



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

**Implementación de AWS en Ámbito Empresarial Tecnológico**

Corporación Universitaria Remington.  
Facultad de Ingeniería de Sistemas  
Seminario AWS Cloud

Miguel Angel Paz Fuentes

Juan Pablo Berrio  
Opción de Trabajo de grado Seminario-Diplomado.  
2024

## Tabla de Contenidos

Resumen.....	3
Marco conceptual y contextual .....	4
Decisión Estratégica de Infraestructura IT .....	4
Costo: .....	4
Datacenter Local. ....	4
Nube.....	4
Escalabilidad.....	5
Datacenter Local. ....	5
Nube.....	5
Seguridad y Cumplimiento .....	5
Datacenter Local. ....	5
Nube.....	5
Mantenimiento y Operaciones .....	6
Datacenter Local. ....	6
Nube.....	6
Desempeño y Conectividad .....	6
Datacenter Local. ....	6
Nube.....	6
Medidas de Mitigación: .....	6
Escalabilidad y Flexibilidad.....	7
Datacenter Local. ....	7
Nube.....	7
Estrategias para Manejar Picos de Demanda y Variabilidad en las Necesidades de Recursos .....	7
Desarrollo e implementación del aprendizaje.....	9
Conclusiones .....	21
Referencias.....	21

## **Resumen**

La plataforma de computación en la nube Amazon Web Services, es una de las más completas a nivel global, la cual, sin ninguna duda, ha transformado de manera significativa la forma en que las empresas funcionan. Con el avance de esta tecnología, han logrado brindar acceso a una amplia gama de servicios, a precios exequibles a la población en general y con la suficiente escalabilidad para adaptarse a las necesidades de cualquier negocio

Entre los servicios de infraestructura más significativos de AWS, encontramos el alquiler de servidores virtuales para ejecutar sistemas operativos y aplicaciones; el servicio de almacenamiento que nos permite guardar datos de forma escalable y segura; gestión de base de datos relacionales y la creación de redes virtuales seguras.

Seguidamente, y en lo que a ventajas concierne, se evidencia que AWS es ágil, escalable, confiable e innovador; en cuanto a la relación costo-efectividad, solamente se paga por los recursos que se utilizan, sin costos preliminares ni compromisos prolongados.

Es por esto AWS está siendo utilizado por millones de empresas a nivel global, ofreciendo a las compañías todo lo necesario para avanzar en la era digital, consolidándose como una plataforma voluble y poderosa.

## **Palabras clave**

Amazon Web Service (AWS), Amazon Simple Storage Service (Amazon S3), Amazon Elastic Compute Cloud (EC2), Contenedor, balanceador.

## **Marco conceptual y contextual**

TechSolutions S.A. es una empresa de mediana escala la cual tiene alrededor de 150 trabajadores, de los cuales 50 son desarrolladores de software que requieren un entorno flexible y robusto, especializándose en el desarrollo de software a medida y soluciones tecnológicas para diversas industrias. La empresa actualmente tiene una infraestructura IT básica, pero está percibiendo un rápido crecimiento; lo cual, he hecho que la empresa se plantee como desafío montar un nuevo datacenter local o migrar sus servicios a la nube.

Para la toma de decisiones, la empresa evalúa factores determinantes en los requisitos y consideraciones como los costos, la escalabilidad, la seguridad y cumplimiento, los mantenimiento y operaciones, y el desempeño y conectividad: además, analizara los riesgos asociados a cada opción, incluyendo cumplimiento de normativas y posibles vulnerabilidades; proponiendo a su vez medidas para mitigar estos posibles riesgos.

### **Decisión Estratégica de Infraestructura IT**

#### ***Costo:***

##### **Datacenter Local.**

Los riesgos al optar por un Datacenter Local, están asociados con la adquisición de hardware robustos, costos de instalación, acondicionamiento del espacio físico, preservación y seguridad del espacio y de todos los dispositivos; pero a su vez, estos costos operativos podrían ser controlados a largo plazo dependiendo como se gestione.

##### **Nube.**

*Los costos iniciales serán bajos al estar ligados al pago por uso, pero con eventuales variaciones a medida que aumenten las exigencias de almacenamiento y recursos.*

## **Escalabilidad**

### **Datacenter Local.**

Limitada por el hardware y capacidad física disponible, lo cual a su vez esta correlacionada con posibles tramites de órdenes de compra, documentación legal, tiempo y disponibilidad de los dispositivos, lo cual hace que su expansión requiriera financiamientos adicionales en infraestructura.

### **Nube.**

Al ser fácilmente accesible y ofrecer la ventaja de poder añadir o reducir sus recursos en función de las necesidades del momento, no requeriría una inversión inicial adicional, lo cual hace que sea altamente escalable.

## **Seguridad y Cumplimiento**

### **Datacenter Local.**

Frente a la seguridad y cumplimiento al optar por un Datacenter Local, se deberá tener cuidado con las configuraciones que se realizan controlando totalmente la seguridad física y lógica, toda vez que estos podrían estar accesibles desde cualquier lugar al estar conectados a internet, lo que se podría traducir en pérdidas económicas si no se gestiona correctamente.

### **Nube.**

Al igual que con el Datacenter Local, si no se gestiona correctamente podría tener implicaciones en los costos y aspectos financieros de la empresa al estar enlazados a la red.

## **Mantenimiento y Operaciones**

### **Datacenter Local.**

Requiere invertir en equipos internos especializados en el cuidado y respaldo de la infraestructura (aires acondicionados, equipos físicos, etc).

### **Nube.**

Los costos serian nulos, toda vez que la gestión de mantenimiento de la infraestructura subyacente, es responsabilidad del proveedor de servicios en la nube.

## **Desempeño y Conectividad**

### **Datacenter Local.**

Si bien el rendimiento es altamente manejable y ajustable para cumplir con las necesidades particulares de la empresa, se deberá contar con al menos dos proveedores de internet, a fin de tener alternativa por si uno de los dos se cae, lo cual influye en los costos de operación de la empresa.

### **Nube.**

Teniendo en cuenta que el proveedor es quien brinda directamente los servicios, el desempeño y la conectividad no deberían ser una preocupación. Aunque si hablamos directamente del proveedor local de internet, el rendimiento estará vinculado a la conexión a internet y al proveedor de servicios, y la latencia será un factor que se debería tener en cuenta.

### ***Medidas de Mitigación:***

- Aplicar rigurosas medidas de seguridad física, como controles de acceso y supervisión continua.

