



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

Implementación de servicios en AWS

**Presentado por:** Edwin Arvey Castillo Camacho

Se presenta a: Juan Pablo Berrio López

Corporación Universitaria Remington.  
Facultad de ingenierías  
Amazon web services  
19/05/2024

## Tabla de Contenidos

|  |   |
|--|---|
| Introducción.....                                | 3 |
| Marco conceptual y textual.....                  | 4 |
| Desarrollo e implementación del aprendizaje..... | 5 |
| Figuras y tablas.....                            | 6 |
| Conclusiones.....                                | 7 |
| Referencias.....                                 | 8 |

## Introducción

Este trabajo se va a realizar con el fin de saber, cómo fueron los orígenes en la virtualización y la computación en la nube, desarrollo del umbral de tiempo y la virtualización, que permitió a los usuarios compartir eficientemente los recursos de computación, también vamos a realizar una breve línea de tiempo. Se va a realizar un cuadro o comparativo sobre AWS, Azure, GCP. organiza en función de varios aspectos clave para ayudar a entender las diferencias y similitudes entre estas plataformas.

Los orígenes

Desde la década de los 60 y al principio de los 70 ARPANET creo una red de computadores que se conectaban entre dos PC desde ese entonces se originó el primer servicio en la nube, años más tardes principio de la década de los 80 llevo la virtualización para mejorar la tecnología que esto permite tener varios sistemas operativos en un mismo sistema que facilito a los usuarios compartir

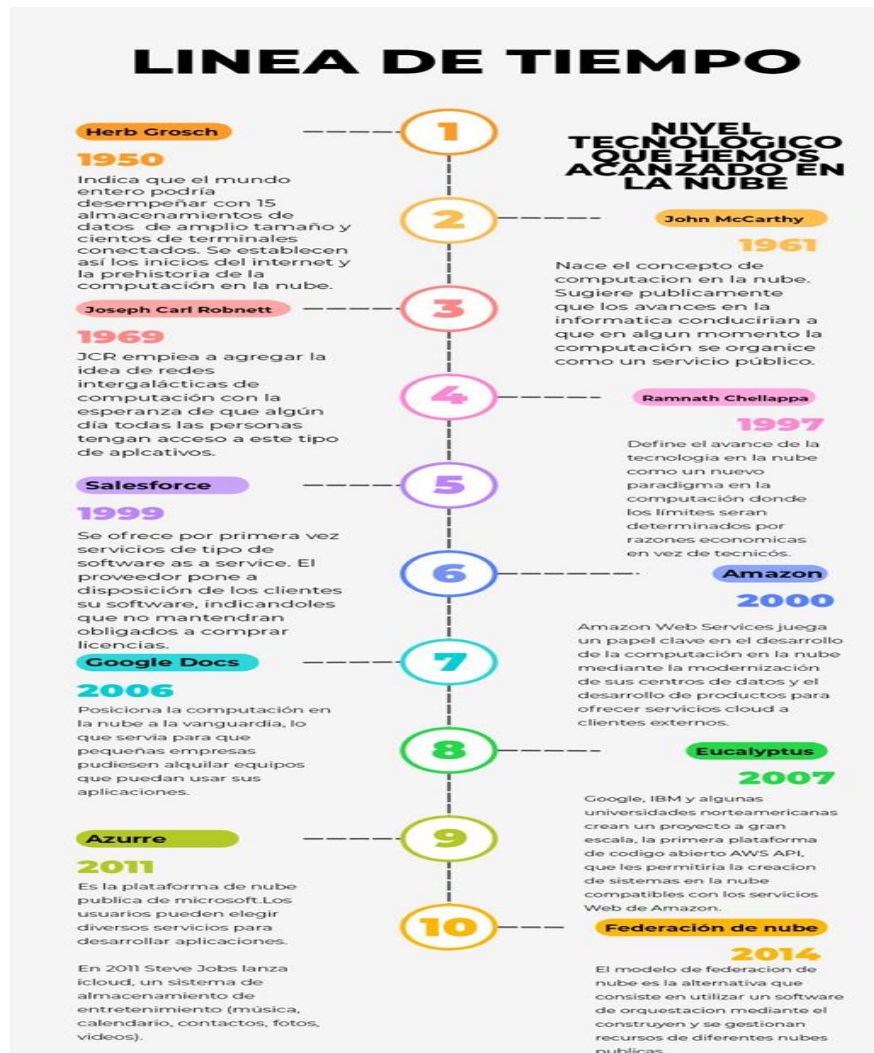
En el 2006 Jeff Bezos fue uno de los creadores de AWS, Marc Benioff dueño de SALESFORCE que es una empresa de software de nube para ofrecer a los clientes en relaciones de ventas y marketing

