



**TRABAJO DE GRADO
Seminario**

Seminario Amazon web Services (AWS)

Corporación Universitaria Remington.
Facultad de Ingeniería
Ingeniería de Sistemas

Nataly Elaine Diaz Martinez
Rafael Eduardo Montes Pérez
Decio Mena Rentería
Nombre del Tutor del trabajo de grado
Juan Pablo Berrío López
Trabajo de grado-Seminario
Año de presentación del trabajo de grado.
2024

Tabla de Contenidos

Resumen.....	3
Marco conceptual.....	5
Marco Contextual.....	7
Desarrollo e implementación del aprendizaje.....	8
Implementación de una cuenta de AWS con contenedores	12
Índice de tablas.....	40
Conclusiones	42
Referencias.....	43

Resumen

El seminario de Amazon Web Services (AWS) comenzó con una introducción sobre la virtualización y la computación en la nube, destacando sus orígenes. AWS es una plataforma integral en constante expansión que ofrece una variedad de servicios de infraestructura en la nube, como cómputo, almacenamiento, bases de datos, redes, analítica e inteligencia artificial.

El seminario se dividió en varias secciones clave. Se enfocó en la implementación de una cuenta de AWS utilizando contenedores, la conexión de una instancia AWS en Linux para cargar archivos en S3, y la conexión a una base de datos RDS MySQL. También se abordó la configuración de políticas de escalado basadas en métricas y la prueba de Auto Scaling aumentando artificialmente la carga en la aplicación para lanzar nuevas instancias EC2.

Se inició con la configuración de una cuenta de AWS, detallando el proceso de creación, la configuración de facturación y la implementación de políticas de seguridad. Se enfatizó la gestión de identidades y accesos (IAM), mostrando cómo crear usuarios, roles y grupos para manejar permisos de manera eficiente.

Luego, se introdujo el uso de contenedores en AWS, presentando Amazon ECS (Elastic Container Service) como solución para ejecutar contenedores Docker. Se mostró cómo crear un clúster ECS, definir tareas y servicios, y desplegar aplicaciones contenedorizadas.

En la siguiente parte, se explicó el proceso de lanzamiento de una instancia EC2 con Linux. Esto incluyó la selección del tipo de instancia, la configuración de grupos de seguridad y el uso de pares de claves SSH para la conexión segura. Se enseñó a instalar y configurar AWS CLI en la instancia EC2 para interactuar con los servicios de AWS y cargar archivos en un bucket S3.

Además, se abordó la creación de una instancia de Amazon RDS con MySQL, incluyendo la configuración de parámetros, seguridad y opciones de escalabilidad. Se demostró cómo instalar el cliente MySQL en la instancia EC2 y conectarse a la base de datos RDS mediante la línea de comandos.

El seminario reafirmo las bondades de aws como servicio de computación en la nube.

Palabras clave

(Amazon Web Services (AWS), Computación en la nube, Infraestructura, Contenedores, Instancias, Auto Scaling, Docker, Amazon ECS, Virtualización.

Marco conceptual

Computación en la Nube: Modelo de entrega de recursos informáticos (servidores, almacenamiento, bases de datos, redes, software, analítica, inteligencia artificial) a través de Internet (la "nube") para ofrecer innovación rápida, recursos flexibles y economías de escala. (Sanabria, 2022)

Virtualización: Tecnología que permite crear una versión virtual (en lugar de real) de algo, como un sistema operativo, un servidor, un dispositivo de almacenamiento o recursos de red, permitiendo que múltiples sistemas operativos coexistan en un solo hardware físico. (Caraves Muñoz, 2014)

Contenedores: Un método para empaquetar una aplicación y sus dependencias de manera que pueda ejecutarse de manera aislada y consistente en diferentes entornos. Docker es una tecnología ampliamente utilizada para la creación y gestión de contenedores. (Pérez López, 2023)

Amazon Web Services (AWS): Plataforma de servicios en la nube de Amazon que ofrece una amplia gama de servicios de infraestructura como computación, almacenamiento, bases de datos, redes, y herramientas de desarrollo para ayudar a las organizaciones a escalar y crecer. (Kamlofsky, 2022)

Amazon ECS (Elastic Container Service): Servicio de AWS que permite ejecutar y gestionar contenedores Docker a gran escala, facilitando el despliegue, gestión y escalado de aplicaciones contenedorizadas. (García Granados, 2023)

Instancias EC2: Servicios de computación en la nube que proporcionan capacidad de procesamiento escalable en la nube. Permiten ejecutar aplicaciones en servidores virtuales configurables y escalables según la demanda. (Bravo Mendoza, 2023)

Amazon S3 (Simple Storage Service): Servicio de almacenamiento de objetos de AWS que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento de alta durabilidad para cualquier cantidad de datos. (Rodríguez Mosquera, 2023)

RDS (Relational Database Service): Servicio gestionado que facilita la configuración, operación y escalado de una base de datos relacional en la nube. Soporta varias bases de datos, incluida MySQL. (Brito Zhunio, 2011)

IAM (Identity and Access Management): Servicio de AWS que permite controlar el acceso a los recursos de AWS de forma segura, mediante la gestión de usuarios, permisos y roles. (Agulló Valls, 2021)

Marco Contextual

Transformación Digital: Las empresas están adoptando la computación en la nube como parte de su estrategia de transformación digital para mejorar la agilidad, reducir costos y acelerar la innovación. AWS, como uno de los proveedores líderes de servicios en la nube, juega un papel crucial en este proceso. (Campos Campos, 2021)

Demanda de Escalabilidad y Flexibilidad: Las organizaciones buscan soluciones que les permitan escalar sus operaciones rápidamente y adaptarse a las fluctuaciones del mercado. La computación en la nube, y en particular AWS, ofrece la flexibilidad necesaria para ajustar los recursos según las necesidades. (Planta Ulloa, 2024)

Seguridad y Conformidad: La seguridad de los datos y el cumplimiento de regulaciones son prioridades esenciales para las empresas que migran a la nube. AWS proporciona herramientas y servicios robustos para garantizar la seguridad de los datos y el cumplimiento de normativas. (Rojas Serrano, 2023)

Innovación y Competitividad: Las empresas están utilizando servicios avanzados de AWS, como inteligencia artificial, machine learning, y analítica de datos, para innovar y mantenerse competitivas en sus respectivos mercados. (Farro Flores, 2017)

Educación y Capacitación: Con la creciente adopción de tecnologías en la nube, existe una demanda creciente de profesionales capacitados en AWS y sus servicios. Este seminario contribuye a llenar este vacío, proporcionando conocimientos y habilidades prácticas necesarias para gestionar infraestructuras en la nube de manera eficiente. (Mendoza Ciprian, 2018)

Desarrollo e implementación del aprendizaje

Los orígenes de la virtualización y la computación en la nube se cree que comenzó en los años de 1950 cuando se desarrollaron las primeras redes de computadoras, por el gobierno de los Estados Unidos, que fue uno de los primeros sistemas de defensa antiaérea automatizados en el cual, se utilizaba varias computadoras conectadas en red para rastrear objetos voladores no identificados. Herb Grosh, un científico informático estadounidense que realizó importantes contribuciones al desarrollo de la computación en la nube, para esa misma época afirmó que las economías se podrían adaptar mejor si confiaban en el almacenamiento de datos centralizado, y no en equipos individuales con almacenamiento de unidades.

Igualmente las diferentes empresas de mainframes se vieron en la necesidad de conectar distintos usuarios a la vez y manejar sus recursos desde distintas terminales o nodos de accesos; porque para ese entonces ya las empresas trabajaban con máquinas independientes o centralizadas; por otro lado, a muchas empresas les era muy complicado tener una gran infraestructura de cómputo por el tamaño de los equipos y su valor económico, esto generó que entre 1956 y 1961 el científico de la computación y matemático, John McCarthy presentara la idea de crear la computación colectiva o compartida en que esos recursos utilizados por las empresas, fueran pagados de la misma forma o parecido a como se contratan los servicios públicos, estos quiere decir que se pagaría de acuerdo al consumo. También para este tiempo, IBM desarrolló el sistema operativo CP-67, un hipervisor inicial que introdujo la memoria virtual a la familia de servidores System 360 de la empresa.

Al alrededor de 1960-1969 Joseph Carl Robnett Licklider, científico de la computación, quien contó con la ayuda de Bob Taylor un ingeniero de sistemas en el laboratorio de investigación de Xerox, propuso la idea de una "oficina sin papeles"; manteniendo la información digitalizada y con acceso desde cualquier lugar y Larry Roberts, también científico de la computación, considerado el padre de la internet; propuso una red intergaláctica o sistemas interconectados de ordenadores, que dada la tecnología existente en ese momento no fue posible implementar por la lentitud del

desarrollo tecnológico de ese tiempo; lideró el trabajo y desarrollo de lo que conocemos como arpanet, que es la precursora del internet; la cual se expandió rápidamente por su ancho de banda para soportar la información en la nube, permitiendo el uso de la interconexión que ofrecía la empresa conocida como Salesforce a diferentes empresas del mercado.

Licklider imaginaba un mundo interconectado y esta visión de poder acceder a programas y datos desde cualquier lugar del mundo se fue desarrollando, convirtiéndose en lo que conocemos hoy como cloud computing.

La idea de arpanet fue una propuesta en primer lugar por Leonard Kleinrock en un artículo titulado: "Information Flow in Large Communication Nets." Él hablaba de redes de ordenadores interconectados que trabajaban desde cualquier lugar comunicándose entre ellos de forma remota.

En 1963, el ingeniero de sistemas John C. Shaw, propuso la misma idea de Red Intergaláctica de Computadoras, donde los usuarios podrían acceder a recursos informáticos remotos desde cualquier lugar.

La aparición del arpanet en 1967-1970 que consistía en una red de ordenadores creada por el Departamento de Defensa de los EEUU y las diferentes instituciones académicas y estatales, tuvo un fuerte impacto en la historia de la información en la nube, lanzando la primera red que usó el protocolo TCP/IP para interconectar redes de computación de diferentes configuraciones, facilitando así la transferencia de datos entre ellas de manera descentralizada mejorando la eficiencia, la fiabilidad de la computación en la nube.

Entre 1970 y 1990 hubo grandes avances tecnológicos en cuanto a Internet, la computación en la nube o las máquinas virtuales, afianzando el concepto de virtualización, como lo hizo IBM desarrollando la primera máquina virtual en 1972; haciendo posible ejecutar varios sistemas operativos a la vez en un espacio aislado y totalmente diferentes; en 1999, se llevó a cabo la 1er ola de la nube por la empresa Salesforce, la cual proporcionaba aplicaciones de gestión y software de automatización para empresas.

Desde el 1999 el Cloud Computing, por lo tanto, ha jugado un papel importante en el apogeo del Big Data, facilitando herramientas que le ha permitido a las organizaciones mayor flexibilidad en su infraestructura y el software, como poder recolectar, almacenar, procesar y analizar cantidades masivas de datos a nivel global y crear nuevas oportunidades para las empresas en cualquier momento y lugar, directamente desde la nube.

En el 2002 Amazon comenzó a ofrecer almacenamiento de datos y uso compartido de recursos en la nube y para el 2006 lleva al mercado de la informática el EC2 para que los clientes pudieran alquilar servidores para uso de aplicaciones virtuales. Igualmente, entre el 2006 y 2008 el EC2 comenzó a usarse de manera comercial para las diferentes empresas no importando su tamaño. Google cloud también en 2009 comenzó a ofrecer servicios basados en la nube, siguiéndole Microsoft Azure, Huawei Cloud

Las empresas como Amazon, Google, Alibaba Cloud y Microsoft fueron las primeras en ofrecer el servicio de recursos TI en la nube a gran escala, creando los ecosistemas digitales que conocemos hoy por medio de la innovación.

En el 2002-2006, Amazon Web Services (AWS) rompió los parámetros de servicios en acceso a la infraestructura de cómputo, permitiendo a diferentes empresas tener más rentabilidad en el uso de la computación en la nube.

En 2006, con la Web 2.0, Google Cloud Platform (GCP) abriendo las puertas al análisis de datos a través de la inteligencia artificial basadas en navegadores y la innovación.