- Gestionar sólidas medidas de seguridad informática, como cortafuegos, antivirus y sistemas para detectar intrusiones.
- Capacitar y especializar a los empleados en el campo de normativo, con el fin de asegurar el acatamiento de las normativas correspondientes.

## **Escalabilidad y Flexibilidad**

### **Datacenter Local.**

Escalabilidad Horizontal: Además de ser limitada, requiere de inversiones en equipamiento y expansión física del centro de datos; además, puede resultar costosa en el futuro, dado que implica la modernización de elementos ya en uso.

Aunque a su vez, podría genera más control y ofrecer la posibilidad de personalizar la infraestructura y ajustarla a las necesidades específicas de la empresa.

### **Nube.**

*Facilita la ampliación rápida de la capacidad de procesamiento y almacenamiento de información, a través de la provisión de recursos extra en la nube; además, ofrece la posibilidad de adaptar de forma dinámica la infraestructura de tecnología de la información a las demandas cambiantes del negocio, pagando únicamente por los recursos empleados.*

### ***Estrategias para Manejar Picos de Demanda y Variabilidad en las Necesidades de Recursos***

- Realizar un seguimiento constante y análisis del desempeño de la infraestructura y las aplicaciones.
- Introducir servicios de escalado automático para adecuar los recursos según la demanda.
- Promover una mentalidad de optimización de recursos en la empresa.

**Tabla 1***Impacto en el personal y operaciones*

<b>Factor</b>	<b>Datacenter Local</b>	<b>Nube</b>
<b>Personal IT</b>	Mayor necesidad de personal técnico	Menor necesidad de personal técnico
<b>Carga de trabajo</b>	Mayor carga de trabajo para el equipo de TI	Menor carga de trabajo para el equipo de TI
<b>Habilidades</b>	Habilidades en administración de datacenters	Habilidades en tecnologías de nube
<b>Seguridad</b>	Mayor control sobre la seguridad	Sólidas medidas de seguridad por parte del proveedor
<b>Disponibilidad</b>	Depende de la infraestructura local	Alta disponibilidad por redundancia en la nube
<b>Operaciones</b>	Complejidad operativa	Operaciones simplificadas
<b>Escalabilidad</b>	Escalabilidad limitada	Escalabilidad elástica
<b>Costos</b>	Inversión inicial significativa	Modelo de pago por uso
<b>Mantenimiento</b>	Requiere un equipo de TI dedicado	Mantenimiento a cargo del proveedor
<b>Monitoreo</b>	Monitoreo manual	Monitoreo automatizado
<b>Integración</b>	Posiblemente compleja	Posiblemente compleja

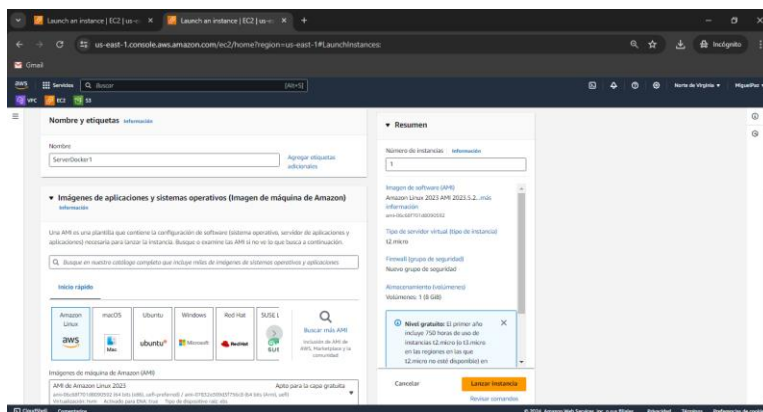


## Desarrollo e implementación del aprendizaje

Se presentan los resultados obtenidos o generados del presente informe técnico, describiendo la ejecución de lo aprendido en el curso en un entorno o contexto, implementando una instancia EC2 con dos contenedores, los cuales contienen un balanceador de carga.

### Figura 1

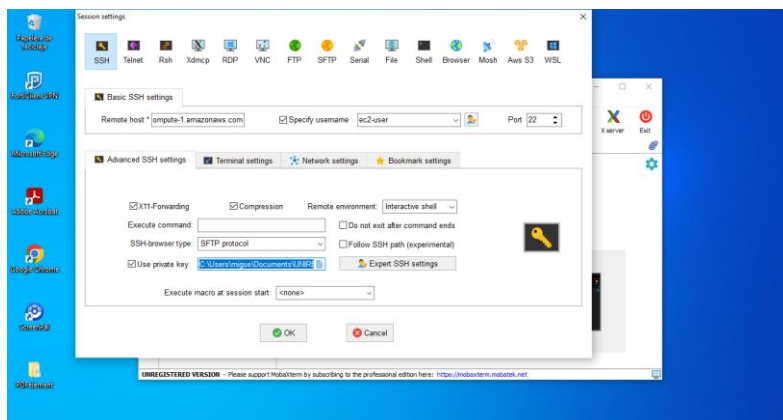
*Creación del recurso EC2 donde se instaurará la máquina virtual o instancia.*



*Nota.* Se monto máquina virtual en Linux Amazon, en una instancia t2.micro y con configuración de red pública.

### Figura 2

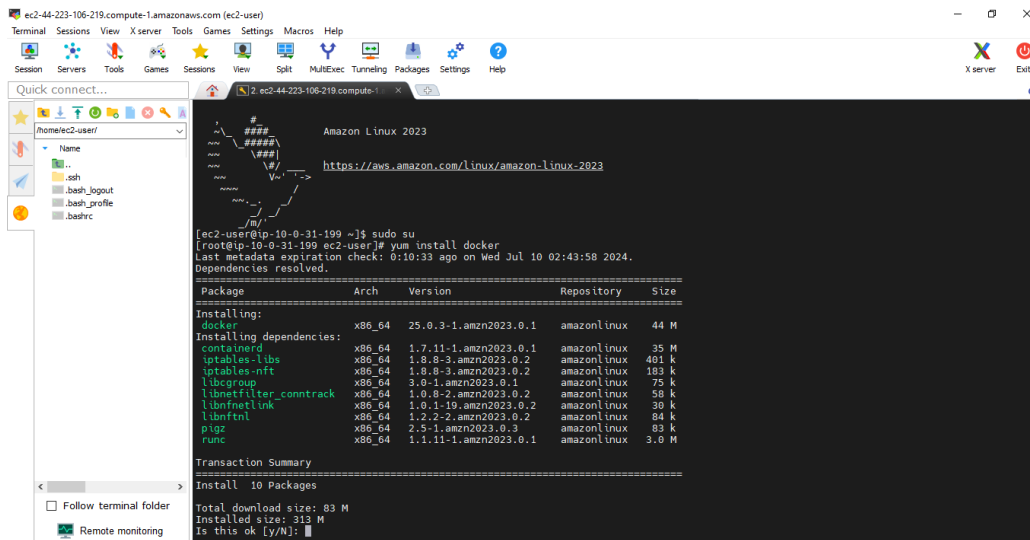
*Creación del recurso EC2 donde se instaurará la máquina virtual o instancia.*



*Nota.* Usamos gestor de conexiones MobaXterm para contactarnos a nuestro servidor.

