



TRABAJO DE GRADO
Opción Seminario-Diplomado.

Amazon Web Services AWS

Corporación Universitaria Remington
Facultad de ingeniería
Ingeniería de sistemas

Wendy Yasmin Landázuri Angulo
Flor Alexa Meza Cabezas
Edwin Alejandro Sapuyes Garreta
Docente: Juan Pablo Berrio Lopez
Amazon web services AWS
San Andrés de Tumaco
2024

Dedicatoria

Agradezco a Dios por haberme permitido culminar con mi carrera profesional. En especial a Javier Moreno quien fue la primera persona en apoyarme moral y económicamente a inicios de mi carrera siendo la apertura de mi logro hoy, a mi mamá por su apoyo incondicional al igual que mi amiga y psicóloga Denis Rincón gracias a ustedes hoy puedo decir logre mi objetivo. No estuvo en mí la grandeza de no caer pero si de levantarme como triunfadora seguir adelante. Ustedes fueron el mejor ejemplo que pude tener. A mi fiel amiga y compañera Flor Meza de quien también me llevo un grato aprendizaje de que cuando se quiere se puede, al sr Julio Moreno quien jugó un papel muy importante en esta etapa de mi vida, me llevo un enorme agradecimiento por la calidad humana brindándome su ayuda cuando lo necesite, por enseñarme que lo más importante en la vida no tiene precio. A Carlos España por su colaboración, al fondo de comunidades negras quienes apoyaron financiando mis estudios conjuntamente con icetex, a mi familia quienes han creído en mí demostrando de la mejor manera sus críticas constructivas y las palabras de aliento en los momentos que lo necesite.

Sin duda alguna a la universidad REMINGTON y al cuerpo docente. En especial a Luis Alberto Reynel quien se preocupaba en el momento que surgía alguna dificultad, siempre presto para tener los consejos idóneos, quienes contribuyeron en mi formación académica y como persona, al igual que Doris Preciado por su colaboración, paciencia y buenos consejos de cada uno me llevo gratos aprendizajes.

WENDY YASMIN LANDAZURI ANGULO

Agradecimientos

Mi agradecimiento en especial para Dios por permitirme terminar mis estudios, igualmente a mi hijo y para un ser muy especial que está en el cielo quien fue una razón más de motivación. Al docente Juan Pablo Berrio encargado de darnos las pautas en el seminario Amazon Web Services AWS, con ello culminar mi trabajo de grado y mi carrera profesional.

Es una verdadera dicha este gran logro, y las personas que hicieron parte de mi formación académica en cada momento, siendo un apoyo para mi proceso de manera significativa, a mi familia por ser fundamental en el momento que lo necesite, al igual que la señora Nelly Angulo quien siempre estuvo atenta y dispuesta en cuidar de mi hijo mientras estudiaba, agradezco a mi papá fue de gran aprendizaje en mi proceso debido a que gracias a el hice resiliencia y me llene de mucho más valor para salir adelante.

WENDY YASMIN LANDAZURI ANGULO

Tabla de Contenido

	Pág
Resumen.....	5
Marco Conceptual.....	6,7
Marco Contextual.....	8
1. Virtualization y la Computación en la Nube.....	9
1.1 Línea de Tiempo.....	10
Servicios entre AWS, Azure y Gcp.nobredelautor.....	12
Tabla 1.....	12
EBS.....	13
Figura 2-2.....	14-23
Servicio de Base de Datos Relacional.....	24
S3.....	24
Figura 1-30.....	25-58
Conclusiones.....	59
Referencias.....	60

Resumen

Amazon Web Services (AWS) es una plataforma de servicios en la nube ofrecida por Amazon.com. Permite a empresas y desarrolladores ejecutar aplicaciones, almacenar datos y llevar a cabo operaciones informáticas en la nube de forma escalable y segura. Algunos puntos clave sobre AWS son: infraestructura sin necesidad de hardware físico; AWS elimina la necesidad de invertir en servidores físicos y otros componentes de infraestructura. Las empresas pueden aprovechar los recursos de AWS según sus necesidades. Flexibilidad y adaptabilidad; AWS ofrece una amplia gama de servicios, desde almacenamiento hasta procesamiento de datos, aprendizaje automático y más. Las organizaciones pueden adaptar su infraestructura según cambien sus requerimientos. Pago por uso; AWS opera bajo un modelo de pago por uso. Esto significa que las empresas solo pagan por los recursos que realmente utilizan, lo que puede ser más eficiente en términos de costos. Red global de centros de datos; AWS tiene una extensa red de centros de datos distribuidos en todo el mundo. Esto garantiza alta disponibilidad y redundancia para las aplicaciones y servicios alojados en la plataforma.

Palabras clave: WSL; Docker; Servicios de AWS; Computación en la nube; Balanceadores de carga.

Marco Conceptual

En el siguiente trabajo se va a profundizar los diferentes conceptos que surgen en esta investigación con el objetivo de ampliar el conocimiento de AWS

WSL. Es una característica de Windows que permite a los desarrolladores instalar distribuciones de Linux (por ejemplo, Ubuntu, OpenSUSE, Kali, Debian, Arch Linux, etc.) y utilizar comandos de línea, herramientas y equipos. En Windows, no hay modificaciones ni actualizaciones de las máquinas virtuales existentes ni de las configuraciones de arranque dual

DOCKER. Docker es una plataforma abierta que ayuda a los desarrolladores a crear, compartir, ejecutar y verificar aplicaciones mediante contenedores. Los contenedores permiten empaquetar una aplicación y sus dependencias juntas, lo que garantiza un comportamiento coherente en diferentes entornos.

SERVICIOS DE AWS. Amazon Web Services (AWS) es una plataforma de computación en la nube líder a nivel mundial. Ofrece una amplia gama de servicios para ayudarte a crear aplicaciones sofisticadas y escalables. Algunos de los servicios más destacados de AWS incluyen:

Amazon EC2: Servidores virtuales en la nube que te permiten ejecutar aplicaciones y servicios.

Amazon S3: Almacenamiento de objetos diseñado para recuperar cualquier volumen de datos desde cualquier ubicación.

Amazon DynamoDB: Base de datos NoSQL administrada.

Amazon CloudFront: Red de entrega de contenido global.

Amazon Lambda: Servicio de cómputo sin servidor que ejecuta código en respuesta a eventos.

Computación en la nube. La computación en la nube es el acceso a demanda, a través de Internet, a recursos informáticos, como servidores (físicos o virtuales), almacenamiento de datos, herramientas de desarrollo y capacidades de creación de redes. Estos recursos

COMPUTACIÓN EN LA NUBE. La computación en la nube es el acceso a demanda, a través de Internet, a recursos informáticos, como servidores (físicos o virtuales), almacenamiento de datos, herramientas de desarrollo y capacidades de creación de redes. Estos recursos están alojados en centros de datos remotos administrados por proveedores de servicios en la nube (CSP). Algunos beneficios clave de la computación en la nube son: **Reducción de costos de TI:** Permite evitar la inversión en infraestructura local y simplifica la administración.

Agilidad y tiempo de creación de valor: Las organizaciones pueden comenzar a usar aplicaciones empresariales en minutos en lugar de semanas o meses.

Escalabilidad más fácil y rentable: La nube ofrece elasticidad, permitiendo ajustar la capacidad según las necesidades.

BALANCEADOR DE CARGA. Un balanceador de carga distribuye el tráfico entre varios servidores para evitar la saturación de uno solo.

Ayuda a mantener altos niveles de servicio al asegurar que las peticiones no se concentren en un único servidor.

Existen diferentes tipos de balanceadores de carga, como los Application Load Balancers, Network Load Balancers y Classic Load Balancers1.

Marco Contextual

Es importante resaltar el contexto donde fue realizado el informe técnico de este trabajo. Realizado en San Andrés De Tumaco, por estudiantes de la Corporación Universitaria Remington. Amazon Web Services (AWS) es una plataforma líder en servicios de computación en la nube que fue lanzada por Amazon.com en 2006. Desde entonces, ha experimentado un crecimiento significativo y se ha convertido en un referente en la industria de la tecnología de la información (TI). En términos de escala y flexibilidad, AWS destaca por su capacidad para ajustar rápidamente los recursos de acuerdo con las necesidades cambiantes de los usuarios. Esta característica permite a las organizaciones adaptarse a la demanda del Mercado y optimizar sus operaciones de manera eficiente. La gama de servicios ofrecidos por AWS es amplia y va desde servicios básicos de infraestructura, como computación, almacenamiento y redes, hasta soluciones avanzadas de inteligencia artificial, analítica de datos y seguridad. Esto brinda a los usuarios la posibilidad de construir y gestionar aplicaciones de forma más efectiva y con mayor agilidad. AWS opera una red global de centros de datos distribuidos estratégicamente en diferentes regiones del mundo. Esta arquitectura permite a los clientes desplegar sus aplicaciones en ubicaciones cercanas a sus usuarios finales, lo que reduce la latencia y garantiza la conformidad con las regulaciones locales. La constante innovación de AWS se refleja en su lanzamiento continuo de nuevos servicios y funcionalidades para satisfacer las demandas cambiantes del mercado. Su evolución desde su lanzamiento en

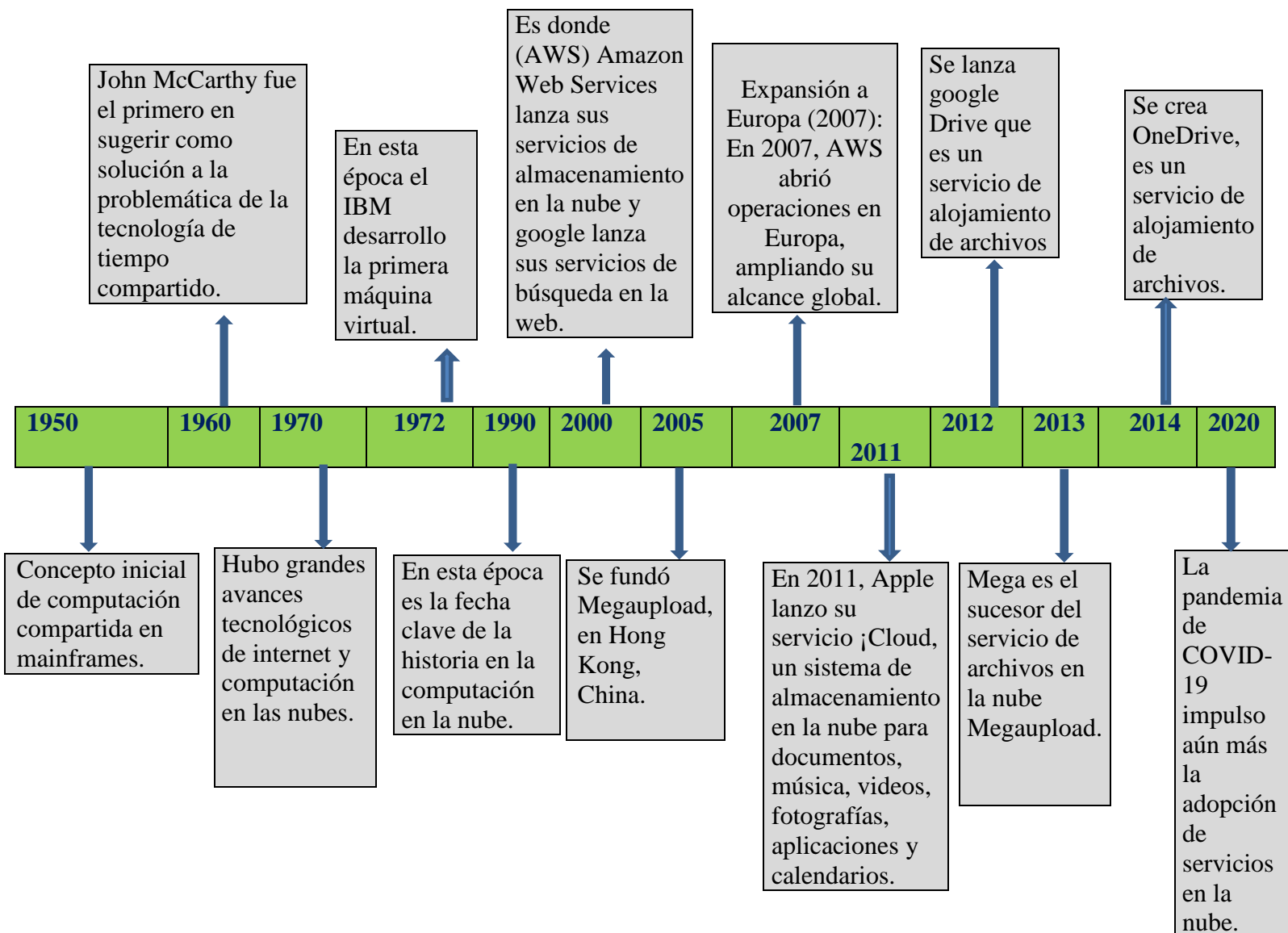
2006, su enfoque en escalabilidad y flexibilidad, la diversidad de servicios ofrecidos, su infraestructura global de centros de datos y su compromiso con la innovación constante.

1. Virtualización y la Computación en la Nube

Los orígenes de la virtualización se remota a la década de 1960, con la creación de la primera máquina virtual por IBM, llamada CP-40. Sin embargo, la virtualización comenzó a ganar popularidad a principios de la década de 2000 con empresas como VMware desarrollando tecnologías de virtualización x86.

En cuanto a la computación en la nube, sus orígenes se pueden rastrear hasta la década de 1960, cuando los grandes mainframes eran compartidos por múltiples usuarios. En los años 60 las empresas vieron la necesidad de consultar grandes cantidades de información desde distintos lugares de acceso, pero, recurrir a infraestructura era muy costoso. Entonces, Jhon McCarthy fue el primero en surgir como solución a este problema. Explicaba que podría venderse como un servicio público. Sin embargo, la noción moderna de la computación en la nube comenzó a surgir a principios de la década de 2000 con servicios como Amazon Web Services (AWS), que permitieron a las empresas acceder a recursos informáticos a baja demanda a través de internet.

1.1 Línea de Tiempo de Orígenes de la Virtualización y computación en la Nube



1.2 Servicios entre AWS, Azure y Gcp.nobredelautor

Tabla 1.

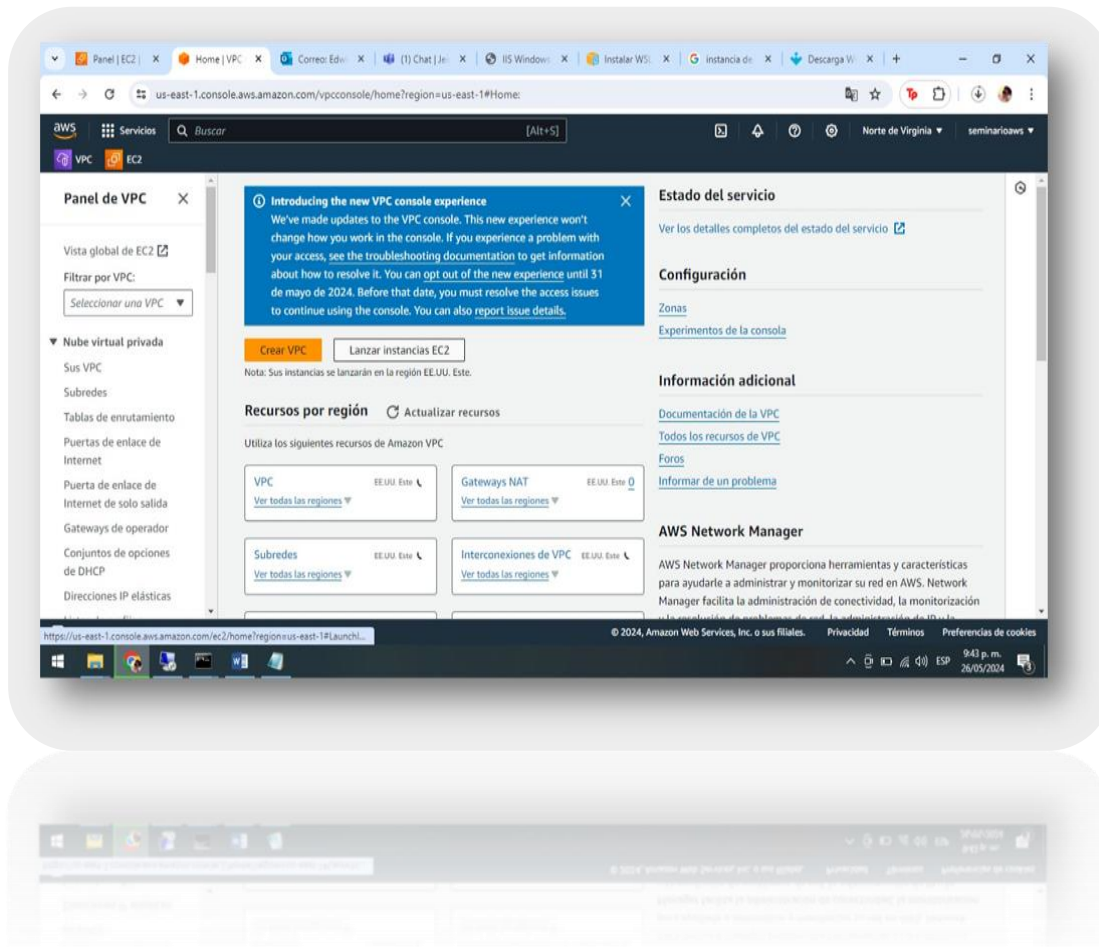
	AWS	Azure	Gcp
Computación	Amazon Elastic Compute Cloud (EC2)	Azure Virtual Machines	Google Compute Engine
Almacenamiento	Amazon Simple Storage Service (s3)	Azure Blob Storage	Google Cloud Storage (GCS)
Base de datos	Amazon Relational Database Service (RDS)	Azure SQL Database	Google Cloud SQL
Redes	Amazon Virtual Private Cloud (VPC)	Azure Virtual Network	Google Virtual Private Cloud (VPC)
Servicios de servidor sin estado	AWS Lambda	Azure Functions	Google Cloud Functions
Servicios de inteligencia artificial y aprendizaje automático	Amazon SageMaker	Azure Machine Learning	Google Cloud AI Platform
Servicios de contenedores	Amazon Elastic Container Service (ECS) / Amazon Elastic Kubernetes Service (EKS)	Azure Kubernetes Service (AKS)	Google Kubernetes Engine (GKE)
Servicios de análisis de datos	Amazon Redshift	Azure Synapse Analytics	Google BigQuery

2. EBS

Amazon Elastic Block Store (EBS) es un servicio de almacenamiento en bloque persistente proporcionado por Amazon Web Services (AWS).