Cloud, M. (2024, 23 febrero). *Origen e historia de la computación en la nube.*

Mainframe Cloud. <https://mainframecloud.co/origen>

| <b>Características</b>                      | <b>AWS</b>  | <b>Azure</b>  | <b>Gcp</b>  |
|---|---|---|---|
| <b>lanzamiento</b>                          | 2006  | 2010  | 2008  |
| <b>empresa</b>                              | Amazon  | Microsoft   | Google  |
| <b>Composición con Materiales de Mejora</b> | Extensa afinidad con compuestos lenguajes y marcos de trabajo               | Excelente integración con Visual Studio, .Net, Azure Devops   | Buen soporte para Kubernetes y Docker, Cloud Funciones para serverless  |
| <b>Seguridad y Desempeño</b>                | Amplia escala de certificaciones y materiales de seguridad, IAM, AWS Shield | Alto guía en desempeño y gobernanza, Active Directory, Security Center                              | Certificaciones fuertes, IAM, Google Cloud Armor                        |
| <b>Almacenamiento para archivado</b>        | Acceso poco frecuente   | Almacenamiento fresco y archivo   | Línea fresca y línea fría   |
| <b>Almacenamiento híbrido</b>               | Puerta enlace de almacenamiento   | Almacenamiento simple   | N/A   |
| <b>Bases no racionales</b>                  | es una BD clave valor y de títulos que es ideal para aplicaciones           | Es una asistencia de base de datos NoSQL distribuido globalmente que acepta varios modelos de datos | servicio de base de datos de columnas amplias y crecidamente escalable. |
| <b>Almacenamiento en caché</b>              | Almacenamiento de archivos: Amazon Elastic File System (EFS).               | Almacenamiento de objeto, Almacenamiento de bloques   | Almacenamiento de objetos y bloques                                     |

<https://prezi.com/p/kqsoydompsjt/?present=1>



Para lograr que dos contenedores ejecutándose en una instancia EC2 respondan a nombres de dominio específicos sin especificar puertos en las consultas, se utilizar un servidor proxy inverso como Nginx para dirigir las solicitudes

1. Configuración DNS: toca configurar los registros DNS para apuntar los nombres de dominio específicos a la dirección IP pública de la instancia EC2.

2. Instalación de Nginx en la instancia EC2: conectarse a la instancia EC2 y asegurarse de que Nginx este instalado. Nginx se puede instalar en una instancia EC2 basada en Amazon Linux 2 utilizando el comando: ```bash sudo amazon-Linux-extras install nginx1```
3. Configuración de Nginx:
  - se crea dos archivos de configuración en la carpeta ``/etc/nginx/conf.d/``, uno por cada nombre de dominio específico.
  - En cada archivo de configuración, se configura el servidor virtual de Nginx para poder redirigir las solicitudes entrantes al contenedor que corresponde.
  - Toca que asegurarse de que los servidores virtuales estén configurados para los puertos 80 y/o 443
4. Se reinicia Nginx: una vez que este configurado los archivos Nginx, se reinicia el servidor de Nginx para que se haga los cambios: ```bash sudo systemctl restart nginx ```
5. Prueba de los dominios: se probará los nombres de dominio específicos en un navegador web sin especificar puertos. Nginx se dirige a la solicitudes de contenedores correspondientes basándose en la configuración que se estableció

“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.”

“Un *equilibrador de carga* actúa como único punto de contacto para los clientes. El equilibrador de carga distribuye el tráfico entrante de aplicaciones entre varios destinos, tales como instancias EC2, en varias zonas de disponibilidad. Esto aumenta la disponibilidad de la aplicación. Puede agregar uno o varios oyentes al equilibrador de carga.”

“Un *oyente* comprueba las solicitudes de conexión de los clientes mediante el protocolo y el puerto configurados. Las reglas que defina para un oyente determinan cómo el equilibrador de carga va a direccionar las solicitudes a sus destinos registrados. Cada regla consta de una prioridad, una o más acciones y una o más condiciones. Cuando se cumplen las condiciones de una regla, se llevan a cabo sus acciones. Debe definir una regla predeterminada para cada oyente y, si lo desea, puede definir reglas adicionales.”

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

¿Qué es un Equilibrador de carga de aplicación? - Elastic Load Balancing. (s. f.).

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

**Reglas de entrada** Información

| ID de la regla del grupo de seguridad | Tipo <small>Información</small> | Protocolo <small>Información</small> | Intervalo de puertos <small>Información</small> | Origen <small>Información</small> | Descripción: opcional <small>Información</small> |          |
|---------------------------------------|---------------------------------|--------------------------------------|---|-----------------------------------|--|----------|
| sgr-0812ad107ff0eed20                 | HTTP                            | TCP                                  | 80  | Perso...<br>0.0.0.0/0 X           |  | Eliminar |
| sgr-030f3fa53800ee47c                 | RDP                             | TCP                                  | 3389  | Perso...<br>0.0.0.0/0 X           |  | Eliminar |

Agregar regla

**VPC** Mostrar detalles  
Su red virtual de AWS

proyecto-vpc

**Subredes (6)**  
Subredes dentro de esta VPC

**us-east-2a**

- proyecto-subnet-public1-us-east-2a
- proyecto-subnet-private1-us-east...