En el 2009 se funda Alibaba Cloud es una empresa que proporciona servicios computacionales en la nube para empresas en línea y a su propio ecosistema de comercio electrónico de Alibaba

En el 2010 es lanzado Microsoft Azure que a permitido por medio del cloud brindar soluciones de migración híbridas en la nube. Ofrece una amplia gama de servicios, que incluyen: cómputo, almacenamiento, redes, bases de datos, análisis, inteligencia artificial y aprendizaje automático

En 2012 Oracle Cloud anunció el lanzamiento de un conjunto de soluciones informáticas en la nube. Google Drive igualmente lanzó el servicio de almacenamiento y

sincronización de archivos que permitía a los usuarios almacenar, compartir y editar documentos colaborativamente.

En 2013, IBM adquirió SoftLayer, un importante proveedor global de infraestructura de nube, ampliando su oferta de soluciones de infraestructura como servicio (IaaS).

La comparación de nombres de servicios entre AWS, Azure y GCP, si bien sabemos que los tres ofrecen buenas soluciones, y su elección se basa mucho en la necesidad que se tenga, el precio de las tres es muy semejantes.

Tabla 1 Computación.

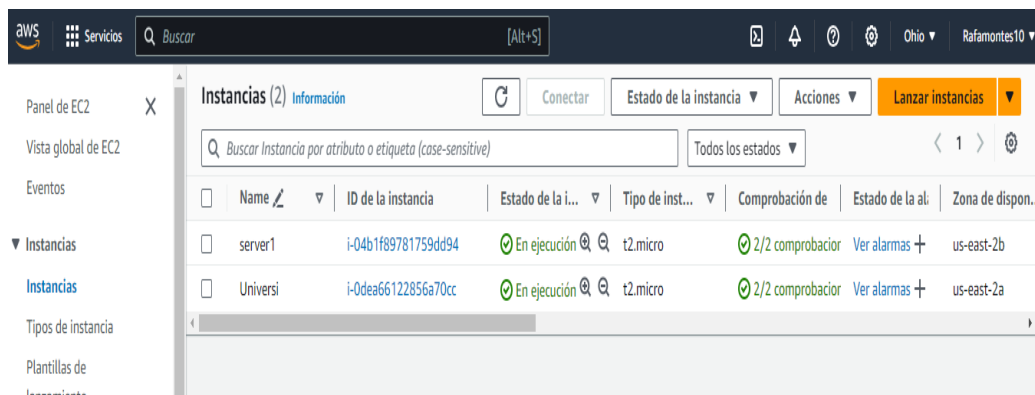
computacion			
servicio	aws	Azure	GCP
Maquinas virtuales	EC2	Virtual Machines	Compute Engine
funciones sin servidor	Lambda	Functions	Cloud Functions
Almacenamiento			
objetos	S3	Blob Storage	Cloud Storage
archivos	EFS	File Storage	Filestore
Bases de Datos			
Base de datos	RDS	SQL Database	Cloud SQL
Base de datos NoSQL	DynamoDB	Cosmos DB	Firestore/Datastore
Data warehouse	Redshift	Synapse Analytics	BigQuery
Redes			
Redes virtuales	VPC	Virtual Network	Virtual Private Cloud
CDN	CloudFront	Azure CDN	Cloud CDN
Herramientas de Gestión			
Monitoreo	CloudWatch	Azure Monitor	Stackdriver
Gestión de identidades	IAM	Active Directory AD	Cloud IAM
Automatización	CloudFormation	Resource Manager (ARM)	Deployment Manager
Machine Learning e Inteligencia Artificial			
Machine Learning	SageMaker	Azure Machine Learning	AI Platform
Servicios de IA	Rekognition	Cognitive Services	Cloud AI Vision, NLP
Analítica y Big Data			
Procesamiento de datos	EMR	HDInsight	Dataproc
Integración de datos	Glue	Data Factory	Dataflow

Fuente: Elaboración propia

Implementación de una cuenta de AWS con contenedores

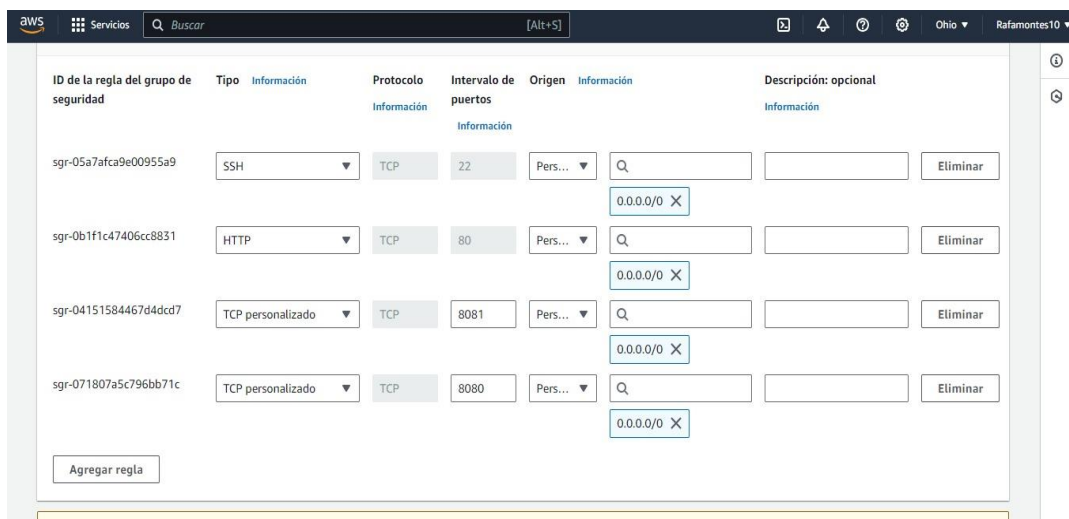
Estos fueron los paso a paso que se implementaron para que, al ejecutar los dos contenedores, cada uno responda solamente a un nombre de dominio específico, la implementación se realiza con el registro de la cuenta en AWS y luego la opción EC2.

Figura 1 Instancias en AWS



Fuente: Elaboración propia en AWS

Figura 2 Las reglas de internet.



Fuente: Elaboración propia en AWS

La regla SSH permite que la WSL pueda conectarse a la instancia de Linux.

Figura 6 Generación de códigos

```

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
#
#
# 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

LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so

```

Fuente: Elaboración propia en AWS

El proxy es la tecnología que se utilizó para realizar este trabajo y este es el intermediario que comunica al cliente con el servidor remoto, actuando como puente para que el cliente pueda acceder a los recursos del del servidor.

Una vez esto creamos el virtual host con el código sudo nano /etc/httpd/conf.d/vhost.conf.

Figura 7 Creación del virtual host

```

root@ip-10-0-10-125/home/e x + v
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

actividad@LAPTOP-459RP311:~$ chmod 400 "trabajo.pem"
actividad@LAPTOP-459RP311:~$ ssh -i "trabajo.pem" ec2-user@ec2-18-119-112-163.us-east-2.compute.amazonaws.com

#####
##### Amazon Linux 2023
#####
##### |
##### \##/
#####  \##/  https://aws.amazon.com/linux/amazon-linux-2023
#####   V##/
#####  /##/
##### /##/

Last login: Sun May 26 18:34:33 2024 from 186.84.91.42
[ec2-user@ip-10-0-10-125 ~]$ sudo su
[root@ip-10-0-10-125 ec2-user]# sudo nano /etc/httpd/conf.d/vhost.conf |

```

Fuente: Elaboración propia en AWS

Figura 8 Virtual host creado

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

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

<VirtualHost *:80>
  ServerName tienda.com

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

Fuente: Elaboración propia en AWS

Dentro de ese código agregamos la configuración para definir dos hosts virtuales. Una vez terminado esto, configuramos con la Ip del computador asignando la Ip de cada propiedad donde una es comercio.com y tienda.com.

Figura 9 Direcciones IP

```
#
127.0.0.1 localhost
::1 localhost
18.119.112.163 comercio.com
18.119.112.163 tienda.com
```

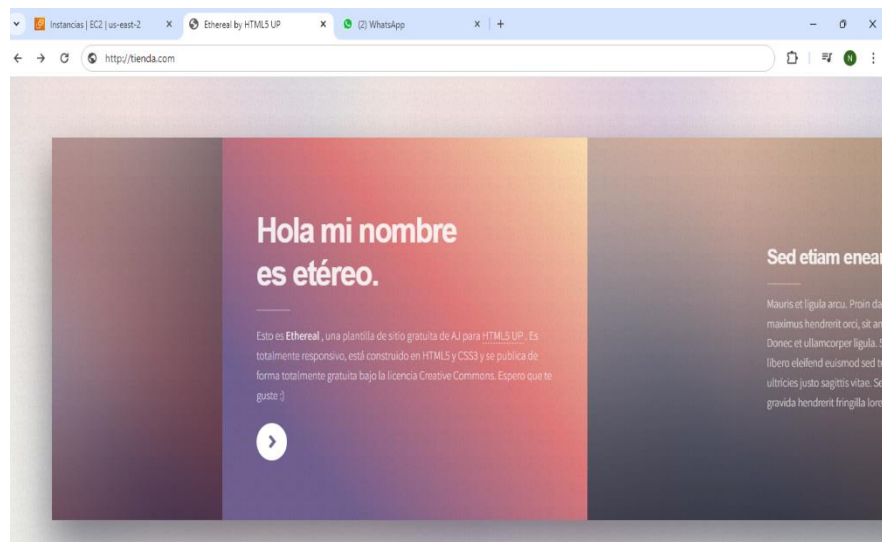
Fuente: Elaboración propia en AWS

Una vez terminado toda la implementación, podemos consultar como queda con las siguientes rutas:

<http://tienda.com>

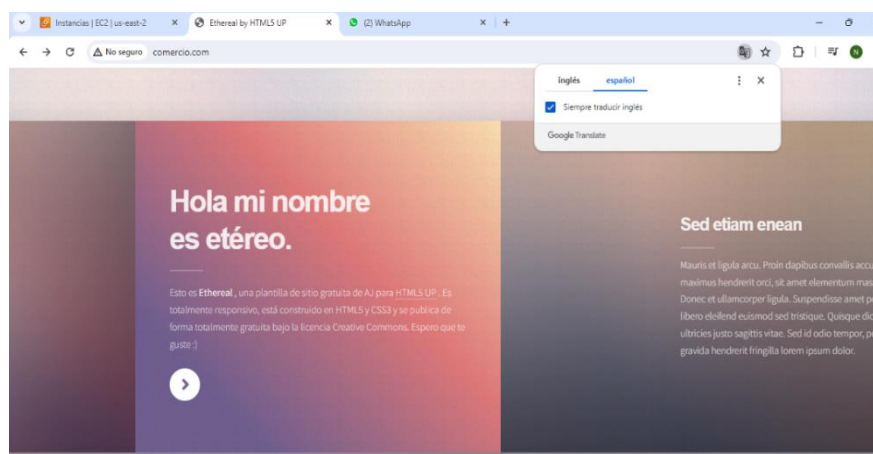
<http://comercio.com>

Figura 10 Imagen de página web tienda.com



Fuente: Elaboración propia en AWS

Figura 11 Imagen de página web Comercio.com

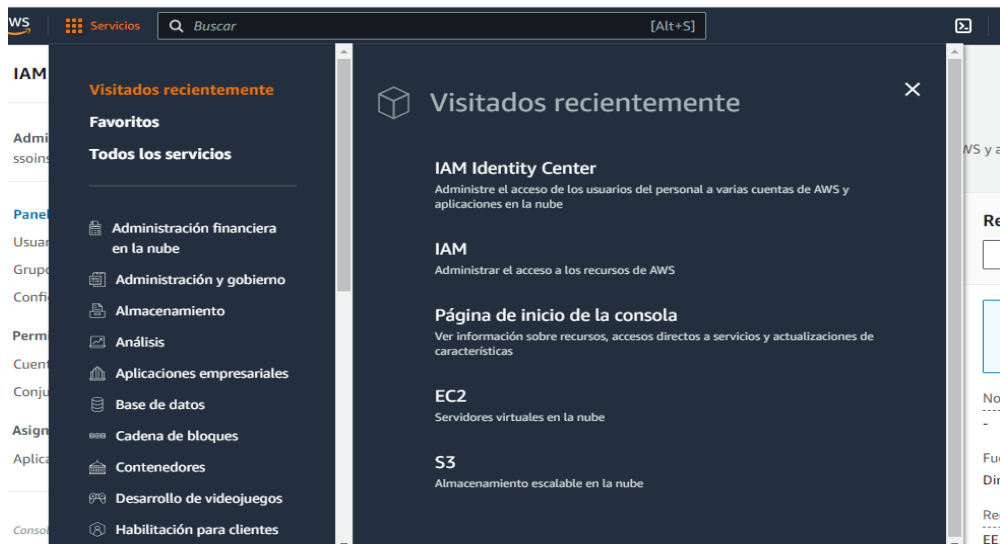


Fuente: Elaboración propia en AWS

Conexión de una instancia AWS en Linux para cargar archivos en S3 y además pueda conectarse a una base de datos RDS Mysql.