Figura 1

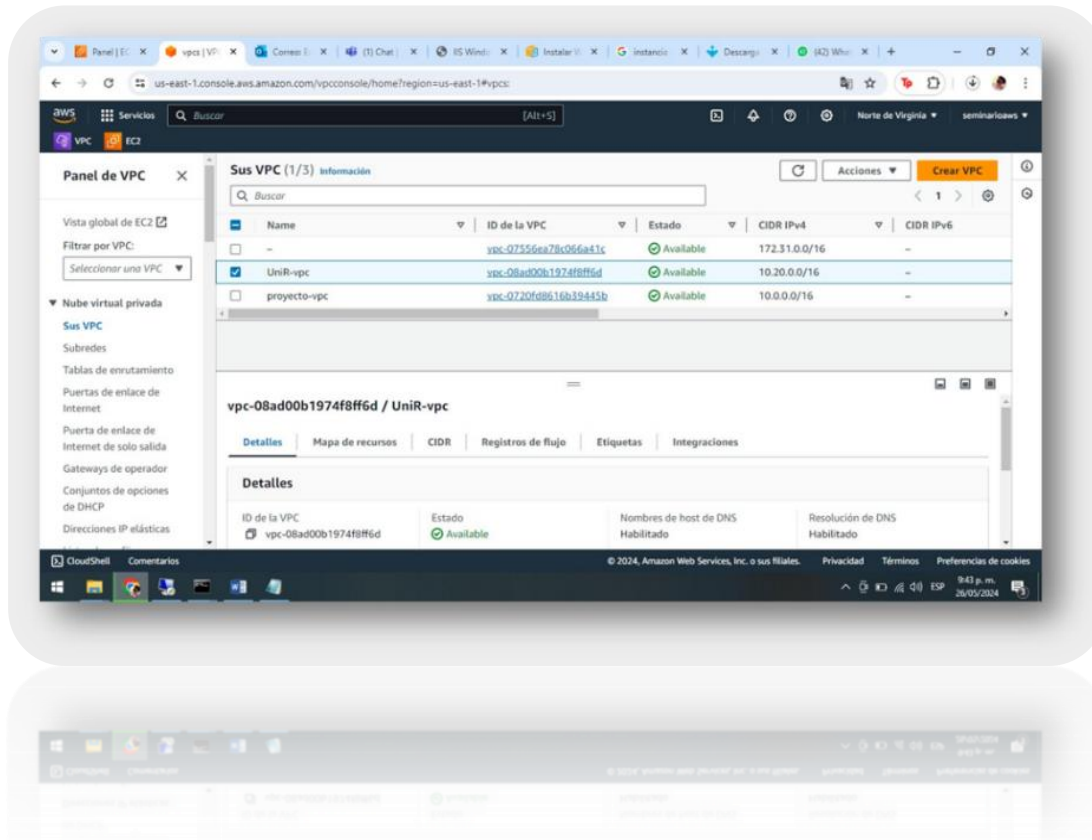
Creación del VPC



Nota.En la imagen anterior vemos como se realiza la creación del VPC.Wendy L, Flor M y Alejandro S, (2024).

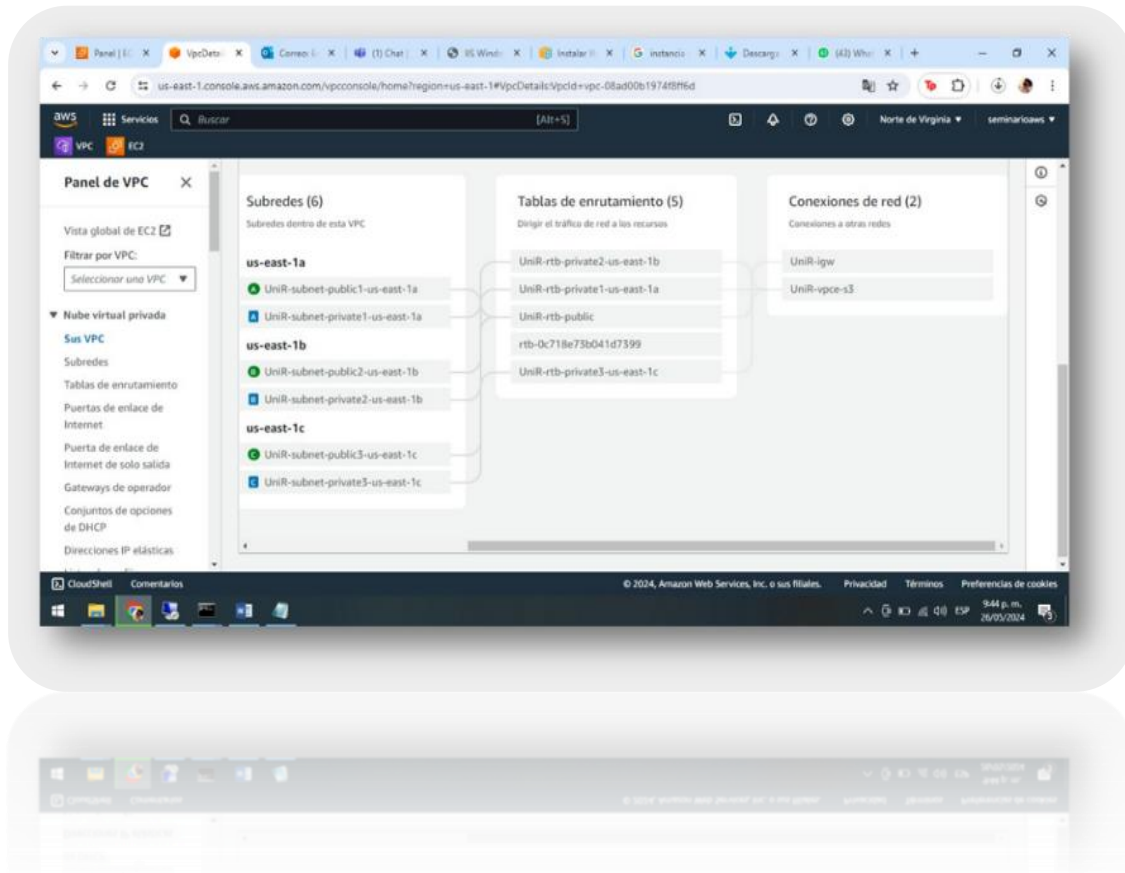
Figura 2

Nombramiento al VPC



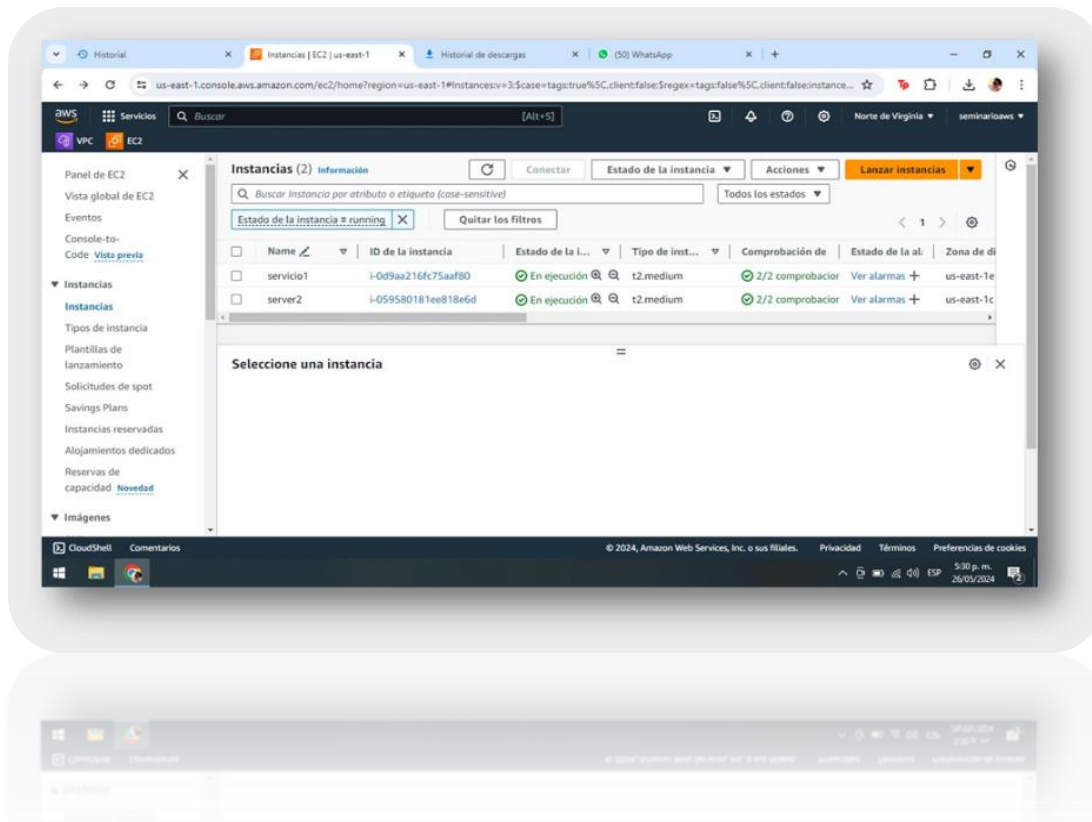
Nota. En la anterior imagen se realiza el nombramiento como uni R al VPC. Wendy L, Flor M y Alejandro S, (2024).

Figura 3
Acceso al VPC



Nota. En la anterior imagen se accede al vpc y se agregan las sub redes tanto públicas como privadas. Wendy L, Flor M y Alejandro S, (2024).

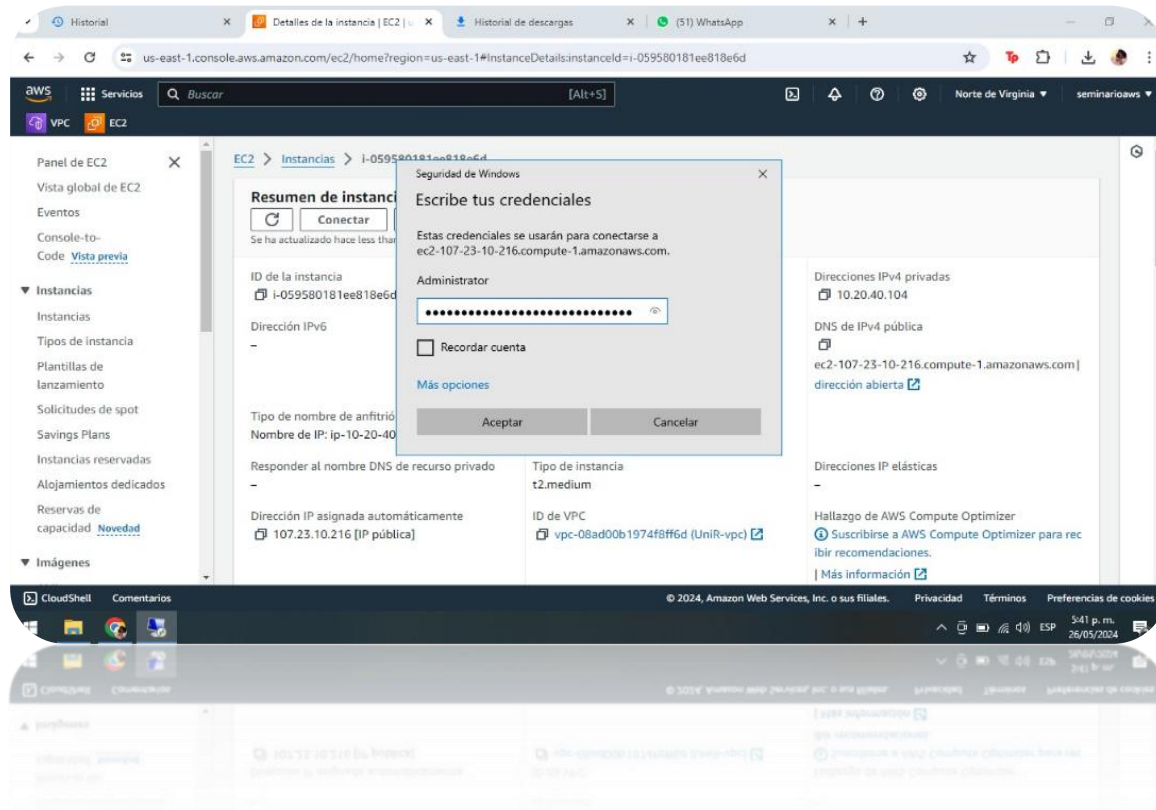
Figura 4
Creación de instancia



Nota.En el paso anterior creamos la instancia del servidor 2 y Verificamos que esté en funcionamiento. Wendy L, Flor M y Alejandro S, (2024).

Figura 5

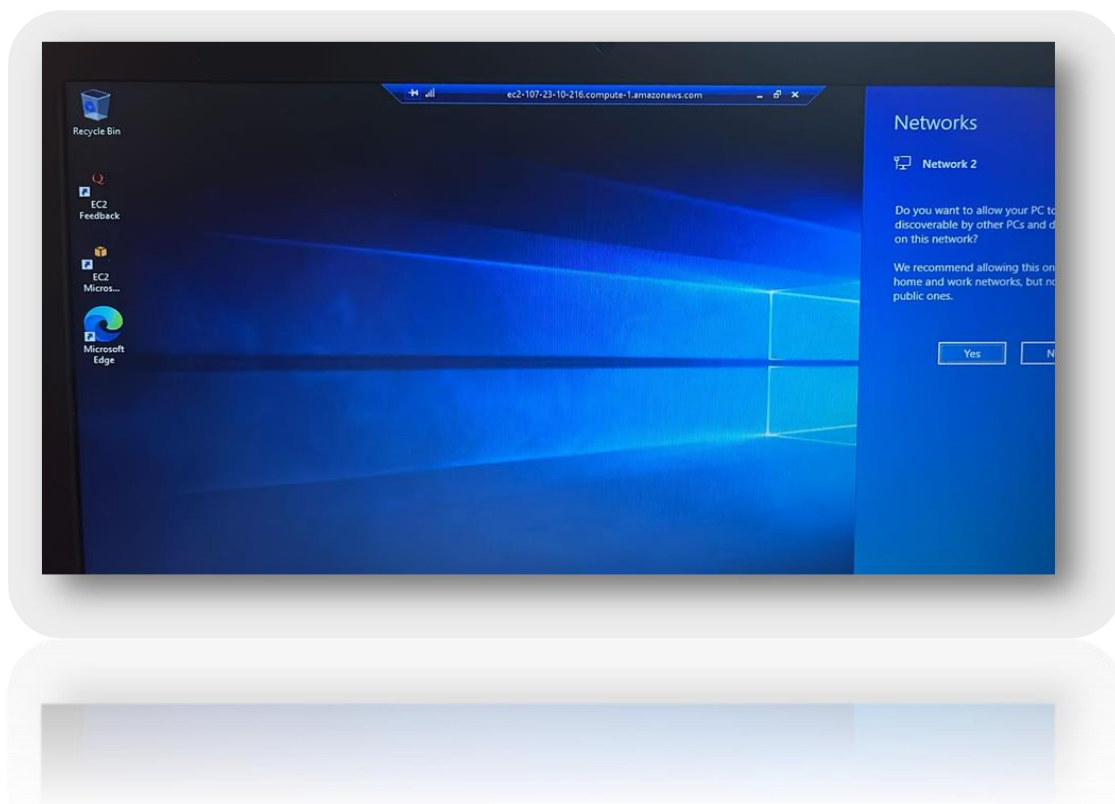
Ingreso de credenciales



Nota. En la anterior imagen anterior Se descargar el archivo remoto y se genera una contraseña y escribimos las credenciales. Wendy L, Flor M y Alejandro S, (2024).

