



TRABAJO DE GRADO

Seminario AWS

Servicios de AWS Para Alojamiento Web

Corporación Universitaria Remington.

Facultad de Ingeniería

Amazon Web Services (AWS)

Briyid Lorena Manrique Merchan

Jahk Daniels Martínez Guerrero

Luis Carlos Ospina Martínez

Juan Pablo Berrido López

Seminario AWS

2024

2

Tabla de Contenido

Resumen	5
Marco conceptual	7
Aws	7
Sitio y aplicación web	8
Data center local	8
Escalabilidad	8
Instancia de computación	8
Contenedores.....	8
Máquina virtual	9
Bases de datos	9

Balanceo de cargas	9
Consola	9
Marco contextual	10
Desarrollo e implementación del aprendizaje	11
Evaluación de Riesgos y Seguridad	11
Riesgos Local	11
Riesgos Nube	11
	3
Medidas para Mitigar Riesgos	12
Local	12
Nube	12
Escalabilidad y Flexibilidad	12
Local	13
Nube	13
Local	13
Nube	14
Impacto en el Personal y Operaciones	14

Local	14
Nube	15
Cambios en Responsabilidades Datacenter Local	15
Nube	15
Impacto en Operaciones Diarias Datacenter Local	16
Nube	16
Productividad de Empleados Datacenter Local	16
Nube	16
Recomendación Final.....	17
Puntos Clave para la Discusión.....	17
Flexibilidad vs. Control	17
	4
Innovación y futuro	18
Instancia EC2	21
Docker	21
Implemente un sitio estatutico en S3	47
EC2	57

Ventajas	
57	
Desventajas	
57	
S3	58
Ventajas	
58	
Desventajas	
58	
¿Por qué en S3?	
59	
Arquitectura de AWS	
60	
Conclusiones	
78	
Referencias	
79	

Resumen

Siempre que una empresa o individuo quiere subir su sitio o aplicación web surge la incertidumbre de cuál es la mejor opción o forma de hacerlo, teniendo en cuenta todas las variables necesarias para que este sitio cumpla con las necesidades tanto de la empresa como del usuario que va a hacer uso de este. Una muy buena opción es Amazon Web Services que permite crear, administrar e implementar sitios web con múltiples formas de hacerlo impulsándolo a ser el más utilizados por las empresas y organizaciones a nivel global.

El propósito de este estudio es analizar las formas en las que se puede subir un sitio web utilizando la creación de instancias y contenedores o sitios estáticos en s3 y ver cuál de estas opciones es la que mejor se adapta a las necesidades requeridas. Dentro de lo que tiene que ver con la creación de instancias se analizará las mejores configuraciones en cuanto a sistema operativo, almacenamiento, procesador, base de datos, seguridad y accesos que cumplan con los requerimientos necesarios para poner en funcionamiento el sitio web. En cuanto a los contenedores se observarán las ventajas de utilizarlos en nuestro sitio y como estos ayudan a que funcione de manera eficiente. Otro elemento necesario que permitirá el desempeño eficiente del sitio o aplicación web son los balanceadores de carga, estos brindan la posibilidad de manejar los recursos que se necesitan a nivel de hardware y software para distribuirlos de manera equitativa entre los contenedores.

Uno de los objetivos de este trabajo de grado sobre AWS es realizar la práctica de lo aprendido en clase e implementarlo a manera de prueba sobre un sitio o aplicación web que

escojamos. Así mismo se presentarán las pruebas hechas con cada una de las opciones implementadas mediante pantallazos que siguen el paso a paso con su debida explicación de estos y las respectivas conclusiones que arrojen estas pruebas. Gracias a la práctica hecha se procederá a realizar un balance sobre que opción es mejor y por qué la escogeríamos para implementar el sitio web teniendo en cuenta los diferentes beneficios que ofrecen cada una de estas.

Palabras clave

Contenedores, instancias, eficiencia, pruebas y necesidades.

Marco conceptual

Muchas son las opciones que tenemos para poner en funcionamiento un sitio o aplicación web, cada una de estas con diferentes beneficios según la necesidad que se tenga, sin embargo una de las mejores es AWS siendo ampliamente la más utilizada en el mundo gracias a sus grandes beneficios como los son la integridad de datos, le flexibilidad, facilidad de uso, rentabilidad, escalabilidad y alto desempeño, además de ofrecer gran variedad de productos como la computación, almacenamiento, bases de datos, redes y entrega de contenido, análisis, machine learning, seguridad, identidad y cumplimiento. Como se puede observar una opción muy completa para todas las necesidades que necesitan cubrir y precisamente se mostrará en este informe una parte de cómo podemos subir al internet un sitio o aplicación web mediante la creación de instancias y contenedores, con sus respectivos balanceadores de carga, también, se verá reflejado las configuraciones pertinentes que se tiene que hacer para lograr el funcionamiento.

Para comprender mejor el funcionamiento de AWS y el objetivo de este seminario se conceptualizar los temas y definiciones empleados.

Aws

Amazon Web Services básicamente es la nube más completa y empleada a nivel global, en la cual se pueden encontrar diferentes servicios como almacenamiento, recursos de computación, bases de datos, aplicaciones móviles en fin una gran cantidad de servicios integrales. Esto ha llevado a que sean millones de usuarios de todo tipo los que adopten AWS para mantener su infraestructura y aplicaciones.

Sitio y aplicación web

Estos se ejecutan en el navegador web siendo el medio por el cual las empresas o un apersona pone a disposición de los usuarios información o productos y de igual forma la manipulación de los mismos de manera remota.

Almacenamiento en la nube: Mediante este servicio se pueden almacenar los datos a través de internet a un sistema de almacenamiento externo que es mantenido por un tercero. Actualmente es un servicio muy utilizado por las empresas, gracias a su excelente capacidad de adaptarse a las necesidades de almacenamiento requeridas, escalabilidad y disponibilidad de los datos.

Data center local

Este viene siendo la infraestructura física de tecnológica que posea una empresa para el alojamiento y procesamiento de datos críticos para la funcionalidad de la misma.

Sistema operativo: Es el software fundamental en los ordenadores y dispositivos que permiten la interacción entre al hardware y el usuario mediante la gestión de los recursos del sistema.

Escalabilidad

disposición que tiene los sistemas para adaptarse y dar respuesta de un sistema al momento de ir creciendo de forma exponencial.

Instancia de computación

Se trata de un recurso de servidor en la nube que tiene la finalidad de realizar cargas de trabajo con uso intensivo de cómputos, por ejemplo, contenedores, bases de datos, máquinas virtuales y microservicios.

Contenedores

viene siendo un paquete o unidad de software en los que se incluyen todos los elementos y dependencias para que un sitio o aplicación web se ejecute en cualquier entorno de forma rápida y fiable.

Máquina virtual

Este es un software capaz de alojar en su interior un sistema operativo haciendo crear que es una computadora de verdad, ya que lograr ser una réplica en cuanto a su comportamiento.

Bases de datos

Estas son fuentes de información en la cual esta recopilada y organizada de forma sistemática todo tipo de datos desde palabras hasta archivos de videos y sobre cualquier tipo de información que haya.

Balanceo de cargas

Es la técnica implementada para distribuir de forma equitativa los recursos de hardware para lograr la disponibilidad, seguridad, escalabilidad y el rendimiento de la maquina o aplicación.

Consola

En la computación es una aplicación o dispositivo electromecánico que sirve para interactuar con el computador.

Marco contextual

Siempre es importante conocer o documentarse bien sobre los sitios de alojamiento web que existen, ya que al haber tantos de estos disponibles cada uno ofrece diferentes tipos de servicios que pueden beneficiar al sitio web que se quiere subir claro está que para esto también debemos tener en cuenta las necesidades que tenemos en el ahora y las que posiblemente vamos a tener en el futuro a medida que el sitio crezca. Como ya se ha explicado anteriormente una de las mejores opciones para el alojamiento web es AWS por sus grandes beneficios. A lo largo de este trabajo se evidencia como podemos utilizar AWS con base en los requerimientos que se tengan y sobre todo controlando los gastos económicos que se presentan al utilizar un alojamiento web.

La finalidad del trabajo es representar los conceptos y las prácticas realizadas a lo largo del seminario en este caso aplicándolos a un sitio web de prueba que se descargó de internet y del cual se presenta evidencia mediante imágenes con su respectiva explicación del funcionamiento de las instancias que se utilizan para poner en funcionamiento el sitio web la creación de contenedores y el respectivo balanceo de cargas de las instancias y contenedores para el funcionamiento óptimo del sitio. Todos estos nuevos conocimientos adquiridos permiten la realización de este trabajo con una característica muy importante y es que en el futuro sirva de guía para otras personas que quieran utilizar AWS como alojamiento web.

Desarrollo e implementación del aprendizaje

Evaluación de Riesgos y Seguridad

Riesgos Local

- Riesgo a fallas eléctricas

- Riesgo al tener acceso a la información de personal no autorizado
- Riesgo de accidentes físico personal durante el uso
- Riesgo de robo de información y brechas de seguridad
- Riesgo a tener errores o fallos del servidor o servidores entre ellos los

componentes o equipos que detiene la operación

- Riesgo al incumplimiento de las normas para obtener la infraestructura para una empresa

Riesgos Nube

- Riesgo a perdida de datos e información.
- Riesgo a incumplimiento de leyes aplicables sobre privacidad de datos.
- Riesgo de fallas del servicio ya sea por conectividad a internet o problemas en el proveedor.
- Riesgos a presentar fallos del servidor o servidores entre ellos los componentes o equipos que detiene la operación.
- Riesgo de que el proveedor no cumpla con las normativas necesarias para el cliente y la región.

Medidas para Mitigar Riesgos

Local o Implementar señales donde indique solo personal autorizado cámaras de seguridad, o personal de vigilancia o Instalar un programa de

detección de accesos no autorizados a un computador o a una red, que detecta amenazas potenciales y genera alertas.

- o Desarrollar políticas de respaldo y recuperación. o Asegurar el cumplimiento de las normas.

Nube o Obtener proveedores de seguridad que estén certificados y reconocidos.

- o Implementar copia de seguridad aparte de la que tiene en el proveedor principal.

Puede ser en otro proveedor diferente.

- o Configurar accesos de autenticación confiables como gestión de identidad y acceso.

- o Generar acuerdos con el proveedor que se utilice para que garanticen alta disponibilidad y tiempos de respuesta.

Escalabilidad y Flexibilidad

- o Evaluar cómo cada opción se adapta a las necesidades de crecimiento y flexibilidad de la empresa. o Proponer estrategias para manejar picos de demanda y variabilidad en las necesidades de recursos.

Local

- Empresa con servidor local necesita de espacio o adaptar un cuarto, carece de espacio físico y el hardware disponible. La empresa crece, se requiere recursos adicionales en infraestructura.

- Cuando se usa el servicio local se requiere de inversión económica al pasar el

tiempo porque a medida que crece la empresa se necesita aumentar los recursos de ese servicio como ram, disco duro, calefacción etc.

Nube

- Es una ubicación mucho más ideal que tenerlos de forma local en la empresa, tienen toda la garantía de seguridad de red contra incendios de aire acondicionado cuentan con proveedores de energía

- Tiene un gasto inicial a gasto variable, se beneficia de las economías de escala masiva

- Aumenta la velocidad y la agilidad

- Deja de gastar dinero en ejecutar y mantener centros de datos

Estrategias para Manejar Picos de Demanda y Variabilidad

Local o Diseñar y analizar para construir una infraestructura donde se van a dejar los servicios que permita manejar picos de demanda sin afectar el rendimiento del sistema.

- o Implementar medidas de reducción y más gestión de recursos para asegurar que se utilicen de manera exacta, donde no se consuman recursos cuando no se esté usando es decir el tiempo de inactividad y así aumentar mucho más el rendimiento.

Nube o Mejora el rendimiento, se deja modificar o cambiar las características de los servicios.

- o Cuenta con notificaciones que alertan sobre picos inusuales en la demanda, esto ocasiona una respuesta rápida para ajustar los recursos.

- o Alta disponibilidad escalabilidad y elasticidad rápida del cómputo en la nube, así poder manejar picos de demanda y variabilidad en las necesidades de recursos.
- o Tolerancia a fallos y recuperación.

Impacto en el Personal y Operaciones

- o Considerar el impacto en el personal IT de la empresa, incluyendo la necesidad de formación y cambios en las responsabilidades.
- o Evaluar cómo cada opción afecta las operaciones diarias de la empresa y la productividad de los empleados.

Local

- Capacitación en Gestión de Infraestructura Física: El personal debe ser entrenado en la instalación, configuración y mantenimiento de hardware y redes, así como en la seguridad física y lógica del datacenter.

Desarrollo de Habilidades en Mantenimiento de Equipos: Es necesario implementar programas de capacitación que incluyan desde la configuración de servidores hasta la solución de problemas de hardware.

Nube

Capacitación en AWS: Proveer programas de certificación como AWS Certified Solutions Architect para asegurar que el personal adquiriera las habilidades necesarias.

Desarrollo de Habilidades en DevOps: Implementar herramientas como AWS CodePipeline y AWS CodeBuild para fomentar prácticas de integración y entrega continua.

Cambios en Responsabilidades Datacenter Local

Intensificación en la Administración del Hardware: El personal de TI deberá dedicar tiempo y recursos significativos al mantenimiento físico del hardware y a la gestión de la infraestructura local, lo que puede limitar su enfoque en actividades estratégicas.

Gestión de Seguridad Física y Lógica: El equipo de TI tendrá control total sobre la implementación y gestión de las medidas de seguridad.

Nube

Disminución en la Administración del Hardware: La utilización de la infraestructura en la nube minimiza la necesidad de realizar tareas de mantenimiento físico, permitiendo al personal enfocarse en actividades estratégicas y de mayor valor para la compañía.

Enfoque en Automatización y Monitoreo: Mayor uso de scripts y herramientas de monitoreo para automatizar procesos y asegurar la continuidad operativa.

Impacto en Operaciones Diarias Datacenter Local

Complejidad en la Gestión de Infraestructura: Requiere un equipo interno dedicado al mantenimiento y soporte, lo que puede complicar las operaciones diarias y aumentar la carga de trabajo del personal de TI.

Optimización Controlada: El desempeño de la infraestructura puede ser optimizado para las necesidades específicas de la empresa, pero requiere constante monitoreo y ajustes.

Nube

Simplificación de la Gestión de Infraestructura: Utilizar la consola de administración de AWS para gestionar recursos de manera eficiente.

Mejora en la Disponibilidad y Redundancia: Implementar arquitecturas multi-AZ para asegurar alta disponibilidad y recuperación ante desastres.

Productividad de Empleados Datacenter Local

Acceso Limitado a Recursos: La capacidad física del datacenter puede limitar el acceso a recursos necesarios para los desarrolladores, afectando la agilidad y capacidad de innovación.

Inversión en Infraestructura Adicional: Expansiones para satisfacer la demanda requieren inversiones adicionales en hardware y espacio.

Nube

Acceso a Recursos Bajo Demanda: Permitir a los equipos de desarrollo acceder a los recursos que necesitan de manera inmediata, mejorando la agilidad y la capacidad de innovación.

Recomendación Final

La mejor opción para la empresa TechSolutions S.A. Es optar por una solución en la nube con AWS ya que esta proporciona una extensa variedad de herramientas y servicios de seguridad que contribuyen a proteger la infraestructura de forma eficaz, teniendo en cuenta la flexibilidad de ajustar automáticamente la capacidad de los recursos según sea la demanda el crecimiento de la empresa, esto llevaría a la reducción de tareas manuales y el enfoque en la automatización de soluciones al personal IT para poder concentrarse en la innovación y mejoras continuas que requiera la empresa.

Puntos Clave para la Discusión

Flexibilidad vs. Control

Gracias a la competitividad que hay en la actualidad por la entrega de productos de forma ágil, escalables y flexibles la nube representa una opción de mucha facilidad flexibilidad para la empresa, facilidad en cuanto a la oferta disponible de obtener el servicio, ya que existen múltiples servicios de computación en la nube, puedes contratar un servicio que se adapte a tus necesidades y así aprovecharlo al máximo, proporciona facilidad en la gestión y control de recursos beneficiando a los usuarios finales. En Cuanto a la flexibilidad permite la elección de los servicios que van a hacer utilizados en incluso cuantas horas al día los vas a utilizar, permite aumentar o disminuir la capacidad de almacenamiento y es muy flexible al momento de permitir que usuarios están permitidos para acceder a los datos.

Lo importante de tener un data center en la empresa es que se puede confiar en los empleados que se especializan en el manejo de este, el data center sigue funcionando en caso de no haber una conexión a internet, podemos tener un control de gastos, ya que en muchas ocasiones la nube representa gastos que no teníamos previstos, en muchos casos se utiliza en proyectos que requieran un alto nivel de confidencialidad permitiendo el control sobre los usuarios autorizados.

En conclusión, cada una de estas opciones tienen sus ventajas y desventajas simplemente debemos tener en cuenta cuales son las necesidades de nuestra empresa para escoger la que más nos convenga.

Innovación y futuro

Las empresas siempre buscan el éxito de sus productos, y precisamente la nube es una tecnología que puede ayudar al éxito de la empresa gracias a que esta garantiza la eficiencia, fiabilidad y seguridad. La nube tiene un panorama muy prometedor gracias a la evolución y publicación periódica de nuevas tecnologías y funcionalidades. Tanto así, que se estima que el 80% de las empresas migren a la nube.

La nube es tan prometedora debido a que, gracias a los avances tecnológicos se ha podido eliminar las amenazas potenciales en los sistemas en la nube, además, permite la automatización de los procedimientos de vigilancia en ataques y defensa, asegurando un futuro estable y resistente.

Gracias a la gerencia flexible de los recursos la computación en la nube se ve direccionada hacia el mejoramiento de la productividad y la coordinación.