Inicialmente es crear una cuenta de IAM estas son para gestionar de manera segura y eficiente el acceso a los recursos de una organización.

Figura 12 Inicialización de cuenta IAM



Fuente: Elaboración propia en AWS

Figura 13 Creación de usuarios.

Información principal	
Clave de atributo	Valor
Nombre de usuario	Usuario1
Correo electrónico	nataly.diaz.1199@miremington.edu.co
Nombre	Manuel
Apellidos	Martinez
Nombre de visualización	Manuel Martinez
▶ Métodos de contacto - opcional	

Fuente: Elaboración propia en AWS

Se crea un respectivo usuario con el fin de tener un mayor control en el acceso y seguridad.


Figura 14 configuración de autenticación.


Registrar dispositivo MFA


Nombre de usuario:
 Usuario1 (¿No es usted?)

La organización requiere la autenticación multifactor (MFA) para mayor seguridad durante el inicio de sesión. Cada vez que inicie sesión, se le pedirá que indique su contraseña y un dispositivo MFA. [Más información](#)

Seleccione una de las siguientes opciones para comenzar:

 **Aplicación de autenticación**
 Autenticar con un código generado por una aplicación instalada en su dispositivo móvil o equipo.

 **clave de seguridad**
 Para autenticar, presione una llave de seguridad de hardware como YubiKey, Feitian, etc.

 **Autenticador integrado**
 Autenticar con un escáner de huellas digitales o una cámara integrada en su equipo, como Apple TouchID, Windows Hello, etc.

Fuente: Elaboración propia en AWS

La autenticación de MFA es un método de autenticación que requiere que un usuario presente dos o más tipos de evidencia para verificar su identidad.

Figura 15 Configuración de políticas y permisos

Tipos

Conjunto de permisos predefinido
 Para crear un conjunto de permisos predefinidos, elija una plantilla definida por AWS. Esta plantilla permite seleccionar una única política administrada por AWS. Por ejemplo, puede seleccionar una política que conceda permisos para una función de trabajo común, como Billing, o un nivel específico de acceso a los servicios y recursos de AWS, como ViewOnlyAccess. Puede actualizar el conjunto de permisos a medida que evolucionen sus necesidades.

Conjunto de permisos personalizado
 Para crear un conjunto de permisos personalizado, seleccione las políticas administradas por AWS y cree una política insertada (opción recomendada). También puede asociar políticas administradas por el cliente y definir un límite de permisos (avanzado).

Política para el conjunto de permisos predefinido

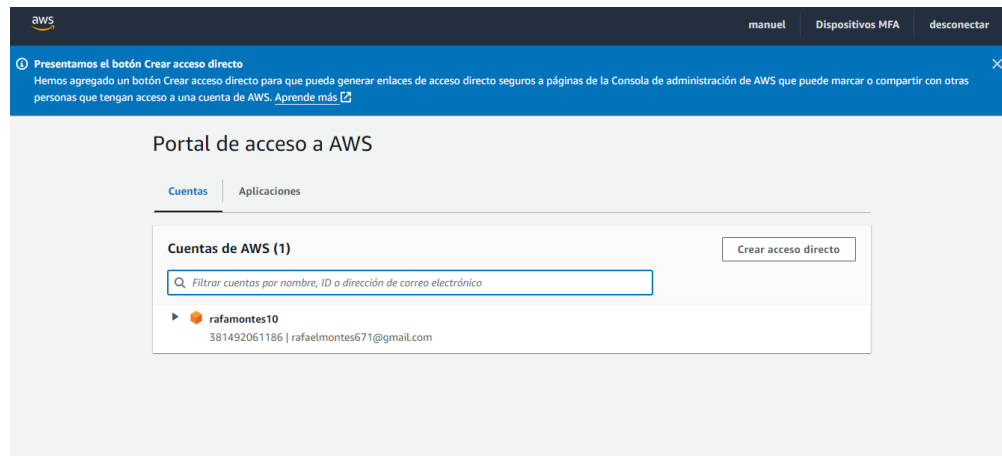
Seleccione una política administrada por AWS

- AdministratorAccess**
 Provides full access to AWS services and resources.
- Billing**
 Grants permissions for billing and cost management. This includes viewing account usage and viewing and modifying budgets and payment methods.
- DatabaseAdministrator**
 Grants full access permissions to AWS services and actions required to set up and configure AWS database services.

Fuente: Elaboración propia en AWS

Una vez confirmada la autenticación, se pasa a configurar las políticas y permisos que se le van a asignar al nuevo usuario que va a ingresar.

Figura 16 Portal de acceso a AWS



Fuente: Elaboración propia en AWS

Finalmente, así quedaría la cuenta en el portal de acceso a AWS este es donde los usuarios pueden gestionar y controlar todos los aspectos de sus recursos en la nube. Y a su vez les Permite a los usuarios acceder y gestionar una amplia gama de servicios de AWS, como EC2, S3, RDS.

Para hacer todos estos ajustes se necesita instalar una aplicación en WSL llamada aws cli donde se instala con el siguiente commando.

Figura 17 Creación del bucket

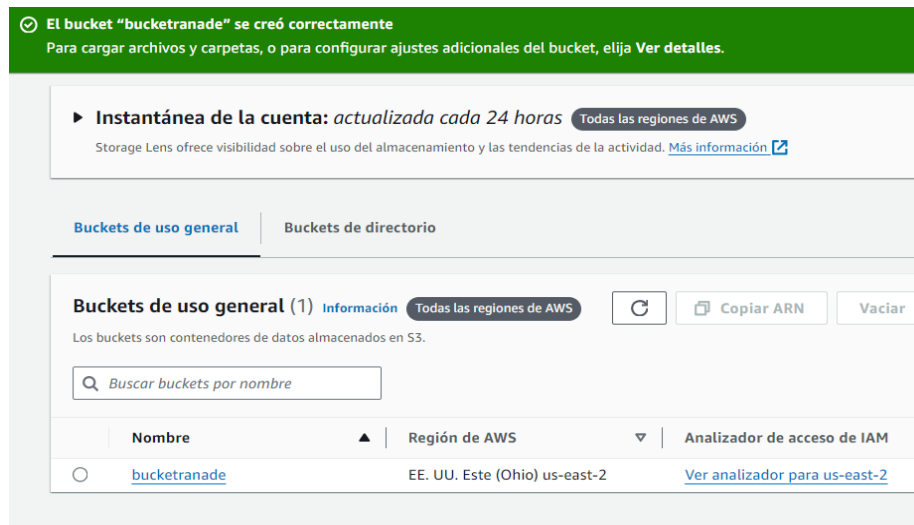
Para instalar AWS CLI, ejecute los siguientes comandos.

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

Fuente: Elaboración propia en AWS

Se crea un bucket en el s3 este es un contenedor utilizado para almacenar objetos, que pueden ser archivos de cualquier tipo y tamaño. Los buckets se utilizan para organizar y agrupar objetos, cada cubo puede contener una cantidad prácticamente ilimitada de datos, a este se le debe asignar un nombre en este caso lo llamamos bucketranade.

Figura 18 Imagen de bucket creado



Fuente: Elaboración propia en AWS

Para subir un archivo al bucket mediante la consola WSL debe ser mediante un código dicho código se obtiene inicialmente del portal de acceso a WSA en la opción claves de acceso.

Figura 19 Clave de acceso

Ejecute los siguientes comandos en su terminal para configurar las variables de entorno de AWS. [Aprende más](#)

```
exportar AWS_ACCESS_KEY_ID = "ASIAVRUVS4QBFIYIG7XW7"
exportar AWS_SECRET_ACCESS_KEY = "KKaG9JZKrLpjjlCJCOGI+B8dUHQY0yUAt
export AWS_SESSION_TOKEN = "IQoJb3JpZ2luX2VjELr////////wEaCXVzLWw"

```

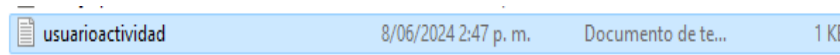
[Copiar](#)

▼ Opción 2: asigne un perfil a su archivo de credenciales de AWS

Fuente: Elaboración propia en AWS

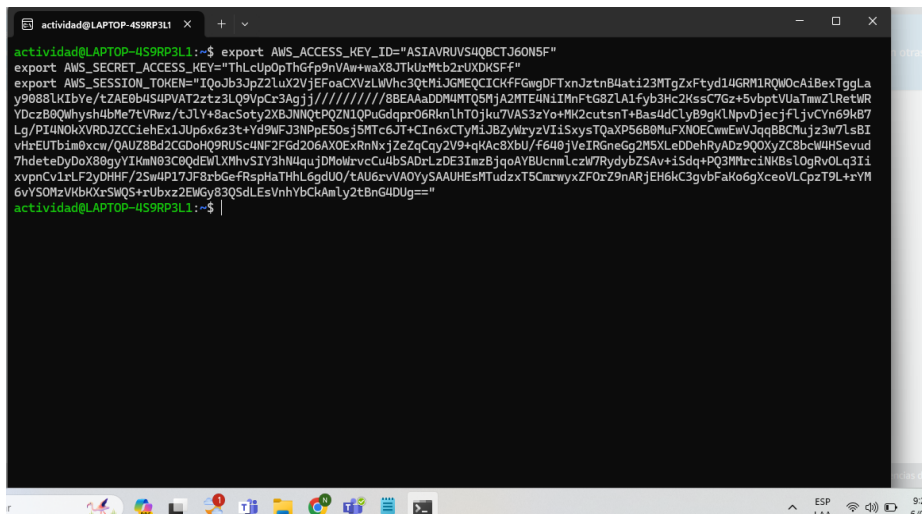
Este es ingresado a la consola y en esta se va a subir un archivo en la dirección del bucket que en este caso es s3://bucketranad.

Figura 20 Archivo



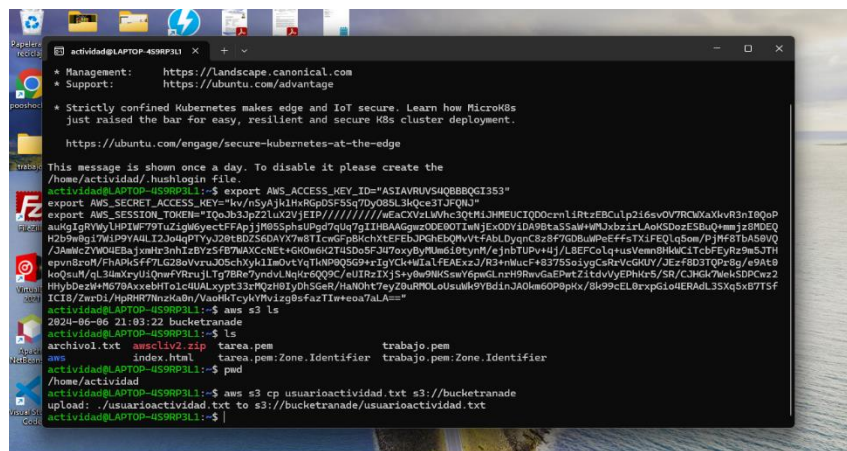
Fuente: Elaboración propia en AWS

Figura 21 Exportación de clave de acceso en WSL



Fuente: Elaboración propia en AWS

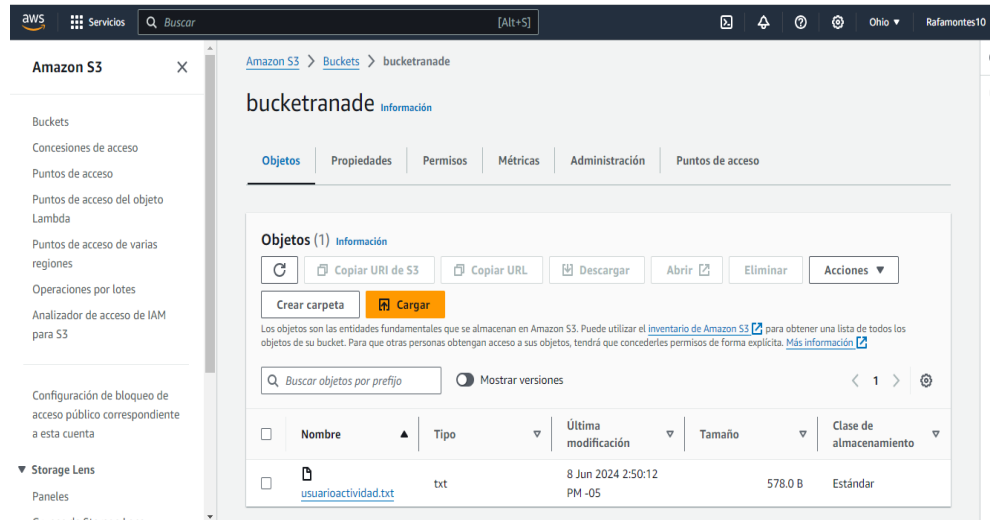
Figura 22 configuración por consola



Fuente: Elaboración propia en AWS

Una vez subido el archivo se vera de esta manera.