**us-east-2b**

- proyecto-subnet-public2-us-east-2b
- proyecto-subnet-private2-us-east...

**us-east-2c**

- proyecto-subnet-public3-us-east-2c
- proyecto-subnet-private3-us-east...

**Tablas de enrutamiento (5)**  
Dirigir el tráfico de red a los recursos

- proyecto-rtb-private3-us-east-2c
- rtb-0176a4c7ea79d5c59
- proyecto-rtb-private2-us-east-2b
- proyecto-rtb-public
- proyecto-rtb-private1-us-east-2a

## Tablas de enrutamiento (5)

Dirigir el tráfico de red a los recursos

proyecto-rtb-private3-us-east-2c

rtb-0176a4c7ea79d5c59

proyecto-rtb-private2-us-east-2b

proyecto-rtb-public

proyecto-rtb-private1-us-east-2a

## Conexiones de red (2)

Conexiones a otras redes

proyecto-igw

proyecto-vpce-s3

Windows Server

## Internet Information Services

Welcome

Bienvenue

Tervetuloa

ようこそ

Benvenuto

歡迎

Bienvenido

Hoş geldiniz

ברוכים הבאים

Welkom

Bem-vindo



Bienvenido

Hoş geldiniz

ברוכים הבאים

Καλώς  
ορίσαστε

Välkommen

환영합니다

Добро  
пожаловать

Üdvözöljük

Vítejte

Καλώς  
ορίσαστε

Välkommen

환영합니다

Добро  
пожаловать

Üdvözöljük

مرحبا  
歡迎

Microsoft

Willkommen

Velkommen



Witamy

### Reglas de entrada Información

| ID de la regla del grupo de seguridad | Tipo <small>Información</small> | Protocolo <small>Información</small> | Intervalo de puertos <small>Información</small> | Origen <small>Información</small>       | Descripción: opcional <small>Información</small> |          |
|---------------------------------------|---------------------------------|--------------------------------------|---|---|--|----------|
| sgr-0812ad107ff0eed20                 | HTTP                            | TCP                                  | 80  | Perso... <input type="text" value="Q"/> | <input type="text"/>                             | Eliminar |
| sgr-030f3fa53800ee47c                 | RDP                             | TCP                                  | 3389  | Perso... <input type="text" value="Q"/> | <input type="text"/>                             | Eliminar |

**ID de la instancia**  
i-0b9c02c3fdbcf0173 (tareaservicio1)

**Dirección IPv6**

**Nombre de instancia**  
t2.micro

**Nombre de nombre de anfitrión**  
Nombre de IP: ip-172-31-32-58.us-east-2.compute.internal

**Responder al nombre DNS de recurso privado IPv4 (A)**  
Dirección IP asignada automáticamente: 3.14.14.192 [IP pública]

**Dirección IPv4 pública**  
3.14.14.192 | dirección abierta

**Estado de la instancia**  
En ejecución

**Nombre DNS de IP privada (solo IPv4)**  
ip-172-31-32-58.us-east-2.compute.internal

**Tipo de instancia**  
t2.micro

**ID de VPC**  
vpc-00441918281fdc41b

**Direcciones IPv4 privadas**  
172.31.32.58

**DNS de IPv4 pública**  
ec2-3-14-14-192.us-east-2.compute.amazonaws.com | dirección abierta

**Direcciones IP elásticas**  
-

**Hallazgo de AWS Compute Optimizer**  
Suscribirse a AWS Compute Optimizer para recibir recomendaciones.  
[Más información](#)





```
[root@ip-172-31-36-81 ec2-user]# docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:266b191e926f65542fa8daaec01a192c4d292bff79426f47300a046e1bc576fd
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
[root@ip-172-31-36-81 ec2-user]# docker images
REPOSITORY      TAG          IMAGE ID       CREATED        SIZE
hello-world     latest      d2c94e258dcb  13 months ago  13.3kB
[root@ip-172-31-36-81 ec2-user]# docker run hello-world
```

Hello from Docker!  
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub. (amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:  
\$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:  
<https://hub.docker.com/>