Un aspecto a tener en cuenta es que la nube permite el ahorro de recursos muy importante como lo es la energía, ya que esta funciona en la web y no se necesitan contar con estructuras que gastan energía en la conectividad y funcionalidad y el mantenimiento del servidor.

En la actualidad se generan datos a niveles masivos y a medida que surgen nuevas tecnologías va a ser muchísimo mayor la cantidad de datos que se generan y esto siempre presenta un reto para almacenarlos. Se espera que los distintos proveedores de la nube aumenten su infraestructura de almacenamiento de datos, claro está asegurando precios competitivos.

Gracias a la constante evolución de las necesidades de las empresas, el incremento en cuanto a las expectativas del cliente y el aumento masivo de los datos que se generan. Se espera

que los servicios en la nube dominen el mercado. Teniendo en cuenta esto es recomendable escoger los servicios en la nube.

Experiencia del usuario

En cuando a los desarrolladores se debe tener en cuenta que estos deben tener mucho más conocimiento en cuanto al manejo de este, además, deben tener en cuenta una serie de variables al momento de trabajar en un sistema como, el crecimiento en el tiempo, análisis de conectividad, establecer sistemas de seguridad y control, ver que el sistema no incumpla con las variables de enfriamiento. Por otro lado, los usuarios tienen asegurado el acceso permanente a los datos gracias a la conectividad confiable y rápida que tienen los data center, permitiendo acceder a los servicios de forma eficiente, pueden estar tranquilos, ya que se cuenta con más seguridad para el robo de datos y daños de los mismos por malware.

La nube

Un beneficio muy grande que tiene para los desarrolladores es que no deben ser grandes expertos en el manejo de esta para poder implementarla, les ofrece un apoyo sólido que impulsa una mejor agilidad en el ciclo de vida de las aplicaciones, les ofrece una mayor seguridad, productividad y escalabilidad al construir las aplicaciones. La experiencia del usuario en la nube es muy positiva, porque obtienen experiencias personalizadas gracias a la optimización y personalización de sus experiencias específicas en momentos importantes, obtienen interacciones más personalizadas, ya que la nube cuenta con herramientas que le permite segmentar al cliente por su comportamiento, los usuarios se benefician de la rapidez y facilidad de la nube debido a las numerosas formas en las que se puede contactar la empresa, correos, sms, chat en línea, redes sociales, etc. Permitiendo al usuario obtener el producto de forma rápida y eficaz.

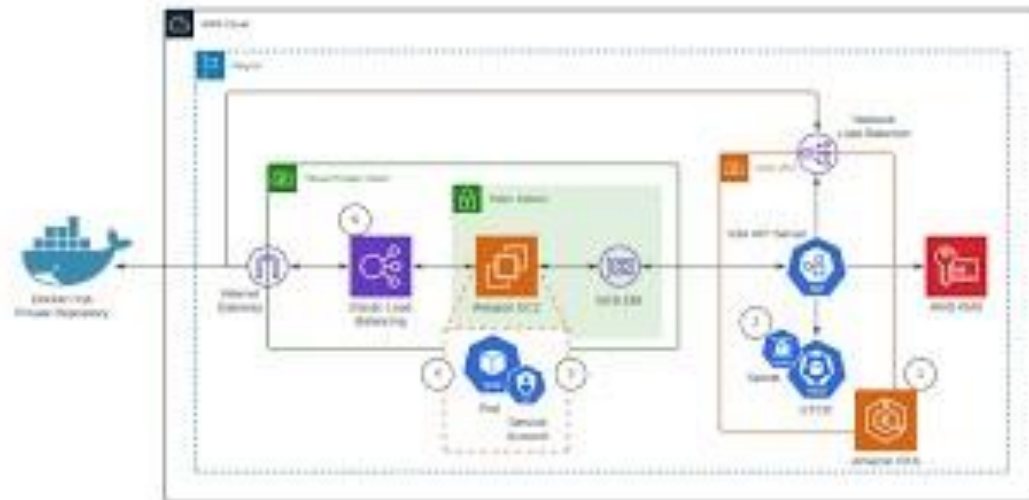
Instancia EC2

Creación en AWS La instancia como se muestra en la Figura 1

Docker

Es una aplicación o plataforma completa que permite ejecutar aplicaciones en un contenedor es una pequeña cajita es un proceso dentro del sistema operativo donde se pueden ejecutar muchos programas o un solo programa. Cuando se ejecuta ese programa deja de llamarse programa y se empieza a llamar proceso esto se basa en que ese programa carga una información en la memoria ram

En esta actividad se va observar que docker va tener un proceso o varios procesos que va permitir ejecutar aplicaciones, desplegarlas, gestionarlas esto ocurre en segundos minutos. Cada una es independiente a la otra que se está ejecutando.



Se va a crear en AWS la instancia que se va llamar ServerDocker

Figura 1

Creación de instancia

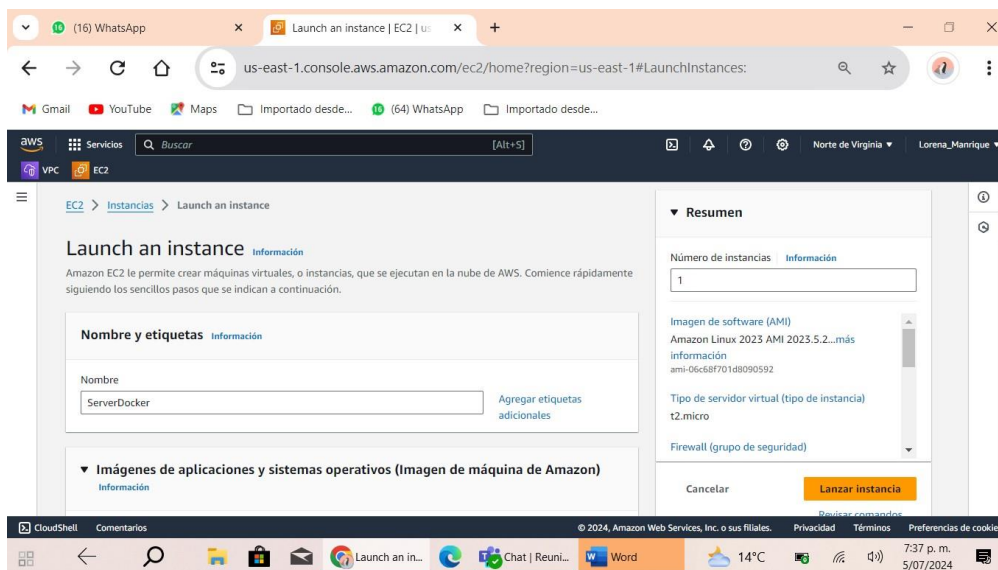


Figura1 escogemos

AWS

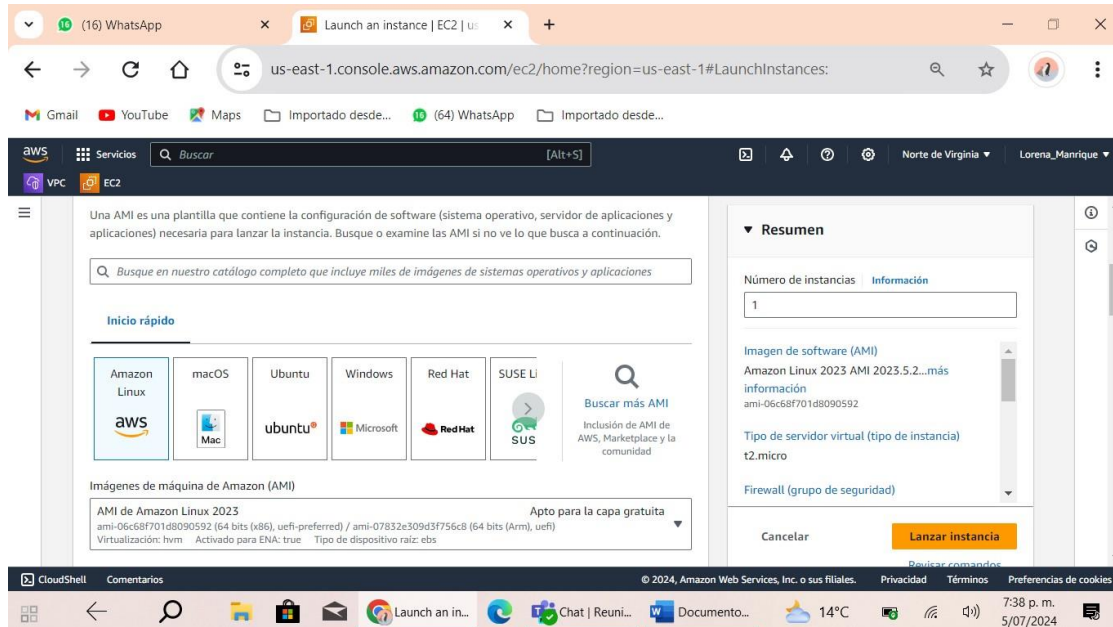


Figura2

Configurar red

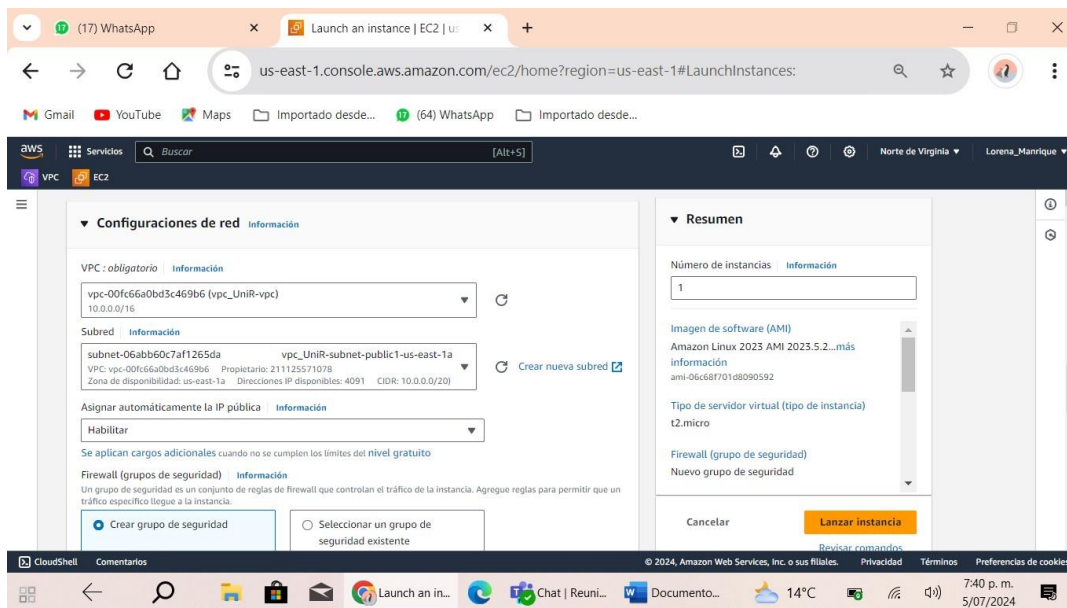


Figura3

Nombre red

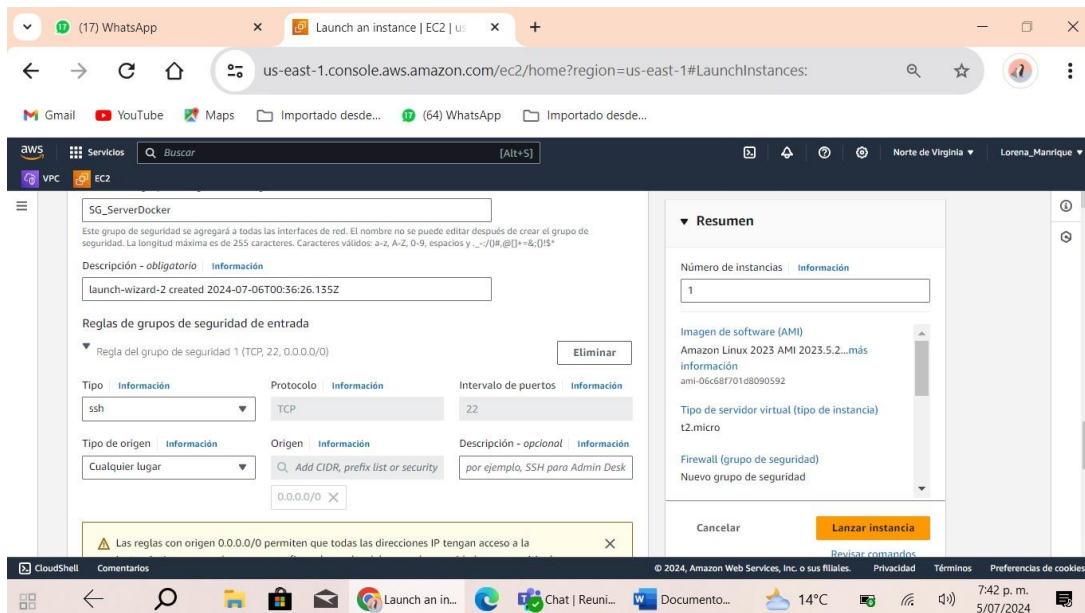


Figura4

Almacenamiento

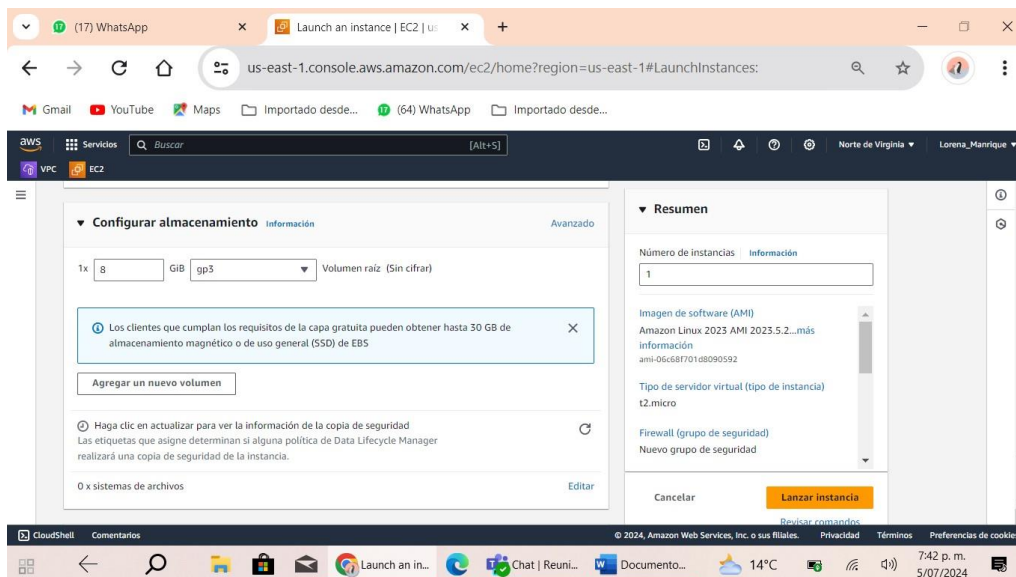
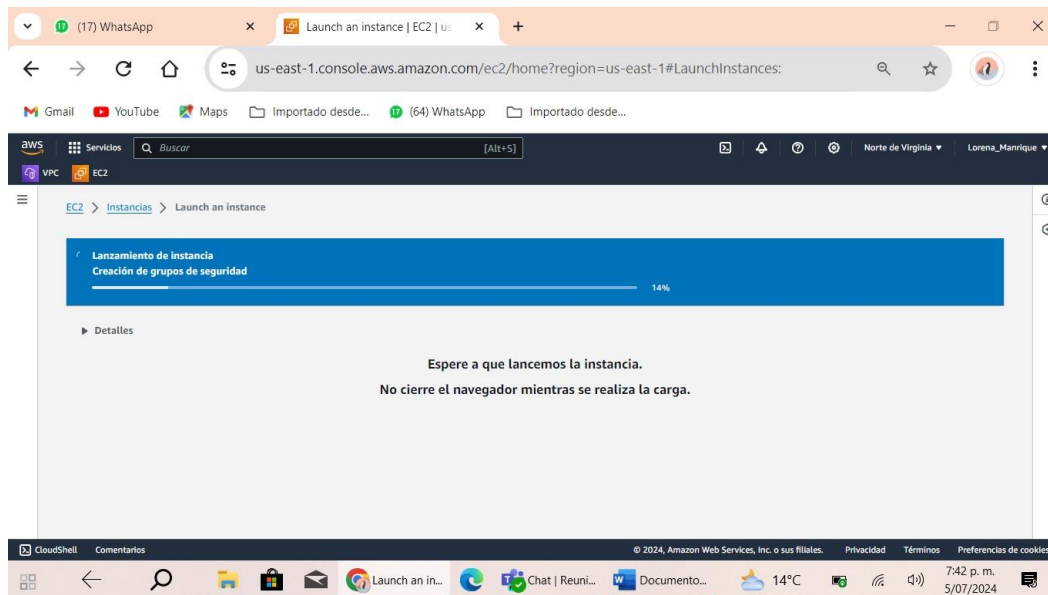


Figura5 creacion de instancia



A Continuación, vemos la forma de conectarnos a Linux si no estamos usando interfaz grafica

SE USA EL WSL (implementación que hizo Microsoft para tener una consola Linux ejecutándose sobre Linux).