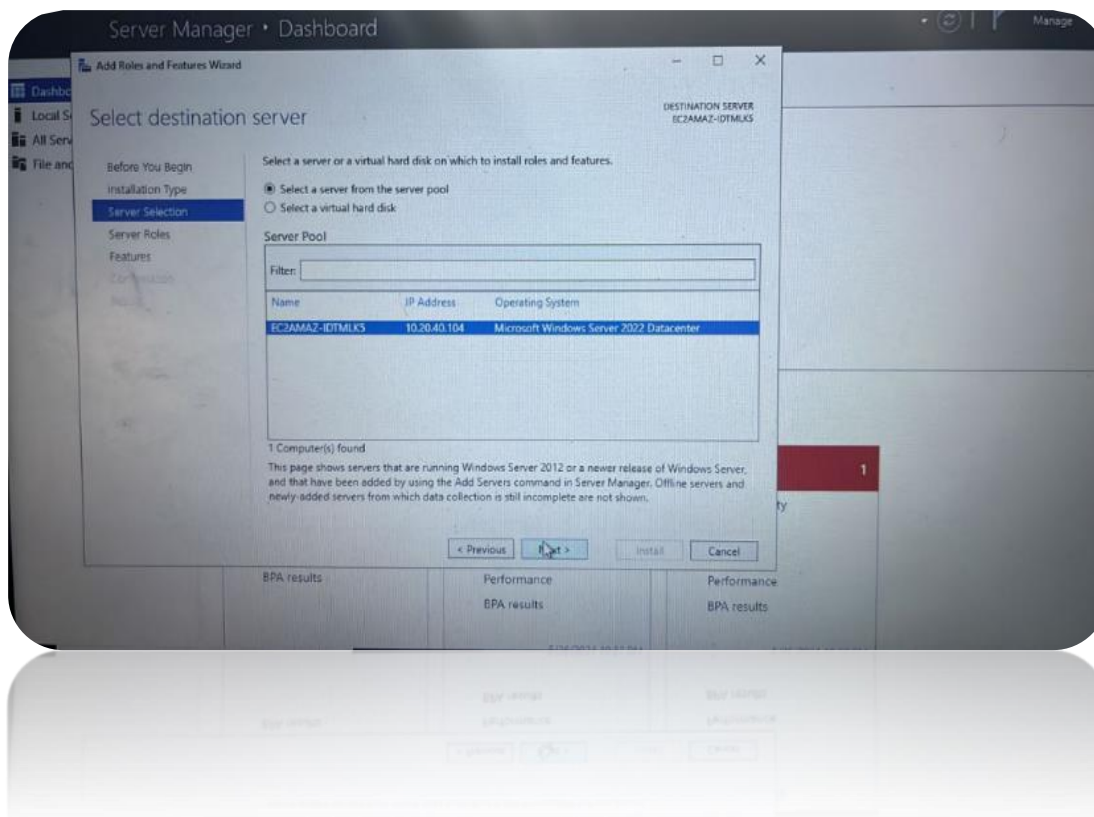
Figura 6

Acceso al escritorio remoto



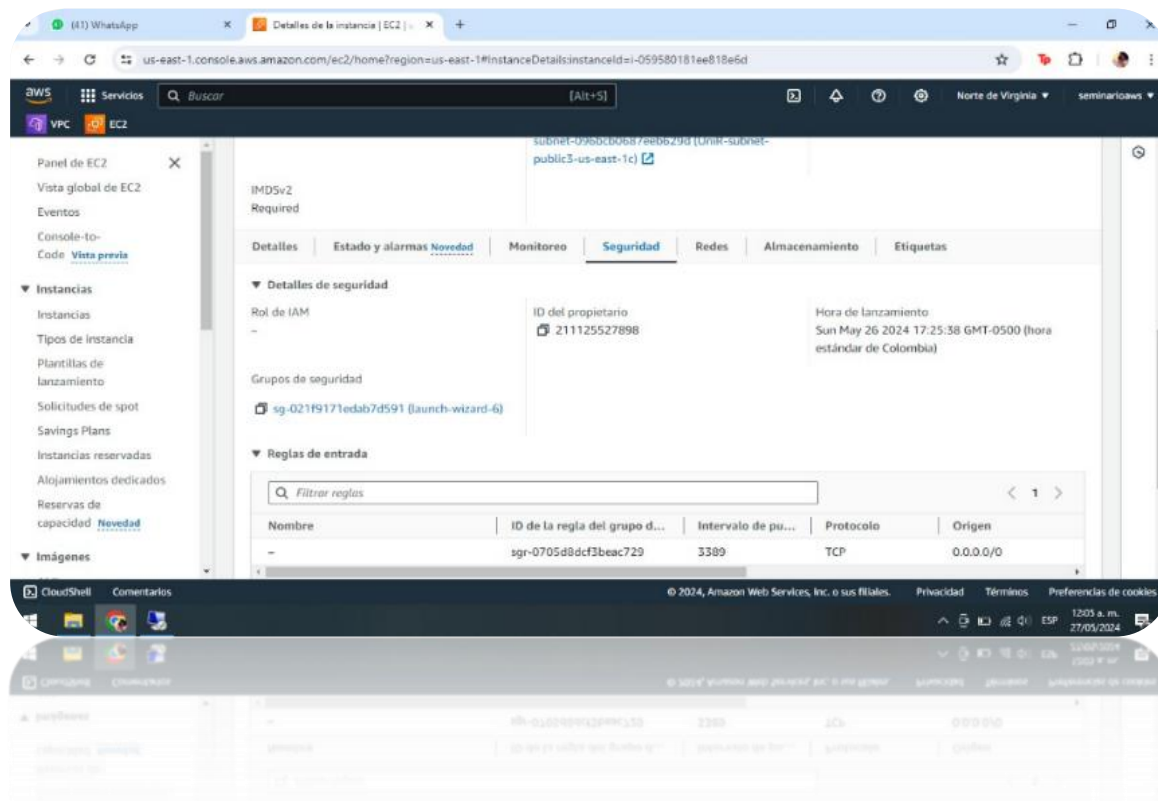
Nota.El paso anterior es donde accedemos al escritorio remoto. Wendy L, Flor M y Alejandro S, (2024).

Figura 7
Server Manager



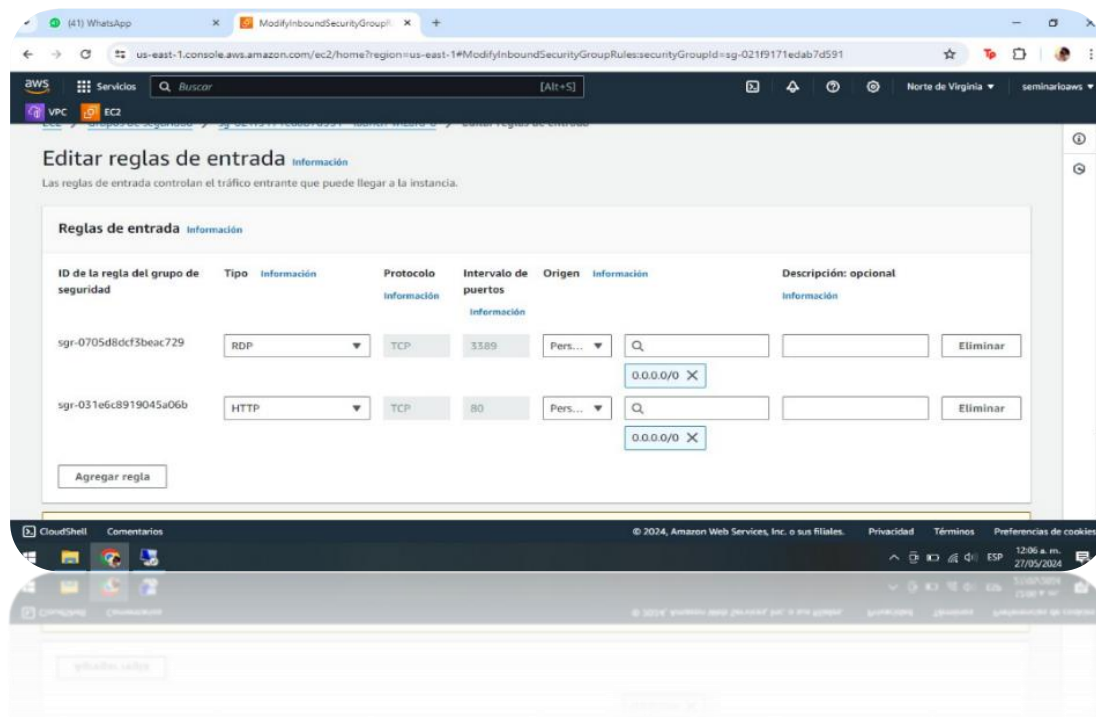
Nota. En el proceso anterior encontramos a server manager, luego en manage y añadiremos roles. Wendy L, Flor M y Alejandro S, (2024).

Figura 8
Grupos de seguridad



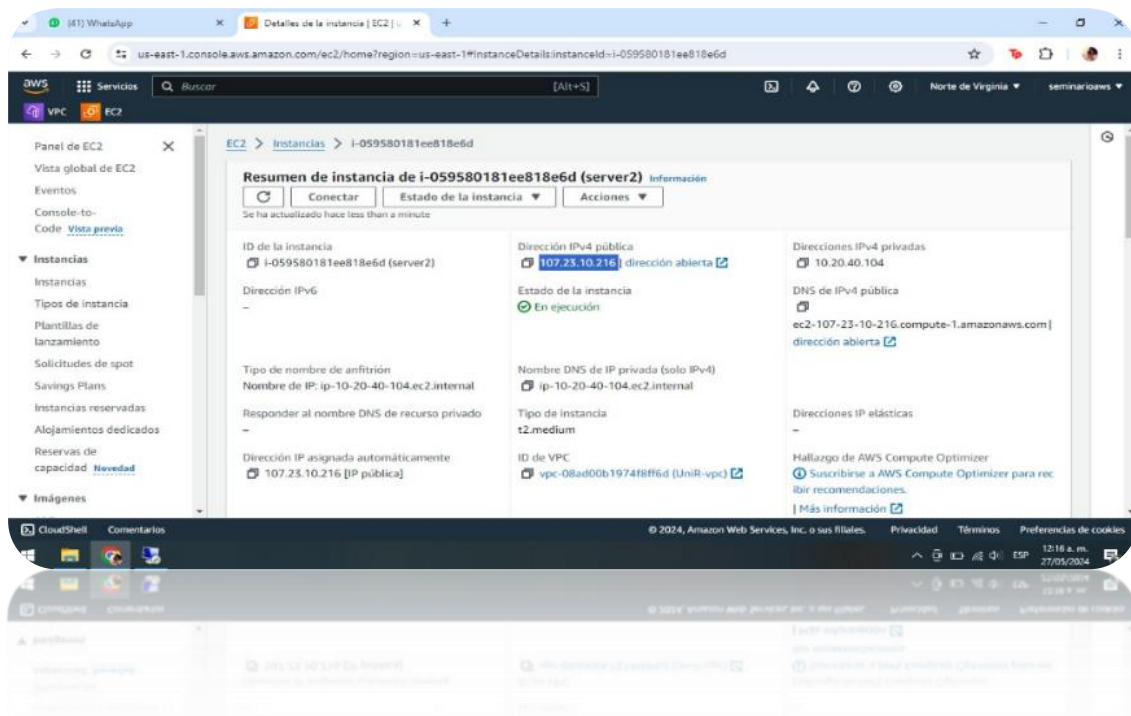
Nota.En el proceso de la imagen anterior nos dirigimos a la instancia del server 2, bajamos hasta Seguridad para acceder a grupos de seguridad. Wendy L, Flor M y Alejandro S, (2024).

Figura 9
Reglas de entrada



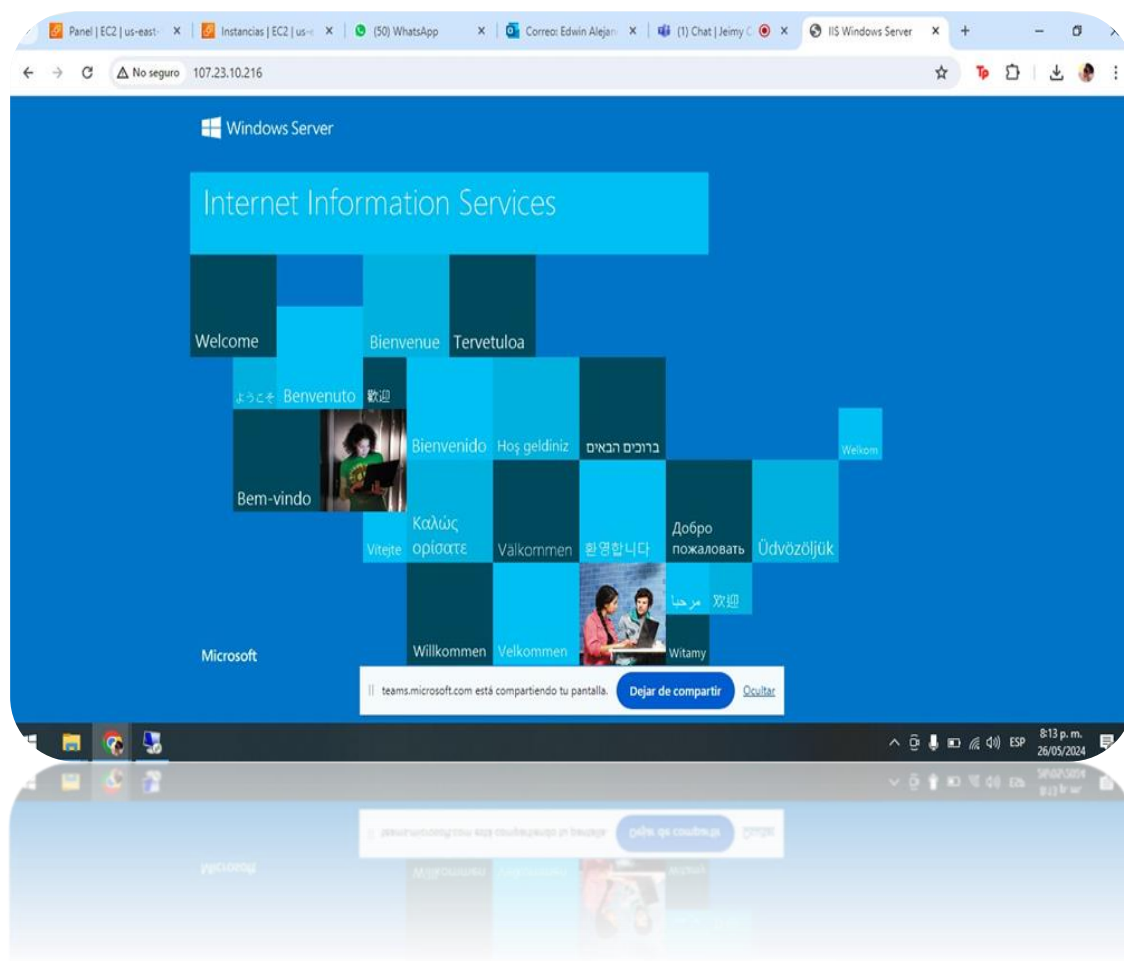
Nota. En la anterior imagen procede a editar las reglas de entrada y se agregará una regla. En "tipo de información" le pondremos como HTTP y luego le damos en guardar regla. Wendy L, Flor M y Alejandro S, (2024).

Figura 10
Resumen de instancias



Nota. En el proceso anterior Volvemos a las instancias del server 2 y copiamos la dirección IPv4 pública. Wendy L, Flor M y Alejandro S, (2024).