For more examples and ideas, visit:  
<https://docs.docker.com/get-started/>

```
[root@ip-172-31-36-81 ec2-user]# systemctl stop docker
Warning: Stopping docker.service, but it can still be activated by:
docker.socket
```

```
[root@ip-172-31-36-81 /]# cd tmp/
[root@ip-172-31-36-81 tmp]# ls
systemd-private-0a4577b81e04411b9bf62b3522c90f72-chronyd.service-dWV6wR
systemd-private-0a4577b81e04411b9bf62b3522c90f72-dbus-broker.service-IGx424
systemd-private-0a4577b81e04411b9bf62b3522c90f72-policy-routes@enX0.service-LxMCGl
systemd-private-0a4577b81e04411b9bf62b3522c90f72-systemd-logind.service-b7QBFg
systemd-private-0a4577b81e04411b9bf62b3522c90f72-systemd-resolved.service-h00eLb
[root@ip-172-31-36-81 tmp]# mkdir contenedor
bash: mkdir: command not found
[root@ip-172-31-36-81 tmp]# mkdir contenedor
[root@ip-172-31-36-81 tmp]# cd contenedor/
[root@ip-172-31-36-81 contenedor]# ls
[root@ip-172-31-36-81 contenedor]# pwd
/tmp/contenedor
[root@ip-172-31-36-81 contenedor]# docker run -dit --name contenedor1 -p 8080:80 -v /home/catherine:/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-172-31-36-81 contenedor]# docker run -dit --name contenedor1 -p 8080:80 -v /home/catherine:/usr/local/apache2/htdocs/ httpd
Docker: Error response from daemon: invalid volume specification: '/home/catherine:/usr/local/apache2/htdocs/': invalid mount config for type "bind": invalid mount path: 'usr/local/apache2/htdocs/' mount path must be absolute.
See 'docker run --help'.
[root@ip-172-31-36-81 contenedor]# docker run -dit --name contenedor1 -p 8080:80 -v /home/catherine:/usr/local/apache2/htdocs/ httpd
29d89f2921695ff4cba2805e6fdab81eabd6ba8902b1e2b9463341b0f8970467
[root@ip-172-31-36-81 contenedor]# docker images
REPOSITORY      TAG          IMAGE ID       CREATED        SIZE
httpd           latest      356125da0595   7 weeks ago    147MB
hello-world     latest      d2c94e258dcb  13 months ago  13.3kB
[root@ip-172-31-36-81 contenedor]# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
29d89f292169   httpd    "httpd-foreground"      38 seconds ago Up 37 seconds  0.0.0.0:8080->80/tcp, :::8080->80/tcp  contenedor1
[root@ip-172-31-36-81 contenedor]#
```

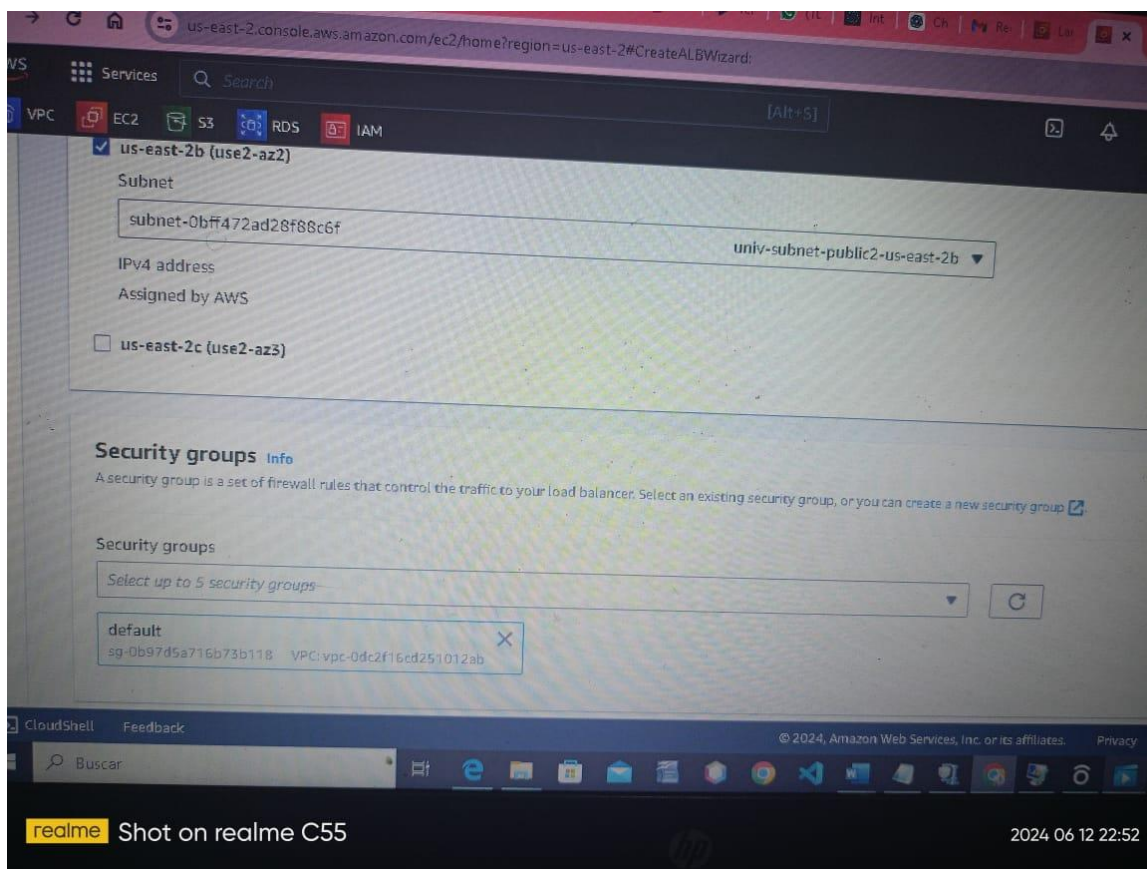
# ¡Funciona!