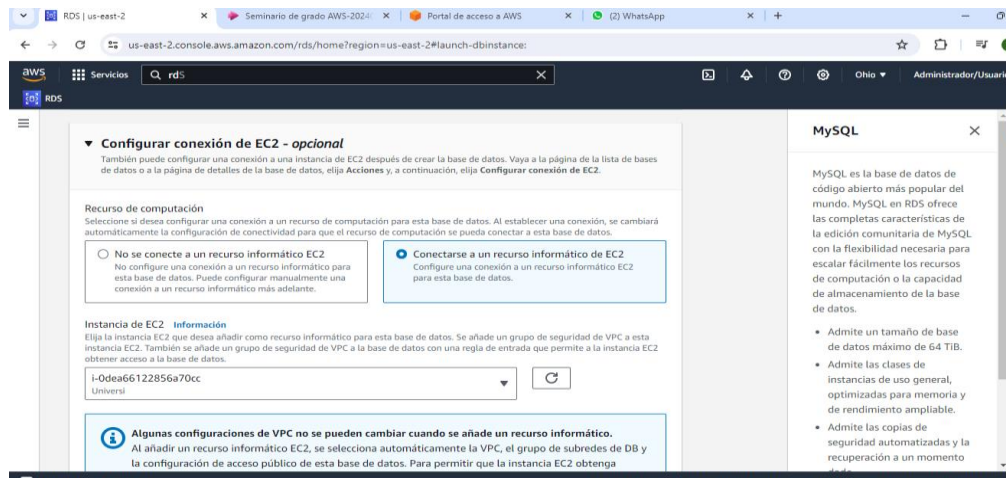
Figura 23 Imagen de bucket



Fuente: Elaboración propia en AWS

El siguiente paso para seguir es crear un RDS y este se conecte con la instancia de ec2 que ya tenemos, y esta esté conectada a la vpc.

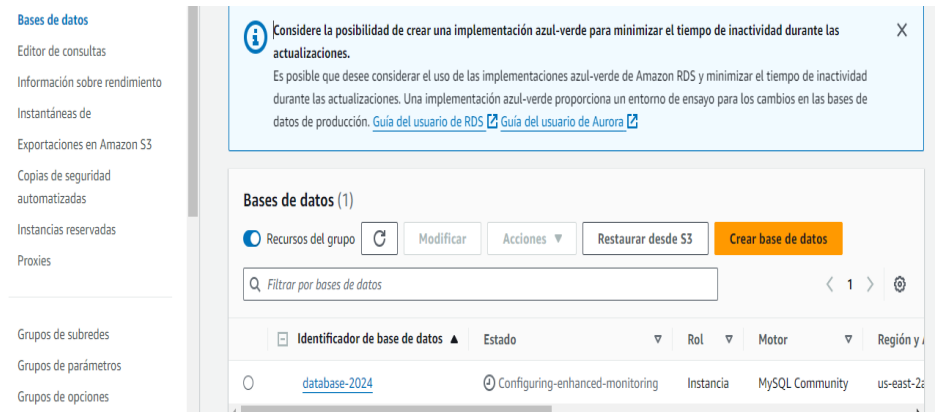
Figura 24 creación del RDS



Fuente: Elaboración propia en AWS

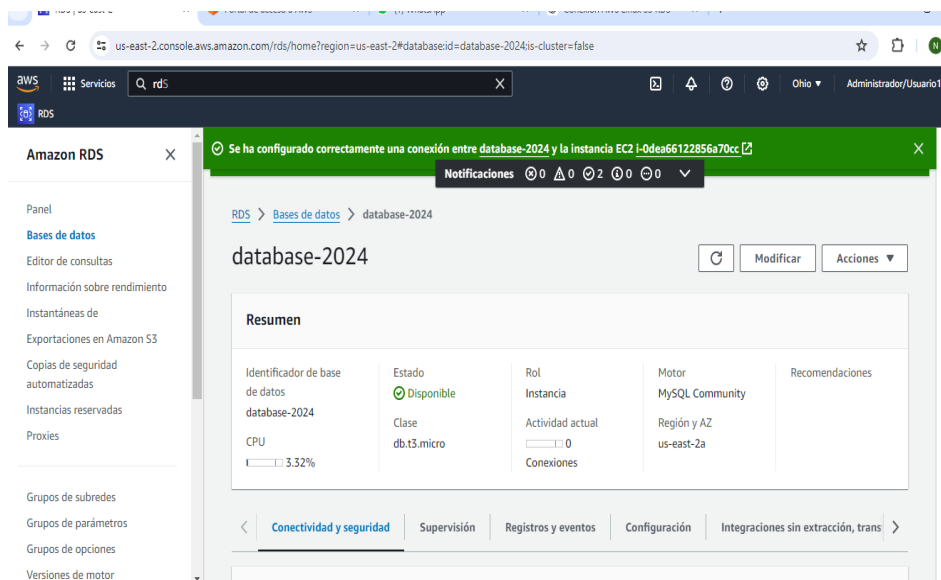
Se crear una base de datos de MySQL, que en este caso se le asigno el nombre de database-2024.

Figura 25 conexión a la base de datos



Fuente: Elaboración propia en AWS

Figura 26 Base de datos creada



Fuente: Elaboración propia en AWS

Una vez creada la base de datos se debe instalar el MySQL en la WSL

Figura 27 Instalación del MySQL

```

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-2.174-460.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-1.1.9998.39-2.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-10-0-10-125 ec2-user]# |

```

Fuente: Elaboración propia en AWS

Figura 28 Imagen 1 de la conexión de la bd

```

[root@ip-10-0-10-125 ec2-user]# sudo dnf install -y mariadb105-server
Last metadata expiration check: 11:22:51 ago on Thu Jun 6 16:34:00 2024.
Dependencies resolved.

```

Package	Architecture	Version	Repository	Size
Installing:				
mariadb105-server	x86_64	3:10.5.23-1.amzn2023.0.1	amazonlinux	11 M
Installing dependencies:				
mariadb-connector-c	x86_64	3.1.13-1.amzn2023.0.3	amazonlinux	196 k
mariadb-connector-c-config	noarch	3.1.13-1.amzn2023.0.3	amazonlinux	9.2 k
mariadb105	x86_64	3:10.5.23-1.amzn2023.0.1	amazonlinux	1.6 M
mariadb105-common	x86_64	3:10.5.23-1.amzn2023.0.1	amazonlinux	30 k
mariadb105-errmsg	x86_64	3:10.5.23-1.amzn2023.0.1	amazonlinux	214 k
mysql-selinux	noarch	1.0.4-2.amzn2023.0.3	amazonlinux	36 k
perl-B	x86_64	1.80-477.amzn2023.0.6	amazonlinux	179 k

Fuente: Elaboración propia en AWS

Se procede a conectarse a la base de datos.

Figura 29 Imagen 2 de la conexión de la bd

```

root@ip-10-0-10-125/home/e  x  +  v  -  o  x
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:          38
Current database:
Current user:           admin@10.0.10.125
SSL:                    Not in use
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server:                 MySQL
Server version:         8.0.35 Source distribution
Protocol version:       10
Connection:             database-2024.c74cke0gvdr.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 2 min 41 sec

Threads: 3  Questions: 13621  Slow queries: 0  Opens: 354  Flush tables: 3  Open tables: 234  Queries per second avg: 3.621

MySQL [(none)]> |

```

Fuente: Elaboración propia en AWS

Figura 30 Imagen 3 de la conexión de la bd

```

Protocol version: 10
Connection: database-2024.c74cke08gvdr.us-east-2.rds.amazonaws.com via TCP/IP
Server characterset: utf8mb4
Db characterset: utf8mb4
Client characterset: utf8mb3
Conn. characterset: utf8mb3
TCP port: 3306
Uptime: 1 hour 2 min 41 sec

Threads: 3 Questions: 13621 Slow queries: 0 Opens: 354 Flush tables: 3 Open tables: 234 Queries per second avg: 3.621
-----
MySQL [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.004 sec)

MySQL [(none)]> |

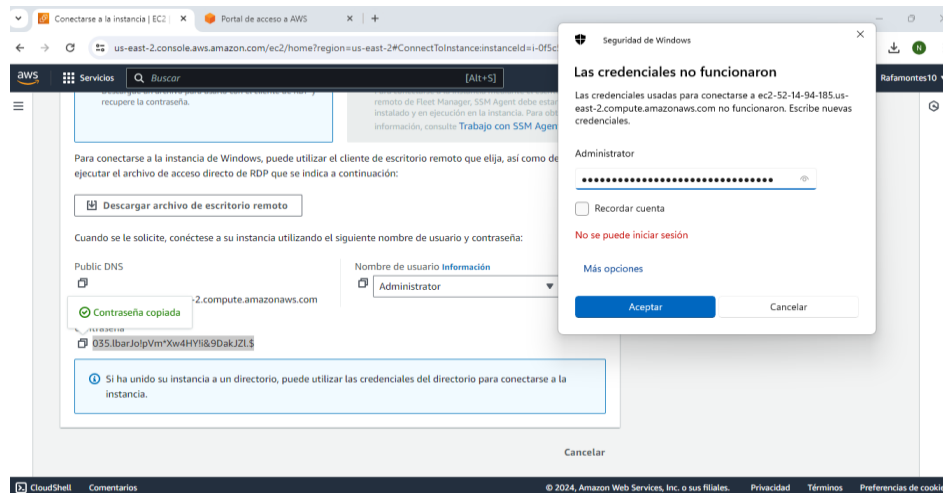
```

Fuente: Elaboración propia en AWS

4to punto

Procedemos a obtener la clave y descarga el archivo para conectarnos al servidor.

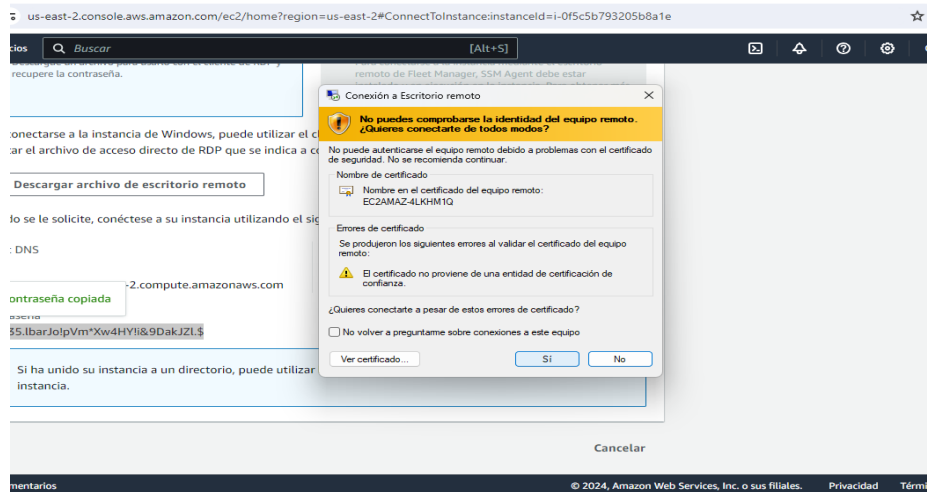
Figura 31 creación de la instancia



Fuente: Elaboración propia en AWS

Luego procedemos a entrar al Windows server.