Figura 3

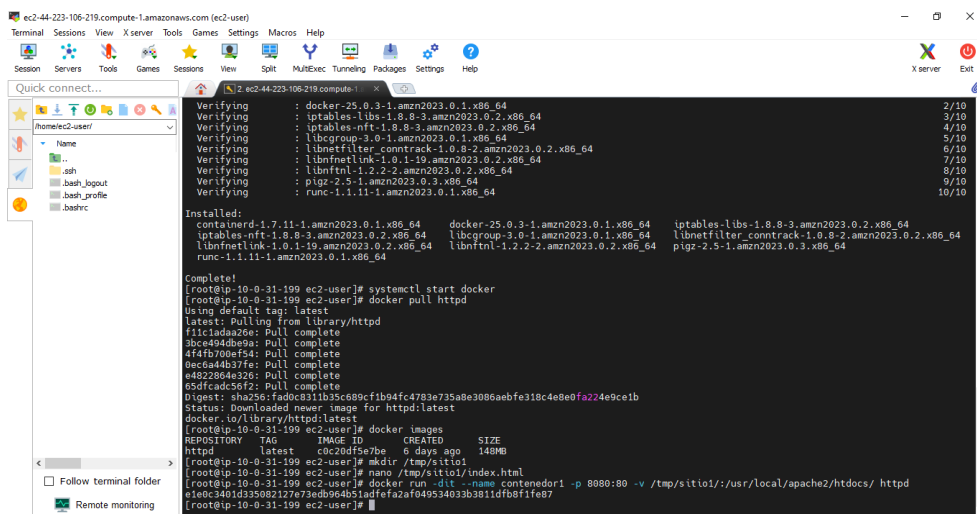
*Logo como administrador*



*Nota.* Ilustración del logeo como administrador e instalación de la plataforma de Docker sobre Linux.

Figura 4

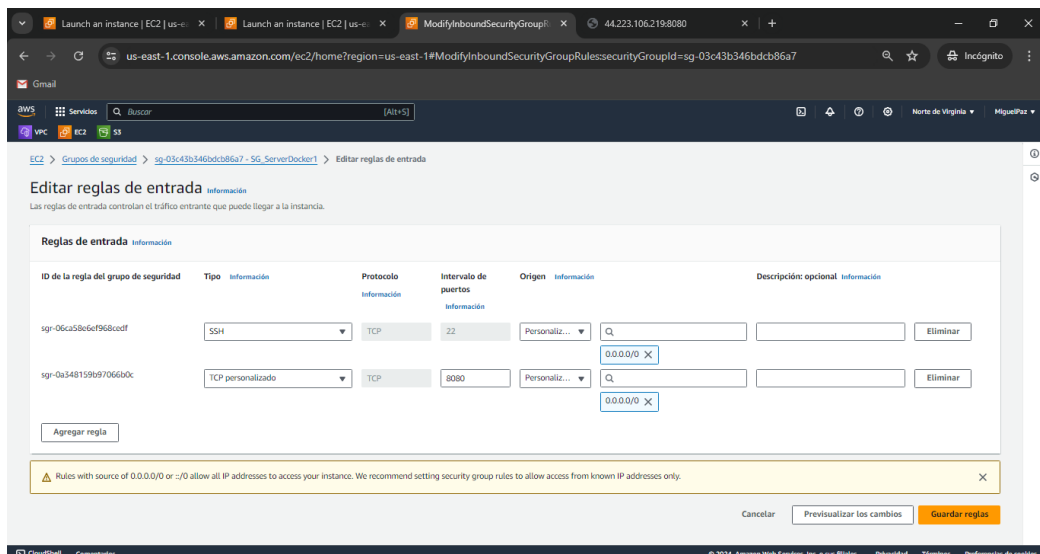
*Instalación de servidor web con un apache y creación de contenedor*



*Nota.* Iniciamos la aplicación con systemctl start Docker y montamos un servidor web con apache; seguidamente creamos el Primer Contenedor para que se ejecute con la imagen.

Figura 5

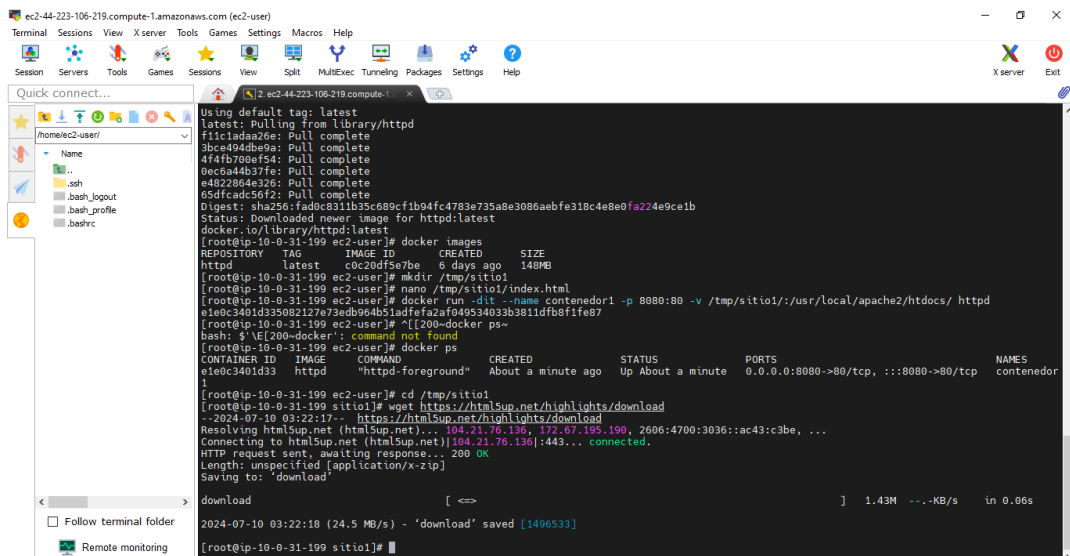
### Inclusión de nueva red con puerto 8080 publico



*Nota.* Se ilustra de la configuración del firewall de la instancia y la inclusión de una red nueva a fin de que con el puerto 8080 ingrese cualquiera.