Figura 11
Funcionamiento de la pagina Windows server



Nota. En la imagen anterior se realiza la prueba de como carga la pagina del windows server. Wendy L, Flor M y Alejandro S, (2024)

RDS, Servicio de Base de Datos Relacional

Amazon Relational Database Service (Amazon RDS) es un servicio administrado que facilita la configuración, operación y escalado de bases de datos relacionales en la nube.

Aquí tienes algunos detalles clave: Fácil de administrar: Amazon RDS automatiza tareas como aprovisionamiento, configuración, copias de seguridad y aplicaciones de revisiones. Elección de motores: Puedes seleccionar entre ocho motores populares, incluyendo Amazon Aurora, PostgreSQL, SQL Server y MySQL. Alta disponibilidad: Implementa Multi-AZ para alta disponibilidad. Experiencia operativa: Benefíciate de más de una década de experiencia operativa comprobada y prácticas recomendadas de seguridad.

S3

S3 es altamente escalable, lo que significa que puedes almacenar y proteger cualquier cantidad de datos, desde pequeñas cantidades hasta grandes volúmenes.

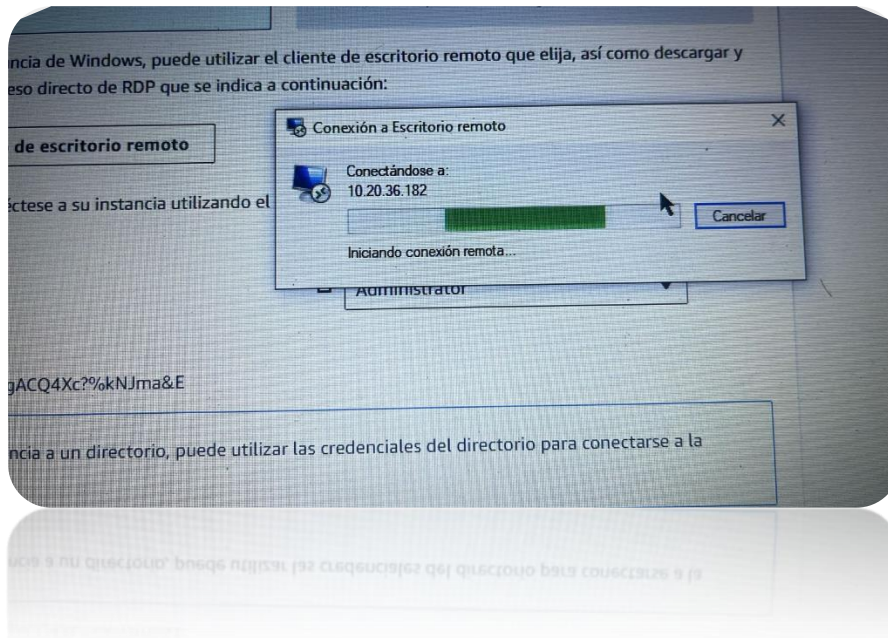
Disponibilidad de datos: Ofrece alta disponibilidad y durabilidad, lo que garantiza que tus datos estén accesibles en todo momento.

Seguridad: Amazon S3 proporciona opciones de cifrado y control de acceso para proteger tus datos.

Rendimiento: Permite un acceso rápido a los datos con baja latencia.

Figura 1

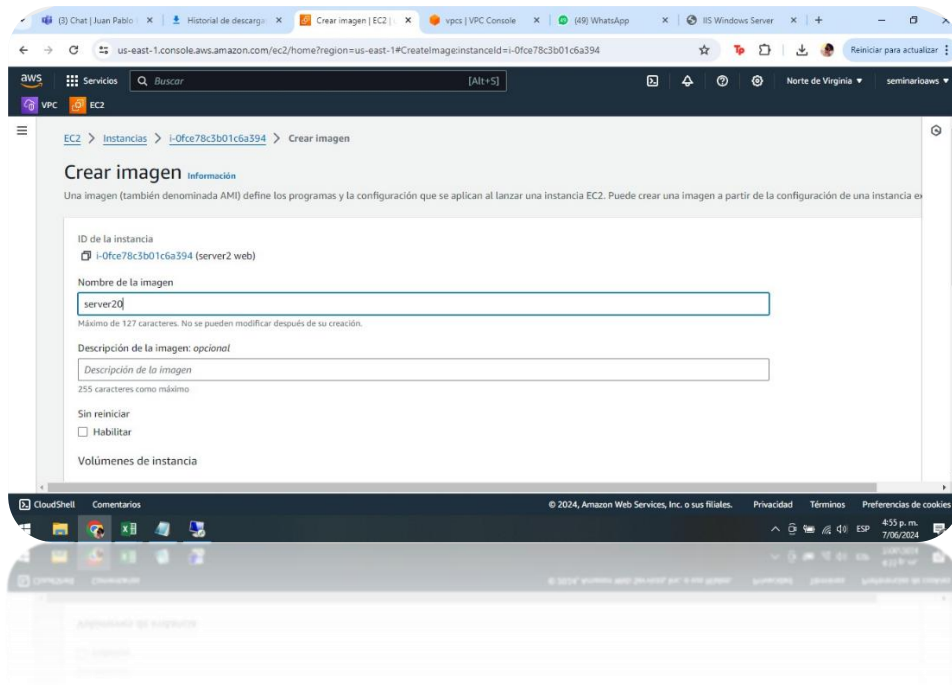
Conexión a escritorio remoto



Nota. En procedimiento anterior se realiza la conexión a escritorio remoto. Wendy L, Flor M y Alejandro S, (2024).

Figura 2

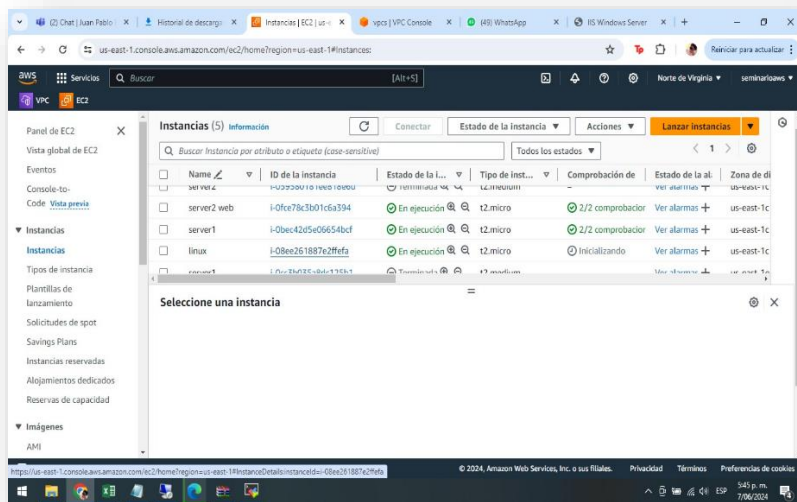
Creación del nombre de la imagen



Nota. En la imagen anterior se procede a crear el nombre de la imagen. Wendy L, Flor M y Alejandro S, (2024).

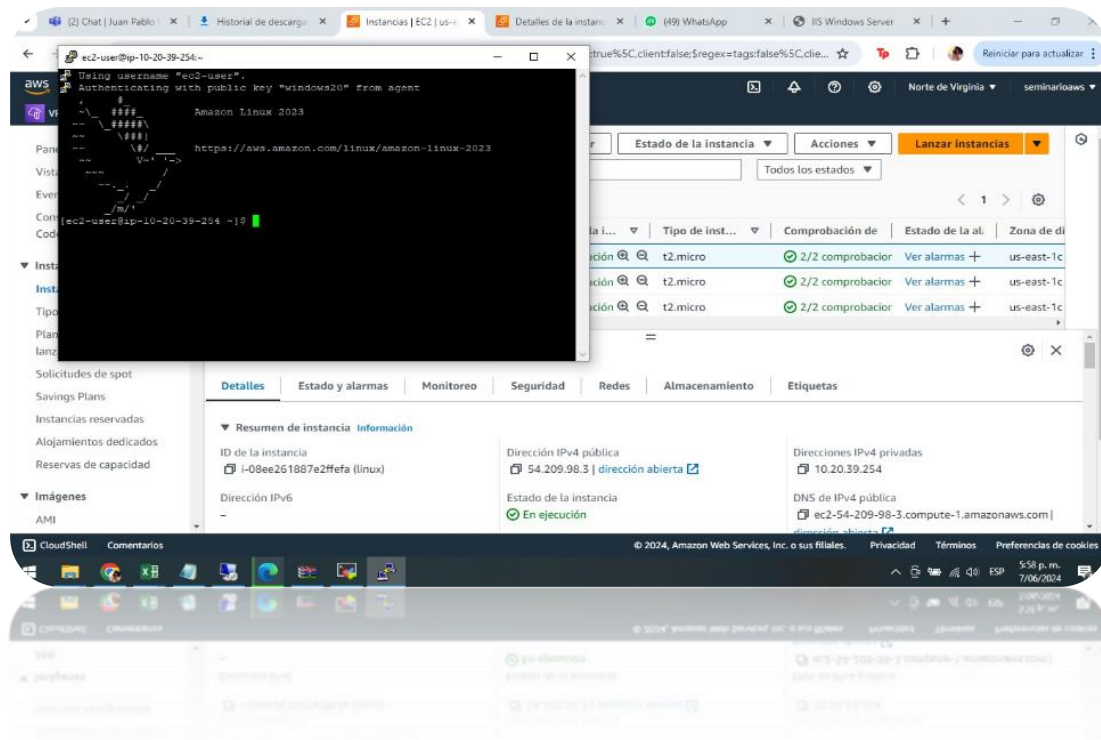
Figura 3

Creación de instancia llamada Linux



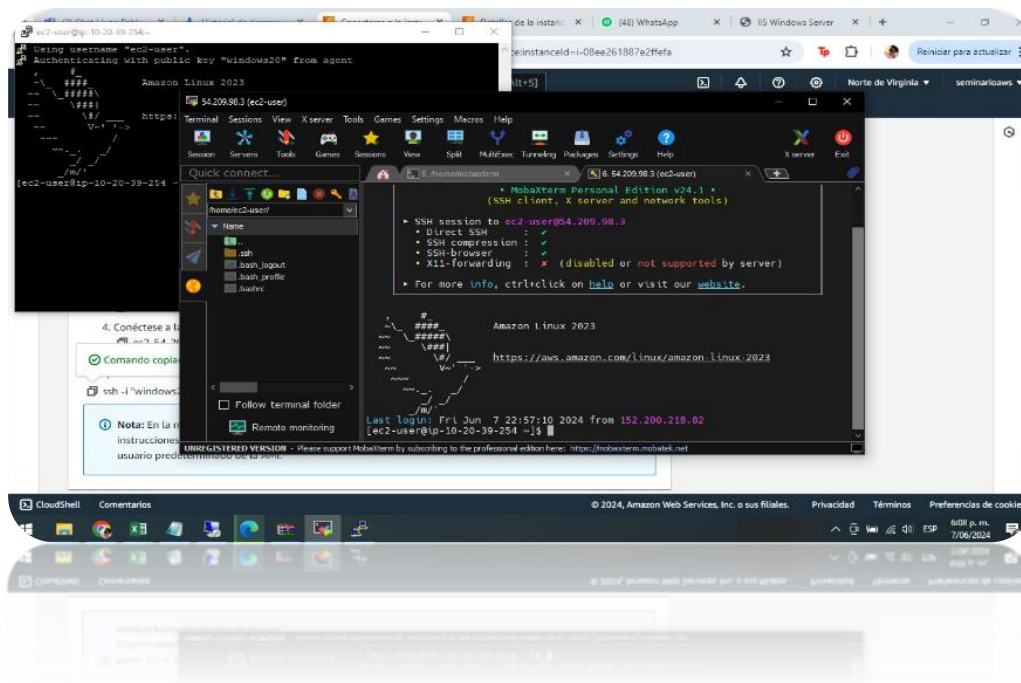
Nota. En la imagen anterior se realiza la Creación de una instancia llamada Linux, con Amazon. Wendy L, Flor M y Alejandro S, (2024).

Figura 4



. Wendy L, Flor M y Alejandro S, (2024).

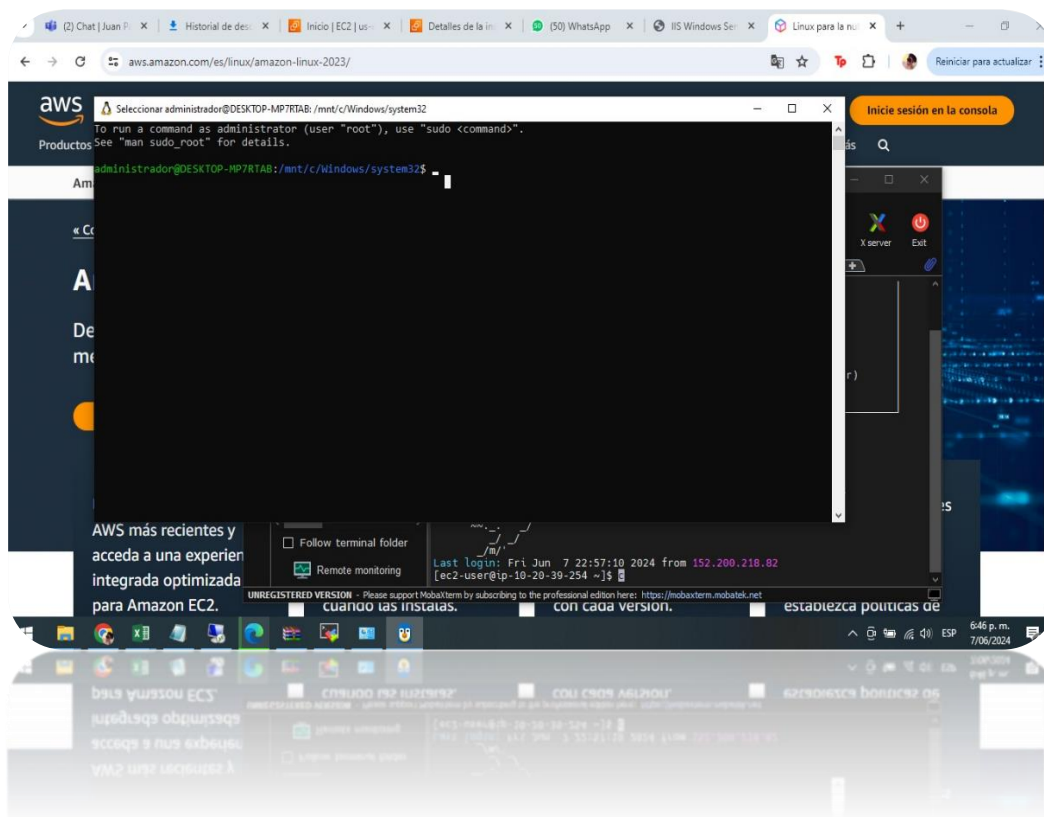
Figura 5
Contenedor



Nota. En este paso entramos al contenedor para extraer capa de virtualización física en la que estamos trabajando. Wendy L, Flor M y Alejandro S, (2024).

Figura 6

Iniciación de wsl



Nota. En la imagen anterior iniciamos el wsl. Wendy L, Flor M y Alejandro S, (2024).

Figura 1

```
ec2-user@ip-10-20-39-254-
Verifying : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 4/10
Verifying : libcgrou-3.0-1.amzn2023.0.1.x86_64 5/10
Verifying : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/10
Verifying : libnftnl-1.0.1-19.amzn2023.0.2.x86_64 7/10
Verifying : libnftnl-1.2.2-2.amzn2023.0.2.x86_64 8/10
Verifying : pigz-2.5-1.amzn2023.0.3.x86_64 9/10
Verifying : runc-1.11-1.amzn2023.0.1.x86_64 10/10
-----
WARNING:
A newer release of "Amazon Linux" is available.

Available Versions:
Version 2023.4.20240611:
Run the following command to upgrade to 2023.4.20240611:

dnf upgrade --releasever=2023.4.20240611

Release notes:
https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.4.20240611.html
-----
Installed:
containerd-1.7.11-1.amzn2023.0.1.x86_64      docker-25.0.3-1.amzn2023.0.1.x86_64      iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
iptables-nft-1.8.8-3.amzn2023.0.2.x86_64    libcgrou-3.0-1.amzn2023.0.1.x86_64      libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
libnftnl-1.0.1-19.amzn2023.0.2.x86_64      libnftnl-1.2.2-2.amzn2023.0.2.x86_64    pigz-2.5-1.amzn2023.0.3.x86_64
runc-1.11-1.amzn2023.0.1.x86_64

Complete!
[root@ip-10-20-39-254 ec2-user]# client_loop: send disconnect: Connection reset by peer
administrador@DESKTOP-MP7RTAB:/mnt/c/Windows/system32$ ssh -i /home/administrador/windows20.pem ec2-user@ec2-54-209-98-3.compute-1.amazonaws.com
#
#####
      _
     _\#####
    _/#####
   _/#####
  _/#####
 _/#####
#####
#####

https://aws.amazon.com/linux/amazon-linux-2023

Last login: Mon Jun 10 23:17:38 2024 from 191.166.191.193
ec2-user@ip-10-20-39-254 ~]$
```

Wendy L, Flor M y Alejandro S, (2024).