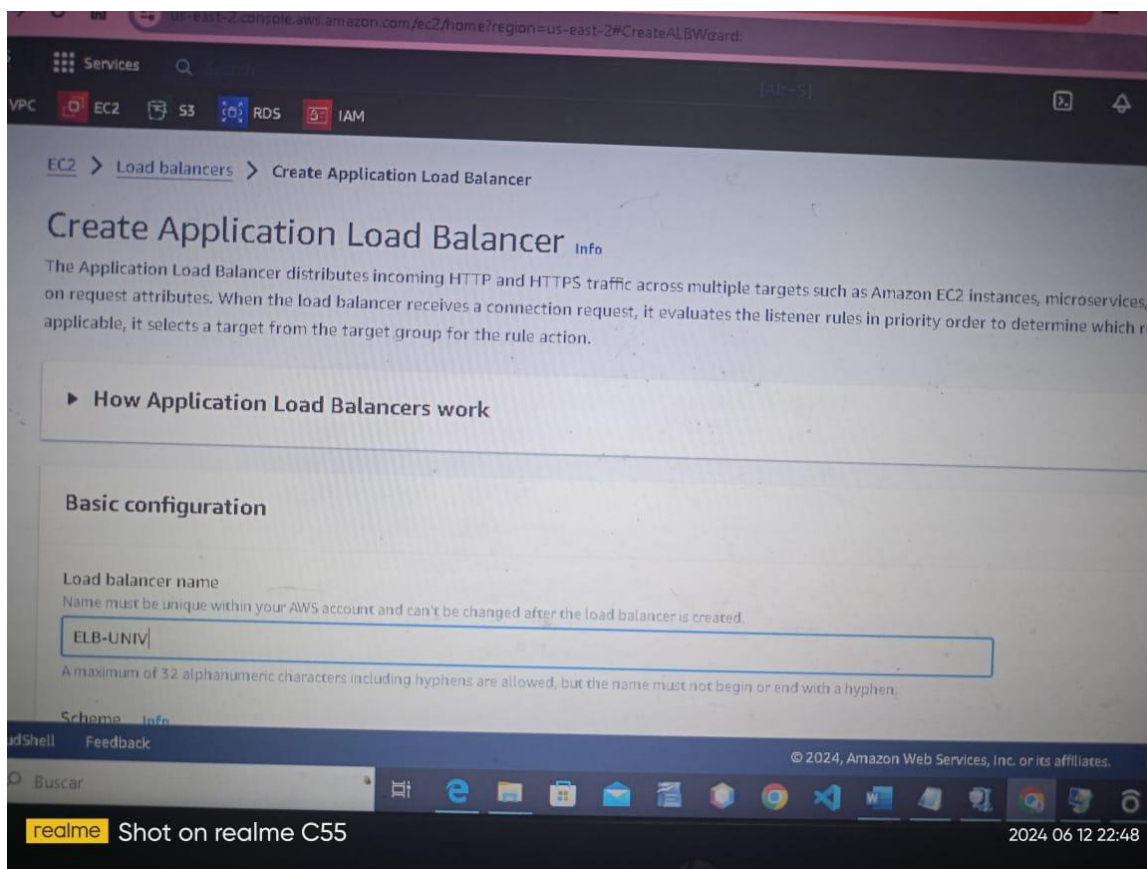
Video

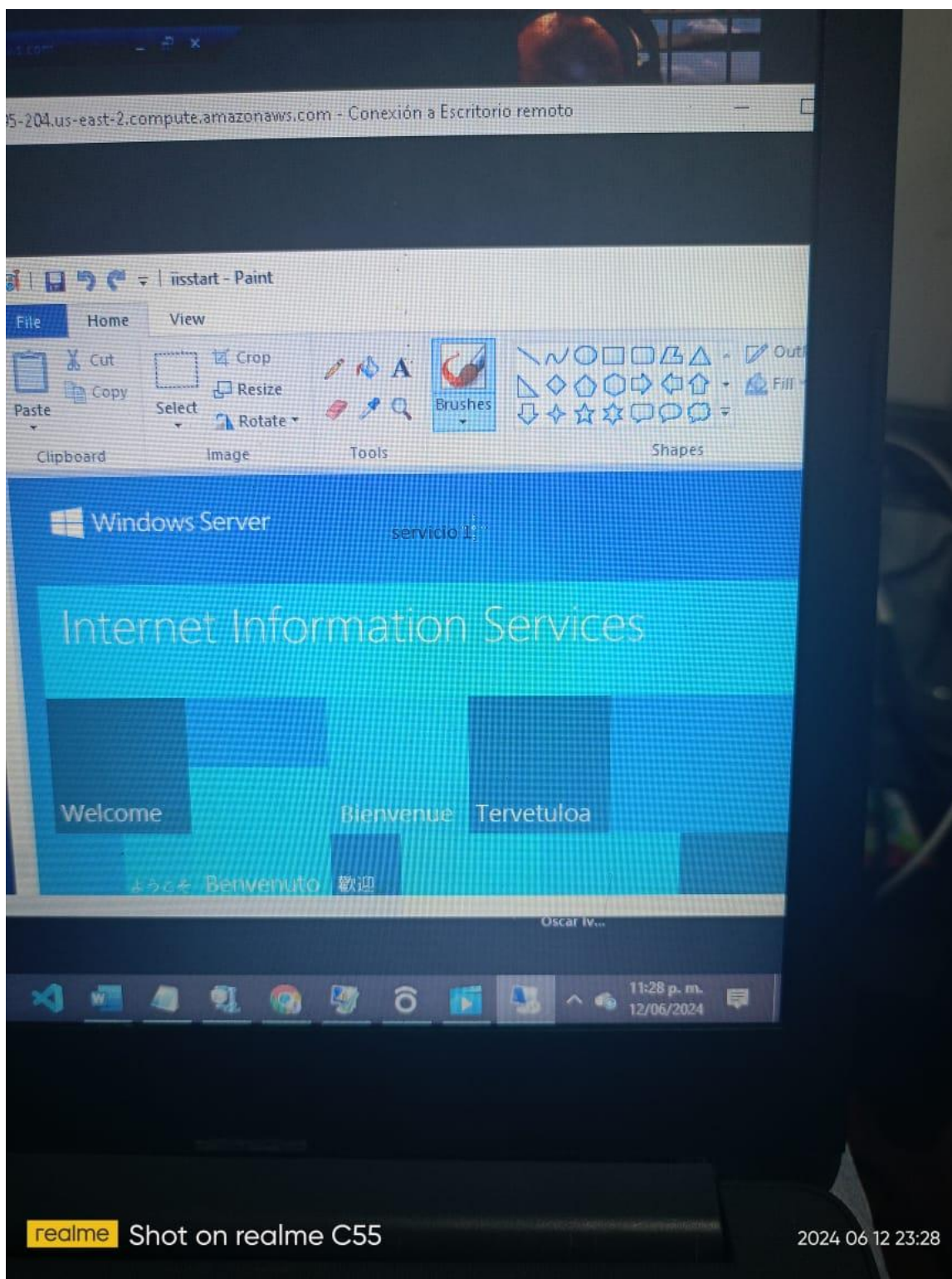
[https://remingtonedu-my.sharepoint.com/:v:/g/personal/jeimy\\_giraldo\\_9109\\_miremington\\_edu\\_co/EeRNhysqr11IjMxLRKGVf14BPKscImad97mWnmlkeEAwgw?referrer=Teams.TEAMS-ELECTRON&referrerScenario=MeetingChicletGetLink.view.view