Figura 6

### Descargue y descompresión de plantilla html en nuestro contenedor



*Nota.* La imagen muestra el proceso de descargue de la plantilla html en la carpeta sitio1.

Figura 7

## Creación de contenedores.

```

creating: assets/css/
unflating: assets/css/fontawesome-all.min.css
unflating: assets/css/main.css
creating: assets/css/images/
unflating: assets/css/images/arrow.svg
unflating: assets/css/images/overlay.png
creating: assets/sass/
unflating: assets/sass/main.scss
creating: assets/sass/libs/
unflating: assets/sass/libs/functions.scss
unflating: assets/sass/libs/_mixins.scss
unflating: assets/sass/libs/_vars.scss
unflating: assets/sass/libs/_utilities.scss
unflating: assets/sass/libs/_vendor.scss
unflating: assets/sass/libs/_breakpoints.scss
unflating: README.txt
replace_index.html? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
unflating: index.html
creating: images/
unflating: images/pic05.jpg
unflating: images/pic03.jpg
unflating: images/pic01.jpg
unflating: images/pic02.jpg
unflating: images/bg.jpg
[root@ip-10-0-31-199 sitio1]# mkdir /tmp/sitio2
[root@ip-10-0-31-199 sitio1]# nano /tmp/sitio2/index.html
[root@ip-10-0-31-199 sitio1]# docker run -dit --name contenedor2 -p 8080:80 -v /tmp/sitio2:/usr/local/apache2/htdocs/ httpd
83d1aefbf29e0cd8f1bf0410a3c36fdbf4e502cf9931ce5b989a793869e
docker: Error response from daemon: driver failed programming external connectivity on endpoint contenedor2 (f626721965705ad6ffbc43bf18a9f41180e851ead7446edeb490a30792): bind for 0.0.0.0:8080 failed: port is already allocated.
[root@ip-10-0-31-199 sitio1]# docker run -dit --name contenedorN2 -p 8081:80 -v /tmp/sitio2:/usr/local/apache2/htdocs/ httpd
b51ec2532f307133f7r3d53f6dba32408adbbe49c017cda0dc304c45c2c40c
[root@ip-10-0-31-199 sitio1]# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
b51ec2532f30   httpd    "httpd-foreground"     10 seconds ago Up 9 seconds    0.0.0.0:8081->80/tcp, :::8081->80/tcp   contenedorN2
e160c3401d33   httpd    "httpd-foreground"     25 minutes ago Up 25 minutes    0.0.0.0:8080->80/tcp, :::8080->80/tcp   contenedor1
[root@ip-10-0-31-199 sitio1]#

```

Nota. Se crearon dos contenedores y se verifico su creación.

Figura 8

## Monte de balanceador de cargas

```

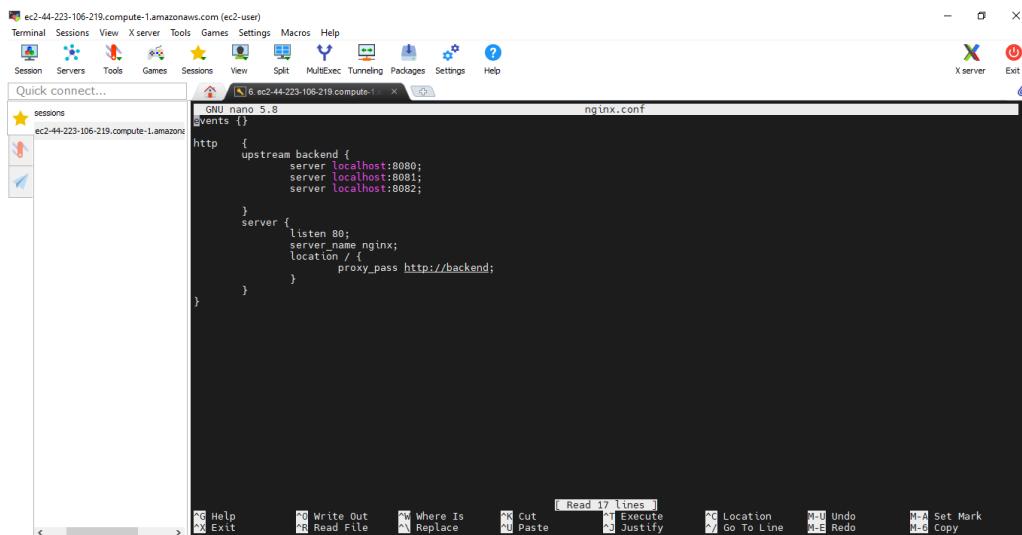
(4/7): nginx-1:24.0-1.amzn2023.0.2.x86_64.rpm          1.5 MB/s | 32 kB  00:00
(5/7): nginxfilesystem-1:24.0-1.amzn2023.0.2.noarch.rpm 413 kB/s | 9.1 kB 00:00
(6/7): nginx-core-1:24.0-1.amzn2023.0.2.x86_64.rpm     15 MB/s | 586 kB 00:00
(7/7): nginx-mtmetypes-2.1.49-3.amzn2023.0.3.noarch.rpm 854 kB/s | 21 kB 00:00
-----
Total: 5.7 MB/s | 1.0 MB 00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      : nginxfilesystem-1:24.0-1.amzn2023.0.2.noarch                1/1
  Installing    : nginxfilesystem-1:24.0-1.amzn2023.0.2.noarch                1/7
  Installing    : nginx-mtmetypes-2.1.49-3.amzn2023.0.3.noarch                2/7
  Installing    : libuwind-1.4.0-5.amzn2023.0.2.x86_64                       3/7
  Installing    : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64                4/7
  Installing    : nginx-core-1:24.0-1.amzn2023.0.2.x86_64                    5/7
  Installing    : generic-logos-httpd-18.0-12.amzn2023.0.3.noarch             6/7
  Installing    : nginx-1:24.0-1.amzn2023.0.2.x86_64                         7/7
Running scriptlet: nginx-1:24.0-1.amzn2023.0.2.x86_64                       7/7
Verifying      : generic-logos-httpd-18.0-12.amzn2023.0.3.noarch             1/7
Verifying      : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64                2/7
Verifying      : libuwind-1.4.0-5.amzn2023.0.2.x86_64                       3/7
Verifying      : nginx-1:24.0-1.amzn2023.0.2.x86_64                         4/7
Verifying      : nginx-core-1:24.0-1.amzn2023.0.2.x86_64                    5/7
Verifying      : nginxfilesystem-1:24.0-1.amzn2023.0.2.noarch                6/7
Verifying      : nginx-mtmetypes-2.1.49-3.amzn2023.0.3.noarch                7/7
Installed:
  generic-logos-httpd-18.0-12.amzn2023.0.3.noarch                gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64
  libuwind-1.4.0-5.amzn2023.0.2.x86_64                          nginx-1:24.0-1.amzn2023.0.2.x86_64
  nginx-core-1:24.0-1.amzn2023.0.2.x86_64                       nginxfilesystem-1:24.0-1.amzn2023.0.2.noarch
  nginx-mtmetypes-2.1.49-3.amzn2023.0.3.noarch
Complete!
[root@ip-10-0-31-199 ec2-user]# systemctl start nginx
[root@ip-10-0-31-199 ec2-user]#