Figura6

Programa consola

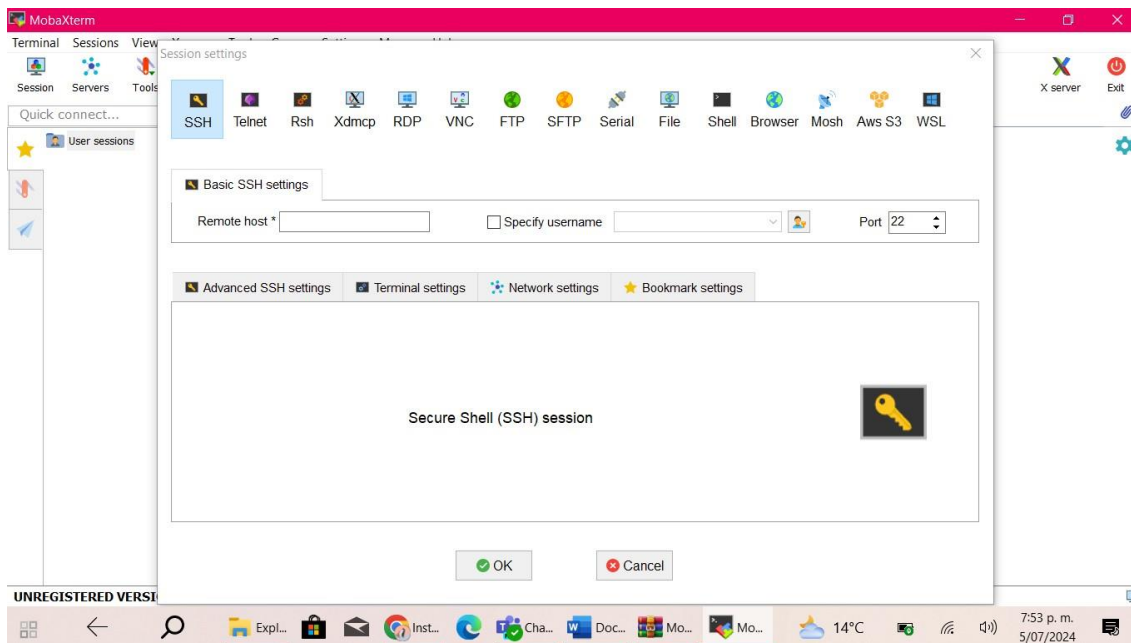


Figura 7

Características de la instancia

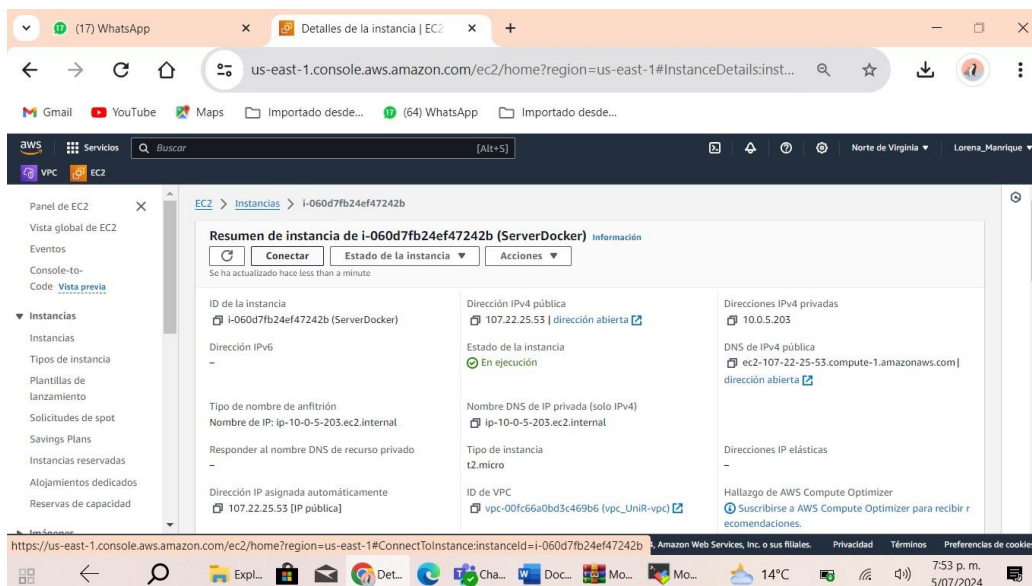
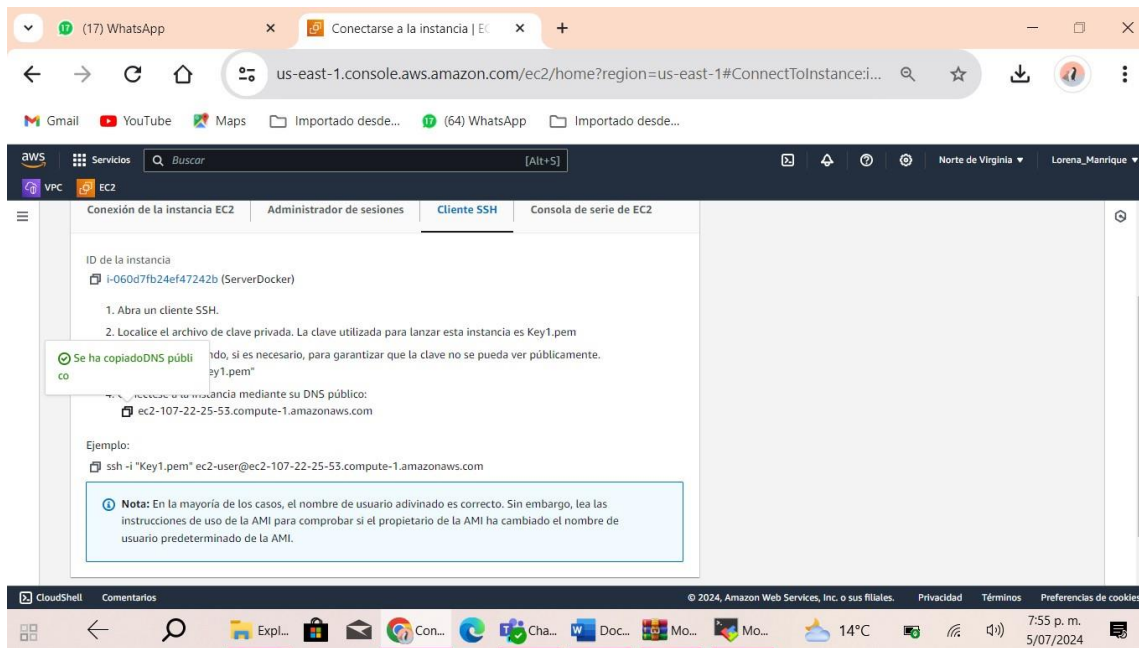


Figura8

ip de red



nombre o ip del equipo, el usuario, certificado de seguridad finalmente se observa que ya

hay conexión Figura9 ejecutar program

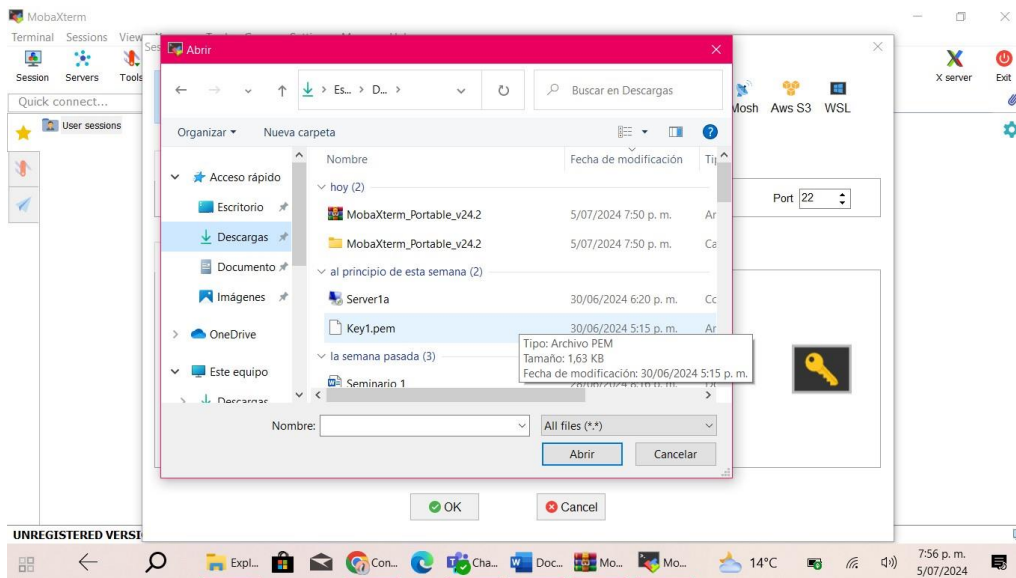


Figura12 linux instalad

```

Installing : libcgrou-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.

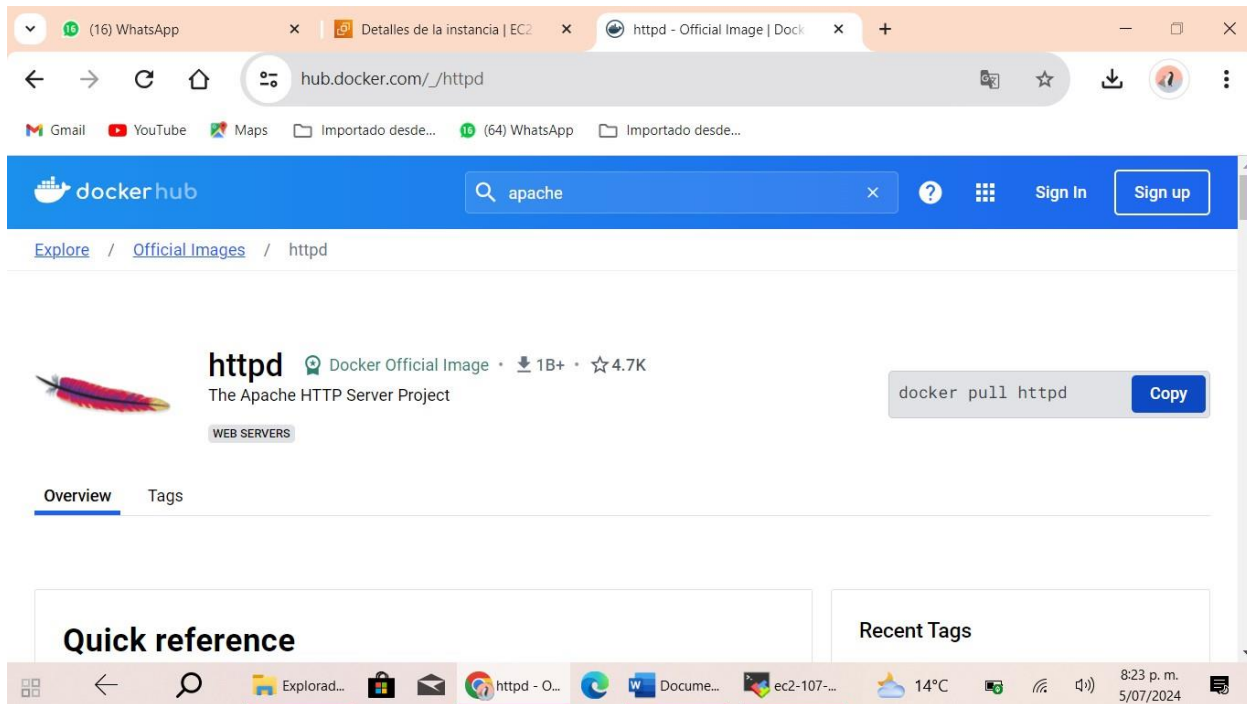
Verifying : containerd-1.7.11-1.amzn2023.0.1.x86_64 1/10
Verifying : docker-25.0.3-1.amzn2023.0.1.x86_64 2/10
Verifying : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64 3/10
Verifying : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 4/10
Verifying : libcgrou-3.0-1.amzn2023.0.1.x86_64 5/10
Verifying : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 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-1.1.11-1.amzn2023.0.1.x86_64 10/10

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-10-0-5-203 ec2-user]#

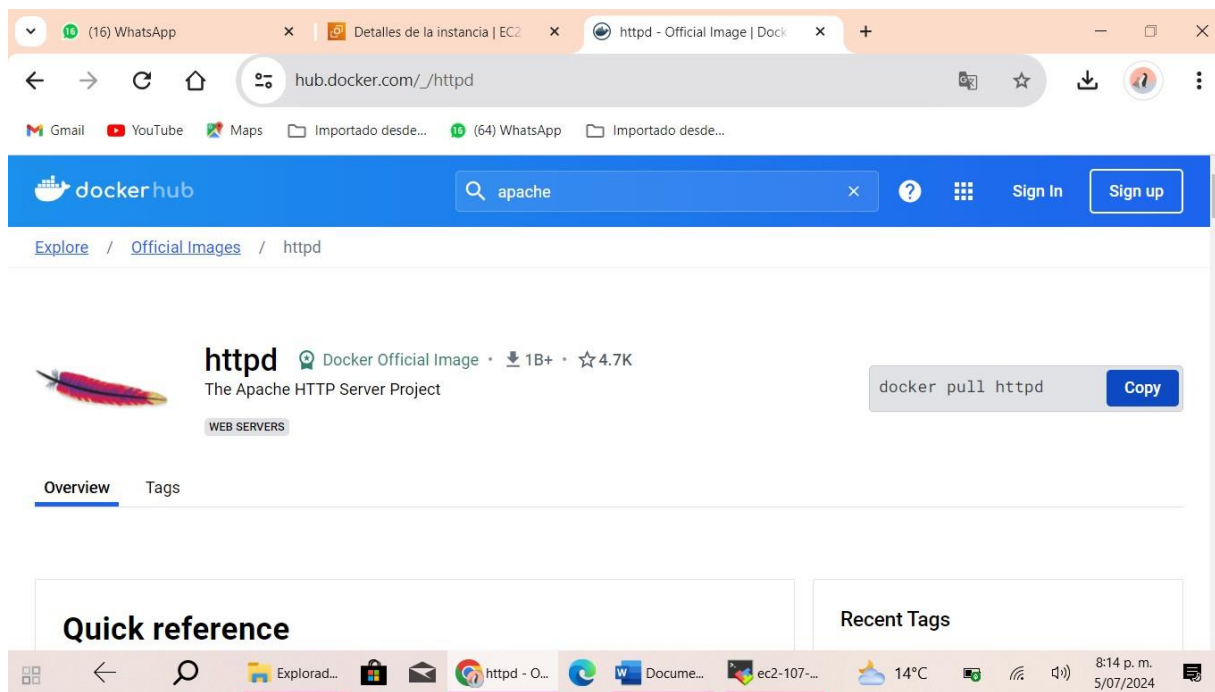
```

Docker hace uso de imágenes estas son básicamente archivos donde tiene al interior cual es la aplicación que va instalar. Esa imagen al crear el contenedor se le dice al Docker necesito crear un contenedor y va hacer uso de esa imagen con un nombre x cuando la inicio a usar en ese punto empieza a existir como tal un contenedor, entonces un contenedor no es posible que exista si no tiene una imagen porque esa imagen al interior tiene es la aplicación esas imágenes uno mismo las puede crear a través de código o hay un repositorio de internet donde puedo descargar imágenes que ya están creada. Esa imagen debe tener uno de esos servidores web además ya que la imagen esta yo al contenedor le paso los archivos de la aplicación. **Figura13 docker**



En obligatoria, eventualmente unos archivos opcionales

Figura14 copy docker



Vamos a crear un contenedor para que se ejecute con esa imagen

Crear carpeta en la raíz (tmp) dentro de tmt creo carpeta sitio 1y dentro de sitio1 voy a crear un archivo que se llame html y al interior de ese archivo voy poner hola a todos ⑦ /tmp/sitio1/index.html (hola a todos). Para acceder a esa carpeta se debe crear un acceso directo http

Figura15
contenedor

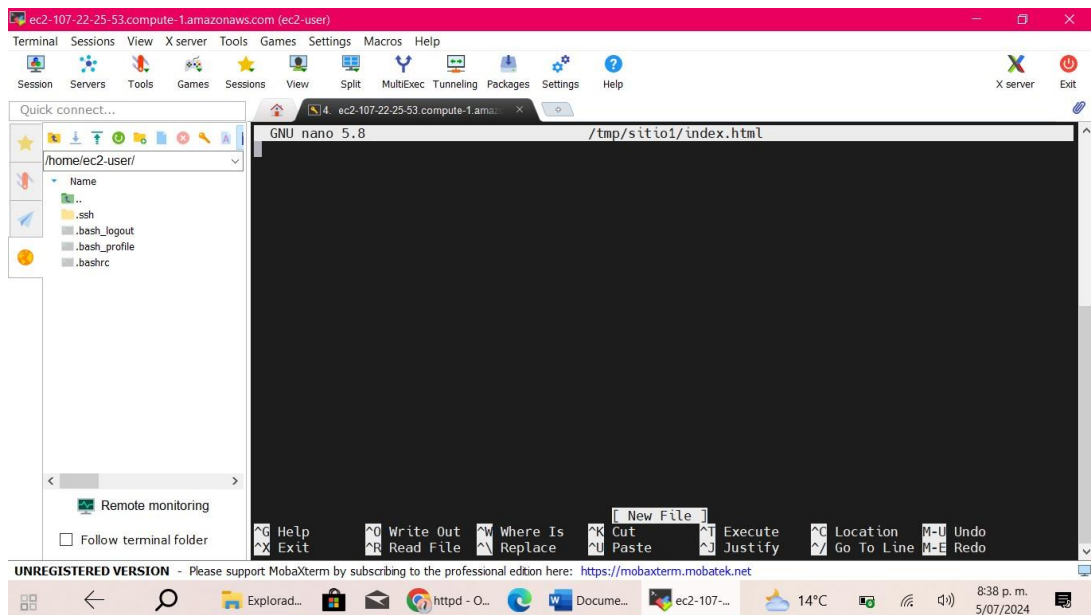
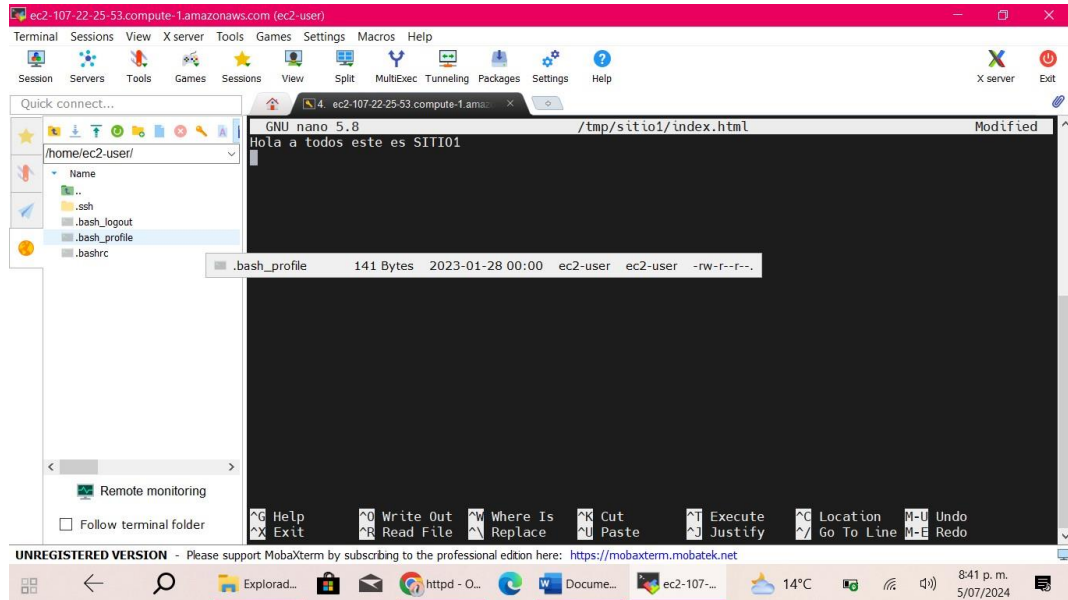