](https://remingtonedu-my.sharepoint.com/:v:/g/personal/jeimy_giraldo_9109_miremington_edu_co/EeRNhysqr11IjMxLRKGVf14BPKscImad97mWnmlkeEAwgw?referrer=Teams.TEAMS-ELECTRON&referrerScenario=MeetingChicletGetLink.view.view)

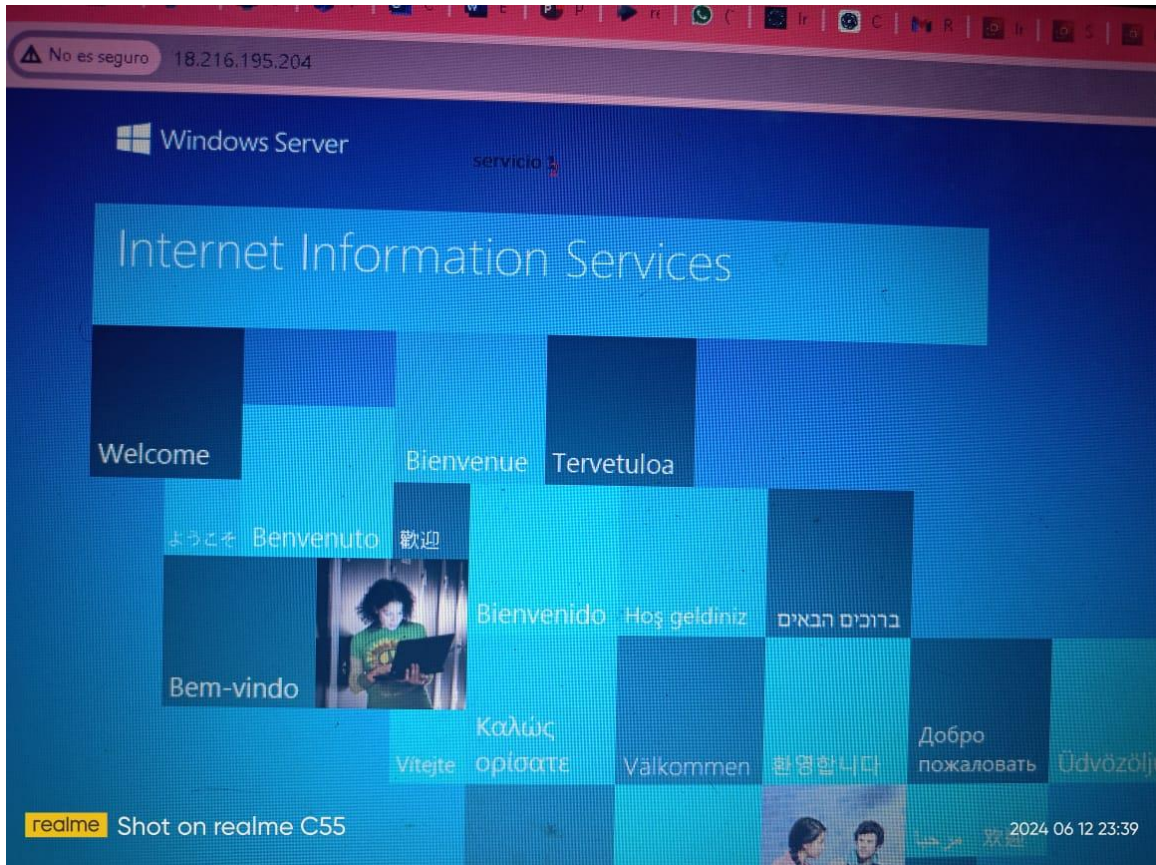
## **Configuración del Balanceador de Carga (ELB)**