```

Nota. Ilustración de monte del balanceador de cargas, e instalación del navegador nginx con acceso al puerto 80 para validar.

**Figura 9**

*Edición del archivo de configuración nginx.conf*



```

GNU nano 5.8 nginx.conf
events {}

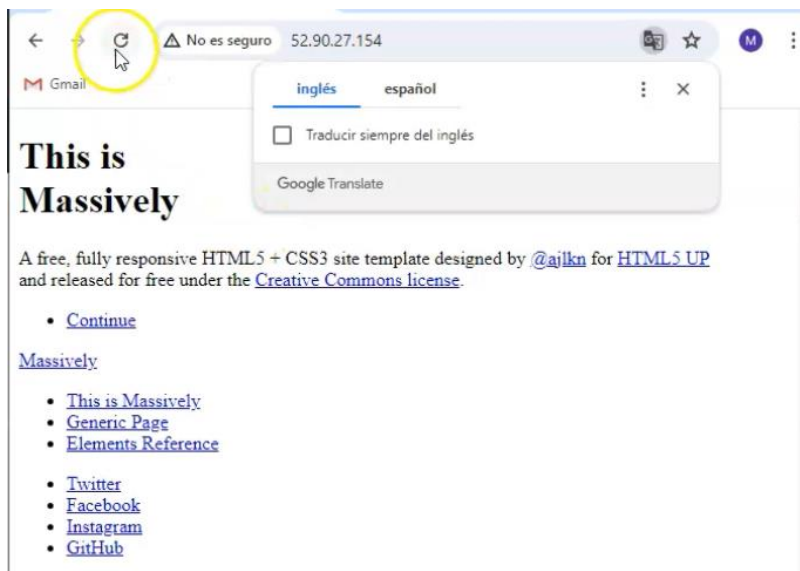
http {
    upstream backend {
        server localhost:8080;
        server localhost:8081;
        server localhost:8082;
    }

    server {
        listen 80;
        server_name nginx;
        location / {
            proxy_pass http://backend;
        }
    }
}
  
```

*Nota:* Representación de la edición del archivo de configuración de nginx llamado nginx.conf.

**Figura 10**

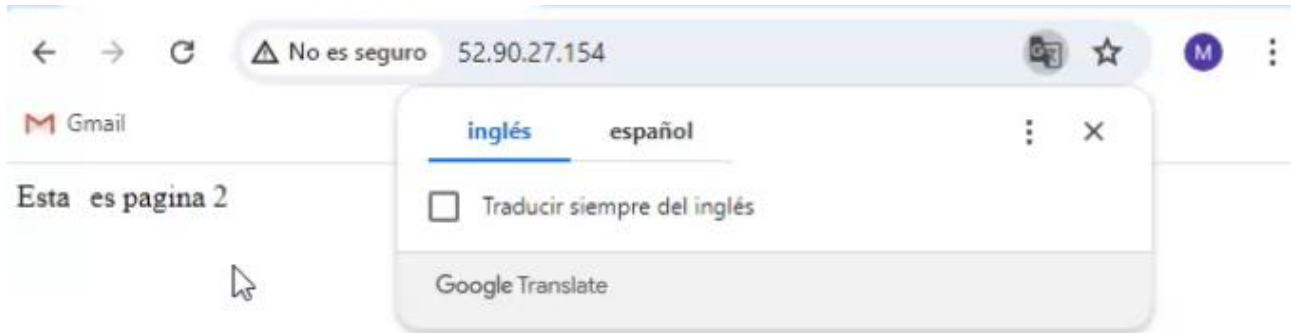
*Visualización del sitio web 1 con la plantilla html incorporada de forma básica*



*Nota.* Contiene la plantilla html descomprimida y descargada en nuestro contenedor, apuntando a la IP pública.

## Figura 11

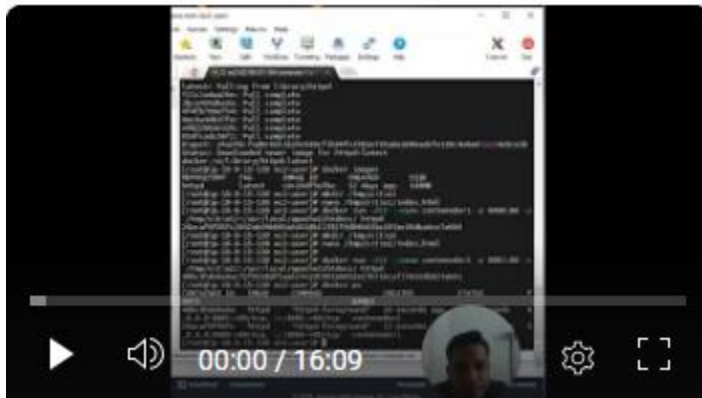
*Visualización del sitio web 2 sin plantilla incorporada*



*Nota.* Contiene el sitio web 2 apuntando a la IP publica, evidenciando así el funcionamiento del balanceador de cargas.