Figura16 conte archivo

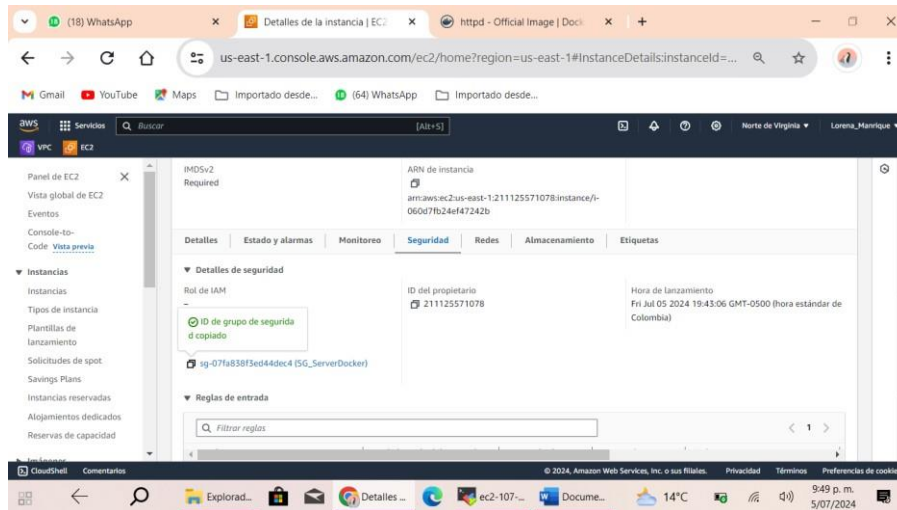


Se necesita un direccionamiento (el puerto)

Puerto: es una entrada o salida hacia un sistema informático

Apache crea un proceso dentro del contenedor y pone el puerto 80 a esperar conexiones

Ya tenemos el contenedor ejecutándose, ahora vamos a ver como se crea este contenedor **Figura19** cracion contenedor



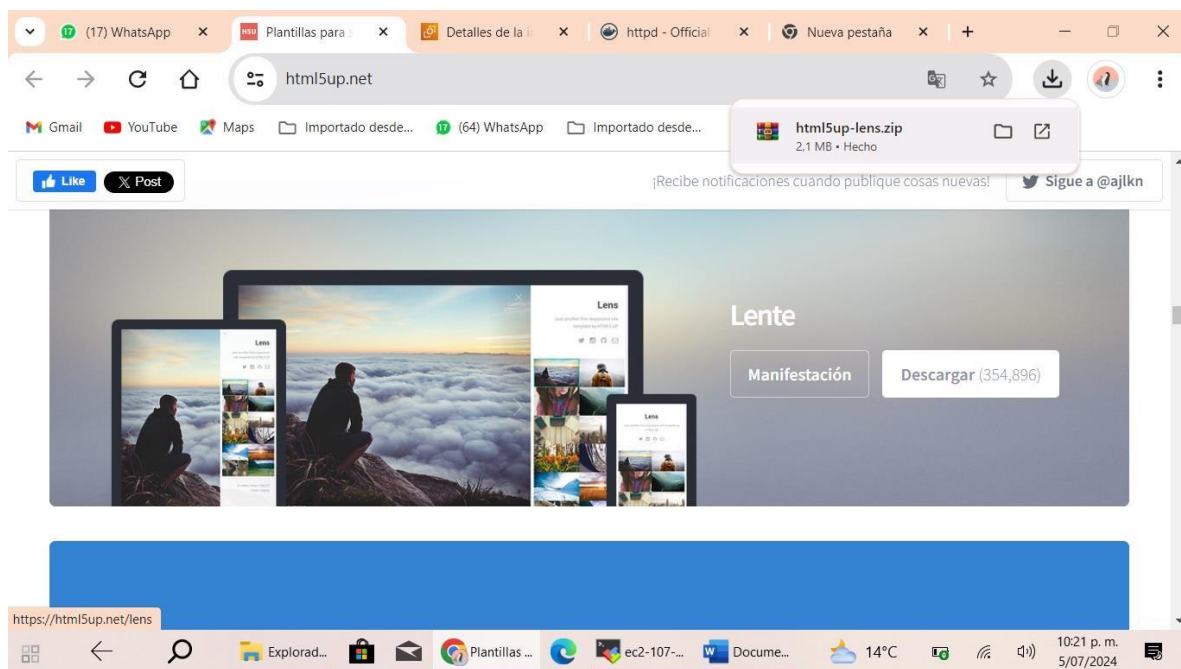
Se le asigna la regla nueva del puerto 8080 entre cualquiera así:

```
run -dit --name contenedor1 -p 8080:80 -v /tmp/sitio1/:/usr/local/apache2/htdocs/ httpd
```

empleamos una Plantilla de HTML 5

Figura20 plantilla

html



Ingresamos a la consola descomprimos el archivo, instalamos para descomprimos y finalmente nos muestra el sitio web

**Figura21 intalacion para
descomprimir**

```

[ec2-107-22-25-53.compute-1.amazonaws.com (ec2-user)]
Terminal Sessions View X server Tools Games Settings Macros Help
[home/ec2-user]
[ec2-107-22-25-53.compute-1.amazonaws.com (ec2-user)]
[root@ip-10-0-5-203 sitio1]# yum install unzip
Last metadata expiration check: 2:53:14 ago on Sat Jul 6 00:43:57 2024.
Package unzip-6.0-57.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-10-0-5-203 sitio1]# unzip download
unzip: cannot find or open download, download.zip or dowload.ZIP.
[root@ip-10-0-5-203 sitio1]# unzip download
Archive:  download
  inflating: LICENSE.txt
   creating: assets/
   creating: assets/webfonts/
  inflating: assets/webfonts/fa-regular-400.woff
  inflating: assets/webfonts/fa-regular-400.eot
  inflating: assets/webfonts/fa-brands-400.eot
  inflating: assets/webfonts/fa-solid-900.svg
  inflating: assets/webfonts/fa-regular-400.ttf
  inflating: assets/webfonts/fa-brands-400.woff2
  inflating: assets/webfonts/fa-regular-400.svg
  inflating: assets/webfonts/fa-solid-900.woff
  inflating: assets/webfonts/fa-brands-400.woff2
  inflating: assets/webfonts/fa-solid-900.eot
  inflating: assets/webfonts/fa-brands-400.svg
  inflating: assets/webfonts/fa-solid-900.woff2
  inflating: assets/webfonts/fa-brands-400.eot
  inflating: assets/webfonts/fa-solid-900.eot
  inflating: assets/webfonts/fa-brands-400.ttf
  extracting: assets/webfonts/fa-regular-400.woff2
  
```

Figura 22 descomprimir

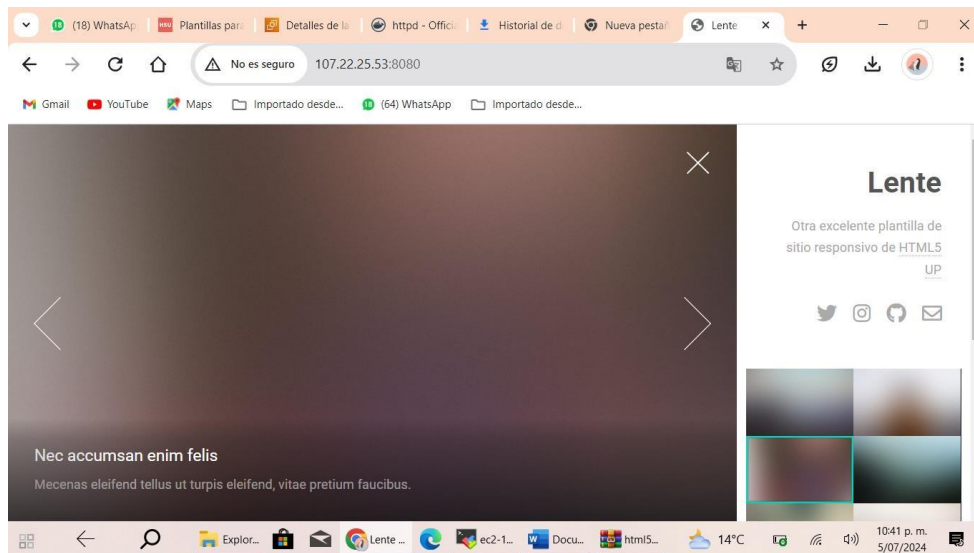
```

[ec2-107-22-25-53.compute-1.amazonaws.com (ec2-user)]
Terminal Sessions View X server Tools Games Settings Macros Help
[home/ec2-user]
[ec2-107-22-25-53.compute-1.amazonaws.com (ec2-user)]
  inflating: images/thumbs/09.jpg
  inflating: images/thumbs/04.jpg
  inflating: images/thumbs/05.jpg
  inflating: images/thumbs/07.jpg
  inflating: images/thumbs/03.jpg
  inflating: images/thumbs/11.jpg
  inflating: images/thumbs/02.jpg
  inflating: images/thumbs/01.jpg
  inflating: images/thumbs/10.jpg
  inflating: images/thumbs/06.jpg
  inflating: images/thumbs/12.jpg
   creating: images/fulls/
  inflating: images/fulls/08.jpg
  inflating: images/fulls/09.jpg
  inflating: images/fulls/04.jpg
  inflating: images/fulls/05.jpg
  inflating: images/fulls/07.jpg
  inflating: images/fulls/03.jpg
  inflating: images/fulls/11.jpg
  inflating: images/fulls/02.jpg
  inflating: images/fulls/01.jpg
  inflating: images/fulls/10.jpg
  inflating: images/fulls/06.jpg
  inflating: images/fulls/12.jpg
[root@ip-10-0-5-203 sitio1]# ls
LICENSE.txt  README.txt  assets  download  images  index.html
[root@ip-10-0-5-203 sitio1]#
  
```

Sitio implementado

Figura23 *implementacion*

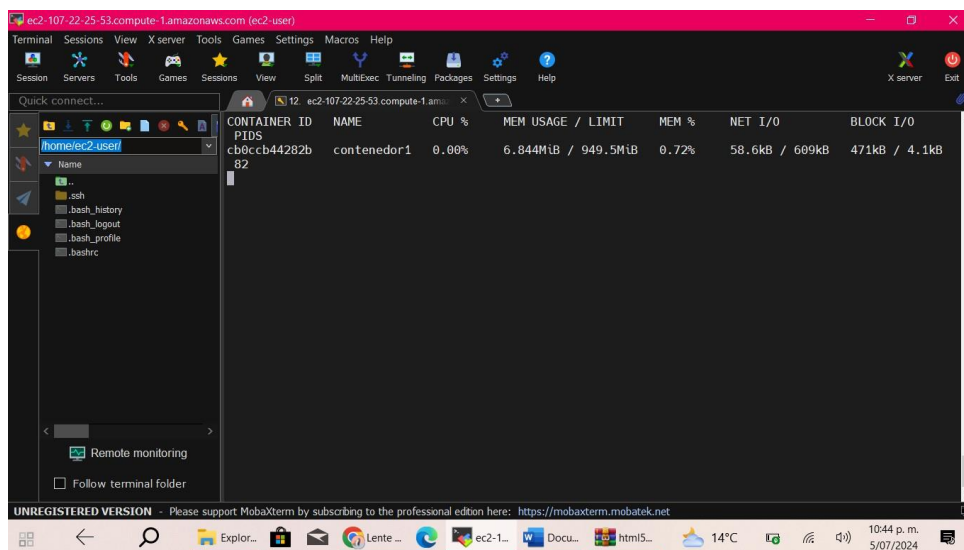
del sitio



Miramos el consumo de memoria ya observamos que va aumentando con el comando

Figura24

Consumo de memoria

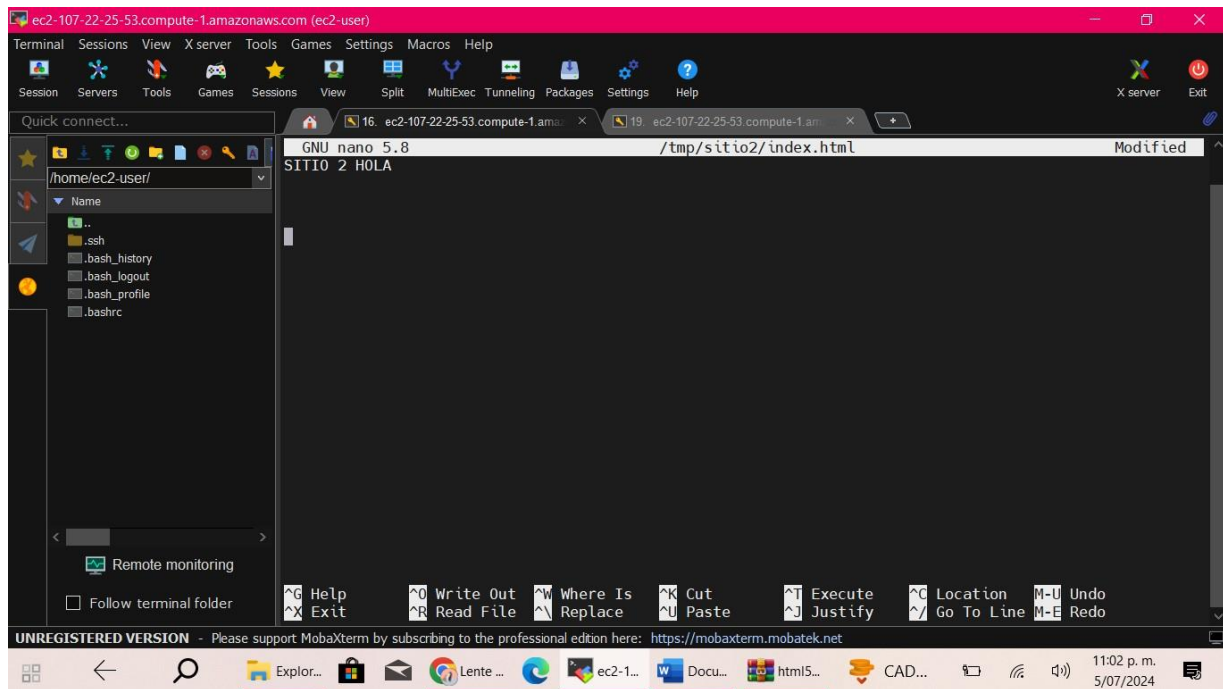


Ahora voy a crear el segundo contenedor

En la misma ubicación solo cambio a sitio2

Figura25

Creacion contenedor 3



Lanzamos un contenedor que va ser contenedor2 y sitio2 hacemos lo mismo que cuando se creo el contenedor1 solo que se cambia a contenedor 2 y sitio2 cambiamos al puerto 8081 porque no se puede repetir el mismo del contenedor 1

Figura26 proceso
contenedor 2

```

Last login: Sat Jul 6 03:57:25 2024 from 186.84.89.143
[ec2-user@ip-10-0-5-203 ~]$ docker stop
"docker stop" requires at least 1 argument.
See 'docker stop --help'.

Usage: docker stop [OPTIONS] CONTAINER [CONTAINER...]

Stop one or more running containers
[ec2-user@ip-10-0-5-203 ~]$ sudo su
[root@ip-10-0-5-203 ec2-user]# docker run -dit --name contenedor2 -p 8080:80 -v /temp/sitio2/:/usr/local/apache2/htdocs/ httpd
345905809eaf98c33abec0dd738d631593eae9fc786aa863cb34134f57773ac4
docker: Error response from daemon: driver failed programming external connectivity on endpoint contenedor2 (17939ec779910c3e600aae2f51608ef23a51f17baaf63d165e6a96b47221b817): Bind for 0.0.0.0:8080 failed: port is already allocated.
[root@ip-10-0-5-203 ec2-user]# docker run -dit --name contenedorN2 -p 8081:80 -v /temp/sitio2/:/usr/local/apache2/htdocs/ httpd
da0be62246f1cb88bf1f0edf9ce43a2b67eb3fe57ef7a41b4d4ac56e0f9ee3d1
[root@ip-10-0-5-203 ec2-user]#