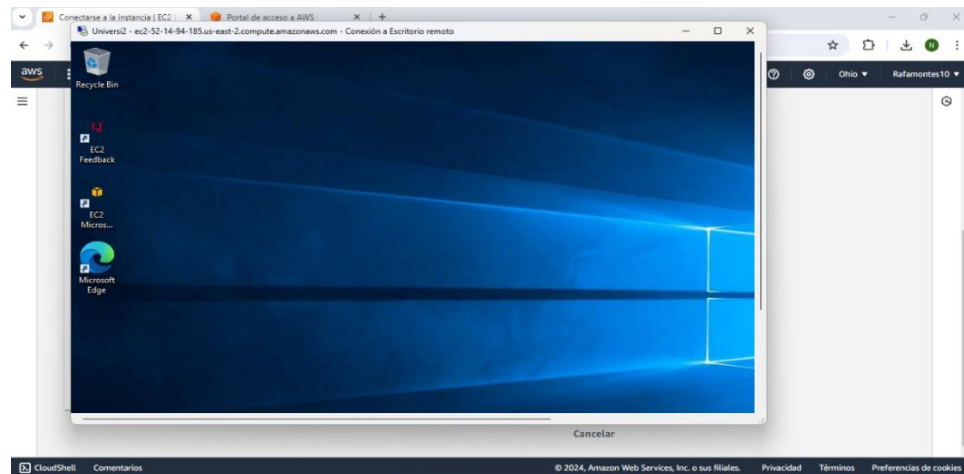
Figura 32 configuración de contraseña



Fuente: Elaboración propia en AWS

Este es Windows server 1.

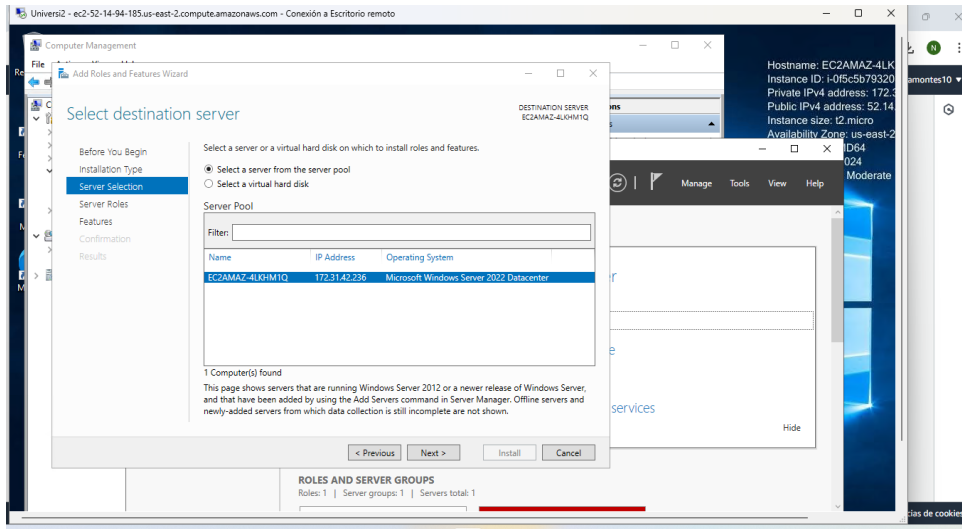
Figura 33 Ventana de Windows server 1



Fuente: Elaboración propia en AWS

Se gestionan los roles y funciones para garantizar la seguridad y la administración eficiente de los recursos en la nube, para optimizando el rendimiento, la escalabilidad y la disponibilidad de las aplicaciones.

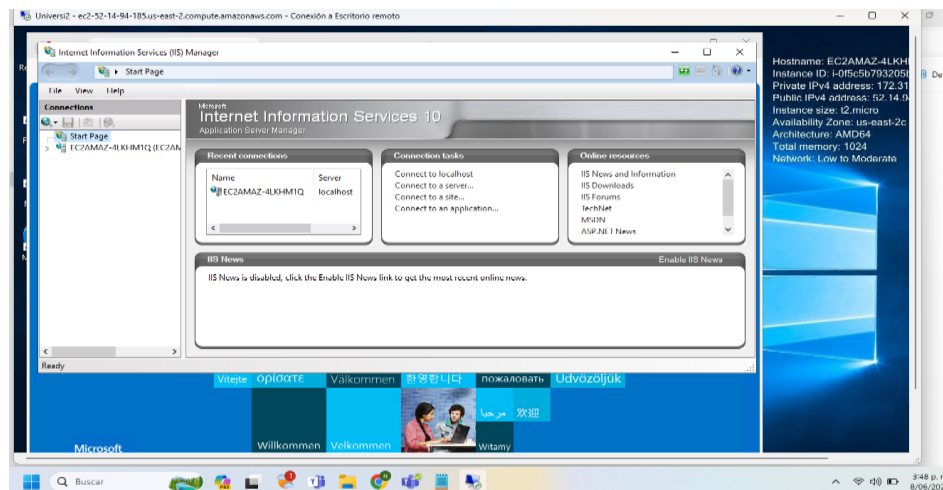
Figura 34 Configuración de roles



Fuente: Elaboración propia en AWS

Se gestionan los roles y funciones para garantizar la seguridad y la administración eficiente de los recursos en la nube, para optimizando el rendimiento, la escalabilidad y la disponibilidad de las aplicaciones

Figura 35 configuración del servidor web en el EC2

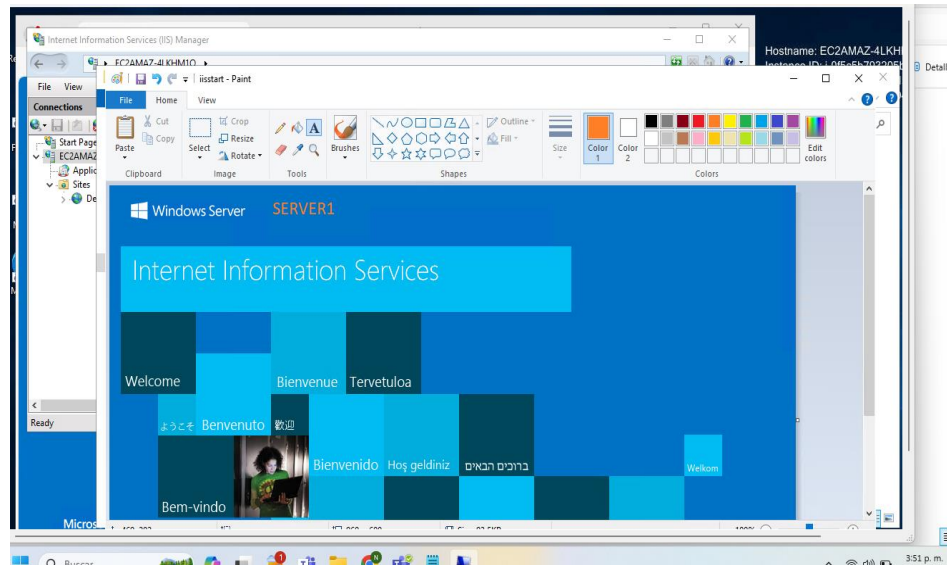


Fuente: Elaboración propia en AWS

Se hace la configuración del servidor web en el EC2 para entregar el contenido web a los usuarios a través de Internet.

Esta es la página de prueba la cual asignamos como servidor 1.

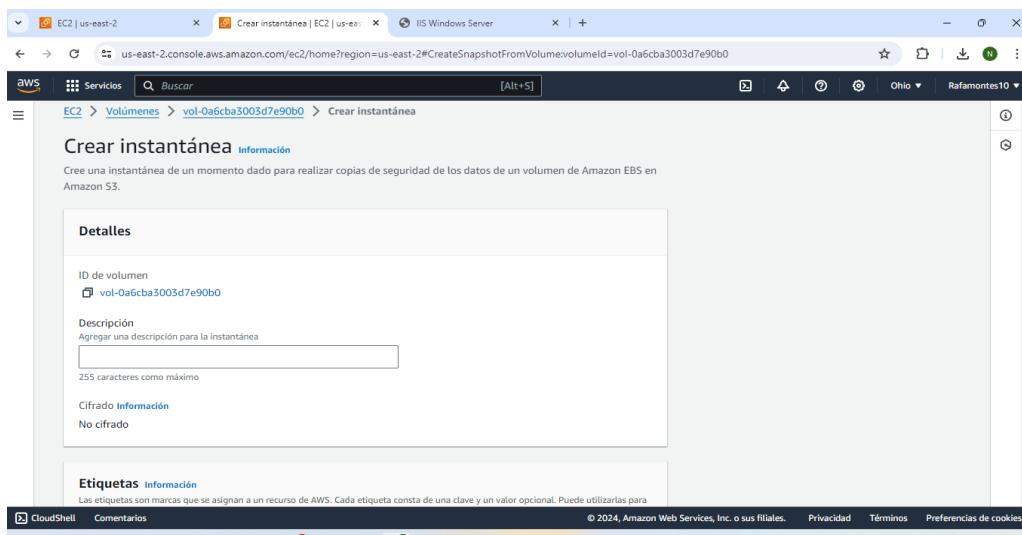
Figura 36 asignación del servidor 1.



Fuente: Elaboración propia en AWS

Creamos una nueva instancia que sería servidor 2.

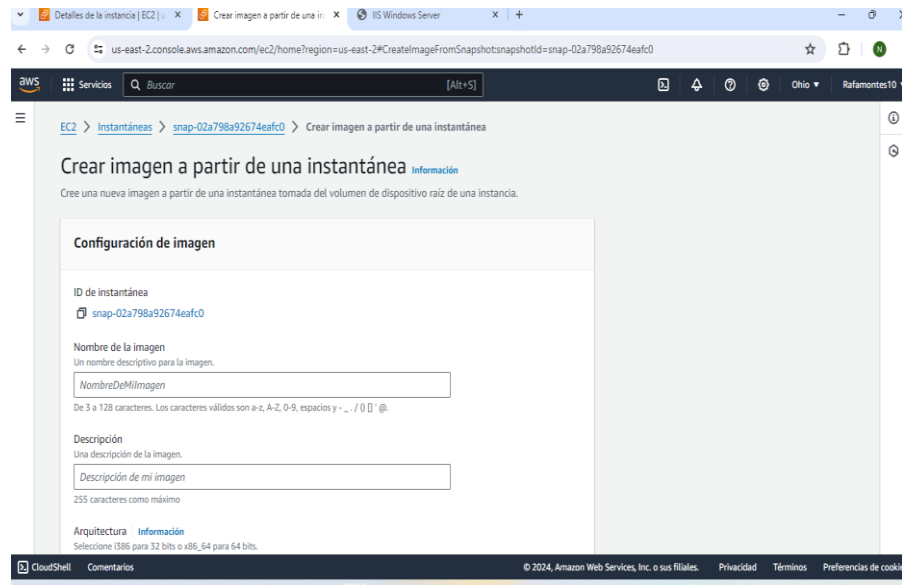
Figura 37 Instancia de servidor 2



Fuente: Elaboración propia en AWS

Creamos una imagen a partir de la instancia y le colocamos servidor 2.

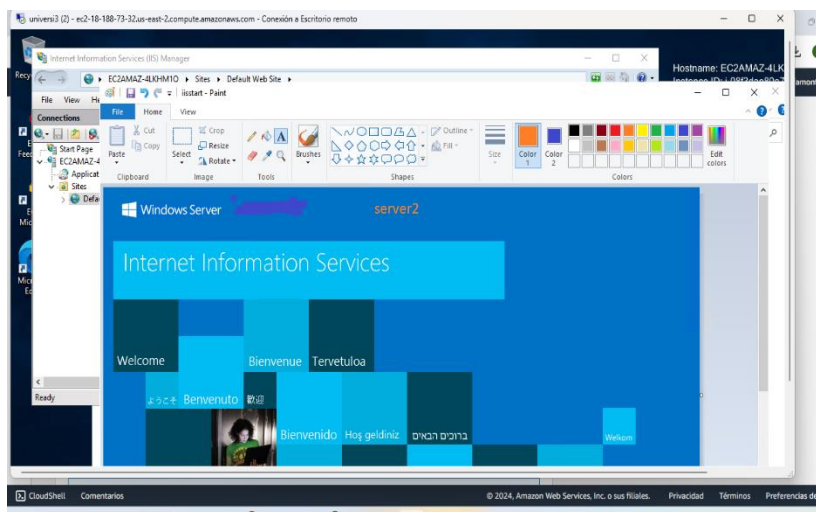
Figura 38 Creación de la imagen del servidor 2



Fuente: Elaboración propia en AWS

Esta es la página de prueba del servidor 2.