## Video 1

*Explicación del ejercicio desarrollado en EC2, mostrando el funcionamiento*

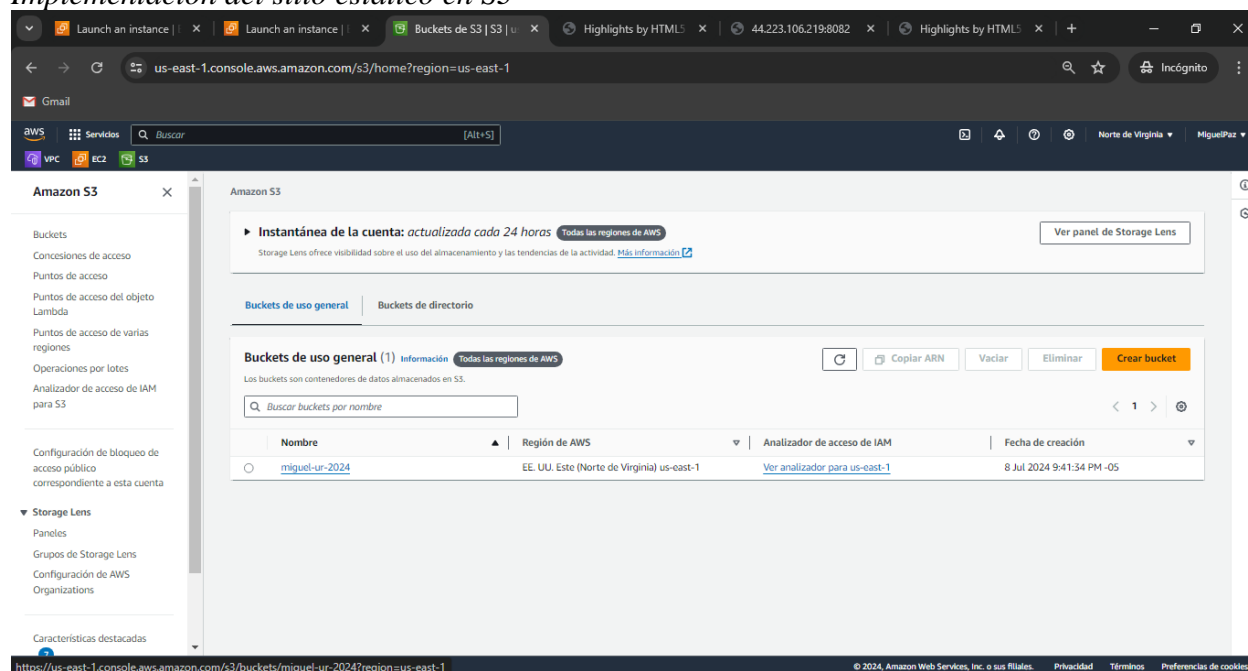


*Nota.* El desarrollo del video se encuentra publicado en <https://youtu.be/IpHIKufCd10>

Seguidamente, se implementó el mismo sitio estático que se usó en los contenedores, utilizando Amazon Simple Storage Service (Amazon S3), explicando las ventajas de cada opción, e indicando cual de las dos alternativas se elegiría para implementar o mantener este sitio web en internet.

## Figura 12

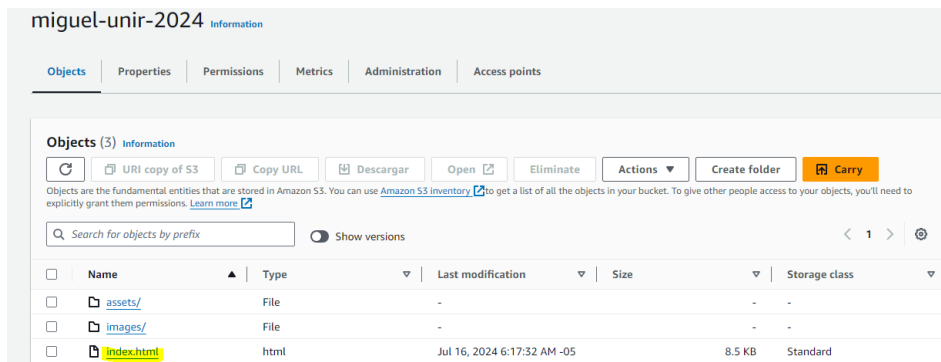
### Implementación del sitio estático en S3



*Nota.* Se llevo a cabo el procedimiento y paso a paso visto, obteniendo como resultado la creación del Bucket.

## Figura 13

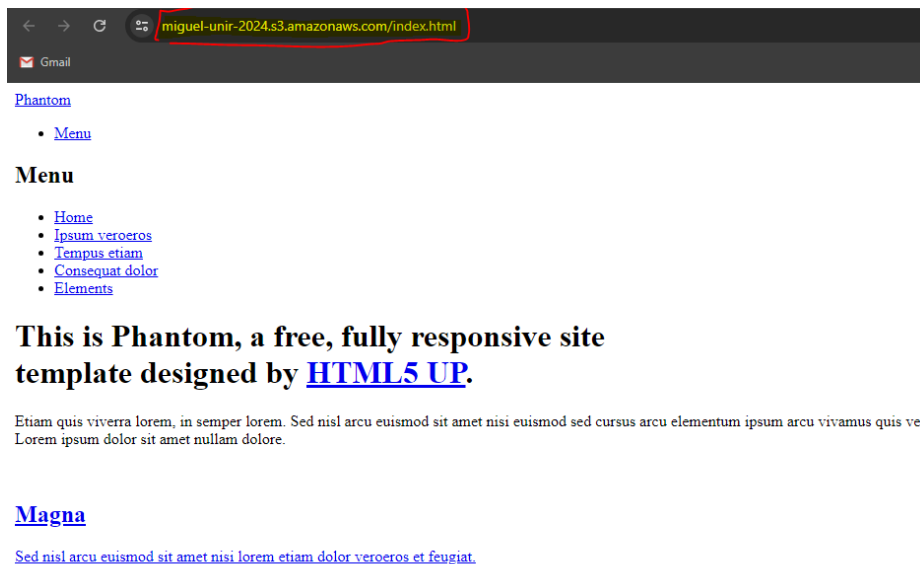
*Cargue preliminar de página web en html*



*Nota.* Se cargaron los objetos básico y principales para la ejecución de la página. t.

## Figura 14

*Visualización del sitio web 1 con la plantilla html incorporada de forma básica*

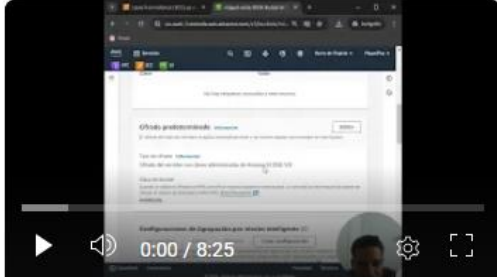


*Nota.* Ilustración final de la conversión del bucket para que sea un sitio web estático, apuntando a la URL del objeto para su visualización.