```

Para poder ver los dos contenedores que tenemos escribimos en la consola (docker ps) y se pueden observar **Figura27 docker ps**

```

Last login: Sat Jul 6 03:57:25 2024 from 186.84.89.143
[ec2-user@ip-10-0-5-203 ~]$ docker stop
"docker stop" requires at least 1 argument.
See 'docker stop --help'.

Usage: docker stop [OPTIONS] CONTAINER [CONTAINER...]

Stop one or more running containers
[ec2-user@ip-10-0-5-203 ~]$ sudo su
[root@ip-10-0-5-203 ec2-user]# docker run -dit --name contenedor2 -p 8080:80 -v /temp/sitio2/:/usr/local/apache2/htdocs/ httpd
345905809eaf98c33abec0dd738d631593eae9fc786aa863cb34134f57773ac4
docker: Error response from daemon: driver failed programming external connectivity on endpoint contenedor2 (17939ec779910c3e600aae2f51608ef23a51f17baaf63d165e6a96b47221b817): Bind for 0.0.0.0:8080 failed: port is already allocated.
[root@ip-10-0-5-203 ec2-user]# docker run -dit --name contenedorN2 -p 8081:80 -v /temp/sitio2/:/usr/local/apache2/htdocs/ httpd
da0be62246f1cb88bf1f0edf9ce43a2b67eb3fe57ef7a41b4d4ac56e0f9ee3d1
[root@ip-10-0-5-203 ec2-user]# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
da0be62246f1   httpd         "httpd-foreground"     49 seconds ago Up 48 seconds  0.0.0.0:8081->80/tcp
cp, :::8081->80/tcp
contenedorN2
cb0cc44282b   httpd         "httpd-foreground"     About an hour ago Up About an hour  0.0.0.0:8080->80/tcp
cp, :::8080->80/tcp
contenedor1
[root@ip-10-0-5-203 ec2-user]#

```

Hay que recordar que para poder ejecutar en los puertos ya sea 8080 o 8081 que es donde tenemos los contenedores si quiero observarlo desde a fuera tenemos que volver al grupo de

seguridad editar en reglas de seguridad o (inbound rules) y agregar los puertos bien sea 8080 o 8081 dar la opción que se vea desde cualquier lugar y le damos guardar reg **Figura28** *pueba puerto 8081*

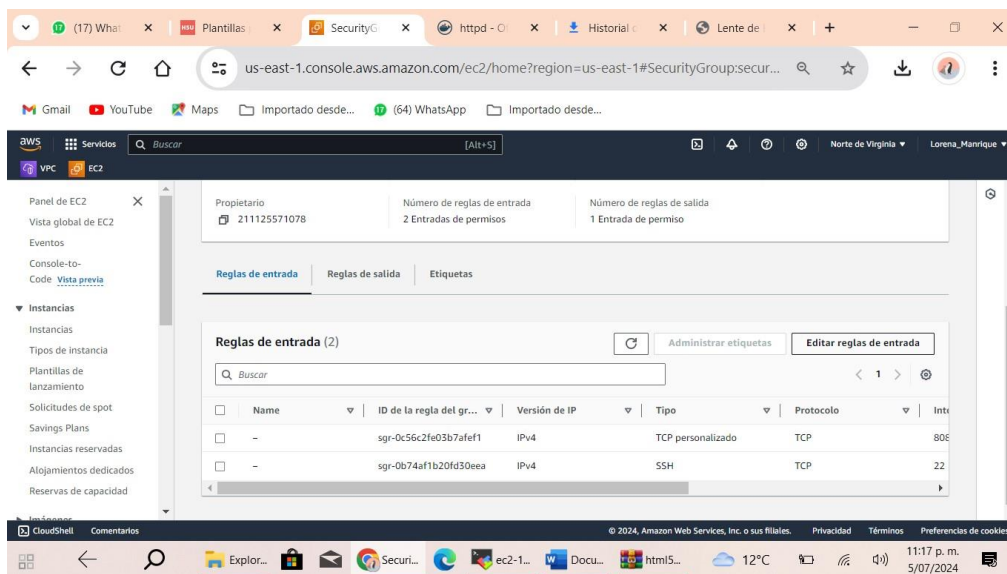


Figura29 *cracion del*

puerto aws

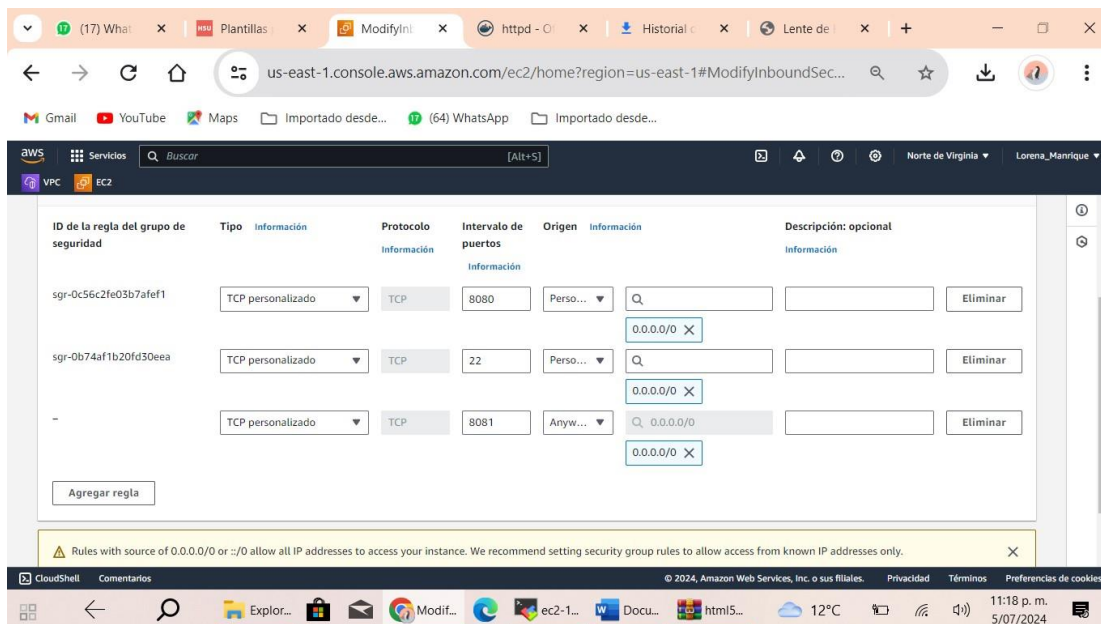
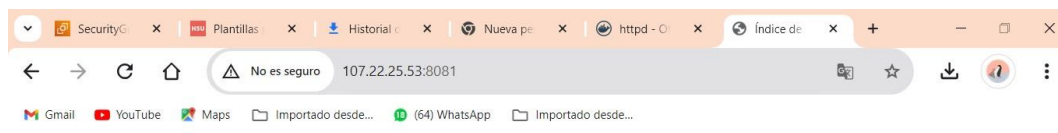


Figura30 vemos la pagina
con el puerto



Finalmente vamos a montar el balanceador de cargas, dentro de la maquina vamos a

instalar nginx.

Figura31 balanceo

de carga

```

ec2-107-22-25-53.compute-1.amazonaws.com (ec2-user)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
/home/ec2-user/
Name
..
.ssh
ç
.bash_history
.bash_logout
.bash_profile
.bashrc
Remote monitoring
Follow terminal folder
Running transaction
Preparing :
Running scriptlet: nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch 1/1
Installing : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch 1/7
Installing : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch 2/7
Installing : libunwind-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-1:1.24.0-1.amzn2023.0.2.x86_64 5/7
Installing : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 6/7
Installing : nginx-1:1.24.0-1.amzn2023.0.2.x86_64 7/7
Running scriptlet: nginx-1:1.24.0-1.amzn2023.0.2.x86_64 7/7
Verifying : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 1/7
Verifying : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64 2/7
Verifying : libunwind-1.4.0-5.amzn2023.0.2.x86_64 3/7
Verifying : nginx-1:1.24.0-1.amzn2023.0.2.x86_64 4/7
Verifying : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64 5/7
Verifying : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch 6/7
Verifying : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch 7/7

Installed:
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64
libunwind-1.4.0-5.amzn2023.0.2.x86_64 nginx-1:1.24.0-1.amzn2023.0.2.x86_64
nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64 nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch
nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

Complete!
[root@ip-10-0-5-203 ec2-user]#
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
Escritorio Indice de... ec2-107-... ENTREG... 12°C 9:55 a.m. 6/07/2024

```

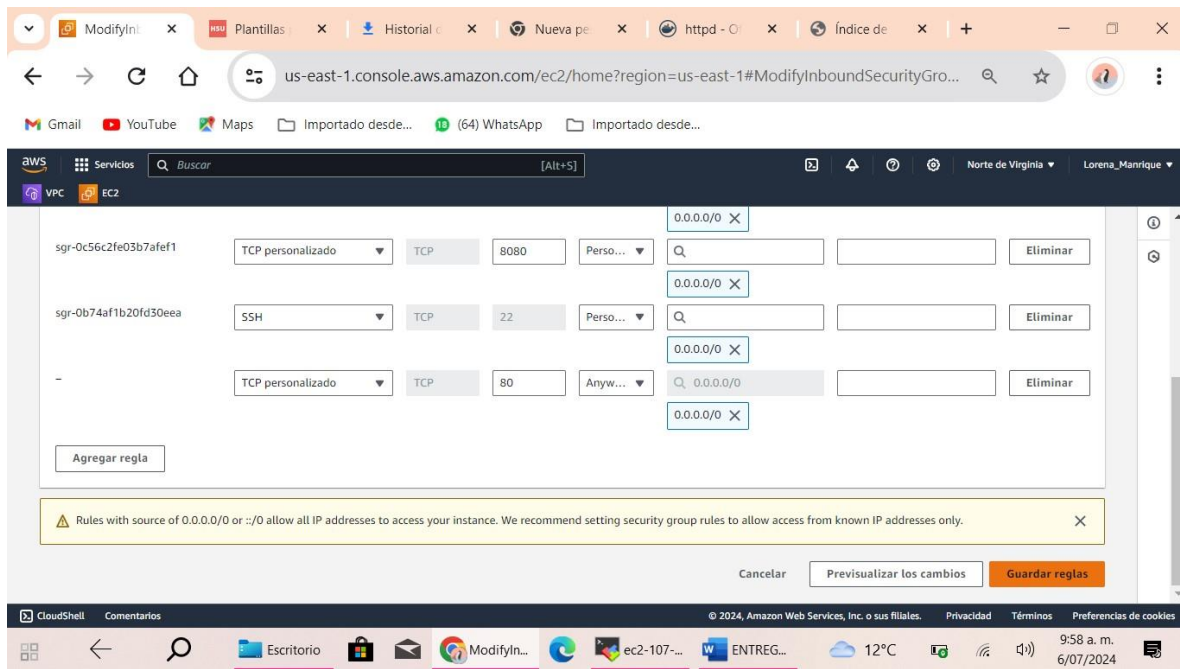
Luego iniciamos el nginx

Nginx por defecto va usar el puerto 80 del Amazon Linux este es independiente

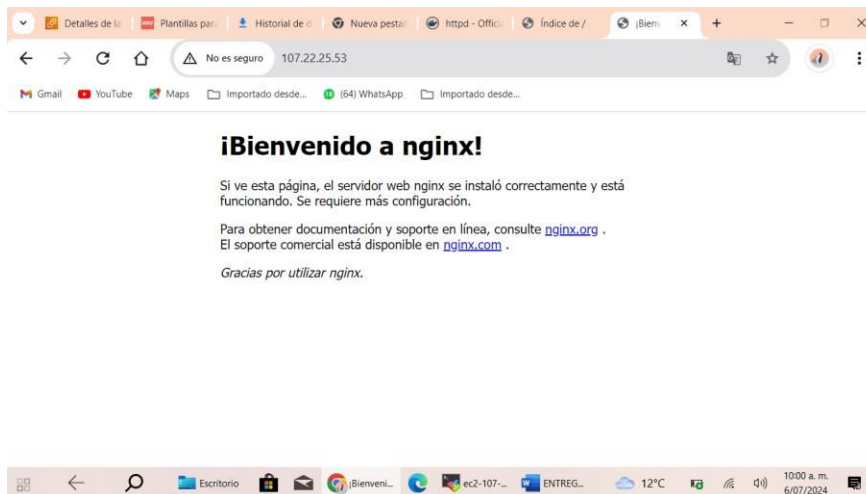
Vamos a crear una regla que nos permita entrar al puerto 80

Figura32

creacion de regla



Con ese pantallazo ya observamos que el nginx ya esta funcionando es un servidor mas es un competidor de apache **Figura33 funcionamiento del nginx**



Lo anterior lo vamos a convertir en un balanceo de carga que todas las peticiones que ingresen a esta maquina por el puerto 8080 se va encargar de distribuirlas a los contenedores que yo quiera

Modificamos el archivo nginx.conf primero hay que hacerle copia

Figura34 *conf*

The screenshot shows a terminal window with the following content:

```

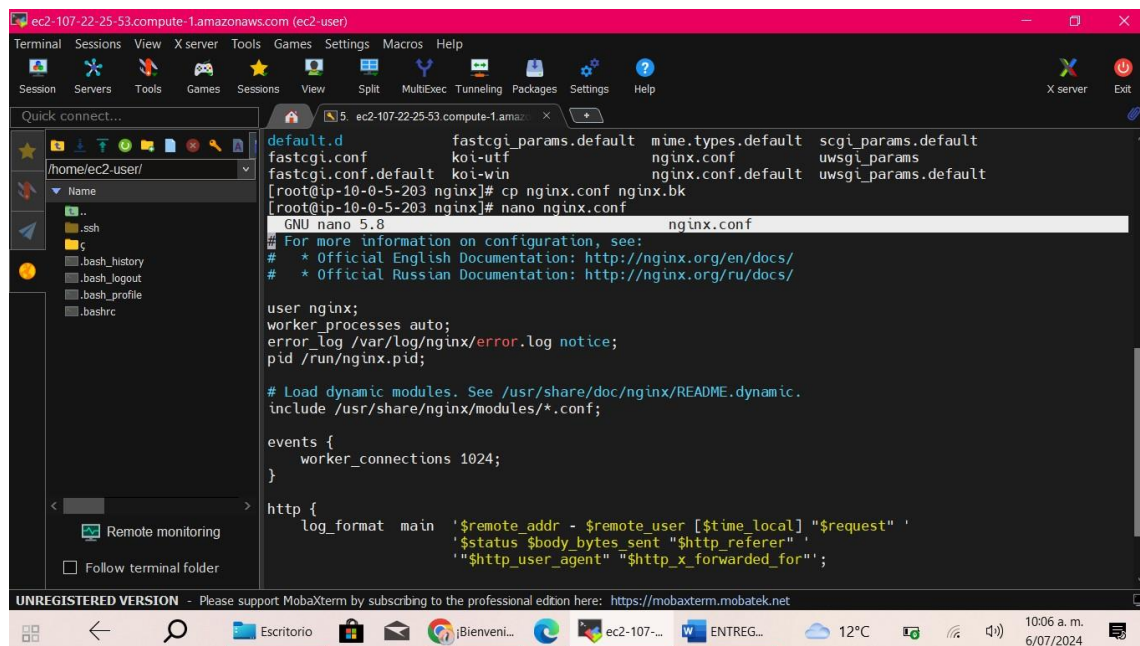
ec2-107-22-25-53.compute-1.amazonaws.com (ec2-user)
Terminal Sessions View X server Tools Games Settings Macros Help
Quick connect...
/home/ec2-user/
Name
..
.ssh
ç
.bash_history
.bash_logout
.bash_profile
.bashrc

Direct SSH : ✓
SSH compression : ✓
SSH-browser : ✓
X11-forwarding : ✗ (disabled or not supported by server)
For more info, ctrl+click on help or visit our website.

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

Last login: Sat Jul 6 14:54:09 2024 from 186.84.88.48
[ec2-user@ip-10-0-5-203 ~]$ sudo su
[root@ip-10-0-5-203 ec2-user]# cd /etc/nginx/
[root@ip-10-0-5-203 nginx]# ls
conf.d          fastcgi_params      mime.types          scgi_params        win-utf
default.d       fastcgi_params.default  mime.types.default  scgi_params.default
fastcgi.conf    koi-utf             nginx.conf          uwsgi_params
fastcgi.conf.default  koi-win             nginx.conf.default  uwsgi_params.default
[root@ip-10-0-5-203 nginx]# cp nginx.conf nginx.bk
  
```

The terminal window is running on an Amazon Linux 2023 instance. The user has successfully switched to root and listed the contents of the /etc/nginx directory. The output shows several configuration files, including conf.d, default.d, fastcgi.conf, fastcgi.conf.default, koi-utf, koi-win, mime.types, mime.types.default, nginx.conf, nginx.conf.default, scgi_params, scgi_params.default, uwsgi_params, uwsgi_params.default, and win-utf. The user has just executed the command to copy nginx.conf to nginx.bk.