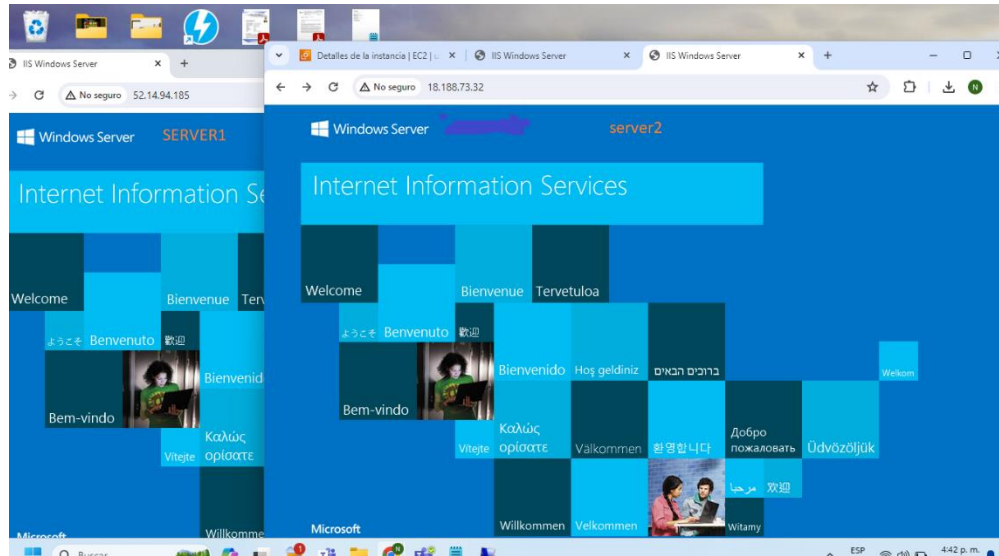
Figura 39 Imagen del servidor 2



Fuente: Elaboración propia en AWS

Mostramos ambos servidores funcionando con sus diferentes ip diferentes.

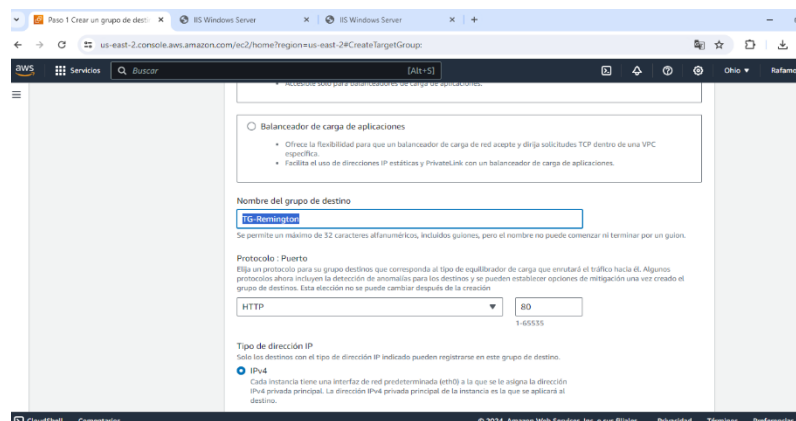
Figura 40 Imagen de ambos servidores



Fuente: Elaboración propia en AWS

Creamos el balanceador de la aplicación y asignamos un nombre en el grupo de destino para ayudará a identificar y administrar fácilmente el grupo, seleccionamos el tipo de balanceador para carga de red, seleccionamos el tipo de dirección ipv4, para distribuir las peticiones de los usuarios.

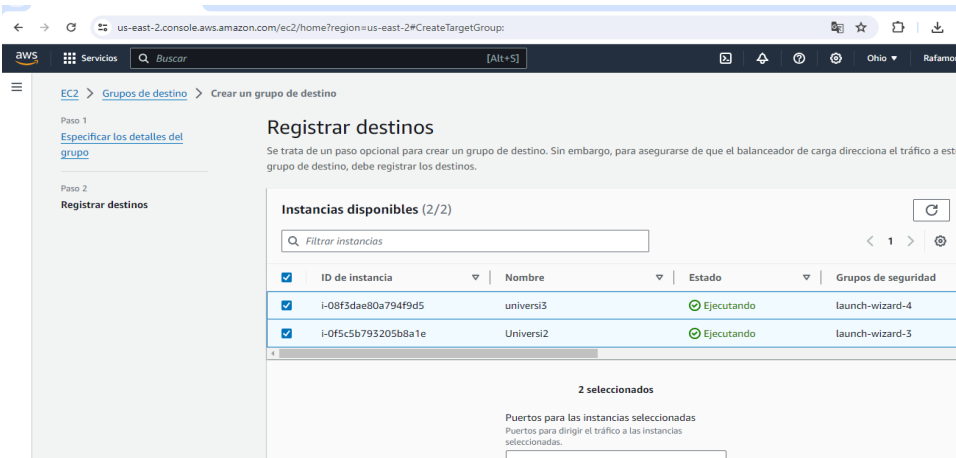
Figura 41 Creación del balaceador



Fuente: Elaboración propia en AWS

Estas son las instancias de usuarios disponibles creados que apuntan al balanceador de carga; donde se encuentra: el id de destino, la dirección ip y el estado de destino entre otros.

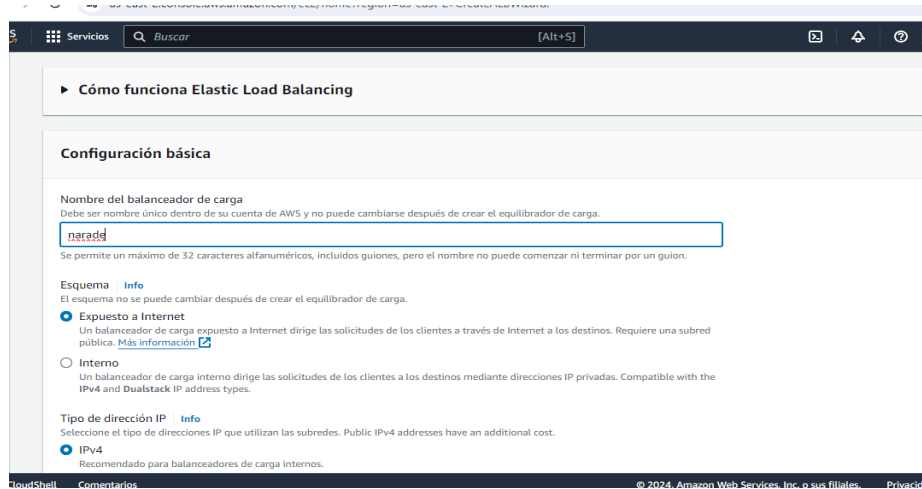
Figura 42 Nombramiento del balanceador



Fuente: Elaboración propia en AWS

Colocamos el nombre del balanceo externo, configuramos dirección IP, el VPC y la zona para permitir el tráfico y la alta disponibilidad desde Internet, permitiendo que los usuarios accedan a los recursos que están detrás del balanceador de carga de las instancias de EC2 con una única dirección para ambos servidores.

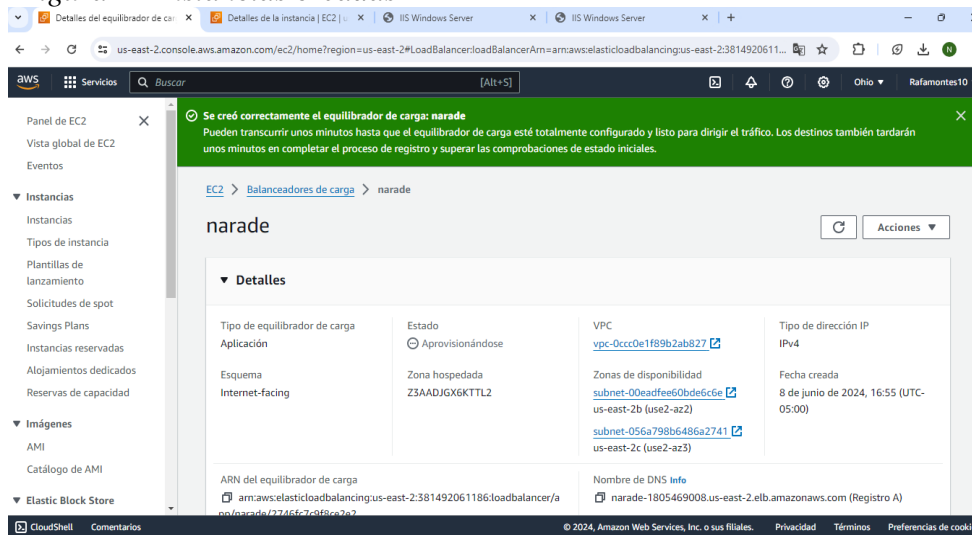
Figura 43 configuración del apuntador



Fuente: Elaboración propia en AWS

Nombre del balanceo narade para apuntar al balanceador con el que se ingresaría directamente a ambos servidores.

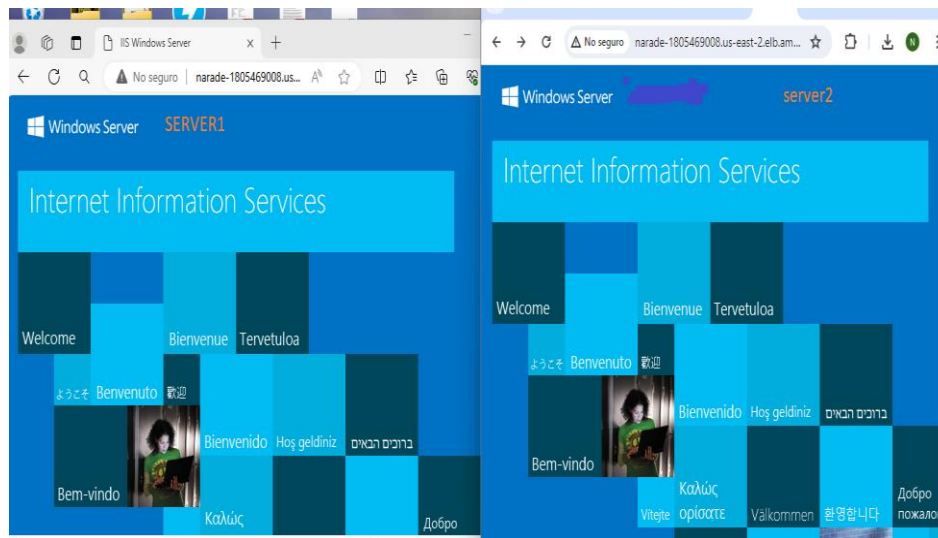
Figura 44 Instancias creadas



Fuente: Elaboración propia en AWS

Se realiza el ingreso a los dos servidores en internet, en diferentes navegadores con el mismo nombre del balanceador.

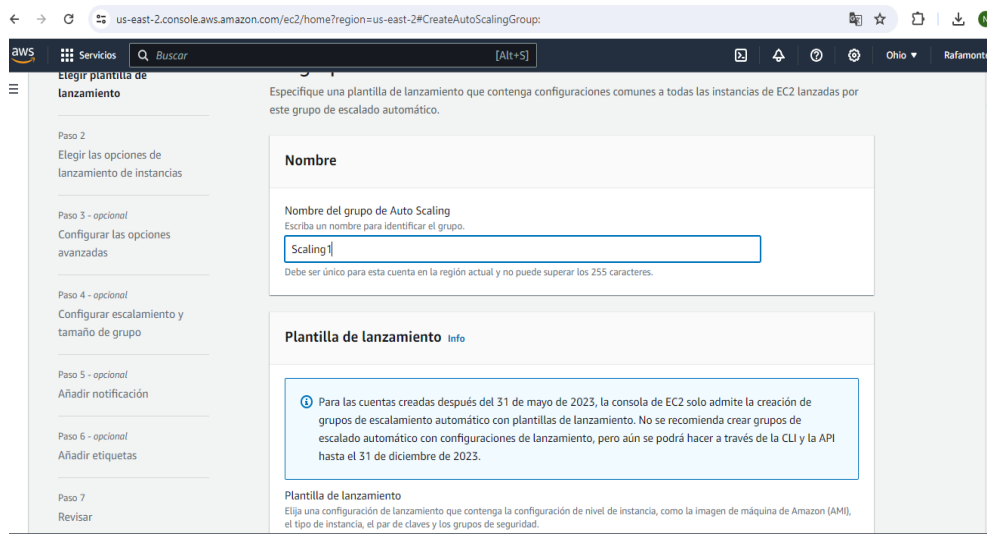
Figura 45 Ingreso con el mismo nombre de balaceo



Fuente: Elaboración propia en AWS

Configuramos el auto escalado del grupo para que se pueda gestionar instancias automáticamente.