## Video 2

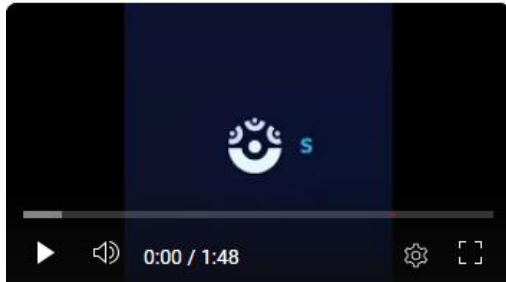
*Explicación del ejercicio desarrollado en S3, mostrando el funcionamiento final*



*Nota.* El desarrollo del video se encuentra publicado en <https://youtu.be/IpHIKufCd10>

## Video 3

*Ejecución del ejercicio 2, mostrando el funcionamiento final*

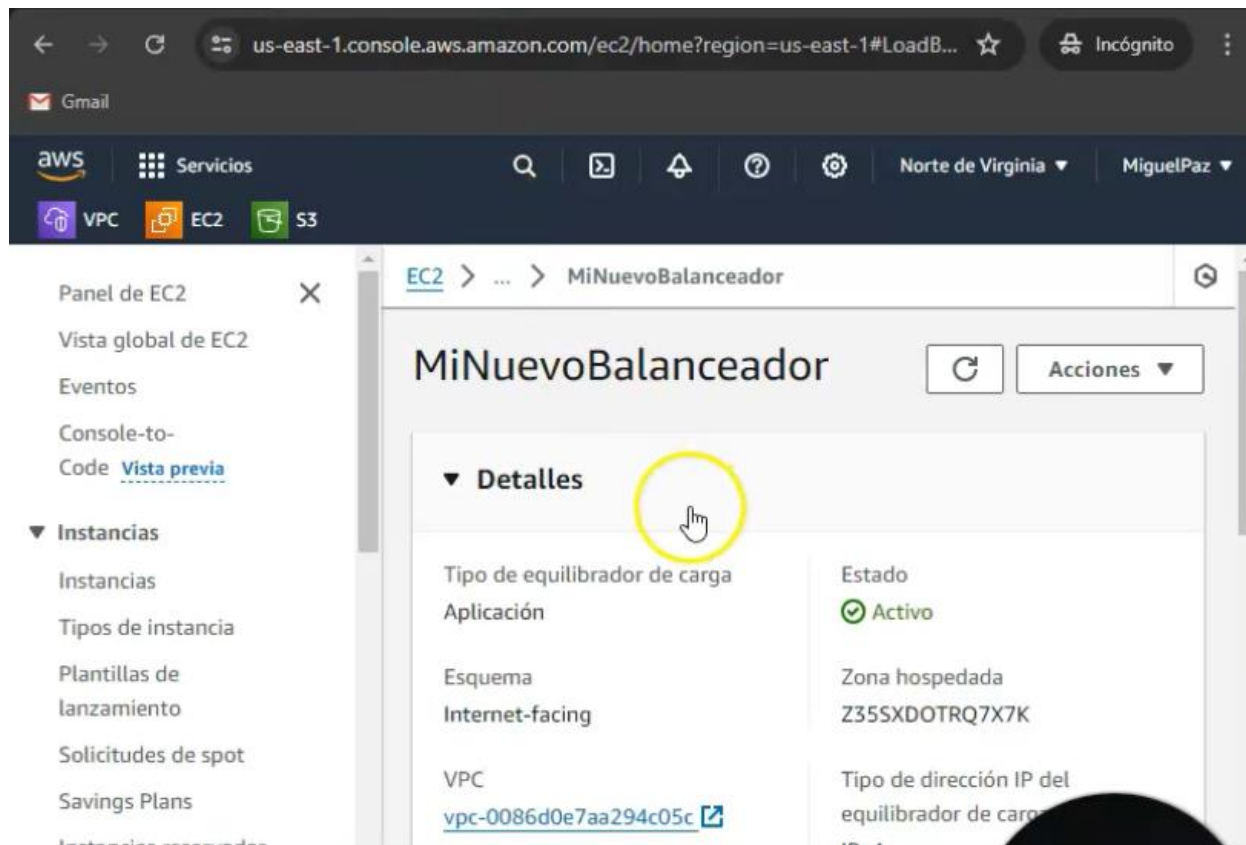


*Nota.* El desarrollo del video se encuentra publicado en <https://youtu.be/YxHyTuP9gcw>

Por último, se implementó una arquitectura de AWS con balanceador de carga de aplicación para al menos dos instancias, y estas a su vez contienen al interior balanceador de carga de contenedores, evidenciando además el funcionamiento del balanceador de carga mediante un video, en el cual se detectan los cambios de salud de los sitios web y las actividades generadas en el grupo de autoescalado.

**Figura 15**

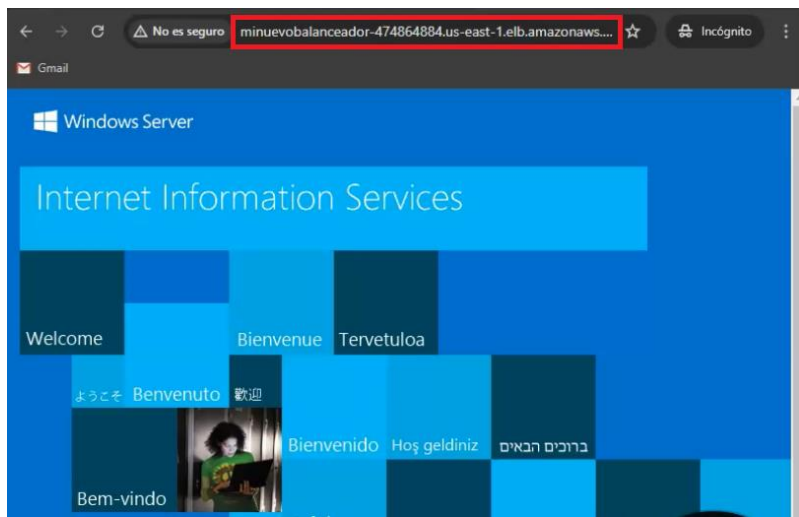
*Creación del balanceador de cargas*



*Nota.* Ilustración del resultado final de la creación del balanceador de cargas, de conformidad con las instrucciones aprendidas en el seminario.