El balanceador de carga es necesario tener mínimo dos instancia para así poder tener sentido lo que es un balanceador de carga la ventaja es que cuando una maquina falla por defecto la otra sigue funcionando





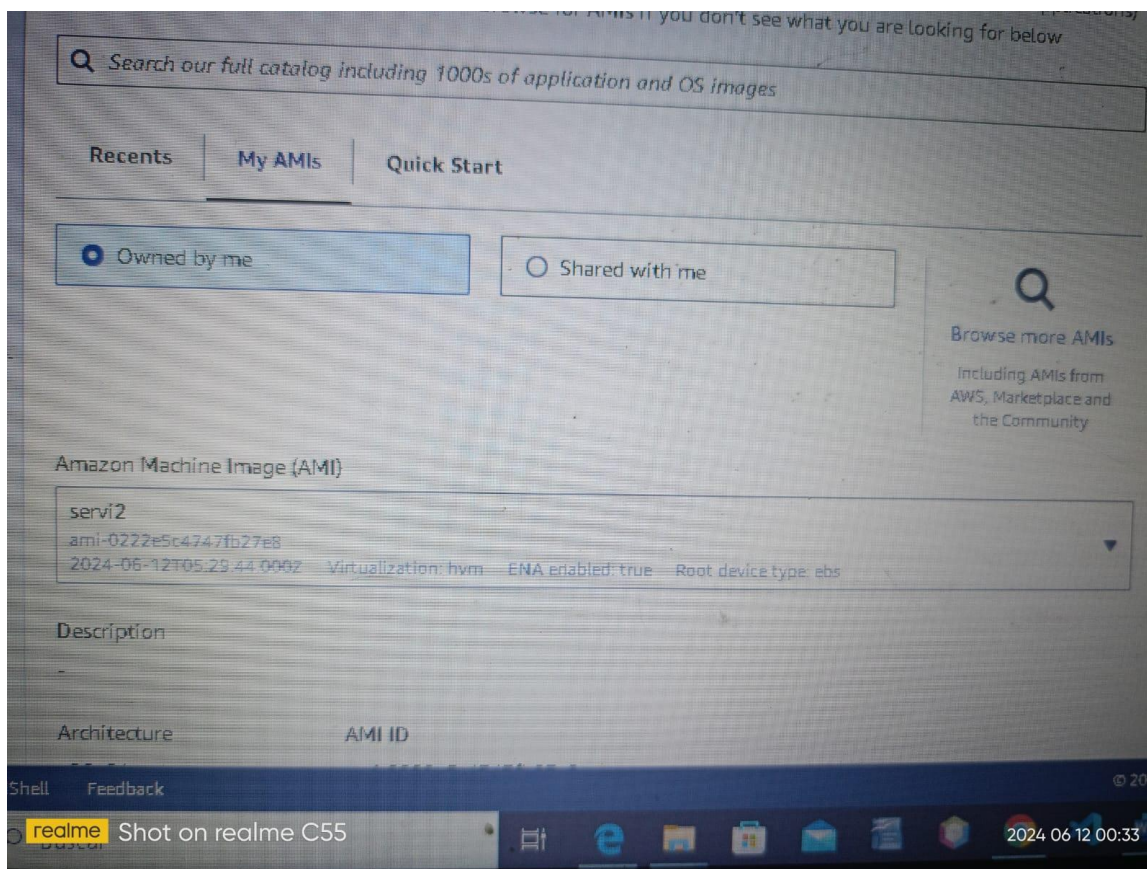


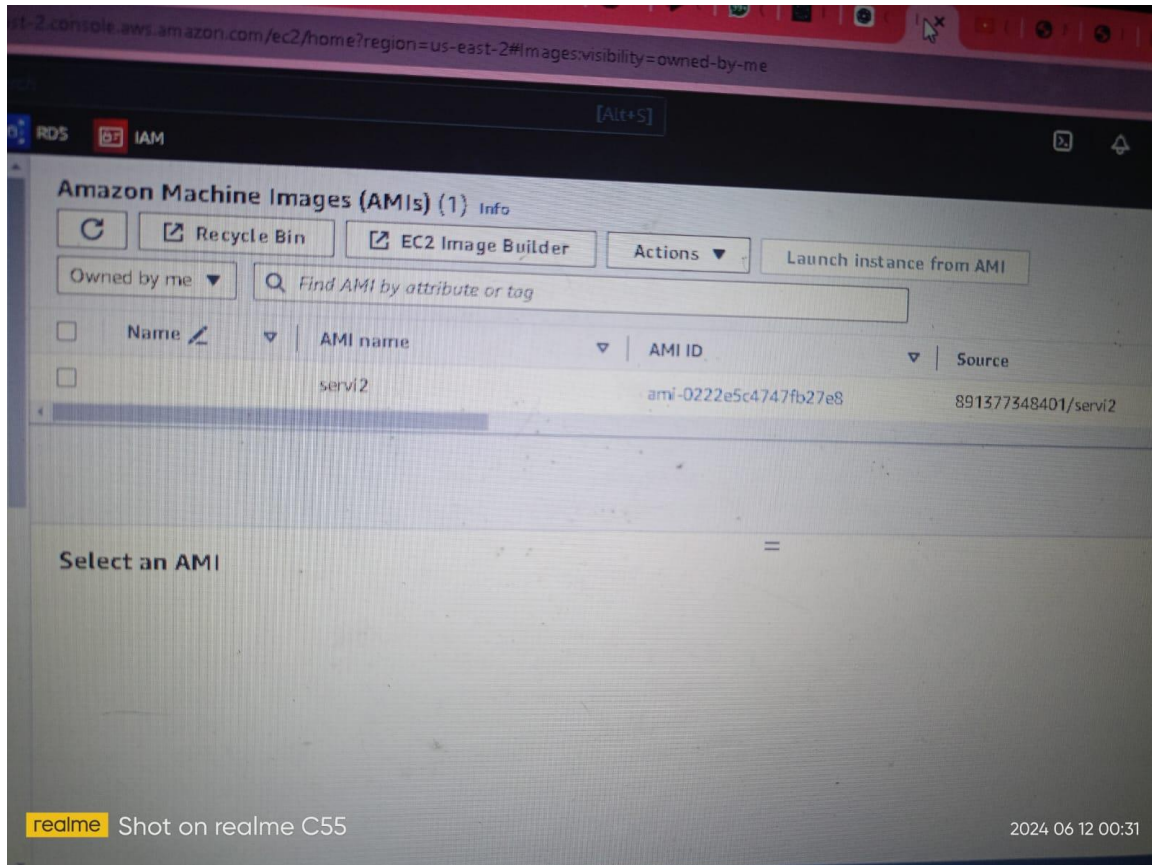


Para crear un par de clave SSH es necesario tener una aplicación putty.  
 Cuando estamos creando una instancia EC2 la damos crear clave

Lanzamiento de instancia EC2: iniciamos

Para lanzar una instancia EC2 para inicial utilizando una AMI Windows: podemos escoger una instancia ya creada, esa misma instancia le creamos una imagen y con el mismo nombre de la instancia que en ese momento es una instancia EC2 Windows. Nos dirigimos en la consola EC2 le damos lanzar instancia no escogemos ningún sistema operativo solo le apuntamos en MY AMIS y automáticamente nos crea la instancia EC2windows





## Conclusiones

En este trabajo se realizaron diferentes tipos de investigaciones como los orígenes de la virtualización, y todo sobre la computación en la nube. Abarcamos todo en una línea de tiempo sobre lo que es hoy en día la nube y las empresas, sistemas, y aplicaciones que se involucraron en el desarrollo de esta. Se evidencia una comparación entre aws, azure, gcp desde su característica hasta su almacenamiento de cada una de estos servidores.



### Referencias

- (Fiz, J. M. (2022, 7 noviembre). AWS vs Azure vs GCP: todos los servicios cloud frente a frente. *Paradigma Digital*.  
<https://www.paradigmadigital.com/dev/comparativa-servicios-cloud-aws-azure-gcp/>
- Castellanos, C. R. R. (2023, 12 noviembre). Bases no relacionales de AWS, GCP y Azure - Carlos Rogelio Reyes Castellanos - Medium. *Medium*.  
<https://medium.com/@carlosreyes2008/bases-no-relacionales-de-aws-gcp-y-azure-534f4448aefb>