Figura 11

Instalación del docker

```

root@ip-10-20-32-116/home/ec2-user
Installing : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/10
Installing : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64 7/10
Installing : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 8/10
Running scriptlet: iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 8/10
Installing : libcgroupp-3.0-1.amzn2023.0.1.x86_64 9/10
Running scriptlet: docker-25.0.3-1.amzn2023.0.1.x86_64 10/10
Installing : docker-25.0.3-1.amzn2023.0.1.x86_64 10/10
Running scriptlet: docker-25.0.3-1.amzn2023.0.1.x86_64 10/10
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.

Verifying : containerd-1.7.11-1.amzn2023.0.1.x86_64 1/10
Verifying : docker-25.0.3-1.amzn2023.0.1.x86_64 2/10
Verifying : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64 3/10
Verifying : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 4/10
Verifying : libcgroupp-3.0-1.amzn2023.0.1.x86_64 5/10
Verifying : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/10
Verifying : libnftnl-1.0.1-19.amzn2023.0.2.x86_64 7/10
Verifying : libnftnl-1.2.2-2.amzn2023.0.2.x86_64 8/10
Verifying : pigz-2.5-1.amzn2023.0.3.x86_64 9/10
Verifying : runc-1.1.11-1.amzn2023.0.1.x86_64 10/10
=====
WARNING:
A newer release of "Amazon Linux" is available.

Available Versions:

Version 2023.4.20240611:
Run the following command to upgrade to 2023.4.20240611:

dnf upgrade --releasever=2023.4.20240611

Release notes:
https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.4.20240611.html
=====
Installed:
containerd-1.7.11-1.amzn2023.0.1.x86_64      docker-25.0.3-1.amzn2023.0.1.x86_64      iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
iptables-nft-1.8.8-3.amzn2023.0.2.x86_64    libcgroupp-3.0-1.amzn2023.0.1.x86_64    libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
libnftnl-1.0.1-19.amzn2023.0.2.x86_64      libnftnl-1.2.2-2.amzn2023.0.2.x86_64    pigz-2.5-1.amzn2023.0.3.x86_64
runc-1.1.11-1.amzn2023.0.1.x86_64

Complete!
root@ip-10-20-32-116 ec2-user]#

```

Nota. En la anterior imagen instalamos el docker en el Linux 2.Wendy Landázuri, Wendy L, Flor M y Alejandro S, (2024).

Figura 12

Activación del servicio

```

root@ip-10-20-32-116/home/ec2-user
https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.4.20240611.html
-----
Installed:
containerd-1.7.11-1.amzn2023.0.1.x86_64      docker-25.0.3-1.amzn2023.0.1.x86_64      iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
iptables-nft-1.8.8-3.amzn2023.0.2.x86_64    libcgrou-3.0-1.amzn2023.0.1.x86_64      libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
libnftnl-1.0.1-19.amzn2023.0.2.x86_64      libnftnl-1.2.2-2.amzn2023.0.2.x86_64    pigz-2.5-1.amzn2023.0.3.x86_64
runc-1.1.11-1.amzn2023.0.1.x86_64

Complete!
[root@ip-10-20-32-116 ec2-user]# systemctl status docker
○ docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; preset: disabled)
   Active: inactive (dead)
   TriggeredBy: ○ docker.socket
   Docs: https://docs.docker.com
[root@ip-10-20-32-116 ec2-user]# systemctl start docker
[root@ip-10-20-32-116 ec2-user]# system ctl status docker
bash: system: command not found
[root@ip-10-20-32-116 ec2-user]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; preset: disabled)
   Active: active (running) since Tue 2024-06-11 20:42:17 UTC; 17s ago
   TriggeredBy: ● docker.socket
   Docs: https://docs.docker.com
   Process: 57966 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
   Process: 57967 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
   Main PID: 57968 (dockerd)
   Tasks: 8
   Memory: 32.0M
   CPU: 332ms
   CGroup: /system.slice/docker.service
           └─57968 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

Jun 11 20:42:16 ip-10-20-32-116.ec2.internal systemd[1]: Starting docker.service - Docker Application Container Engine...
Jun 11 20:42:16 ip-10-20-32-116.ec2.internal dockerd[57968]: time="2024-06-11T20:42:16.791621503Z" level=info msg="Starting up"
Jun 11 20:42:16 ip-10-20-32-116.ec2.internal dockerd[57968]: time="2024-06-11T20:42:16.854796408Z" level=info msg="Loading containers: start."
Jun 11 20:42:17 ip-10-20-32-116.ec2.internal dockerd[57968]: time="2024-06-11T20:42:17.376767365Z" level=info msg="Loading containers: done."
Jun 11 20:42:17 ip-10-20-32-116.ec2.internal dockerd[57968]: time="2024-06-11T20:42:17.404032349Z" level=info msg="Docker daemon" commit=f417435 containerd-snapshotter
Jun 11 20:42:17 ip-10-20-32-116.ec2.internal dockerd[57968]: time="2024-06-11T20:42:17.404369704Z" level=info msg="Daemon has completed initialization"
Jun 11 20:42:17 ip-10-20-32-116.ec2.internal dockerd[57968]: time="2024-06-11T20:42:17.448305415Z" level=info msg="API listen on /run/docker.sock"
Jun 11 20:42:17 ip-10-20-32-116.ec2.internal systemd[1]: Started docker.service - Docker Application Container Engine.
lines 1-22/22 (END)

```

Nota. En el procedimiento anterior vemos como activamos el servicio con el comando `systemctl start docker` y verifico el estado con el `systemctl status docker`.

Wendy L, Flor M y Alejandro S, (2024).

Figura 13

Contenedor 1, Con dirección 0.0.0 en el puerto 8080

```
[root@ip-10-20-32-116 contenedor]# docker run -dit --name contenedor1 -p 8080:80 -v /tmp/contenedor:/usr/local/apache2/htdocs/httpd
"docker run" requires at least 1 argument.
See 'docker run --help'.

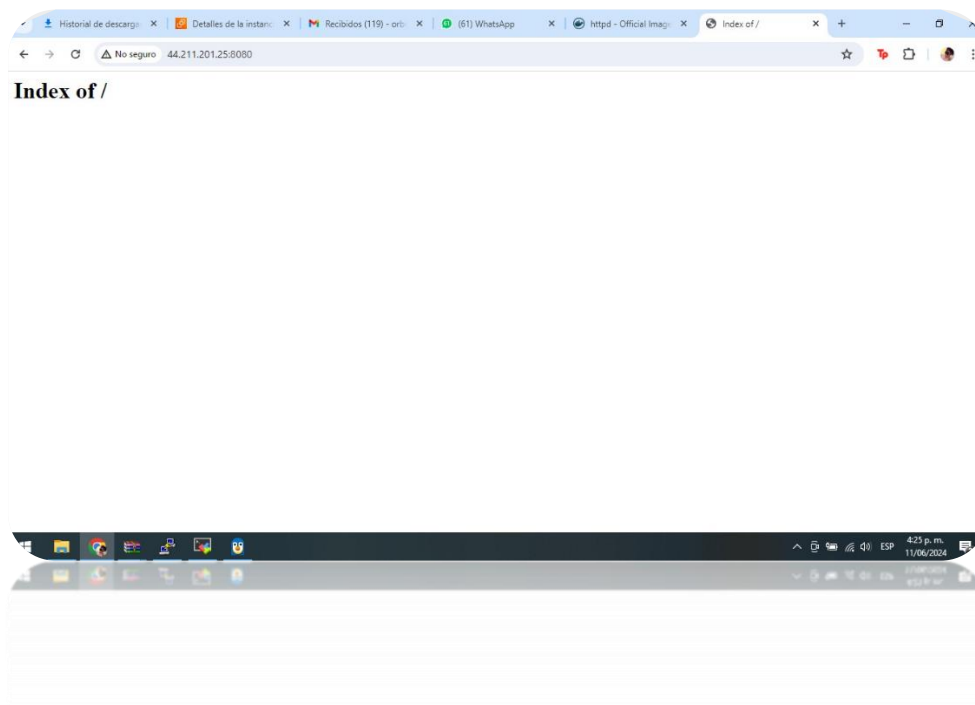
Usage: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Create and run a new container from an image
[root@ip-10-20-32-116 contenedor]# docker run -dit --name contenedor1 -p 8080:80 -v /tmp/contenedor:/usr/local/apache2/htdocs httpd
fcdcd5f860b4d551e48416fc328d4fc77f928b69b05639cd3f92a63f7ae6841
[root@ip-10-20-32-116 contenedor]# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
httpd latest 356125da0595 2 months ago 147MB
hello-world latest d2c94e258dcb 13 months ago 13.3kB
[root@ip-10-20-32-116 contenedor]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
fcdcd5f860b httpd "httpd-foreground" About a minute ago Up About a minute 0.0.0.0:8080->80/tcp, :::8080->80/tcp contenedor1
[root@ip-10-20-32-116 contenedor]#
```

Nota. En la imagen anterior se crea la imagen httpd, y un contenedor y con la dirección 0.0.0 en el puerto 8080 del anfitrión. Se llama contenedor 1. Wendy L, Flor M y Alejandro S, (2024).

Figura 14

Primera prueba del Index of



Nota. En la imagen anterior se muestra la Primera prueba del Index of. Wendy L, Flor M y Alejandro S, (2024).

Figura 15

Contenedor 2

```

for more examples and details, visit
https://docs.docker.com/get-started/

[root@ip-10-20-32-116 contenedor]# docker run -dit --name contenedor1 -p 8080:80 -v /tmp/contenedor/:/usr/local/apache2/htdocs/httpd
"docker run" requires at least 1 argument.
See "docker run --help".

Usage: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

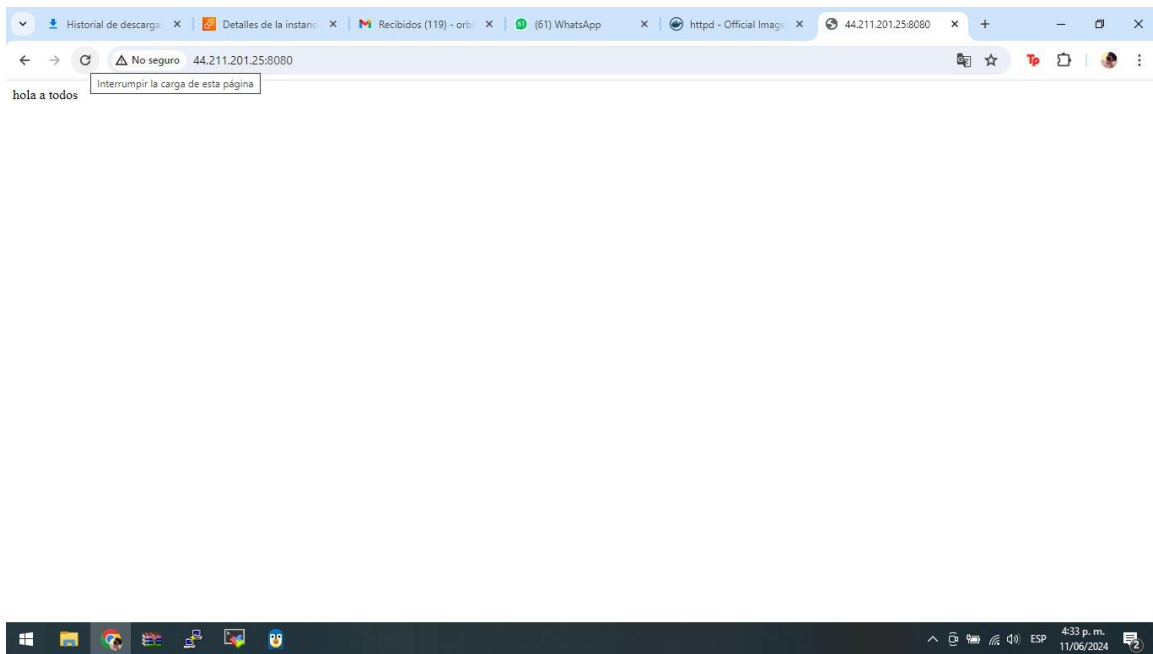
Create and run a new container from an image
[root@ip-10-20-32-116 contenedor]# docker run -dit --name contenedor1 -p 8080:80 -v /tmp/contenedor/:/usr/local/apache2/htdocs httpd
fcddcd5f860b4d51e48416fc328d4fc77f928b09b05639cd3f92a63f7ae6841
[root@ip-10-20-32-116 contenedor]# docker images
REPOSITORY   TAG       IMAGE ID       CREATED        SIZE
httpd        latest   356125da0595   2 months ago   147MB
hello-world  latest   d2c94e258dcb   13 months ago  13.3kB
[root@ip-10-20-32-116 contenedor]# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
fcddcd5f860b   httpd    "httpd-foreground"      About a minute ago   Up About a minute   0.0.0.0:8080->80/tcp, :::8080->80/tcp   contenedor1
[root@ip-10-20-32-116 contenedor]# nano index.html
737171f26b563bca2ba0a21daccf43303bc30a5aa6ca44e4108c220a3f8981
[root@ip-10-20-32-116 contenedor]# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
737171f26b5   httpd    "httpd-foreground"      6 seconds ago   Up 5 seconds   0.0.0.0:8082->80/tcp, :::8082->80/tcp   contenedor2
fcddcd5f860b   httpd    "httpd-foreground"      23 minutes ago   Up 23 minutes   0.0.0.0:8080->80/tcp, :::8080->80/tcp   contenedor1
[root@ip-10-20-32-116 contenedor]#

```

Nota. En la imagen anterior creamos el segundo contenedor, en el puerto 8082 del anfitrión se llama contenedor 2. Wendy L, Flor M y Alejandro S, (2024).

Figura 16

Prueba del sitio web

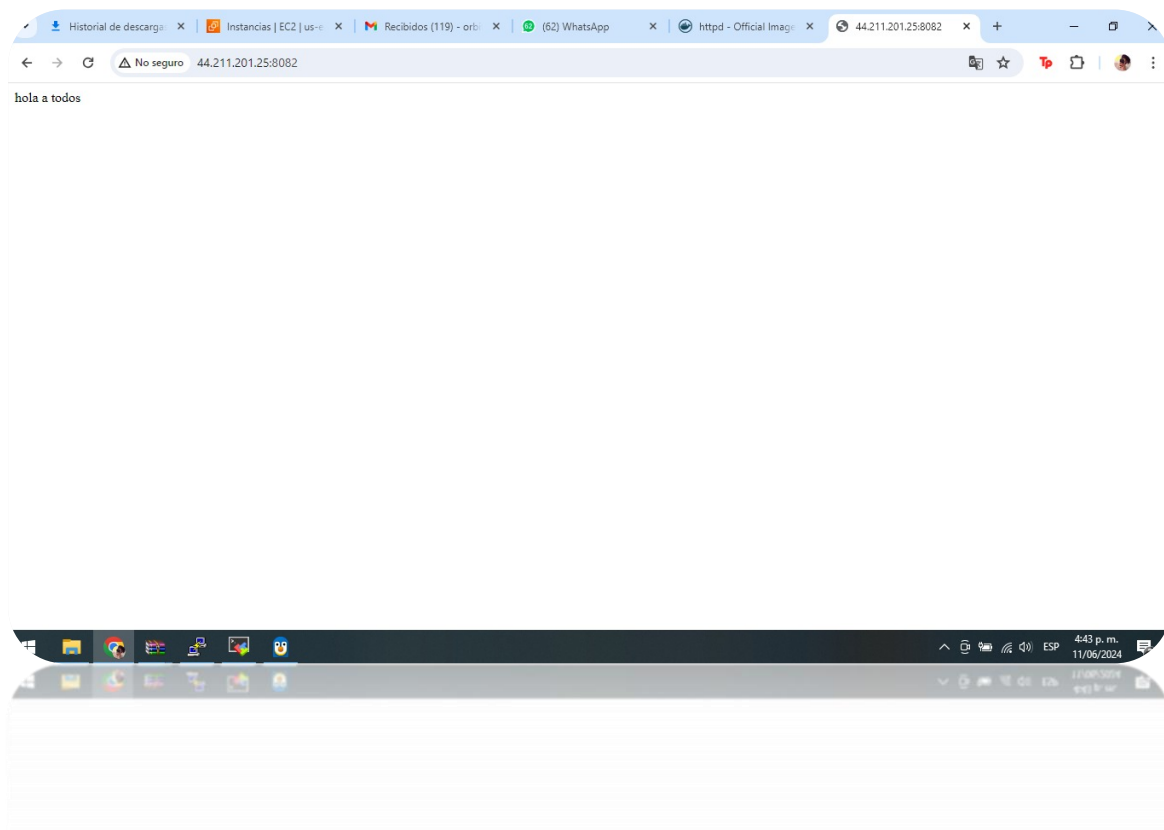


Nota. En la imagen anterior se muestra el proceso de la creación del archivo del sitio web de prueba.