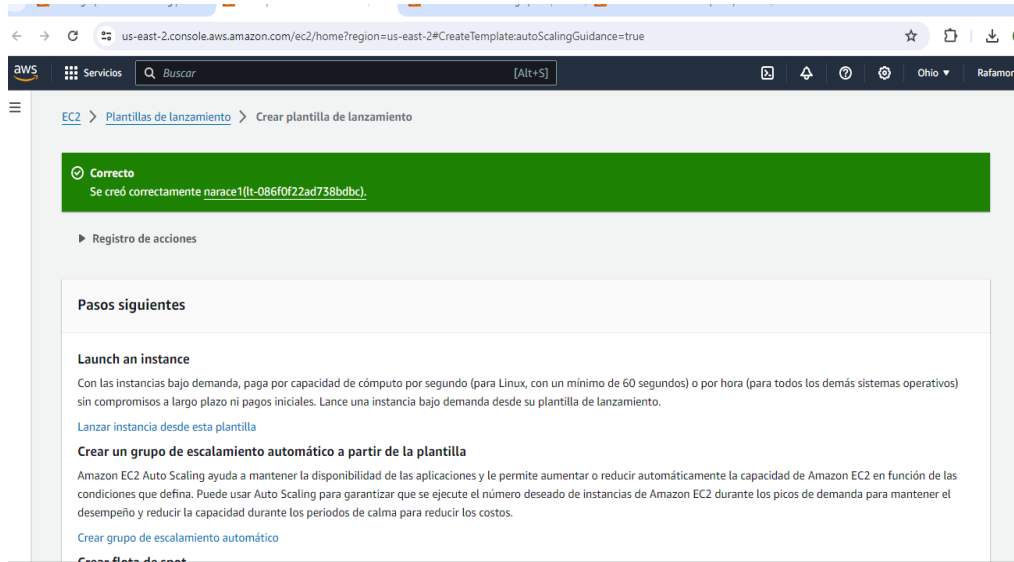
Figura 46 Configuración de auto escalado.



Fuente: Elaboración propia en AWS

Abrimos las instancias para que se creen varias instancias.

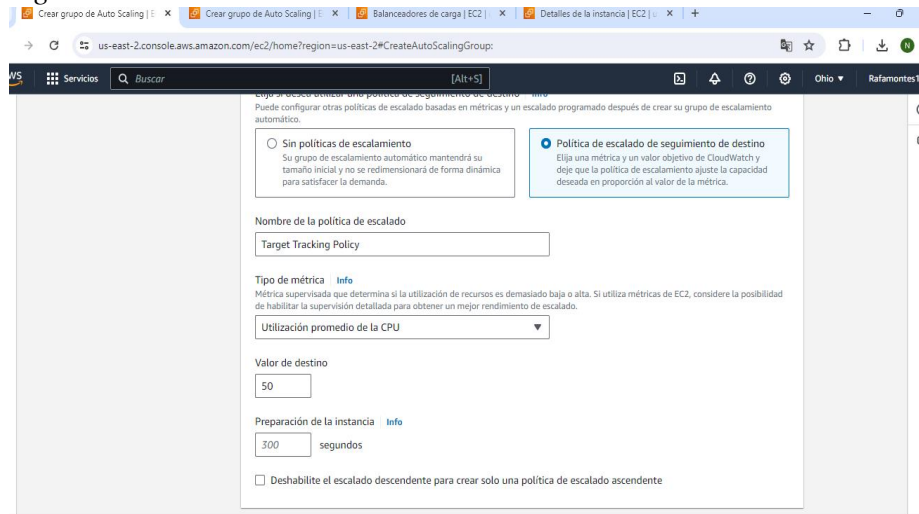
Figura 47 Imagen de la creación del balancidor externo



Fuente: Elaboración propia en AWS

Configura entonces la política de escalado de seguimiento de destino para que se ajuste automáticamente de las instancias con una cantidad mínima y una cantidad máxima, para controlar el rendimiento mientras se minimizan los costos.

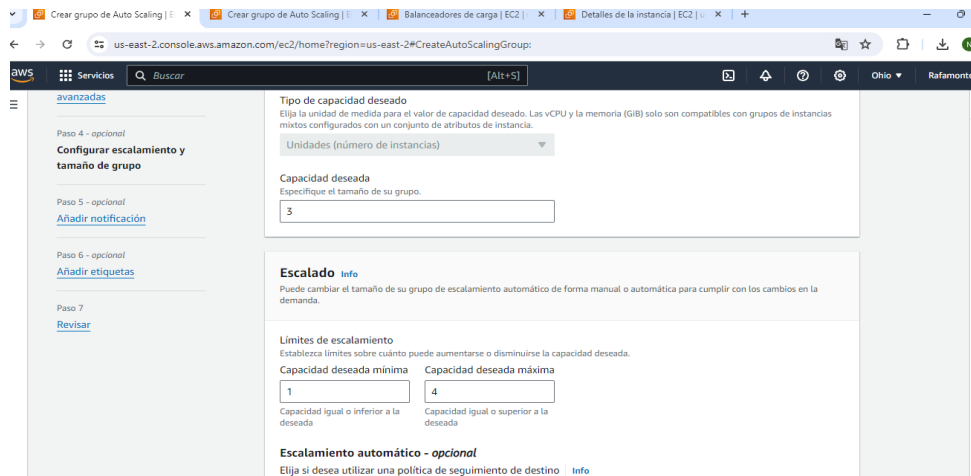
Figura 48 Políticas de auto escalado



Fuente: Elaboración propia en AWS

Se designa el límite de escalamiento, con una capacidad deseada de 3 instancias en el grupo, con mínimo 1 y máximo 4 instancias.

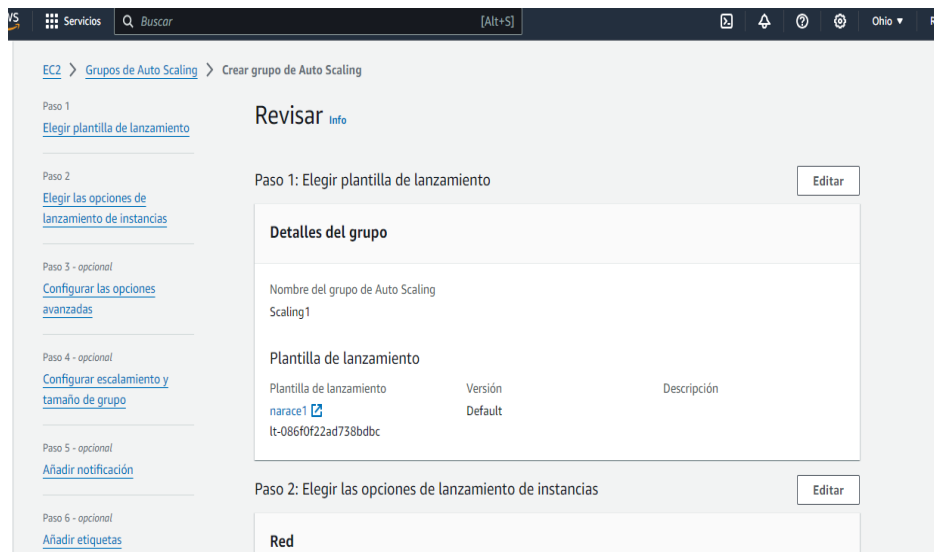
Figura 49 Asignación de límites



Fuente: Elaboración propia en AWS

Configuramos los detalles del grupo para el auto escalado.

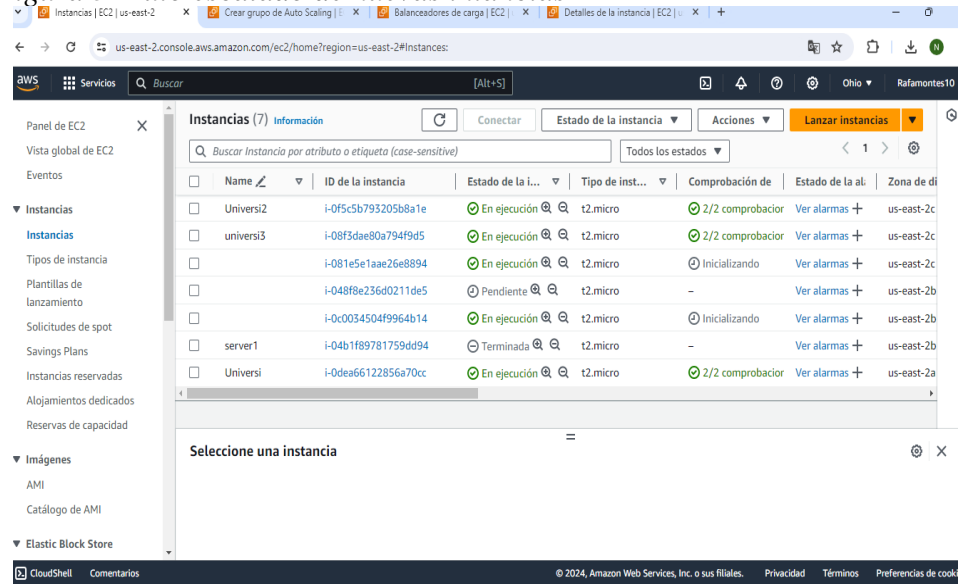
Figura 50 configuración de detalles de grupo



Fuente: Elaboración propia en AWS

Como vemos, el auto escalado crea 3 nuevas instancias.

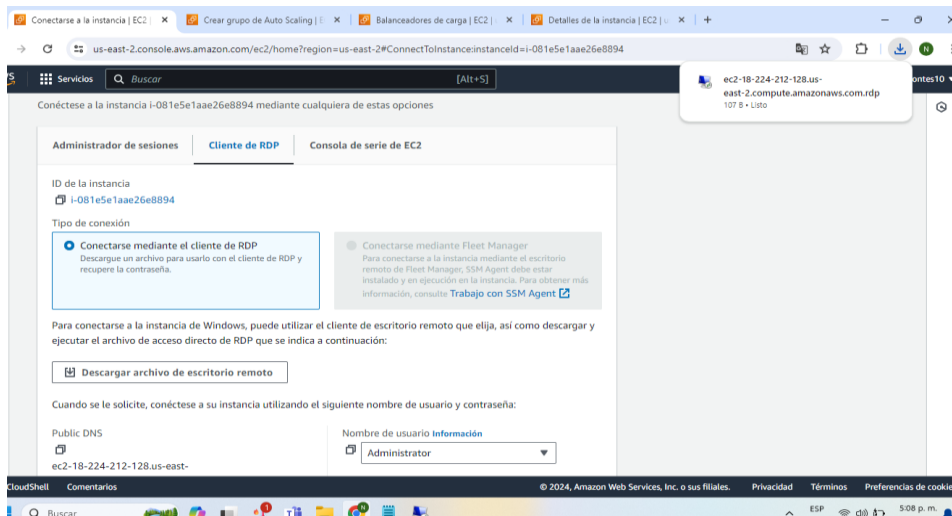
Figura 51 Auto escalado de nuevas instancias



Fuente: Elaboración propia en AWS

Se configura el funcionamiento de las instancias con el RDP, se descarga el archivo para poder ingresar a la máquina virtual.

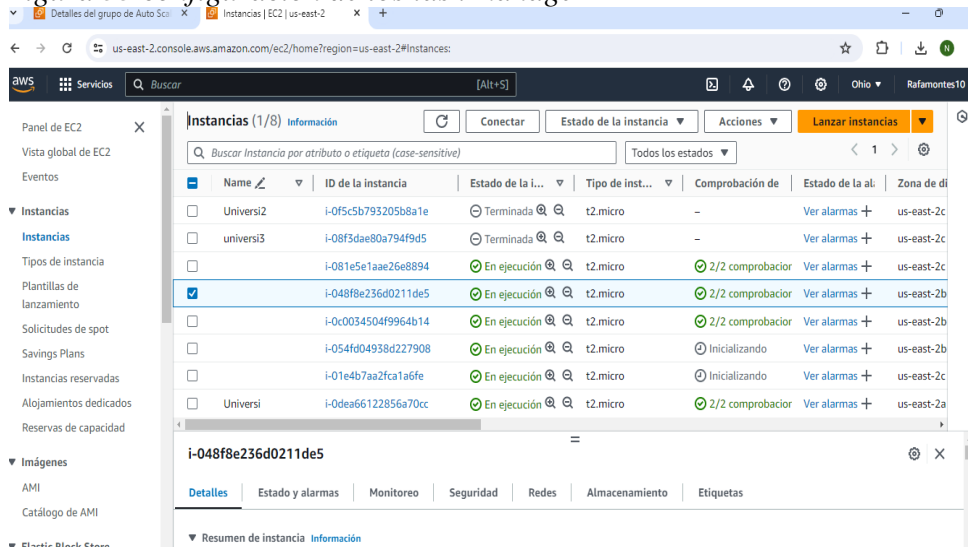
Figura 52 configuración del RDP



Fuente: Elaboración propia en AWS

Aquí se muestra las instancias que han dejado de funcionar y las que se están creando automáticamente.

