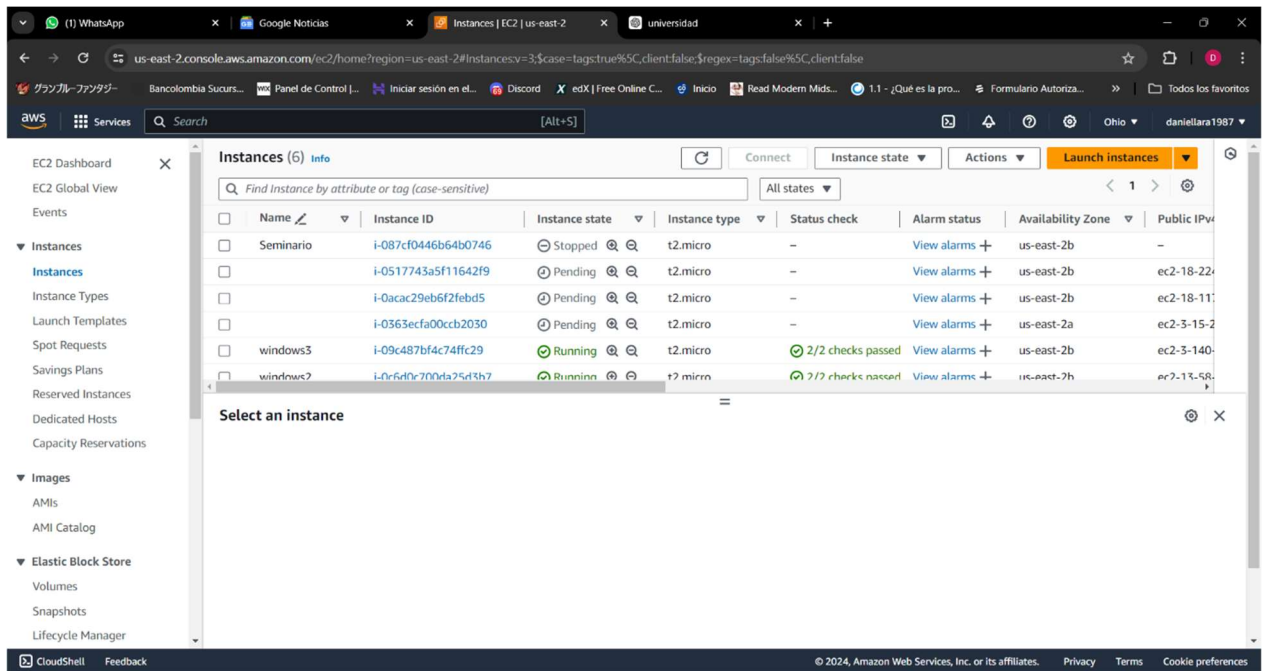


Figura 39. Creación automática de instancias deseadas según auto escalado

Fuente: Elaboración propia

Para concluir el ejercicio, se propone probar las políticas de auto escalado al forzar el uso de CPU en una de las instancias y así obligar a la creación de otras para reemplazarla. Esto se logrará al logear una de las instancias y usar un servicio de pruebas de estrés para la CPU, en este caso se usará Silverbench.com para forzar la CPU al 100% y disparar la creación de nuevas instancias

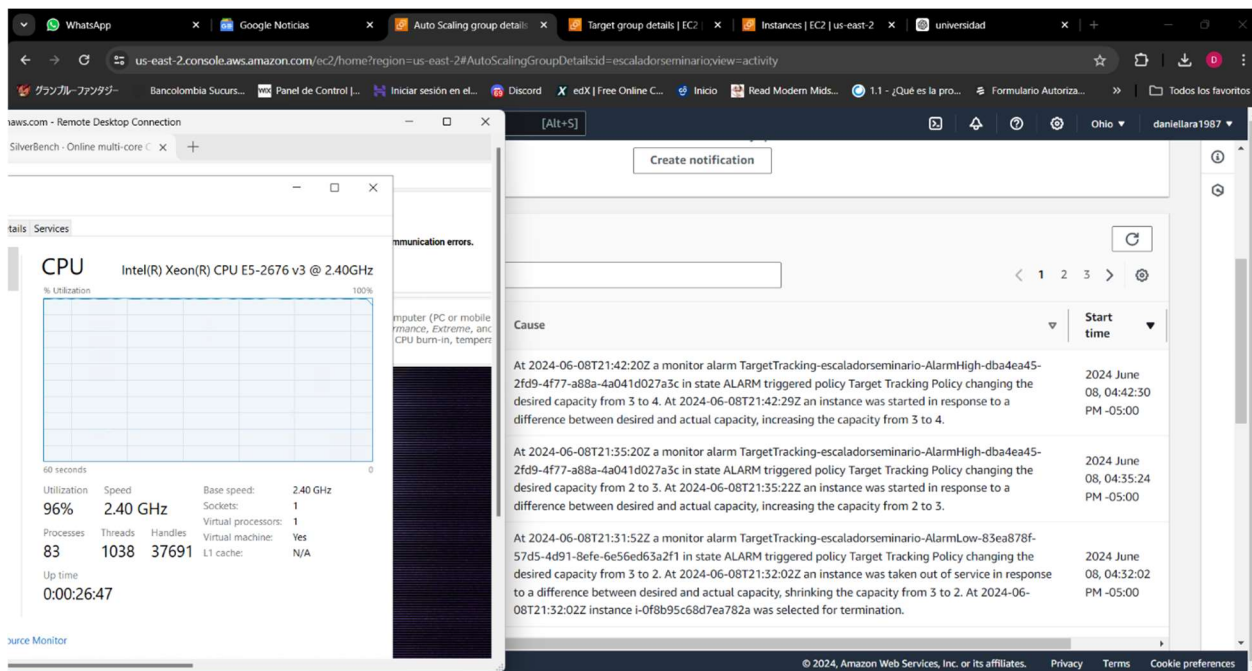
The screenshot shows a remote desktop connection to an Amazon EC2 instance. The browser window displays the SilverBench website, which is a free online network tools and utilities site. The website has a navigation menu with options like Tools, Utilities, Misc, My IP, Smart Whois, Hash & CRC, Response header, and SilverBench. The SilverBench tool is running in the browser, and a Windows Task Manager window is open in the foreground, showing the Performance tab with CPU usage at 81% and Memory usage at 88%. The Task Manager window lists various processes, including Microsoft Edge (14) and Task Manager, with their respective CPU and memory usage.