Wendy L, Flor M y Alejandro S, (2024).

Figura 17

Prueba del puerto 8082

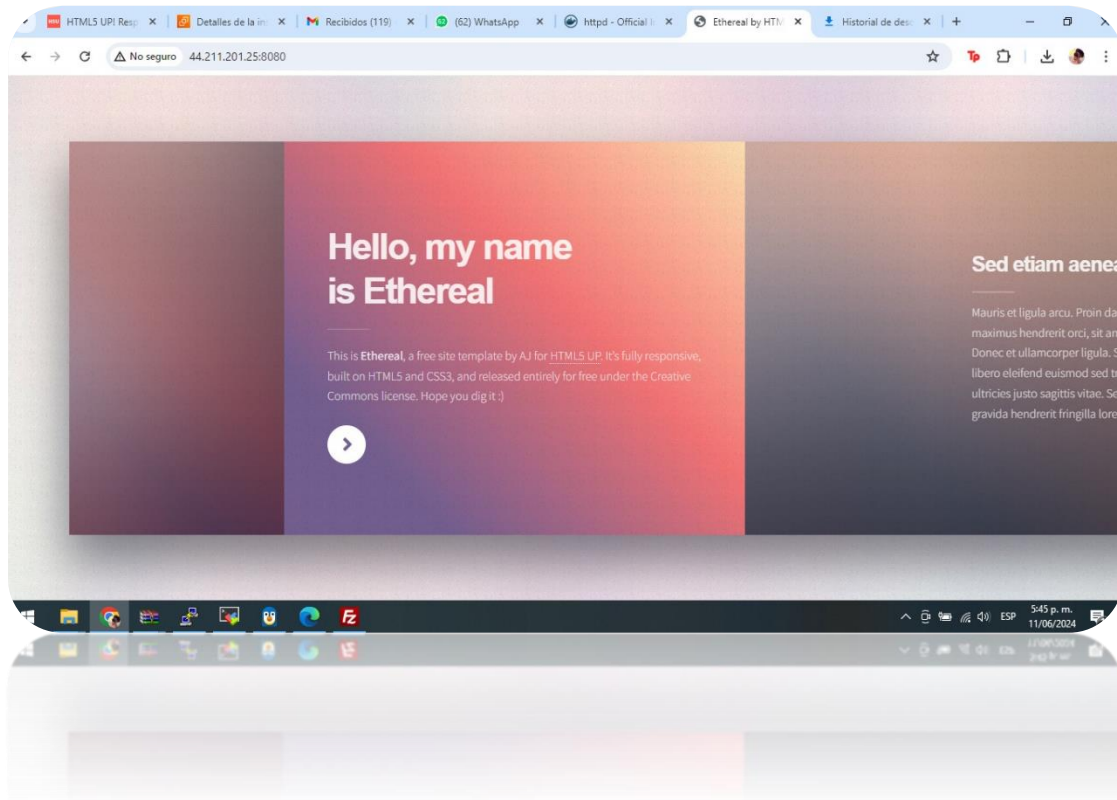


Nota. En la imagen anterior comprobamos el sitio web de prueba del puerto 8082.

Wendy L, Flor M y Alejandro S, (2024).

Figura 18

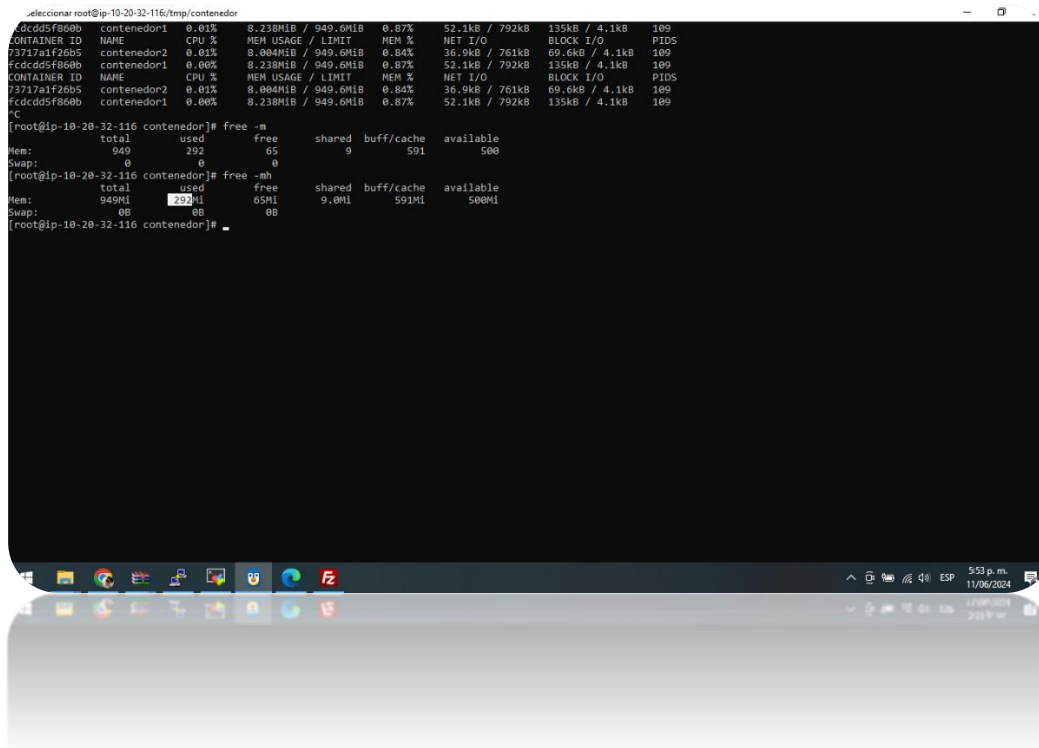
Acceso al puerto 8080 y 8082



Nota. En la imagen anterior observamos como carga la página que se descargó de prueba del puerto tanto como 8080 como 8082. Wendy L, Flor M y Alejandro S, (2024).

Figura 19

Máquina virtual



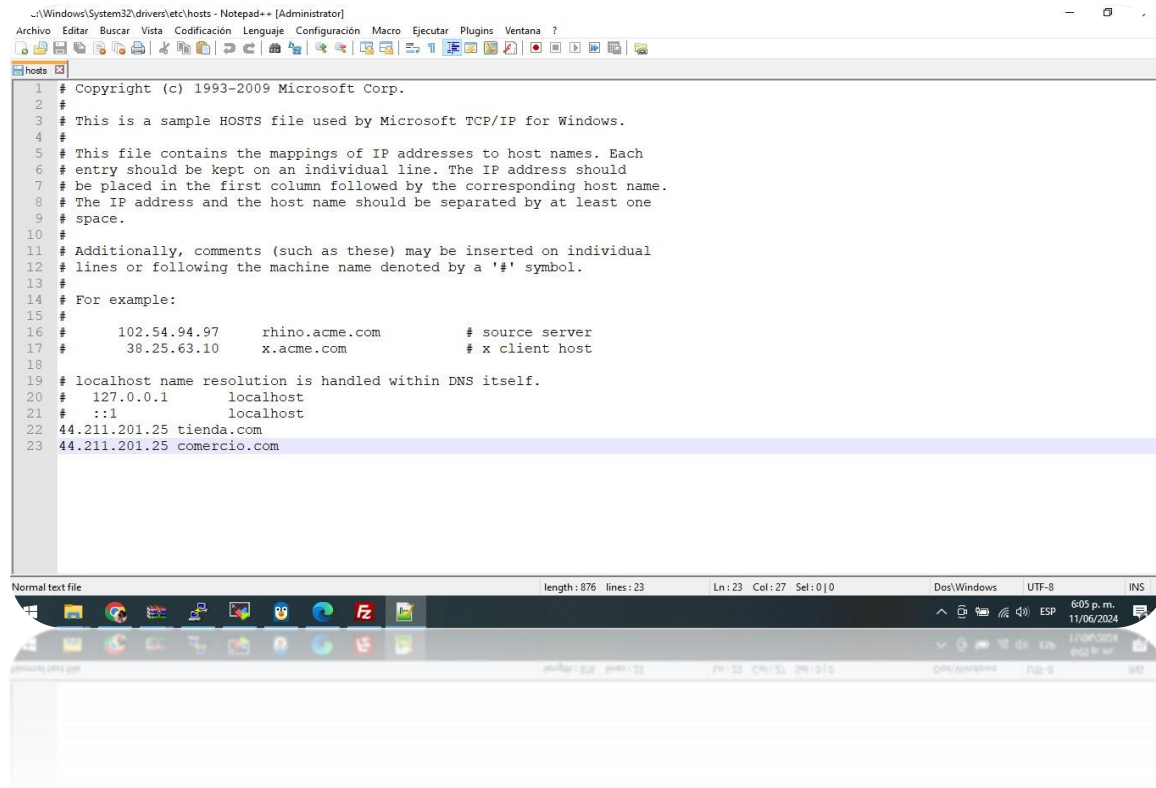
The image shows a terminal window with the following content:

```
seleccionar root@ip-10-20-32-116/tmp/contenedor
fcdd5f80b contenedor1 0.01% 8.238MiB / 949.0MiB 0.87% 52.1kB / 792kB 135kB / 4.1kB 109
CONTAINER ID NAME CPU % MEM USAGE / LIMIT MEM % NET I/O BLOCK I/O PIDS
73717a1f26b5 contenedor2 0.01% 8.004MiB / 949.0MiB 0.84% 36.9kB / 761kB 69.6kB / 4.1kB 109
fcdd5f80b contenedor1 0.00% 8.238MiB / 949.0MiB 0.87% 52.1kB / 792kB 135kB / 4.1kB 109
CONTAINER ID NAME CPU % MEM USAGE / LIMIT MEM % NET I/O BLOCK I/O PIDS
73717a1f26b5 contenedor2 0.01% 8.004MiB / 949.0MiB 0.84% 36.9kB / 761kB 69.6kB / 4.1kB 109
fcdd5f80b contenedor1 0.00% 8.238MiB / 949.0MiB 0.87% 52.1kB / 792kB 135kB / 4.1kB 109
VC
[root@ip-10-20-32-116 contenedor]# free -m
total used free shared buff/cache available
Mem: 949 292 65 9 591 500
Swap: 0 0 0
[root@ip-10-20-32-116 contenedor]# free -mh
total used free shared buff/cache available
Mem: 949Mi 292Mi 65Mi 9.0Mi 591Mi 500Mi
Swap: 0B 0B 0B
[root@ip-10-20-32-116 contenedor]#
```

Nota. En la imagen anterior vemos que la máquina virtual como se está utilizando 292 megas y se está ejecutando procesos propios del sistema y del docker.

Wendy L, Flor M y Alejandro S, (2024).

Figura 20

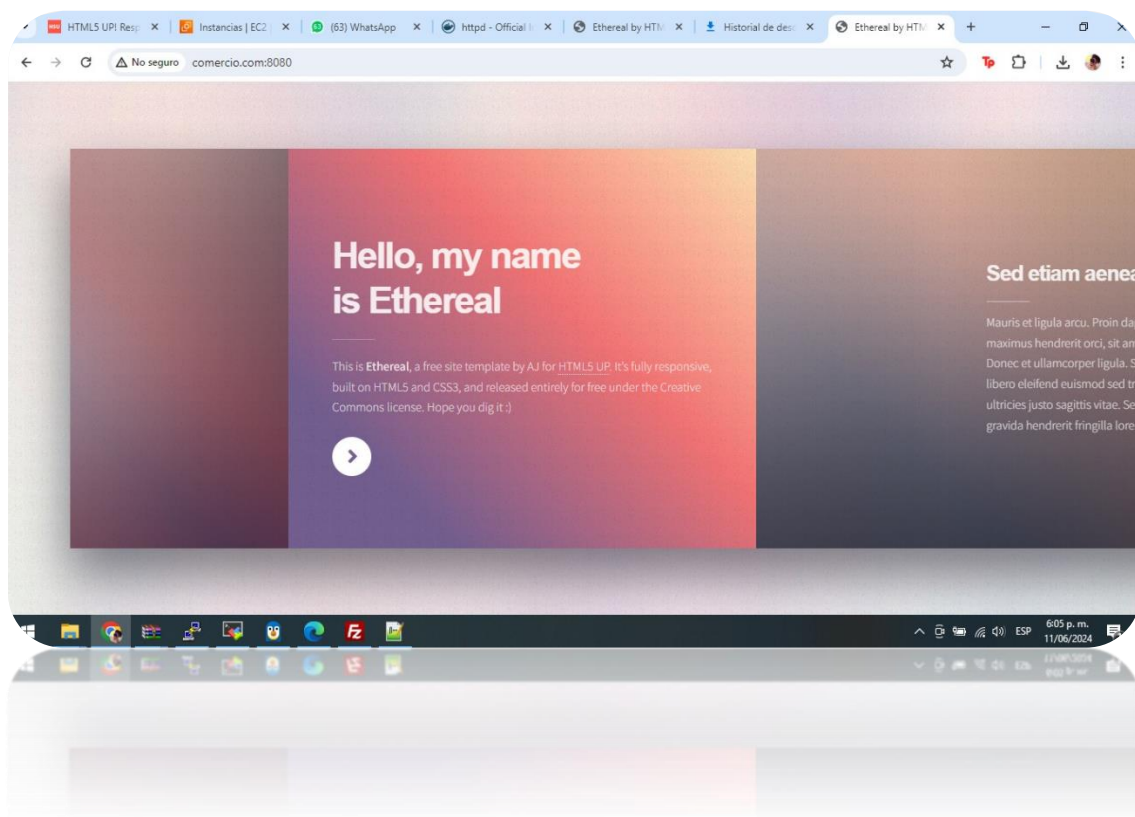


```
1 # Copyright (c) 1993-2009 Microsoft Corp.
2 #
3 # This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
4 #
5 # This file contains the mappings of IP addresses to host names. Each
6 # entry should be kept on an individual line. The IP address should
7 # be placed in the first column followed by the corresponding host name.
8 # The IP address and the host name should be separated by at least one
9 # space.
10 #
11 # Additionally, comments (such as these) may be inserted on individual
12 # lines or following the machine name denoted by a '#' symbol.
13 #
14 # For example:
15 #
16 #       102.54.94.97       rhino.acme.com          # source server
17 #       38.25.63.10      x.acme.com            # x client host
18
19 # localhost name resolution is handled within DNS itself.
20 #   127.0.0.1           localhost
21 #       ::1             localhost
22 44.211.201.25 tienda.com
23 44.211.201.25 comercio.com
```

Wendy L, Flor M y Alejandro S, (2024).