Figura 53 configuración de los task manager



Fuente: Elaboración propia en AWS

Los task manager de las instancias, son diversas herramientas y técnicas para controlar, monitorear y optimizar la ejecución de los procesos y aplicaciones que se ejecutan en las instancias. Como podemos ver en la imagen, la CPU está siendo utilizada en un 100% en la instancia seleccionada que se encuentra en ejecución.

Figura 54 Imagen de uso de EPU

The image shows a remote desktop connection to an Amazon EC2 instance. On the left, the Windows Task Manager is open, displaying system performance metrics. On the right, the AWS Management Console is visible, showing the state of the EC2 instance.

Windows Task Manager Performance:

Component	CPU	Memory
System	100%	80%
Apps (2)		
Microsoft Edge (13)	97.4%	134.6 MB
Task Manager	1.6%	9.2 MB
Background processes (19)		
amazon-ssm-agent	0%	3.7 MB
Antimalware Service Executable	0%	63.2 MB
CTF Loader	0%	1.4 MB
Host Process for Windows Tasks	0%	0.6 MB
IIS Worker Process	0%	1.2 MB
Microsoft Distributed Transaction...	0%	0 MB
Microsoft Network Realtime Ins...	0%	1.3 MB
RDP Clipboard Monitor	0%	0.9 MB

AWS Management Console Instance List:

Tipo de inst...	Comprobación de	Estado de la al.	Zona de di
t2.micro	-	Ver alarmas +	us-east-2c
t2.micro	-	Ver alarmas +	us-east-2c
t2.micro	2/2 comprobador	Ver alarmas +	us-east-2c
t2.micro	2/2 comprobador	Ver alarmas +	us-east-2b
t2.micro	2/2 comprobador	Ver alarmas +	us-east-2b
t2.micro	Inicializando	Ver alarmas +	us-east-2b
t2.micro	Inicializando	Ver alarmas +	us-east-2c
t2.micro	2/2 comprobador	Ver alarmas +	us-east-2a

Fuente: Elaboración propia en AWS

Índice de tablas

Tabla 1 Computación.-----	11
Figura 1 Instancias en AWS-----	12
Figura 2 Las reglas de internet. -----	12
Figura 3 Creación de contenedores-----	13
Figura 4 Instalación de apache-----	13
Figura 5 Configuración de apache-----	14
<i>Figura 6 Generación de códigos</i> -----	15
Figura 7 Creación del virtual host-----	15
Figura 8 Virtual host creado-----	15
Figura 9 Direcciones IP -----	16
Figura 10 Imagen de página web tienda.com -----	17
Figura 11 Imagen de página web Comercio.com -----	17
Figura 12 Inicialización de cuenta IAM -----	17
Figura 13 Creación de usuarios. -----	18
Figura 14 configuración de autenticación. -----	18
Figura 15 Configuración de politicas y permisos -----	19
Figura 16 Portal de acceso a AWS-----	20
Figura 17 Creación del bucket -----	20
Figura 18 Imagen de bucket creado -----	21
Figura 19 Clave de acceso-----	21
<i>Figura 20 Archivo</i> -----	22
Figura 21 Exportación de clave de acceso en WSL-----	22
Figura 22 configuración por consola-----	22
Figura 23 Imagen de bucket -----	23
Figura 24 creación del RDS -----	23
Figura 25 conexión a la base de datos -----	24
Figura 26 Base de datos creada -----	24
Figura 27 Instalación del MySQL-----	24
Figura 28 Imagen 1 de la conexión de la bd -----	25
Figura 29 Imagen 2 de la conexión de la bd -----	25
Figura 30 Imagen 3 de la conexión de la bd -----	25
Figura 31 creación de la instancia -----	26
Figura 32 configuración de contraseña-----	26
Figura 33 Ventana de Windows server 1 -----	27
Figura 34 Configuración de roles-----	27
Figura 35 configuración del servidor web en el EC2-----	28
Figura 36 asignación del servidor 1. -----	29
Figura 37 Instancia de servidor 2-----	29
Figura 38 Creación de la imagen del servidor 2-----	30
<i>Figura 39 Imagen del servidor 2</i> -----	30
Figura 40 Imagen de ambos servidores -----	31

Figura 41 Creación del balaceador -----	31
Figura 42 Nombramiento del balanceador -----	32
Figura 43 configuración del apuntador-----	32
Figura 44 Instancias creadas-----	33
Figura 45 Ingreso con el mismo nombre de balaceo -----	33
Figura 46 Configuración de auto escalado. -----	34
Figura 47 Imagen de la creación del balancidor externo -----	35
Figura 48 Políticas de auto escalado -----	35
Figura 49 Asignación de límites-----	36
Figura 50 configuración de detalles de grupo -----	36
Figura 51 Auto escalado de nuevas intancias-----	37
Figura 52 configuración del RDP-----	37
Figura 53 configuración de los task manager -----	38
Figura 54 Imagen de uso de EPU-----	38

Conclusiones

El seminario de Amazon Web Services (AWS) proporcionó una visión integral y práctica de cómo aprovechar las capacidades de AWS para implementar y gestionar infraestructuras en la nube de manera eficiente y segura. A través de una serie de sesiones detalladas, los participantes pudieron aprender sobre la configuración de una cuenta de AWS, el uso de contenedores para la ejecución de aplicaciones, y la conexión de instancias Linux a servicios clave como Amazon S3 y RDS MySQL.

La configuración inicial de una cuenta de AWS y la gestión de identidades y accesos (IAM) establecen una base sólida para garantizar la seguridad y la organización adecuada de los recursos en la nube.

La demostración práctica del lanzamiento de una instancia EC2 con Linux y la instalación de AWS CLI permitió a los participantes comprender cómo interactuar con los servicios de AWS y gestionar datos en S3 de forma eficaz. Asimismo, la creación y conexión a una base de datos RDS MySQL desde una instancia EC2 ilustró la facilidad con la que se pueden manejar bases de datos en la nube, garantizando la escalabilidad y seguridad de los datos.

El seminario también abordó la configuración de políticas de Auto Scaling, mostrando cómo AWS puede ajustar automáticamente los recursos en respuesta a las métricas de carga, lo que es crucial para mantener la eficiencia operativa y minimizar costos.

El seminario de AWS no solo proporcionó conocimientos técnicos profundos sino también habilidades prácticas esenciales para implementar y gestionar infraestructuras en la nube. Estos conocimientos capacitan a los profesionales para diseñar, desplegar y mantener aplicaciones robustas y escalables en AWS, facilitando así la transformación digital y la innovación en sus respectivas organizaciones.

Referencias

- (Agulló Valls, A. 26 de Julio de 2021). *Universidad Politécnica de Valencia*.
<https://riunet.upv.es/handle/10251/170172>
- (Arsys en Servicios. 02/07/2020). Breve historia y evolución del Cloud Computing. Arsys.
<https://www.arsys.es/blog/historia-cloud>
- (Bravo Mendoza, A. I. 05 de Octubre de 2023). *Escuela Politécnica Nacional*.
<https://bibdigital.epn.edu.ec/handle/15000/24917>
- (Brito Zhunio, D. M. 2011). *Universidad Israel*.
<https://repositorio.uisrael.edu.ec/handle/47000/140>
- (Campos Campos, C. E. 2021).
<https://repositorio.ulacit.ac.cr/bitstream/handle/20.500.14230/10649/REF-1631831537-2.pdf?sequence=2>
- (Caraves Muñoz, E. &. Febreoo de 2014). *Universidad Autonoma del Estado de Mexico*.
<http://ri.uaemex.mx/handle/20.500.11799/99992>
- (Clarcat. 2023, January 11). Comparativa: AWS vs. Microsoft Azure vs. Google Cloud Platform. clarcat. <https://www.clarcat.com/comparativa-aws-vs-microsoft-azure-vs-google-cloud-platform/>
- (Compara los servicios de AWS y Azure con Google Cloud. n.d.). Google Cloud. <https://cloud.google.com/docs/get-started/aws-azure-gcp-service-comparison?hl=es-419>
- (Esteban Canle Fernández. 05/08/2021). Historia de la computación en la nube. tokioschool. <https://www.tokioschool.com/noticias/historia-computacion-nube/>
- (Enrique Miguel. 24/09/20). Historia y evolución de la nube. blog smartekh .
<https://blog.smartekh.com/historia-y-evoluci%C3%B3n-de-la-nube>
- (Farro Flores, C. A. Mayo de 2017). *Universidad del Pacífico*. Obtenido de
<https://repositorio.up.edu.pe/handle/11354/1832>
- (García Granados, A. 15 de Enero de 2023). *Universitat Oberta de Catalunya*. Obtenido de
<https://openaccess.uoc.edu/handle/10609/147969>
- (Gear Technology CorpTérminos. 2024). ¿Cuál es el origen de la computación en la nube? Gear Plug. <https://www.gearplug.io/blog/cual-es-el-origen-de-la-computacion-en-la-nube>

(Hewlett Packard Enterprise Development LP. 2024). ¿Qué es la virtualización? Hewlett Packard. <https://www.hpe.com/lamerica/es/what-is/virtualization.html>

(Kamlofsky, J. 23 de Diciembre de 2022). *Revista Abierta de Informática Aplicada*. Obtenido de <https://raia.revistasuai.ar/index.php/raia/article/view/2>

(Mainframe Cloud. 22 de febrero de 2024). Origen e historia de la computación en la nube. Mainframe Cloud. <https://mainframecloud.co/origen-computacion-en-la-nube/#:~:text=de%20la%20informaci%C3%B3n,-El%20concepto%20inicial%20de%20la%20nube%20y%20su%20evoluci%C3%B3n,eficientemente%20los%20recursos%20de%20computaci%C3%B3n>.

(Mendoza Ciprian, V. O. 2018). <https://cybertesis.unmsm.edu.pe/item/1ac66513-3ff5-4d29-b421-8d4bd0c6b834>

(Michelle Laverick. 07 abr 2014). Cómo la evolución de la virtualización redefinió los centros de datos eficientes. [computerweekly.es. https://www.computerweekly.com/es/cronica/Como-la-evolucion-de-la-virtualizacion-redifinio-los-centros-de-datos-eficientes](https://www.computerweekly.com/es/cronica/Como-la-evolucion-de-la-virtualizacion-redifinio-los-centros-de-datos-eficientes)

(Pablo Blanco. 16/11/2023). Historia de la nube: desde los años 50 hasta nuestros días. educaopen. <https://www.educaopen.com/digital-lab/blog/software/historia-de-la-nube>

(Pérez López, J. A. 18 de Julio de 2023). *Repositorio Institucional de la Universidad de Alicante*. Obtenido de <https://rua.ua.es/dspace/handle/10045/136214>

(Planta Ulloa, J. P. 2024). *Universidad Técnica de Babahoyo*. Obtenido de <http://dspace.utb.edu.ec/handle/49000/15739>

(Rodríguez Mosquera, S. A. 09 de Noviembre de 2023). *Escuela Politécnica Nacional* . <https://bibdigital.epn.edu.ec/handle/15000/25020>

(Rojas Serrano, D. P. 06 de Diciembre de 2023). *Universidad Santo Tomás*. <https://repository.usta.edu.co/handle/11634/53660>

(Sanabria, M. &. 2022). *Tecnologías claves para la transformación digital en las organizaciones*. Obtenido de https://www.researchgate.net/profile/Rafael-Mendez-Romero/publication/361166456_Tecnologias_clave_para_la_transformacion_digital_en_las_organizaciones/links/62a09482a3fe3e3df866a120/Tecnologias-clave-para-la-transformacion-digital-en-las-organizaciones.pdf