Figura35 *conf**consola*

```
ec2-107-22-25-53.compute-1.amazonaws.com (ec2-user)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
/home/ec2-user/
Name
...
.ssh
ζ
.bash_history
.bash_logout
.bash_profile
.bashrc
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

default.d      fastcgi_params.default  mime.types.default  scgi_params.default
fastcgi.conf   koi-utf                 nginx.conf           uwsgi_params
fastcgi.conf.default  koi-win                nginx.conf.default  uwsgi_params.default
[root@ip-10-0-5-203 nginx]# cp nginx.conf nginx.bk
[root@ip-10-0-5-203 nginx]# nano nginx.conf
GNU nano 5.8 nginx.conf
# For more information on configuration, see:
# * Official English Documentation: http://nginx.org/en/docs/
# * Official Russian Documentation: http://nginx.org/ru/docs/

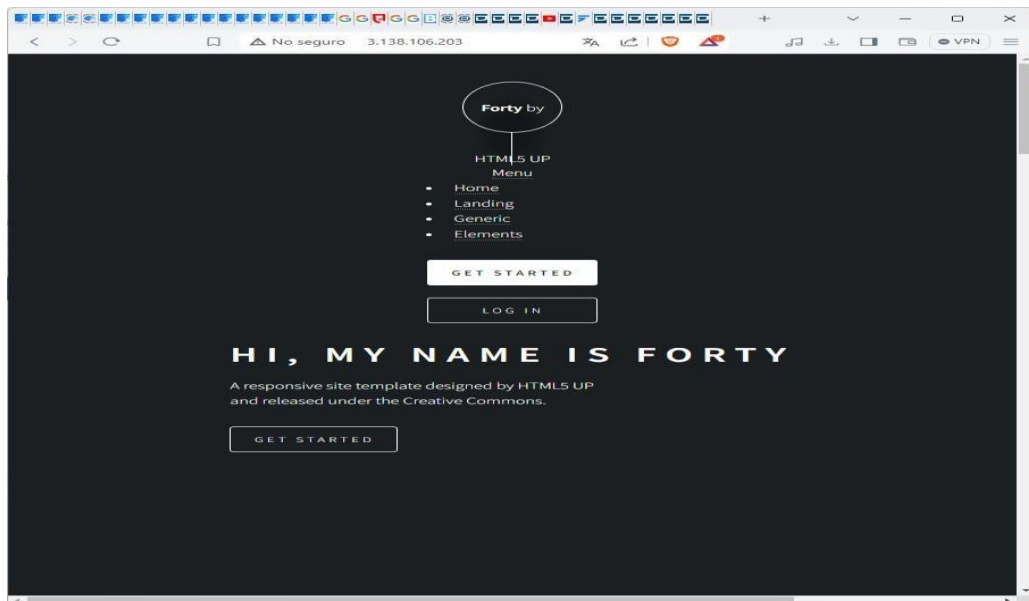
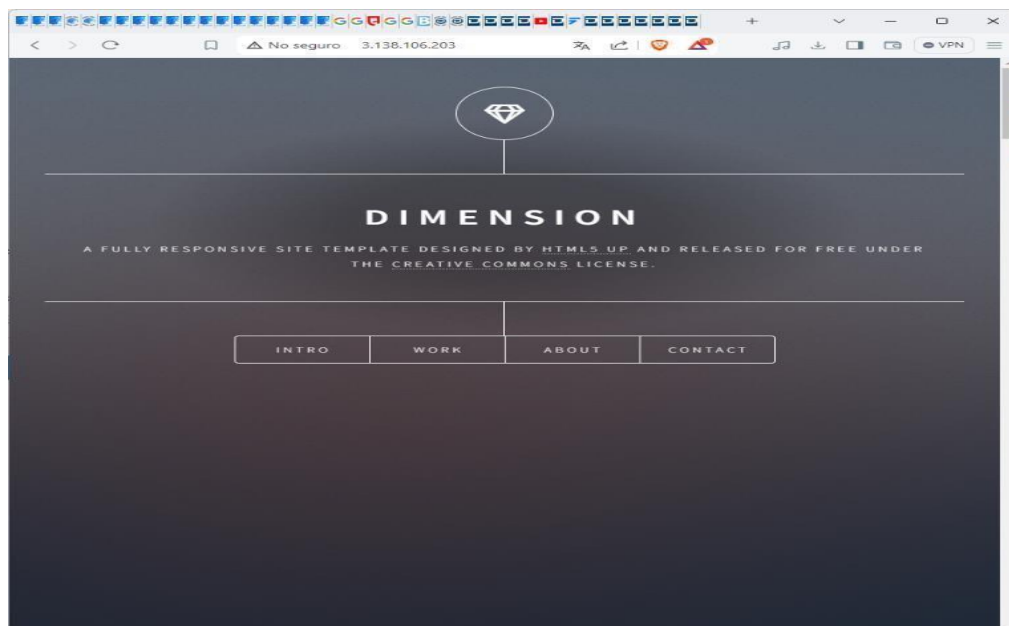
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

http {
    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                   '$status $body_bytes_sent "$http_referer" '
                   '"$http_user_agent" "$http_x_forwarded_for"';
```

Apunte a la ip publica que tiene al Amazon Linux la petición ingresa al puerto 80 esta funcionando como un balanceador en una petición muestra el sitio que está en el contenedor 1 se le da actualizar a la página eso es una petición nueva y muestra el sitio que está en el contenedor 2

Figura36*Balanced 1***Figura37***Balanced 2*

Bien sabemos que la utilidad de los contenedores se puede tener contenedores con muchas aplicaciones ejecutándose en esa maquina virtual o puedo tener contenedores ejecutándose con la misma aplicación para tenerlos funcionando como redundancia ósea que los contenedores tendrían la misma aplicación llegan las peticiones donde se balancean las cargas si se esta usando mucho recursos cuando se hace ese balanceo pues el uso de los recursos en los contenedores se disminuye si se tiene mas de un contenedor se distribuyen de forma equitativa en los diferentes contenedores que haya es una de las ventajas de usar los contenedores y hacer uso de las balances de carga y eso nos permite

Implemente un sitio estatutico en S3

S3: Servicio que es muy simple utiliza pocas cosas, pero es muy útil S3 sirve para almacenar objetos diferente a como se utiliza un disco duro, no va definir un espacio límite de almacenamiento ingresa la información y se expande la capacidad de forma limitada

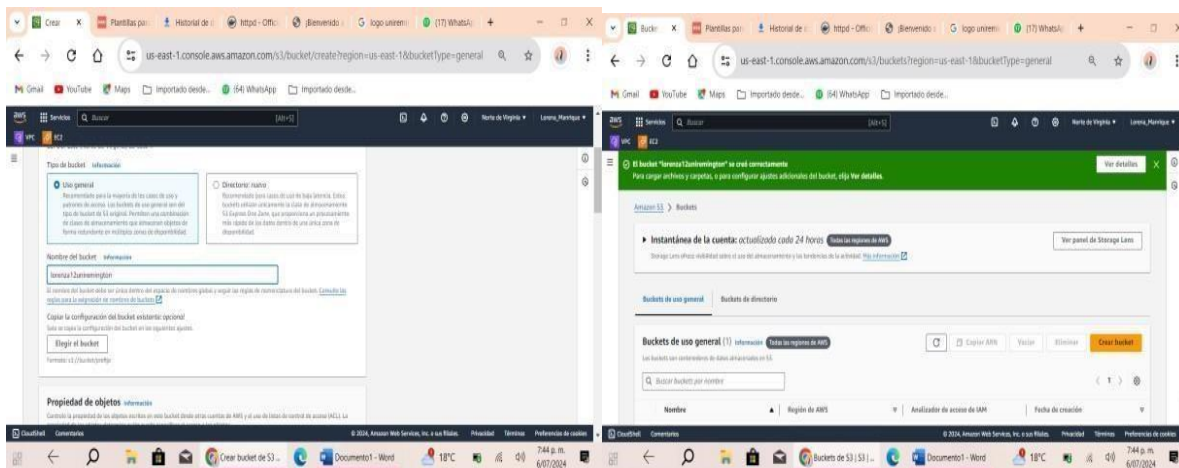
Objeto: archivo de cualquier tipo de extensión en aws solo se paga el espacio que ocupa por ese objeto, aws se encarga de administrar la copia de ese objeto en otro lado si hay un daño en la administración principal que se está trabajando ellos se encargan de recuperar esa información. Los objetos pueden tener una capacidad de 5 teras

Buckets: es una carpeta donde se guardan objetos es un servicio regional, cuando se creen necesitan un nombre y ese nombre es único en el mundo. Ellos tienen varias sesiones para proteger como ACLS (listas de control de acceso)-> por defecto siempre es privada y la quiero hacer pública se tiene que hacer una configuración. Puede tener habilitado el versionamiento de archivos, puede guardar la cantidad de versiones que quiera, pero, así como se guardan hay que tener en cuenta que también las cobran.

Creo el buckets

Figura 38

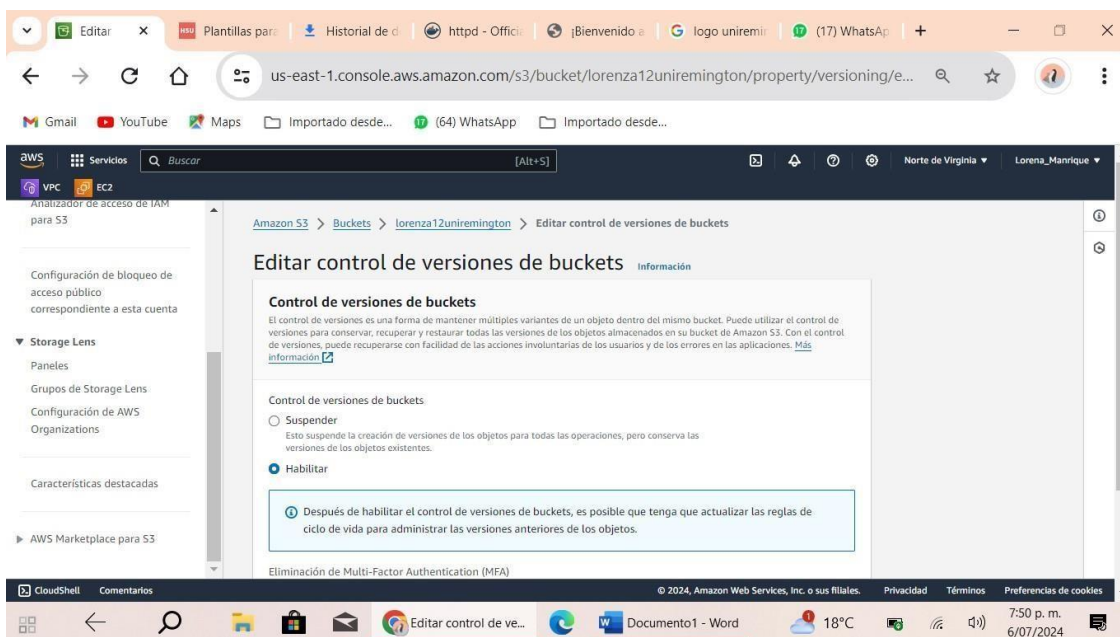
Buckets



Habilitamos el versionamiento

Figura 39

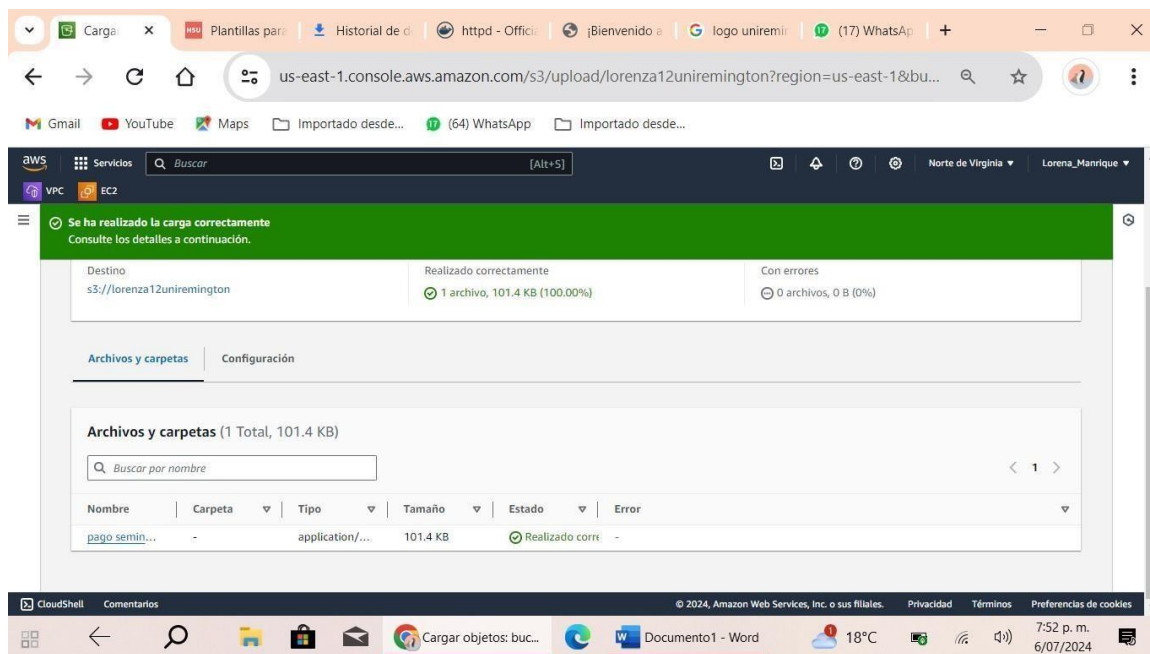
Versionamiento



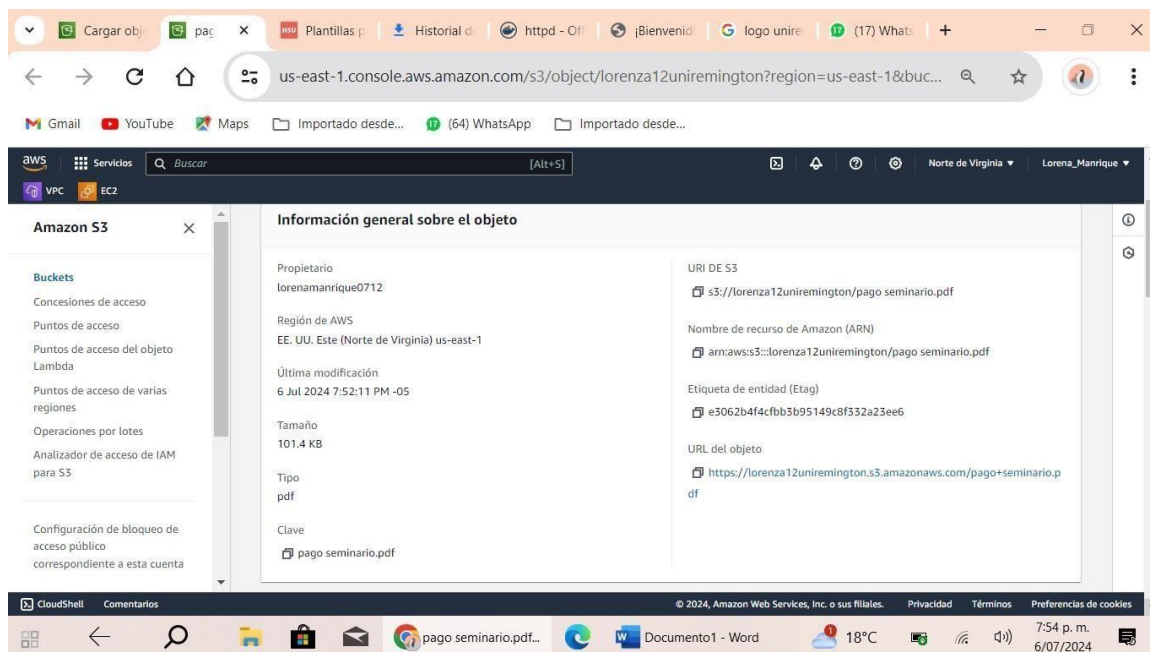
En objeto cargamos un archivo y tenemos un archivo de prueba

Figura 40

Cargue de archivo



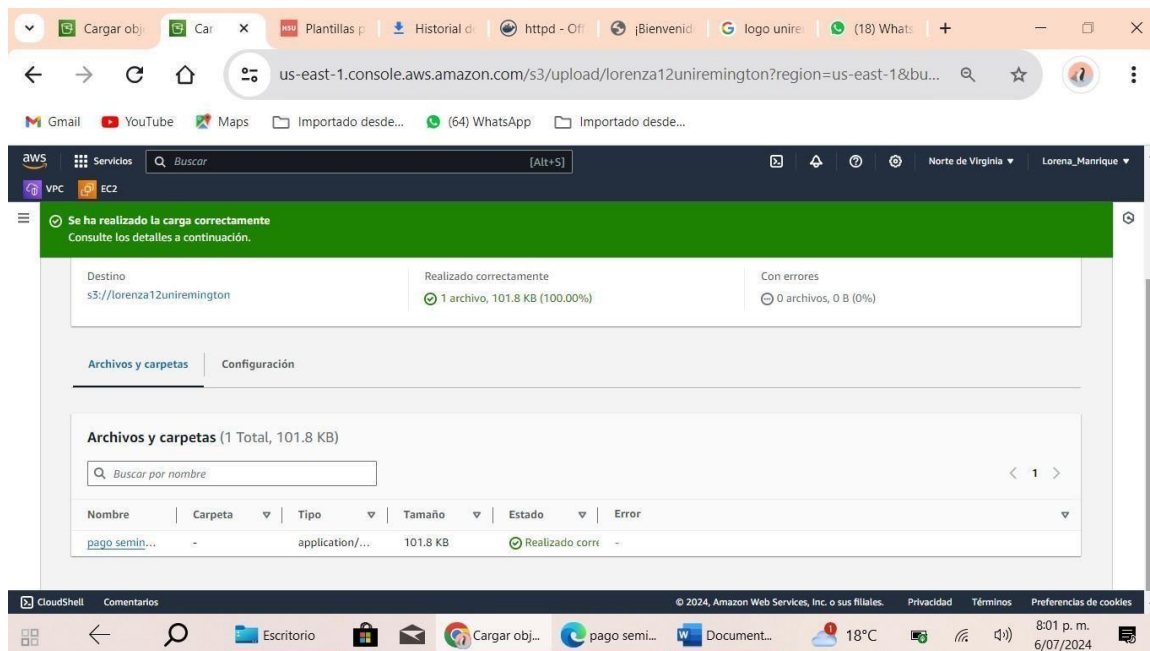
Ingreso al archivo que se creó y no va a mostrar el contenido del archivo sino la información de S3 respecto a ese objeto