Name	Status	81% CPU	88% Memory
<b>Apps (2)</b>			
Microsoft Edge (14)		53.1%	185.5 MB
Task Manager		2.0%	10.1 MB
<b>Background processes (38)</b>			
.NET Runtime Optimization Serv...		12.1%	8.6 MB
amazon-ssm-agent		0%	0.2 MB
AntiMalware Definition Update		0%	0.1 MB
Antimalware Service Executable		0.5%	94.4 MB
CTF Loader		0%	1.2 MB
Host Process for Windows Tasks		0.5%	1.0 MB
Host Process for Windows Tasks		0%	0.1 MB
Host Process for Windows Tasks		0%	0.2 MB
IIS Worker Process		0%	0.4 MB
Microsoft .NET Framework opti...		0%	0.6 MB

*Figura 40. Uso de Silverbench para forzar CPU*

*Fuente:* Elaboración propia

En el apartado de actividad que se encuentra dentro del grupo de escalado, se pueden observar las respuestas automáticas creadas por las reglas que se han configurado. El uso de la CPU al 100% en la instancia dispara la creación de otras instancias debido a que esta instancia se considera como inalcanzable.



*Figura 41. historial de actividad del grupo de auto escalado*

*Fuente:* Elaboración propia

Para terminar, el auto escalado creara las instancias necesarias para cumplir la política creada y asegurar la disponibilidad del servicio. Luego de que detecte que se ha estabilizado, terminara las instancias extra y regresara al mínimo que se ha definido de manera automática, manteniendo un optimo uso de recursos y evitando sobrecostos

The screenshot shows the AWS Management Console interface for EC2 instances. The main content area displays a table of instances with the following data:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
	i-0f8b95c68d7ea782a	Terminated	t2.micro	-	View alarms	us-east-2b	-
	i-08d8100197ee24568	Running	t2.micro	2/2 checks passed	View alarms	us-east-2b	ec2-52...
	i-0517743a5f11642f9	Terminated	t2.micro	-	View alarms	us-east-2b	-
	i-0aac29eb6f2febd5	Terminated	t2.micro	-	View alarms	us-east-2b	-
	i-05ab5e019dc7455ec	Running	t2.micro	Initializing	View alarms	us-east-2a	ec2-3-11...
	i-0363ecfa00ccb2030	Terminated	t2.micro	-	View alarms	us-east-2a	-
windows3	i-09c487bf4c74ffc29	Terminated	t2.micro	-	View alarms	us-east-2b	-
windows2	i-0c6d0c700da25d3b7	Terminated	t2.micro	-	View alarms	us-east-2b	-
	i-0ae69c41d619c8e7f	Running	t2.micro	2/2 checks passed	View alarms	us-east-2b	ec2-13-...

Below the table, the details for instance **i-00aeecd7782508d9** are shown. The Instance ID is **i-00aeecd7782508d9**. The Public IPv4 address is **3.12.73.112** (with a link to open address). The Private IPv4 addresses are **10.20.4.84**.

Figura 42. Manejo de instancias del grupo de auto escalado

Fuente: Elaboración propia

## Conclusiones

El uso de Amazon Web Services (AWS) ha cambiado la manera en que las empresas y desarrolladores implementan y gestionan infraestructuras tecnológicas. A lo largo de este proyecto, se ha explorado múltiples aspectos de AWS, desde la configuración básica de instancias EC2 hasta la implementación de soluciones complejas como el balanceo de carga y el escalado automático. La flexibilidad y el escalado proporcionados por AWS permiten a las organizaciones responder de manera efectiva a las demandas dinámicas de sus aplicaciones y servicios.