**Figura 16**

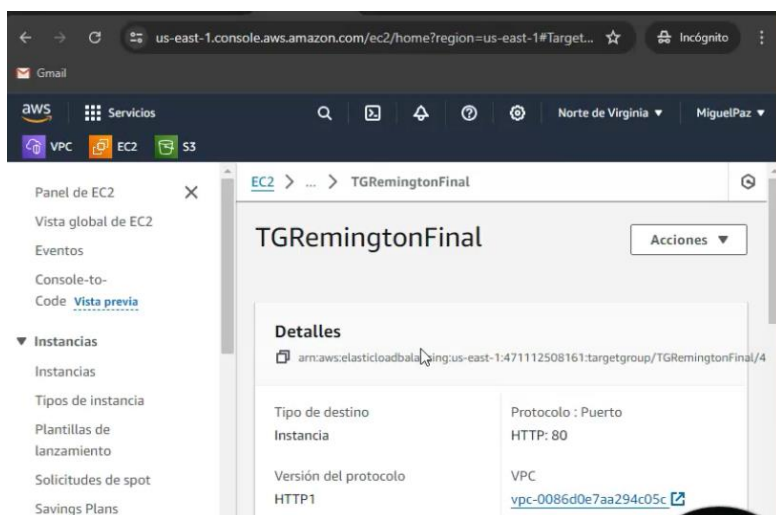
*Ejecución del balanceador de cargas*



*Nota.* Ilustración de la ejecución del balanceador de cargas, apuntando a su DNS.

**Figura 17**

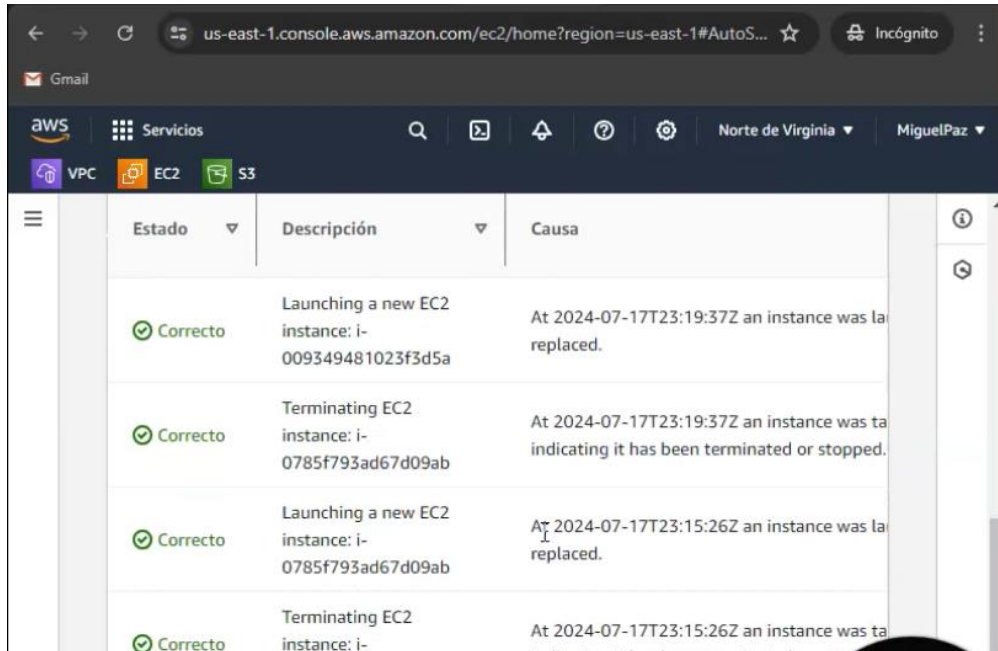
*Creación del Target group*



*Nota.* Ilustración del resultado final de la creación del Target groups, de conformidad con las instrucciones aprendidas en el seminario.

## Figura 17

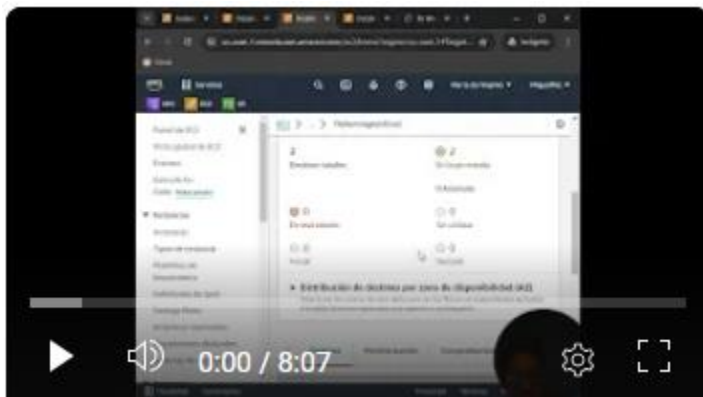
*Demostración de actividades en el grupo de Auto scaling*



*Nota.* Se valido el funcionamiento del balanceador de carga, en el cual se detectaron los cambios de salud de los sitios web y estos a su vez generan actividad en el grupo de autoescalado.

## Video 3

*Explicación del entregable 3, mostrando el funcionamiento requerido*



*Nota.* El desarrollo del video se encuentra publicado en <https://youtu.be/Wt-syVgS7dE>

## Conclusiones

Basándome en los análisis anteriores, se recomienda que TechSolutions S.A. realice la migración a la nube a través de la implementación de Amazon Virtual Private Cloud (Amazon VPC) e instancias en Amazon Elastic Compute Cloud (EC2) y Amazon Simple Storage Service (Amazon S3) con la ejecución de Contenedores y balanceadores, ya que estratégicamente se plantea como la alternativa más idónea para alcanzar sus metas de expansión, eficiencia, seguridad y productividad en un mercado empresarial en constante evolución.

## Referencias

Whaley, B. (2023). *Amazon Web Services in Action, Third Edition: An In-depth Guide to AWS*.

Sitio web:

<https://books.google.es/books?hl=es&lr=&id=joK3EAAAQBAJ&oi=fnd&pg=PA1&dq=amazon+web+services+aws&ots=DsOjRv9ttV&sig=9TsxS1tLjJ0VzM2nRDZoYaZyOFs#v=onepage&q=amazon%20web%20services%20aws&f=false>