Figura 21

Comercio.com

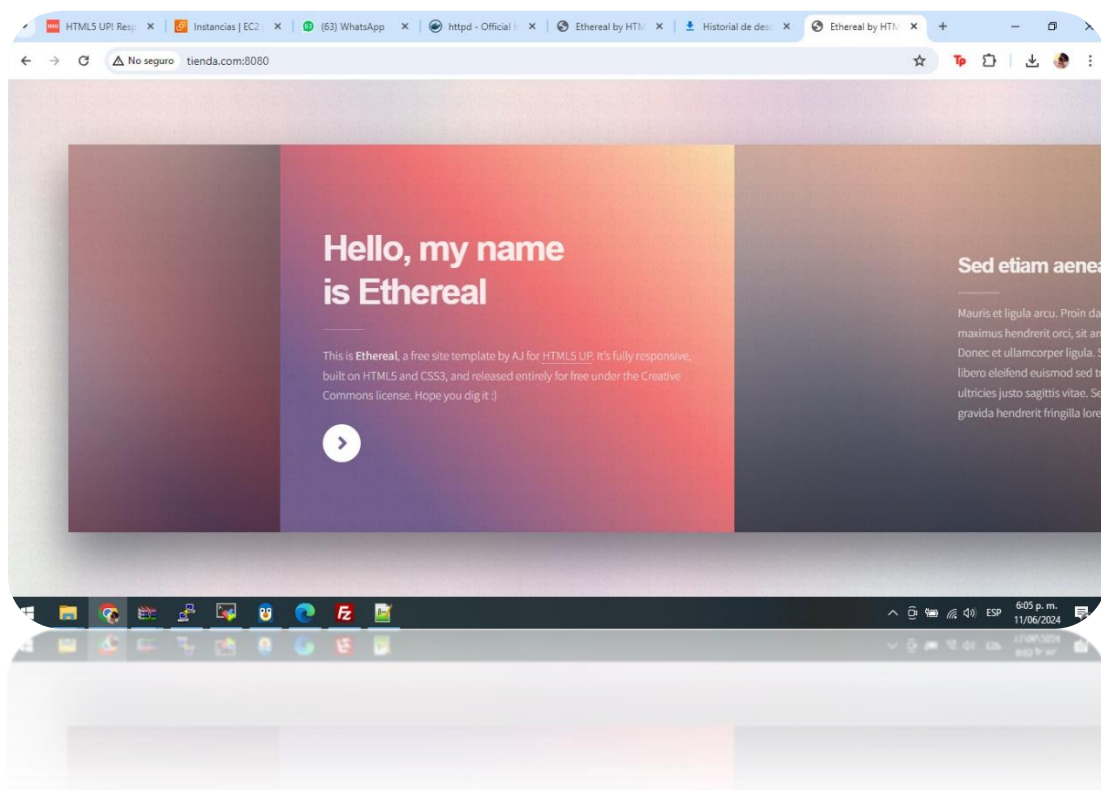


Nota. En la imagen anterior pueden observar la página de comercio.com.

Wendy L, Flor M y Alejandro S, (2024).

Figura 22

Tienda y comercio

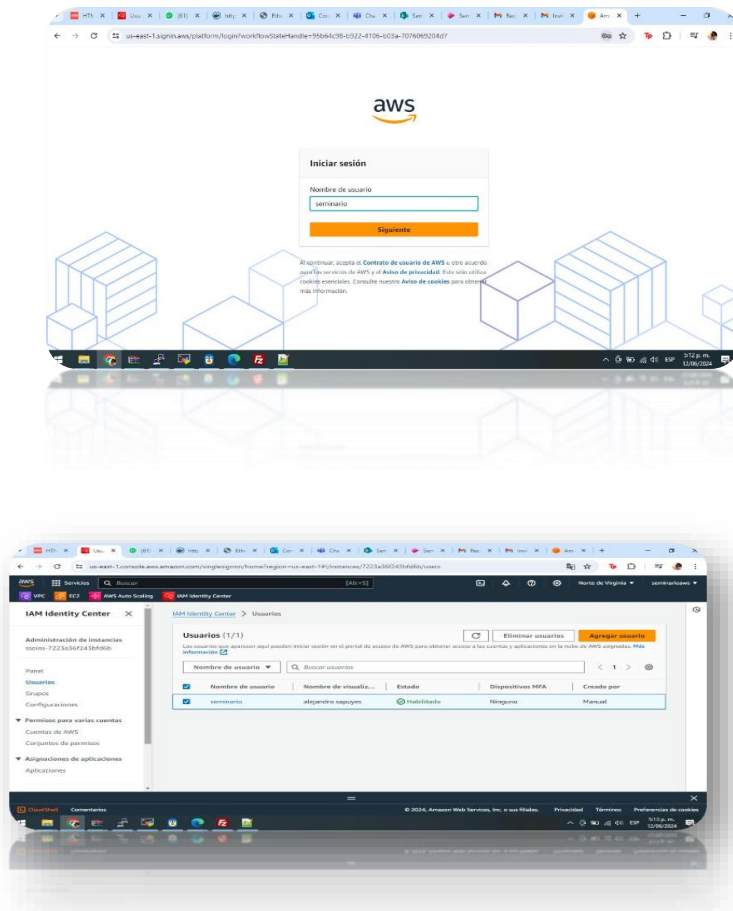


Nota. En la imagen anterior se crea la aplicación de tienda y comercio.

Wendy L, Flor M y Alejandro S, (2024).

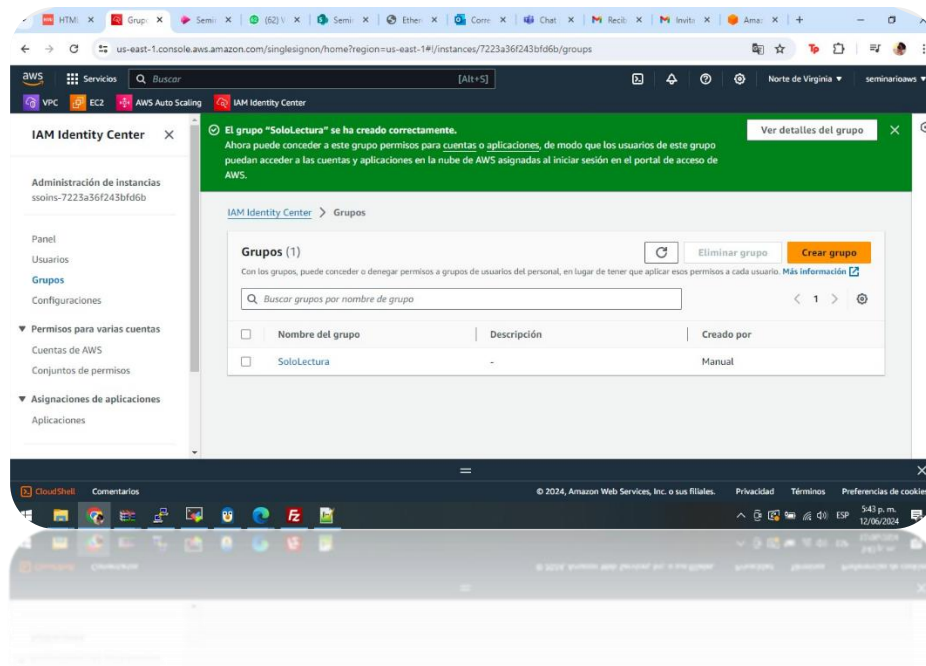
Figura 23

Creación del usuario en I Am Center



Nota. En la imagen anterior Wendy L, Flor M y Alejandro S, (2024).

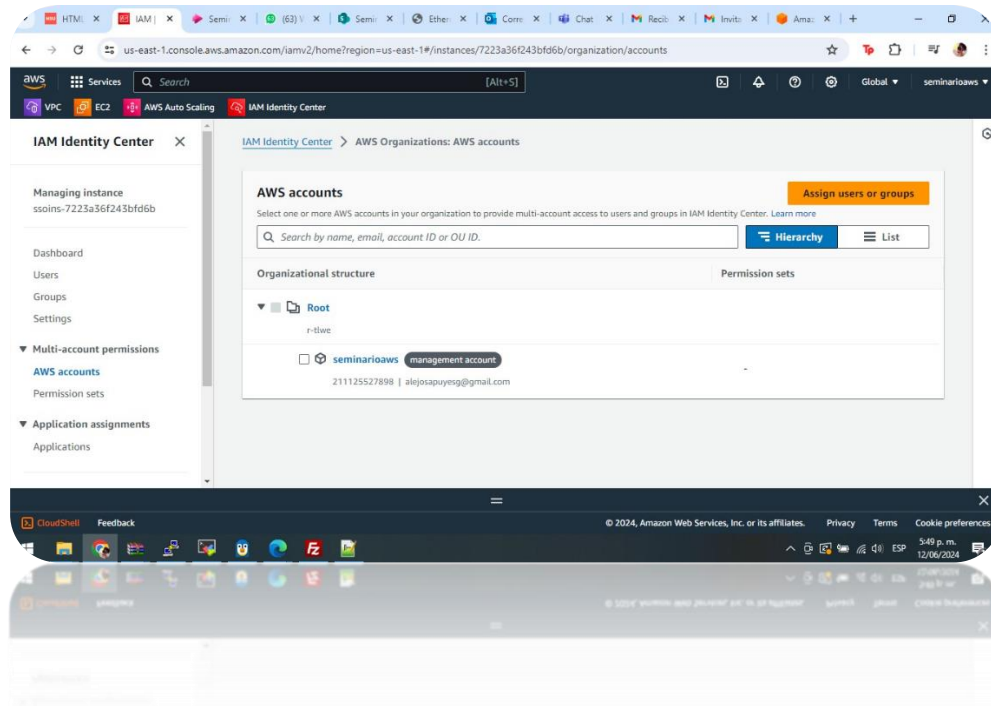
Figura 24



Nota. En la imagen anterior creamos un grupo y le nombramos como Solo Lectura
Wendy L, Flor M y Alejandro S, (2024).

Figura 25

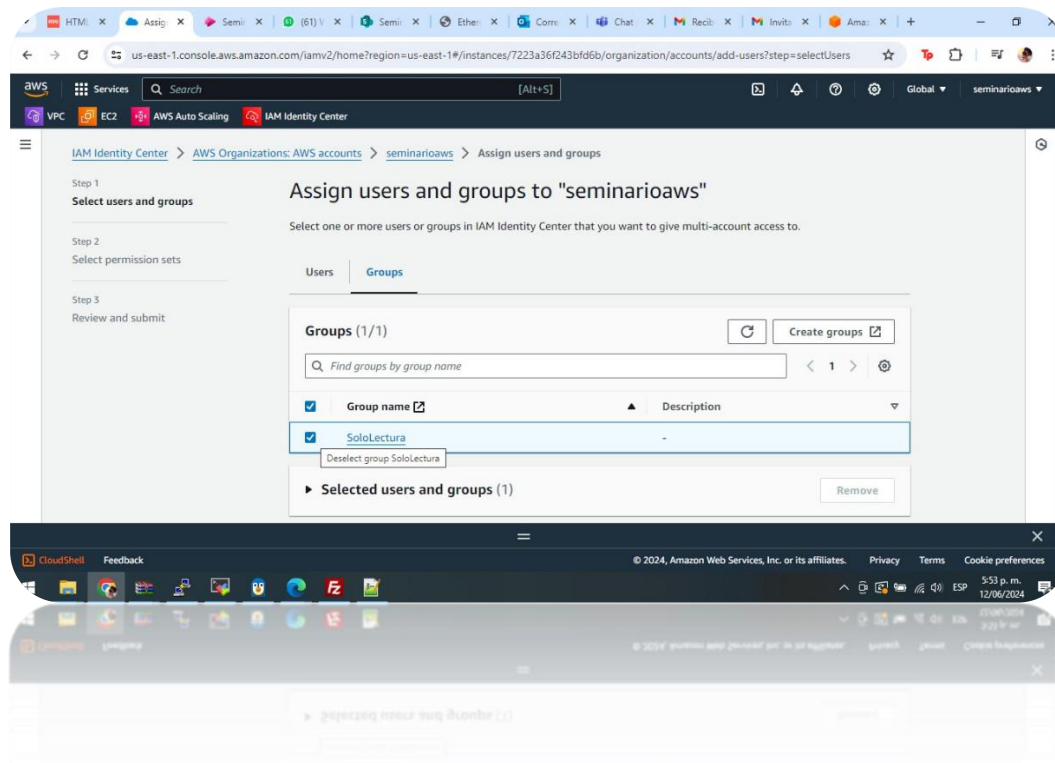
Cuenta organizacional



Nota. En la imagen anterior se muestra la creación de nuestra cuenta organizacional del seminario aws. Wendy L, Flor M y Alejandro S, (2024).

Figura 25

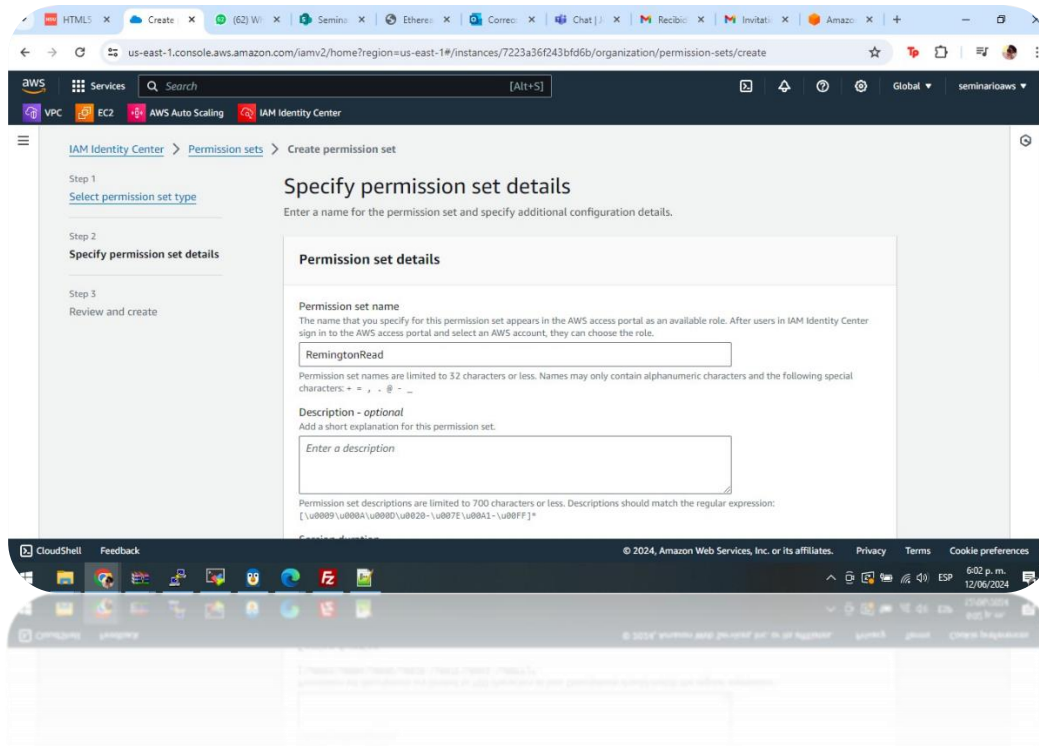
Asignación solo lectura



Nota. En la imagen anterior se le asigna el grupo Solo Lectura. Wendy L, Flor M y Alejandro S, (2024).

Figura 25

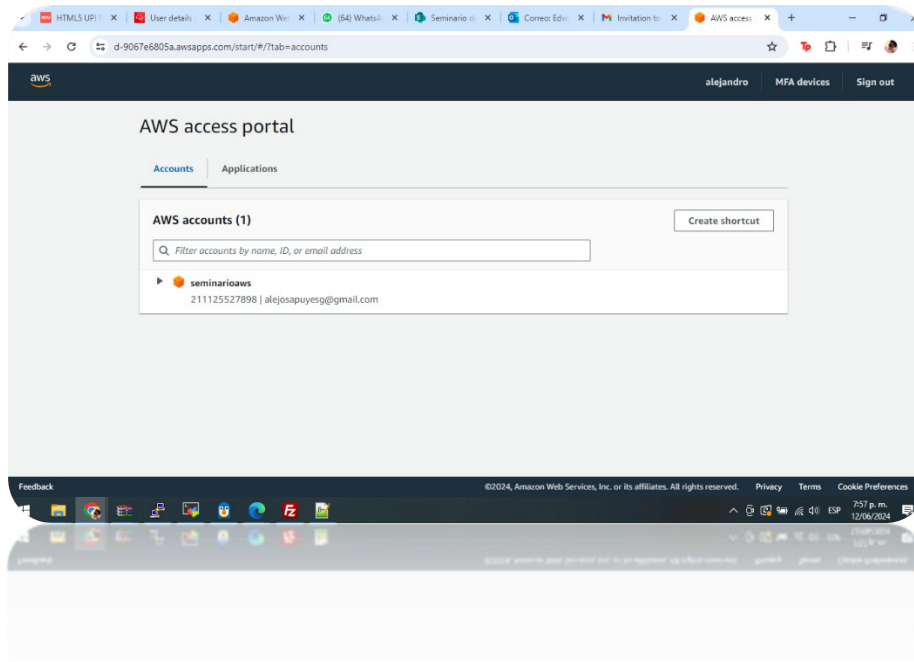
Creación de permisos



Nota. En la imagen anterior Creamos unos permisos y le nombramos remingtonRead.
Wendy L, Flor M y Alejandro S, (2024).