Figura**41***Informacion S3*

Luego vuelvo a subir el mismo archivo, pero modificado dentro de el

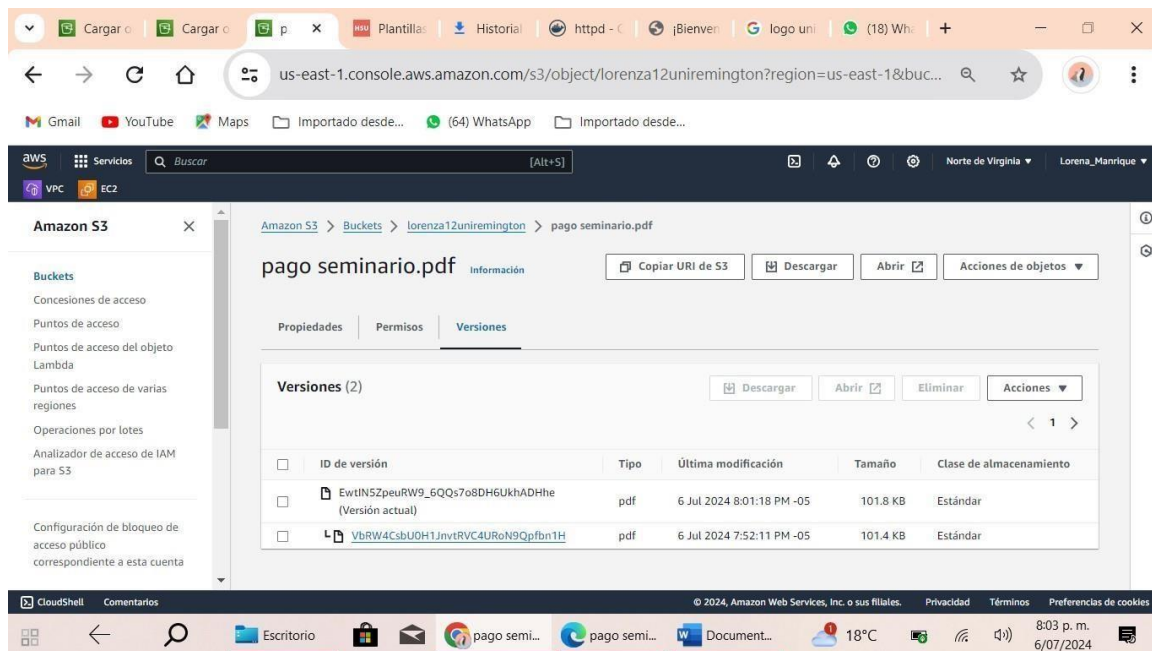
42

Figura
Archivo modificado



A continuación, podemos observar las dos versiones identifica fecha y la hora en la que subió el archivo y el peso, pero ahí cobran el almacenamiento de las dos ahí se guardan las respectivas versiones antiguas o recientes modificadas

Figura
versiones de los archivos

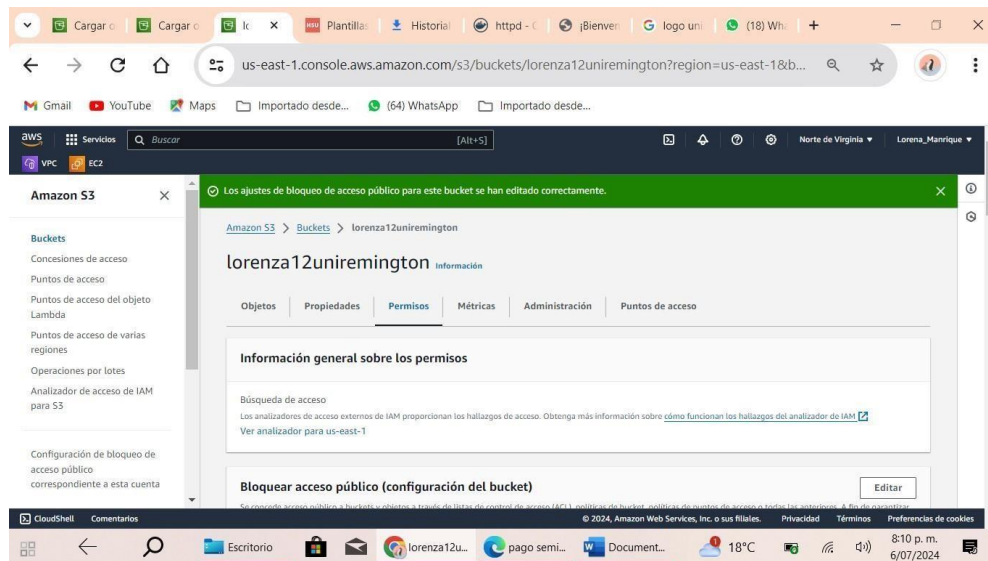


Damos permisos y confirmamos los cambios entonces ese objeto

44

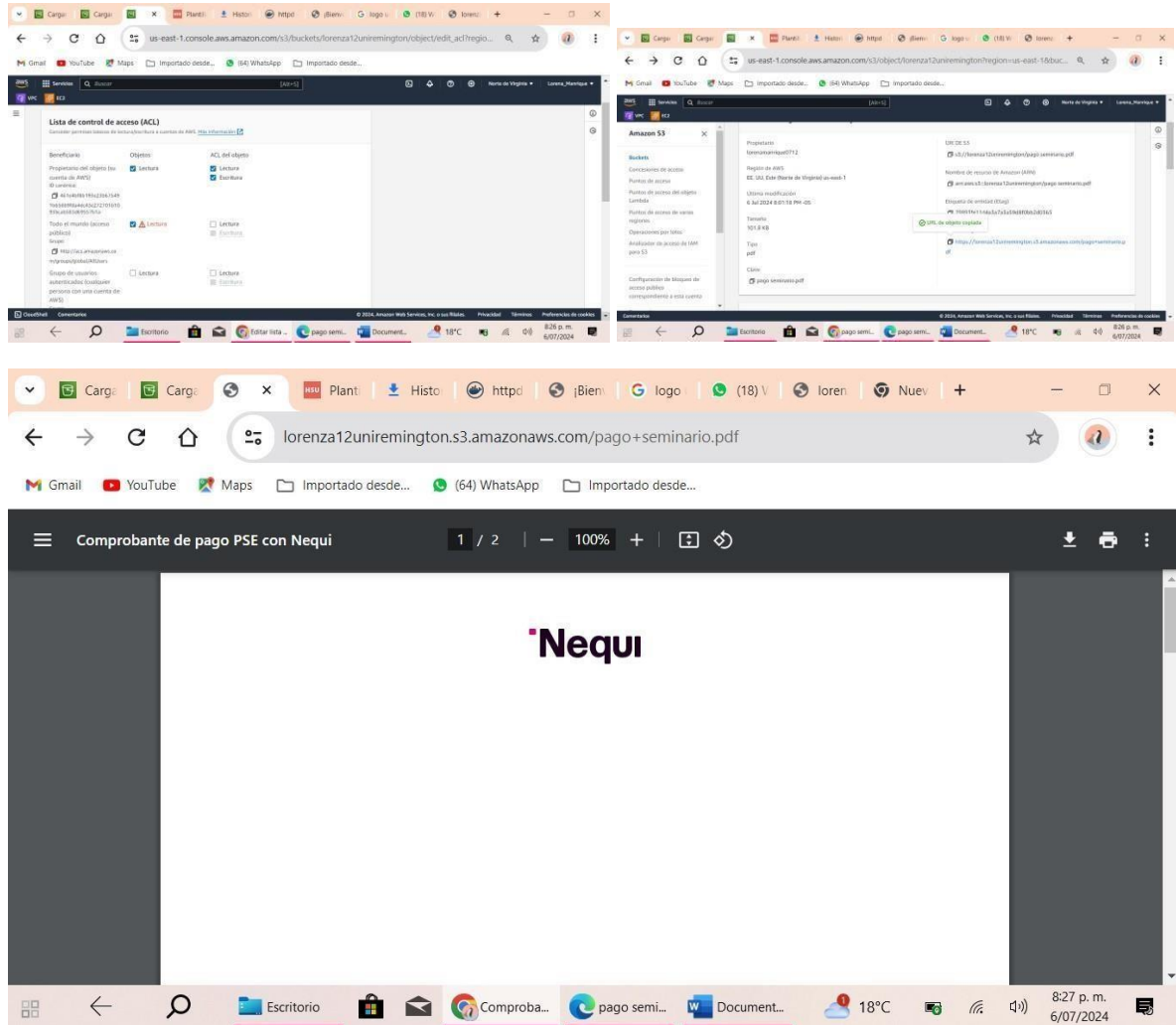
Configuración de cambios

Figura



Habilitamos las ACL que son las que nos permiten entrar dentro del objeto editar los permisos y decir que ese buckets va a tener acceso público desde internet y ya poder ver el contenido. Quiere decir que ya tengo acceso público.

Figura
Contenido del archivo con acceso publico



Convertir que un Buckets sea un sitio web estático, que mientras información en el navegador sin funciones que necesite cálculos desde algún lado algún servidor, editamos los permisos habilitamos y ingresamos el `index.html` y guardamos cambios

Figura
Sitio web estatico

The figure consists of two screenshots from the AWS Management Console, showing the configuration and details of an Amazon S3 bucket named 'lorenza12uniremington'.

Top Screenshot: Editar alojamiento de sitios web estáticos

The page title is "Editar alojamiento de sitios web estáticos". The main content area shows the configuration options for static website hosting:

- Alojamiento de sitios web estáticos:** Habilitar
- Tipo de alojamiento:** Alojamiento de sitios web estáticos. Subtext: "Utilice el punto de enlace del bucket como dirección web."
- Redirigir las solicitudes de un objeto. Subtext: "Redirija las solicitudes a otro bucket o dominio."

A blue information box at the bottom states: "Para que sus clientes puedan obtener acceso al contenido en el punto de enlace del sitio web, debe hacer que todo el contenido sea legible públicamente. Para ello, puede editar la configuración Bloquear acceso".

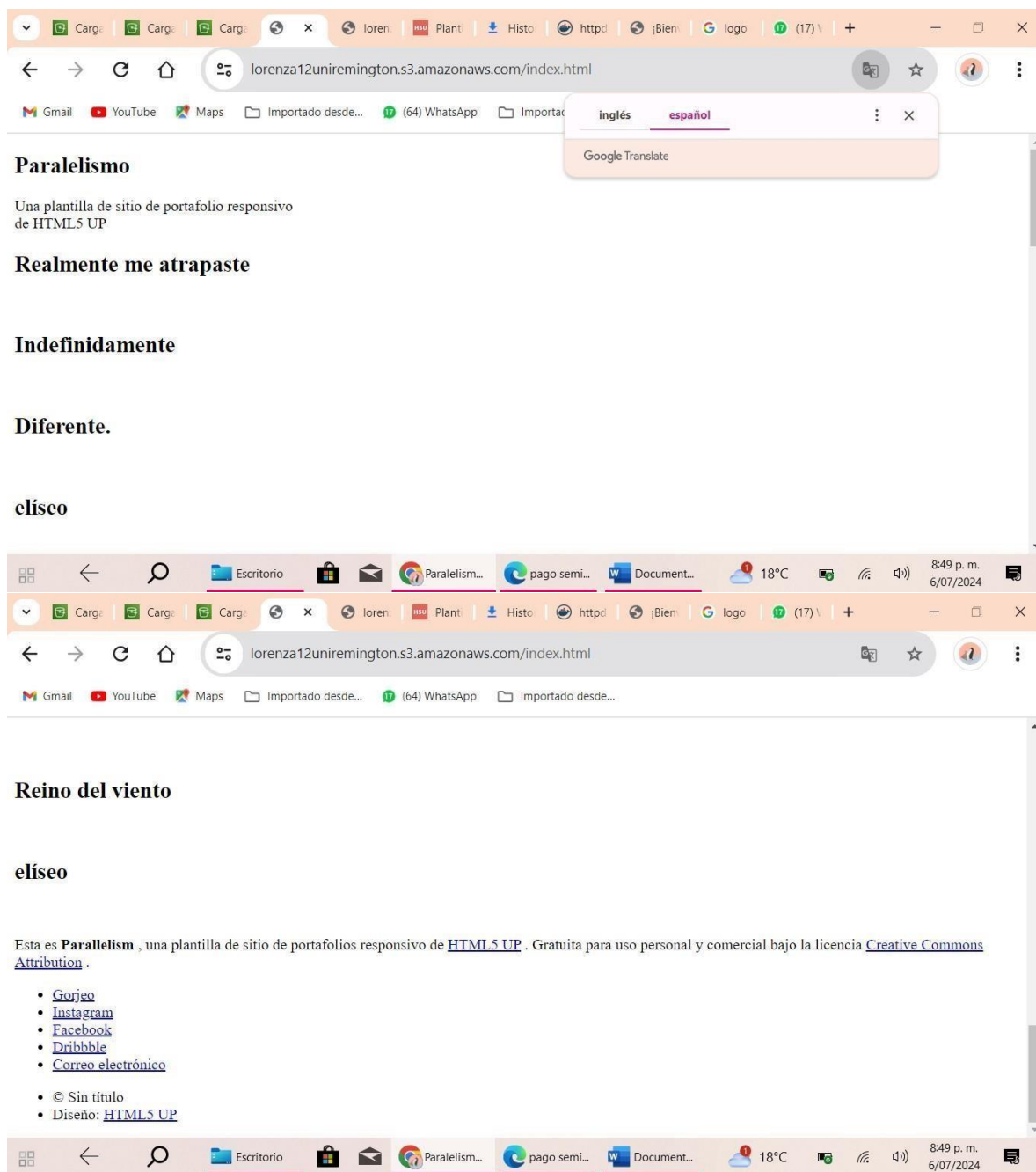
Bottom Screenshot: Información general sobre el bucket

The page title is "lorenza12uniremington". A green notification banner at the top says "Se editó correctamente el alojamiento de sitios web estáticos." The "Propiedades" tab is selected. The "Información general sobre el bucket" section displays the following details:

Región de AWS	Nombre de recurso de Amazon (ARN)	Fecha de creación
EE. UU. Este (Norte de Virginia) us-east-1	arn:aws:s3::lorenza12uniremington	6 Jul 2024 7:44:19 PM -05

The "Control de versiones de buckets" section is visible at the bottom, with an "Editar" button.

Después de ver cargado el archivo html con sus respectivas carpetas y dar permisos al sitio web se observa en el navegador la plantilla de html **Figura 47** *plantilla en el navegador html implementada*



Ventajas

Agilidad y flexibilidad mediante la implementación de controles de seguridad mejorados y la capacidad de actualizar una corriente de trabajo completa.

Eficiencia y optimización de recursos, ya que al crear más contenedores e implementar un balanceo de cargas el uso de recursos se distribuye en los contenedores de forma equitativa.

Portabilidad. Uno de los beneficios que ofrecen los contenedores es que se pueden ejecutar en cualquier, esto es aprovechado por los desarrolladores por que pueden desplegarlos en Linux, Windows, Mac, servidores físicos, máquinas virtuales, la nube, etc.

Protección contra fallos, como los contenedores de un sistema trabajan de forma independiente, el fallo de uno no afectaría el funcionamiento de los demás.

Desventajas

Fragilidad de hosts compartidos, ya que estos comparten el mismo sistema de hardware por debajo de la capa del sistema operativo, una vulnerabilidad que salga de algunos de estos podría afectar el hardware compartido.

Riesgos de seguridad, como los contenedores comparten el ingreso a un solo sistema operativo de hospedaje. Se corre el riesgo de que si se ejecuta un contenedor en un aislamiento incompleto podría ocasionar que un código malicioso acceda a la memoria del computador.

Inconvenientes con la recolección de datos, como cada contenedor recopila información sobre las operaciones, esta permanece en el contenedor a menos que se recoja y centralice.

Ventajas

Durabilidad, permite guardar archivos con un 99,999999999% de permanencia, haciendo que sea robusto para prevenir la pérdida y corrupción de la información.

Bajos costos, ya que se paga por lo que se usa y poco a poco se puede ir ampliando la aplicación como se desee.

Es muy flexible en cuanto al tipo y cantidad de almacenamiento de datos que se desea usar.

Seguridad, como este admite la autenticación de usuario, para controlar el acceso a los datos garantiza privacidad y seguridad de toda la información.

Desventajas

No hay almacenamiento a nivel de bloque disponible.

Es posible que sea obligatorio realizar cambios en la aplicación para aceptar el almacenamiento de Amazon S3.

Teniendo en cuenta la práctica que se hizo con cada una de estas alternativas, sus ventajas y desventajas. Pudimos observar que cada una de estas ofrece muy buenos beneficios en cuanto a la aplicación que utilizamos de ejemplo, siendo este un sitio que simplemente ofrece información la mejor opción para implementarlo es mediante un sitio estático en S3.

¿Por qué en S3?

Es una opción de bajo coste siendo algo muy importante para un sitio pequeño y además nos permite paso a paso con el crecimiento de la misma, ya que no tiene límite de tamaño, los datos estarán casi siempre disponibles y protegidos para los usuarios, es fácil de usar, siguiendo una serie de sencillos pasos en 20 minutos puedes implementar tu sitio en S3 algo muy esencial cuando se trata de un sitio pequeño, además nos concede acceso a la misma infraestructura de alojamiento de datos, económica, rápida y de alta escalabilidad que implementa Amazon para realizar su propia red global de sitios web

Arquitectura de AWS

Implementar una arquitectura de AWS con balanceador de carga de aplicación para al menos dos instancias y que estas a su vez, tengan al interior balanceador de carga de contenedores (puede ser el mismo que se solicitó en la entrega 2).