Durante este proyecto, se configuraron instancias EC2 que actúan como servidores para aplicaciones web simples. A través de la implementación de los balanceadores de carga, también conocidos como Elastic Load Balancer (ELB), pudimos distribuir eficazmente el tráfico de red entre varias instancias, mejorando así la disponibilidad y la tolerancia a fallos de las aplicaciones. El ELB, junto con el Auto Scaling, permitió automatizar el ajuste de la capacidad de las instancias según la carga de trabajo, asegurando un rendimiento óptimo sin intervención manual continua.

La configuración de roles IAM y las políticas adecuadas aseguraron que las operaciones, como la carga de archivos en S3 y el acceso a bases de datos RDS, se realizaran de forma segura y con el nivel adecuado de permisos. Este enfoque resaltó la importancia de la seguridad y la administración granular de accesos en entornos de nube.

El uso de distintos sistemas operativos como Linux o Windows demostró la versatilidad de los servicios de AWS y su capacidad para ajustarse a las distintas necesidades del usuario.

Aunque los servicios usados para este proyecto han sido todos dentro de la capa gratuita de AWS, las distintas configuraciones y herramientas de monitoreo proporcionan control a los costos que se pueden generar por su uso.

La capacidad de AWS para ofrecer servicios escalables, seguros y de alta disponibilidad reafirma su posición como un líder en la computación en la nube, impulsando a organizaciones de todos los tamaños a innovar y crecer en el competitivo entorno digital de hoy.

## Referencias

Amazon (2024) Amazon Relational Database Service

[https://aws.amazon.com/es/rds/?nc2=h\\_ql\\_prod\\_fs\\_rds](https://aws.amazon.com/es/rds/?nc2=h_ql_prod_fs_rds)

Amazon (2024) Amazon S3 [https://aws.amazon.com/es/s3/?nc2=h\\_ql\\_prod\\_fs\\_s3](https://aws.amazon.com/es/s3/?nc2=h_ql_prod_fs_s3)

Amazon (2024) AWS Identity and Access Management

<https://aws.amazon.com/es/iam/?c=sc&sec=srvm>

Amazon (2024) Computación en la nube con AWS <https://aws.amazon.com/es/what-is-aws/>

Amazon (2024) Grupos de escalado automático

[https://docs.aws.amazon.com/es\\_es/autoscaling/ec2/userguide/auto-scaling-groups.html](https://docs.aws.amazon.com/es_es/autoscaling/ec2/userguide/auto-scaling-groups.html)

Amazon (2024) Informática para cualquier carga de trabajo

<https://aws.amazon.com/es/products/compute/>

Amazon (2024) Interfaz de línea de comandos de AWS <https://aws.amazon.com/es/cli/>

Amazon (2024) ¿Qué es un Equilibrador de carga de aplicación?

[https://docs.aws.amazon.com/es\\_es/elasticloadbalancing/latest/application/introduction.html](https://docs.aws.amazon.com/es_es/elasticloadbalancing/latest/application/introduction.html)

Amazon (2024) ¿Qué es una instancia de computación en la nube?

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

Azure (2024) Azure products <https://azure.microsoft.com/en-us/products/>

Cloudflare (2024) ¿Que es un proxy inverso? <https://www.cloudflare.com/es-es/learning/cdn/glossary/reverse-proxy/>

Docker (2024) httpd [https://hub.docker.com/\\_/httpd](https://hub.docker.com/_/httpd)

Fernández E. (2021, 8 de mayo) Historia de la computación en la nube

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

Foot K. (2021, 17 de diciembre). A brief history of cloud computing

<https://www.dataversity.net/brief-history-cloud-computing/>

García F. (2023, 15 de marzo) WSL (Windows Subsystem Linux) o cómo y por qué trabajar con Linux en Windows <https://www.arsys.es/blog/wsl-windows-subsystem-linux>

Globaldata (2020, 18 de junio) Cloud Computing: Timeline

<https://www.verdict.co.uk/cloud-computing-timeline/?cf-view>

Google (2024) La nueva forma de la nube comienza aquí

<https://cloud.google.com/>

Google (2024) ¿Qué es la computación en la nube? <https://cloud.google.com/learn/what-is-cloud-computing?hl=es-419>

Oracle (2024) ¿Qué es una base de datos? <https://www.oracle.com/mx/database/what-is-database/>

Pippol (2024) ¿Qué es el escritorio remoto? <https://www.pippol.es/blog/que-es-el-escritorio-remoto/>

Redhat (2023, 31 de enero) Las máquinas virtuales

<https://www.redhat.com/es/topics/virtualization/what-is-a-virtual-machine>

Ticportal (2023, 7 de diciembre) Servidores <https://www.ticportal.es/glosario-tic/servidores>

Slingerland C. (2023, 15 de diciembre) The Simple Guide To The History Of The Cloud <https://www.cloudzero.com/blog/history-of-the-cloud/>