Figura 25

Acceso al aws acces central

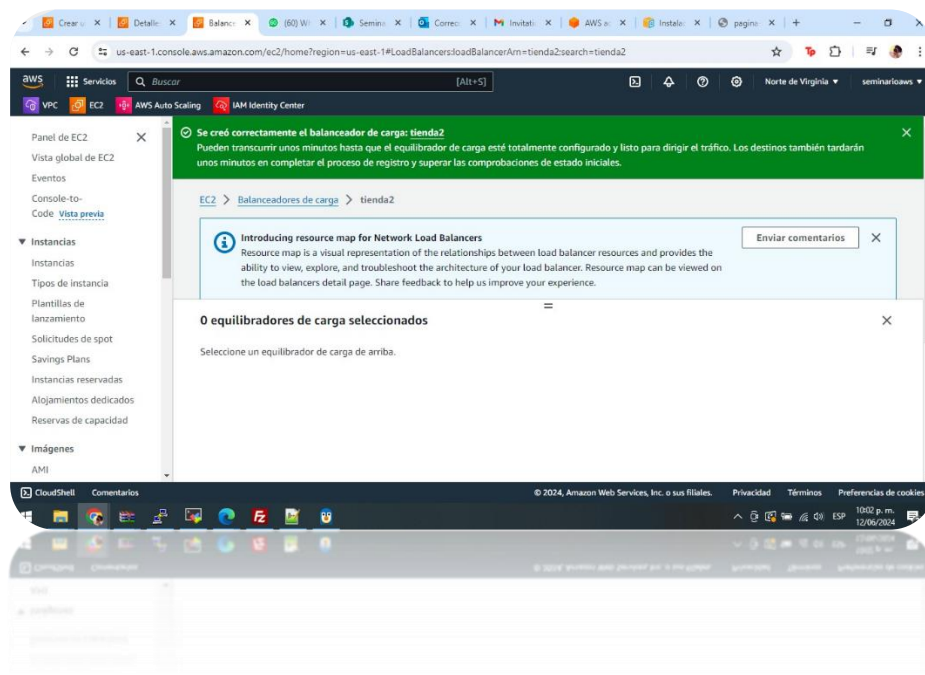


Nota. En la imagen anterior se accedimos al aws acces central y aparece nuestra cuenta.

Wendy L, Flor M y Alejandro S, (2024).

Figura 26

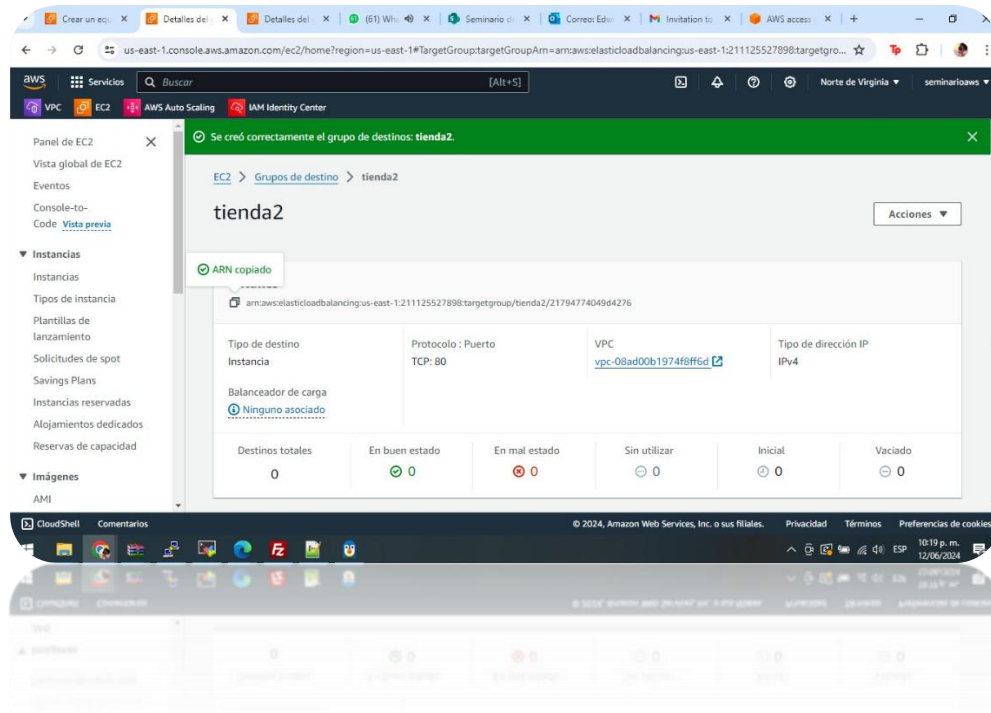
Creación del Contenedor



Nota. En el proceso anterior se crea el balanceador de carga. Wendy L, Flor M y Alejandro S, (2024).

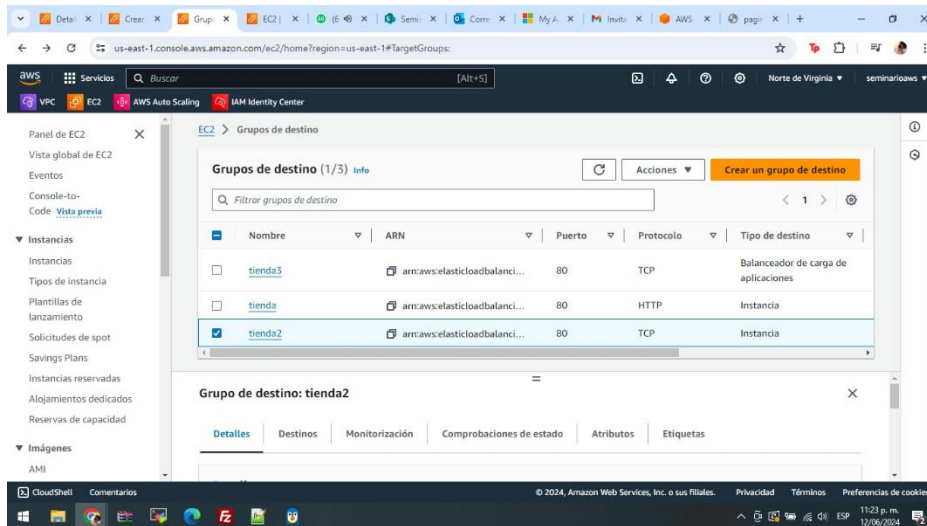
Figura 27

Grupo de destino



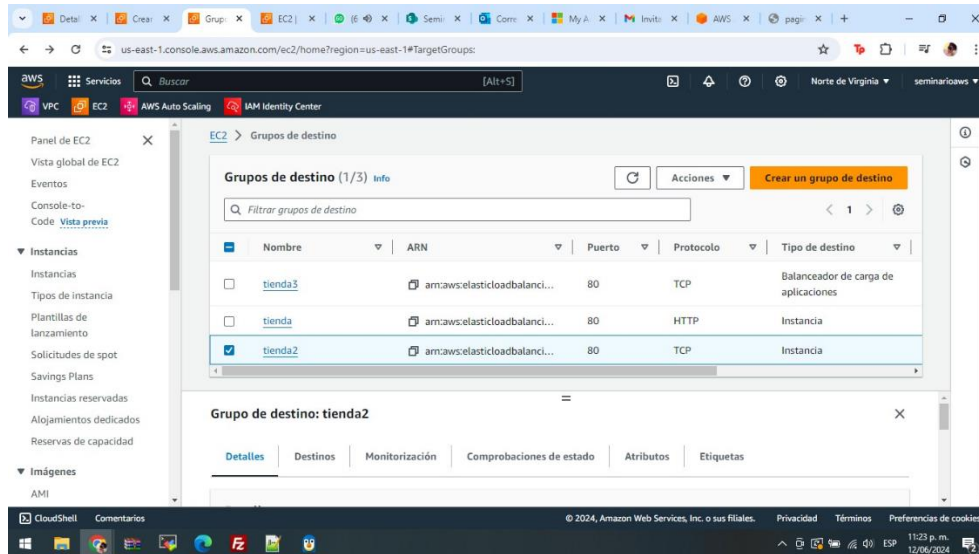
Nota. En el proceso anterior Creamos el grupo de destino. Wendy L, Flor M y Alejandro S, (2024).

Figura 27



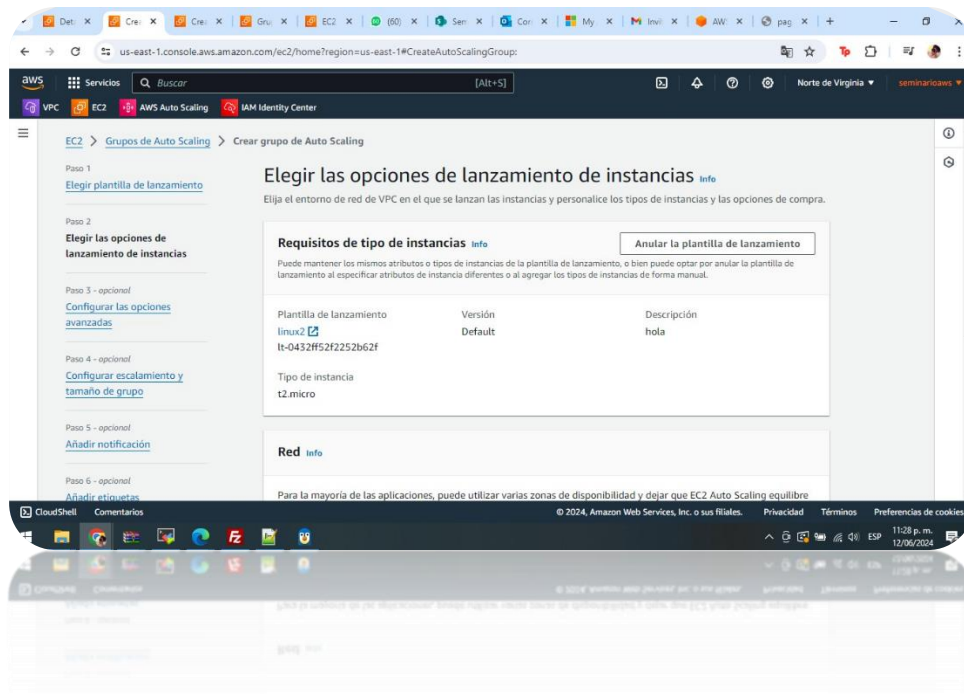
Nota. En el proceso anterior Creamos creación plantilla de lanzamiento Wendy L, Flor M y Alejandro S, (2024).

Figura 28



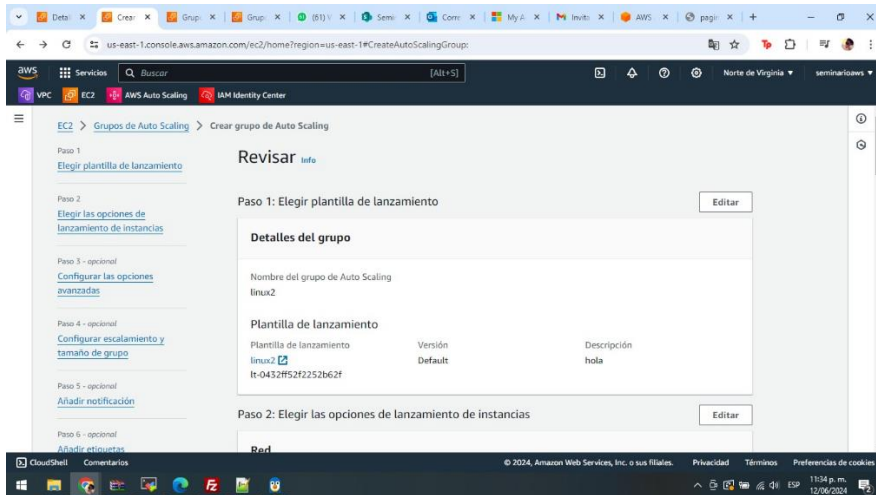
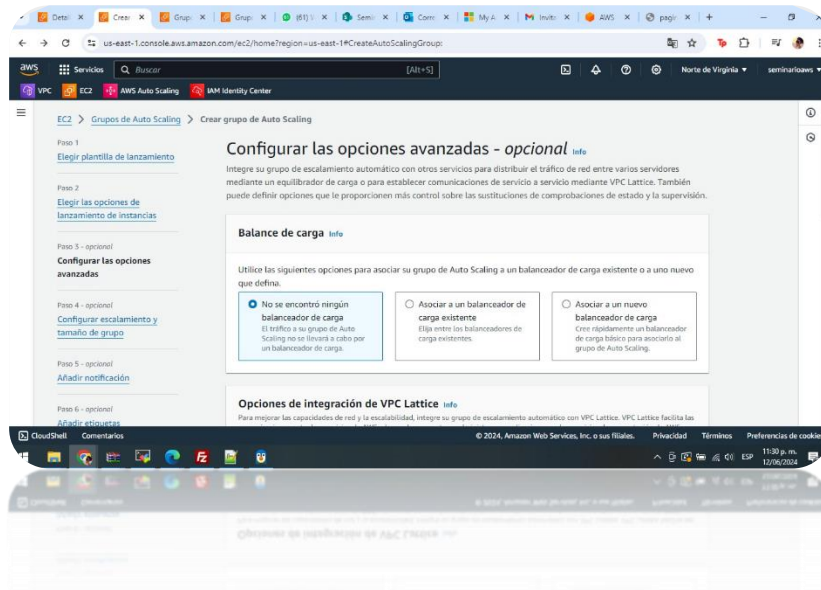
Nota. En el proceso anterior Creamos creacion plantilla de lanzamiento. Wendy L, Flor M y Alejandro S, (2024).

Figura 29



Nota. creacion de auto scaling groups Wendy L, Flor M y Alejandro S, (2024).

Figura 30



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#AutoScalingGroups:

linux2, 1 Balanceador de carga, 1 Grupo de destino, 1 Agente de escucha se creó correctamente. Se asoció 1 nuevo grupo de destino al ASG. La recopilación de métricas de grupo está habilitada.

EC2 > Grupos de Auto Scaling

Grupos de Auto Scaling (2) Info

Configuraciones de lanzamiento Plantillas de lanzamiento Acciones Crear grupo de escalado automático

Buscar sus grupos de Auto Scaling

<input type="checkbox"/>	Nombre	Plantilla de lanzamiento/config...	Instanc...	Estado	Capacidad des...	M...	M...
<input type="checkbox"/>	linux2	linux2 Versión Predeterminado	1	-	1	1	1
<input type="checkbox"/>	tienda.com	tienda.com	1	-	1	0	1

0 Grupos de escalado automático seleccionados

© 2024, Amazon Web Services, Inc. o sus filiales. Privacidad Términos Preferencias de cookies

11:34 p. m. 12/06/2024

Wendy L, Flor M y Alejandro S, (2024).

Conclusiones

Amazon web services AWS, proporciona servicios tecnológicos de baja demanda por medio de internet con precios bajos por su uso, De esa forma se hace más asequible para mover nuestras aplicaciones a la nube y construcción e implementación de tecnologías para la infraestructura, por ende se optimiza recursos informáticos, en cuanto al almacenamiento y bases de datos como relacionales entre otros.

AWS es líder en la industria de servicios en la nube, ofreciendo una amplia gama de servicios y soluciones. En su flexibilidad y escalabilidad permiten a las empresas u organizaciones adaptarse rápidamente a las demandas cambiantes del mercado.

En cuanto a la seguridad y fiabilidad de AWS son altamente valoradas por sus clientes. La capacitación y el soporte son aspectos importantes a considerar para maximizar los beneficiarios.

Referencias

Amazon, (s. f). Amazon.com. Recuperado el 12 de junio de 2024, de https://aws.amazon.com/es/rds/?gclid=CjwKCAjwjqWzBhAqEiwAQmtgT0DNicJvneeazC_z2bcTP2oJhF2Woz87p3THnjhxmlChM3adJWfWBoCh-gQAvD_BwE&trk=e3f15a34-5555-4b5c-882f-f779f28fda06&sc_channel=ps&ef_id=CjwKCAjwjqWzBhAqEiwAQmtgT0DNicJvneeazC_z2bcTP2oJhF2Woz87p3THnjhxmlChM3adJWfWBoCh-gQAvD_BwE:G:s&s_kwcid=AL!4422!3!648114139145!e!!g!!amazon%20relational%20db!19677234681!143348665582

Cloud computing: origen y evolución de la nube. (2017, Junio 12). cisga. <https://www.cisga.es/conoces-origen-del-cloud-computing-nube-informatica/>

Juan B. (2024). Amazon Web Services AWS. Corporación Universitaria Remington

Juan María, (2019). AWS vs Azure vs GCP: todos los servicios cloud frente a frente. Paradigmadigital.com. Recuperado el 12 de junio de 2024, de <https://www.paradigmadigital.com/dev/comparativa-servicios-cloud-aws-azure-gcp/>

LINEA DEL TIEMPO DE LA NUBE timeline. (s.f). Timetoast Timelines. Recuperado el 12 de junio de 2024, de <https://www.timetoast.com/timelines/nube>