Creación en AWS La instancia o usamos con la que trabajamos en la entrega 2 **Figura**

48

AWS

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

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
Server01	i-009070ab3fe88b741	Running	t2.micro	2/2 checks passed	View alarms	us-east-2

The left sidebar contains a navigation menu with the following items:

- EC2 Dashboard
- EC2 Global View
- Events
- Instances
 - Instances
 - Instance Types
 - Launch Templates
 - Spot Requests
 - Savings Plans
 - Reserved Instances
 - Dedicated Hosts
 - Capacity Reservations
- Images
 - AMIs
 - AMI Catalog
- Elastic Block Store
 - Volumes
 - Snapshots
 - Lifecycle Manager
- Network & Security
 - Security Groups
 - Elastic IPs
 - Placement Groups
 - Key Pairs
 - Network Interfaces
- Load Balancing
 - Load Balancers
 - Target Groups
 - Trust Stores *New*
- Auto Scaling
 - Auto Scaling Groups
- Settings

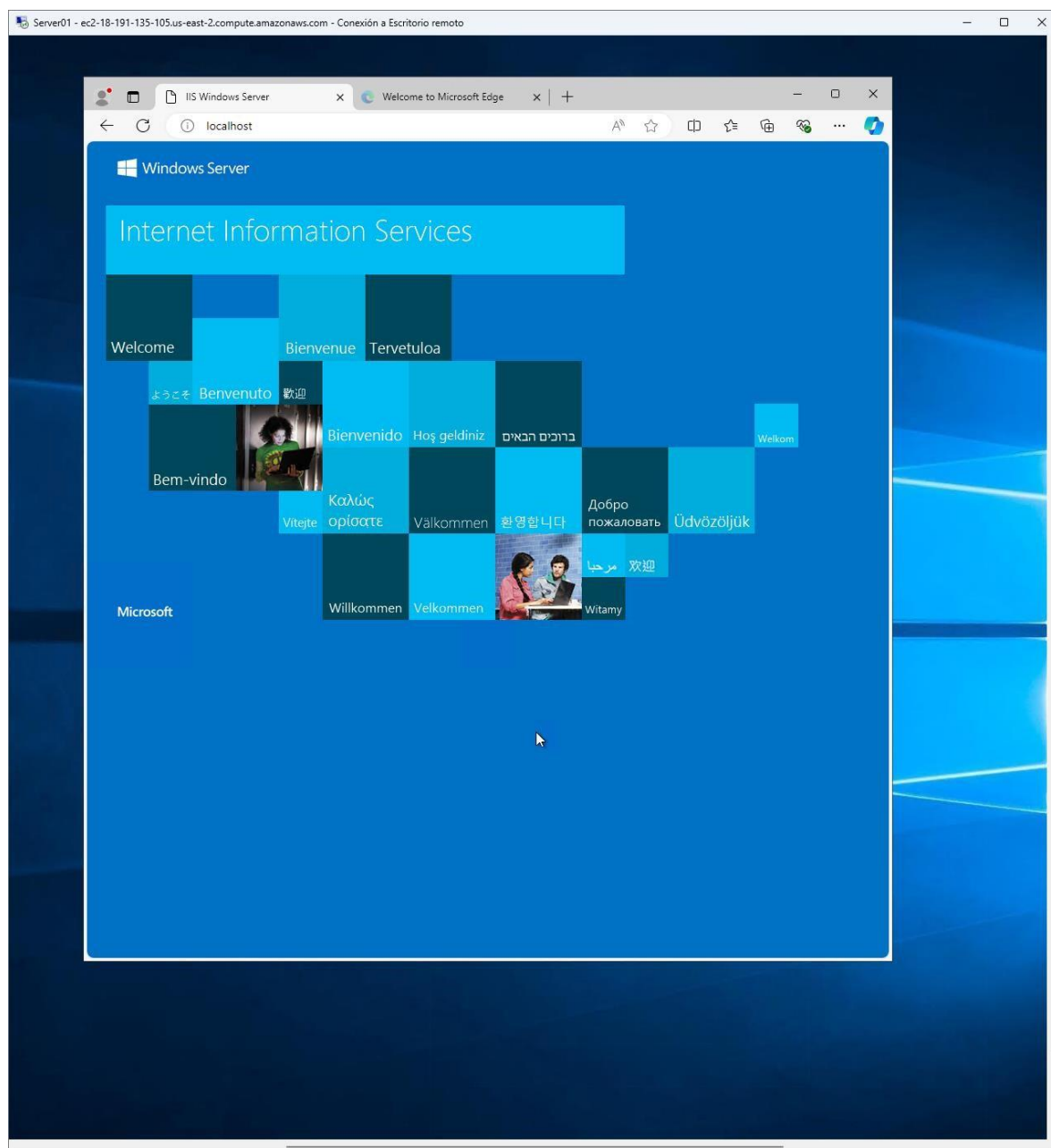
The footer of the page contains the following text:

https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#LoadBalancers © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Una vez configurada la instancia con Windows server 2022 y el servicio IIS activo

Figura 49

Windows server



Procedemos a crear una imagen de la instancia

Figura 50

Imagen instancia

The screenshot shows the AWS Management Console interface for creating an image from an existing EC2 instance. The browser tabs include 'Create application load balancer', 'Create Image | EC2 | us-east-2', and 'Home | EC2 | us-east-2'. The URL is <https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#CreateImage:instanceId=i-009070ab3fe88b741>. The breadcrumb navigation is 'EC2 > Instances > i-009070ab3fe88b741 > Create image'.

Create image [Info](#)

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.

Instance ID
i-009070ab3fe88b741 (Server01)

Image name

Maximum 127 characters. Can't be modified after creation.

Image description - optional

Maximum 255 characters

No reboot
 Enable

Instance volumes

Storage type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/...	Create new sna...	30	EBS General Pu...	100		<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Tag image and snapshots together
Tag the image and the snapshots with the same tag.

Tag image and snapshots separately
Tag the image and the snapshots with different tags.

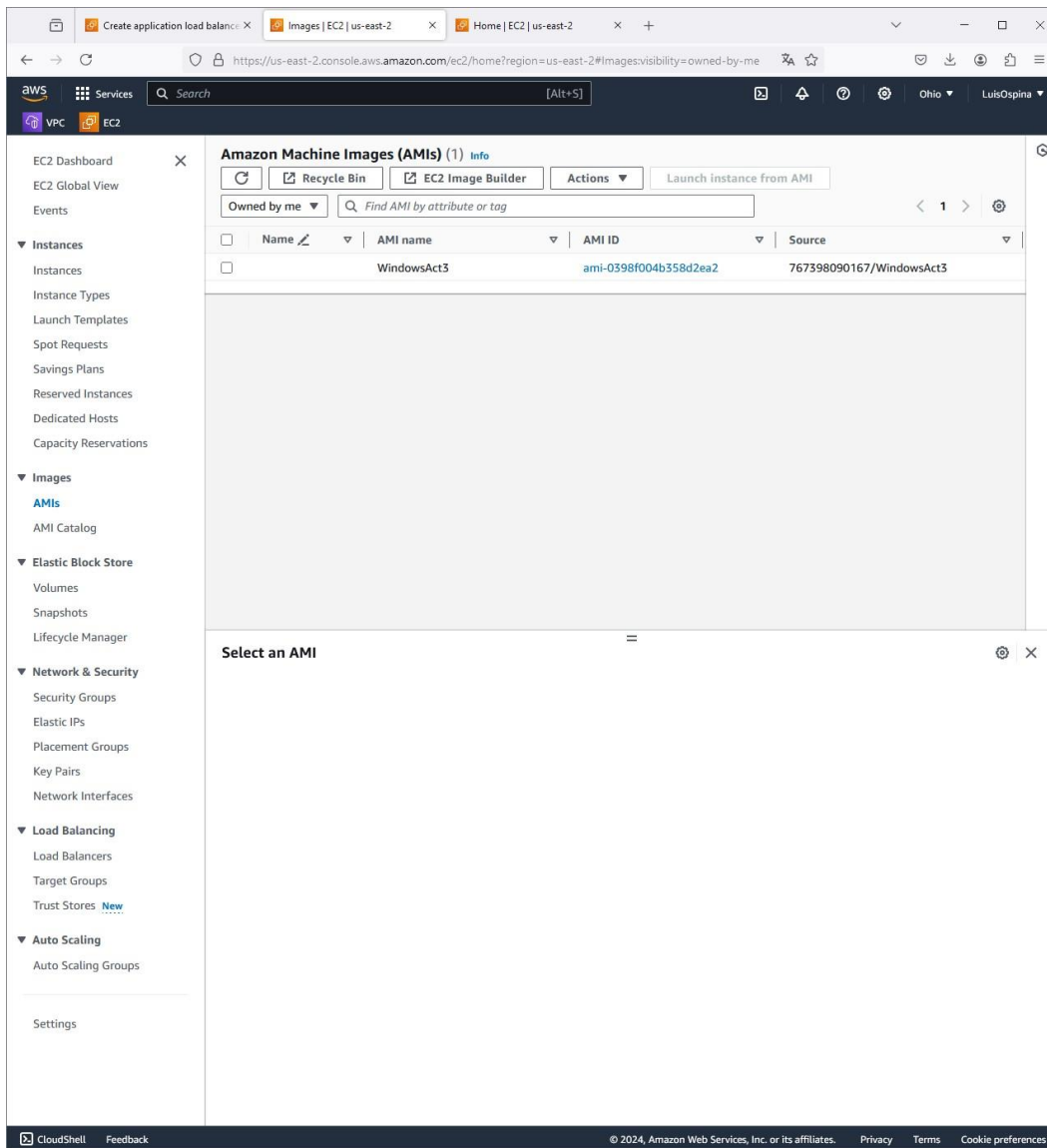
No tags associated with the resource.

You can add up to 50 more tags.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Figura 51

Instancia aws



Procedemos a crear la segunda instancia conforme a la imagen creada.

Figura 52

Segunda instancia

The screenshot displays the AWS Management Console interface for launching an EC2 instance. The main content area is titled "Launch an instance" and includes a "Name and tags" section with a text input field containing "Server02". Below this is the "Application and OS Images (Amazon Machine Image)" section, which features a search bar and a list of AMIs. The selected AMI is "WindowsAct3" (ami-0398f004b358d2ea2). The "Instance type" section is partially visible at the bottom. On the right side, a "Summary" panel provides a quick overview of the configuration: 1 instance, WindowsAct3 software image, t2.micro virtual server type, New security group firewall, and 1 volume(s) - 30 GiB storage. A "Free tier" notification is shown, detailing the benefits of the first year of usage. At the bottom right, there are "Cancel" and "Launch instance" buttons, with "Launch instance" being the primary action.

Una vez creada y corriendo las 2 instancias procedemos a crear el registro el balanceador de carga y sus configuraciones.

Figura 53

Registro balanceador

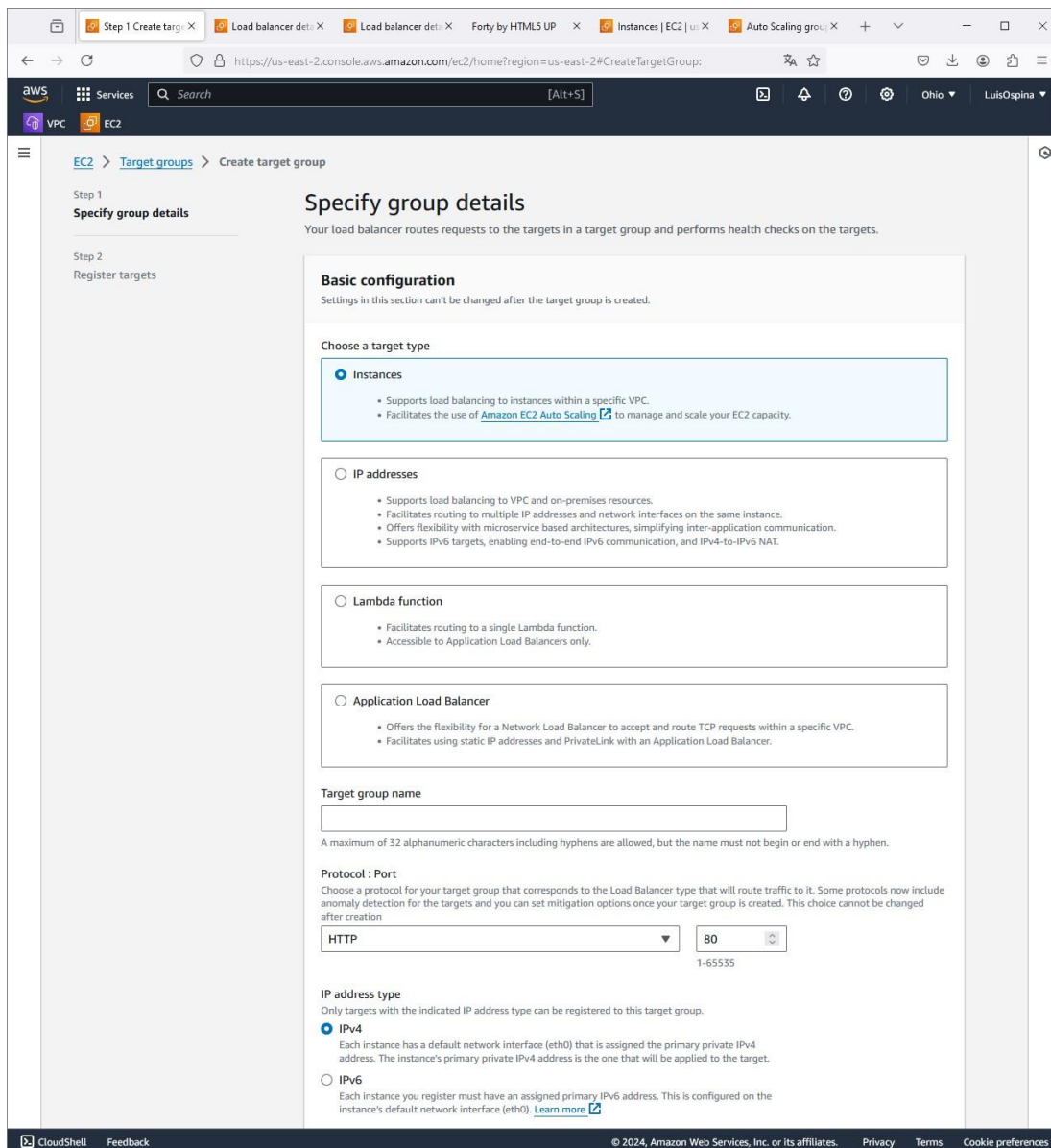


Figura 54

Target

The screenshot shows the AWS Management Console interface for Target Groups. The left sidebar contains navigation options for EC2, including Dashboard, Global View, Events, Instances, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main content area is titled 'EC2 > Target groups' and features a 'Create target group' button and a search filter. A table lists the existing target groups:

<input type="checkbox"/>	Name	ARN	Port	Protocol	Target type
<input type="checkbox"/>	Act3	arn:aws:elasticloadbalanci...	80	HTTP	Instance

Below the table, a message indicates that no target groups are selected:

0 target groups selected
Select a target group above.

The footer of the console shows 'CloudShell Feedback' and '© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

Figura 55

ACT3

Successfully created the target group: **Act3**. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the **Targets** tab.

EC2 > Target groups > Act3

Act3

Actions

Details

arn:aws:elasticloadbalancing:us-east-2:767398090167:targetgroup/Act3/a8a1e9a570386372

Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC vpc-00c2eba354222edfb
IP address type IPv4	Load balancer None associated		

2 Total targets	0 Healthy	0 Unhealthy	2 Unused	0 Initial	0 Draining
--------------------	--------------	----------------	-------------	--------------	---------------

0 Anomalous

Distribution of targets by Availability Zone (AZ)

Select values in this table to see corresponding filters applied to the Registered targets table below.

Targets | Monitoring | Health checks | Attributes | Tags

Registered targets (2) Info [Anomaly mitigation: Not applicable](#) [Deregister](#) [Register targets](#)

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

Filter targets

<input type="checkbox"/>	Instance ID	Name	Port	Zone	Health status	Health status
<input type="checkbox"/>	i-0adc33c9d265093b4	Server02	80	us-east-2a	Unused	Target group
<input type="checkbox"/>	i-009070ab3fe88b741	Server01	80	us-east-2a	Unused	Target group

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Una vez creado el balanceador de carga y sus respectivas configuraciones y funcionamiento, procedemos a crear la auto escala

Figura 56

auto escala

The screenshot displays the AWS Management Console interface for creating an Auto Scaling group. The breadcrumb navigation shows the path: EC2 > Auto Scaling groups > Create Auto Scaling group. The current step is Step 1: Choose launch template. The main heading is 'Choose launch template' with an 'Info' icon. Below the heading, a sub-heading reads 'Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.'

The 'Auto Scaling group name' field is filled with 'PantillaAct3'. A warning message is displayed in a light blue box: 'For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.'

The 'Launch template' dropdown menu is open, showing a search bar and a list of templates. The template 'Pantilla3' is selected and highlighted. The 'Next' button is visible at the bottom right of the form.

Figura 57

configure group

The screenshot shows the AWS Management Console interface for creating an Auto Scaling group. The page is titled "Configure group size and scaling - optional" and is part of a 7-step process. The current step, Step 4, is highlighted in the left-hand navigation pane.

Group size Info
Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type
Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.
Units (number of instances)

Desired capacity
Specify your group size.
2

Scaling Info
You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits
Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity	Max desired capacity
1	4
Equal or less than desired capacity	Equal or greater than desired capacity

Automatic scaling - optional
Choose whether to use a target tracking policy Info
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

- No scaling policies**
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.
- Target tracking scaling policy**
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

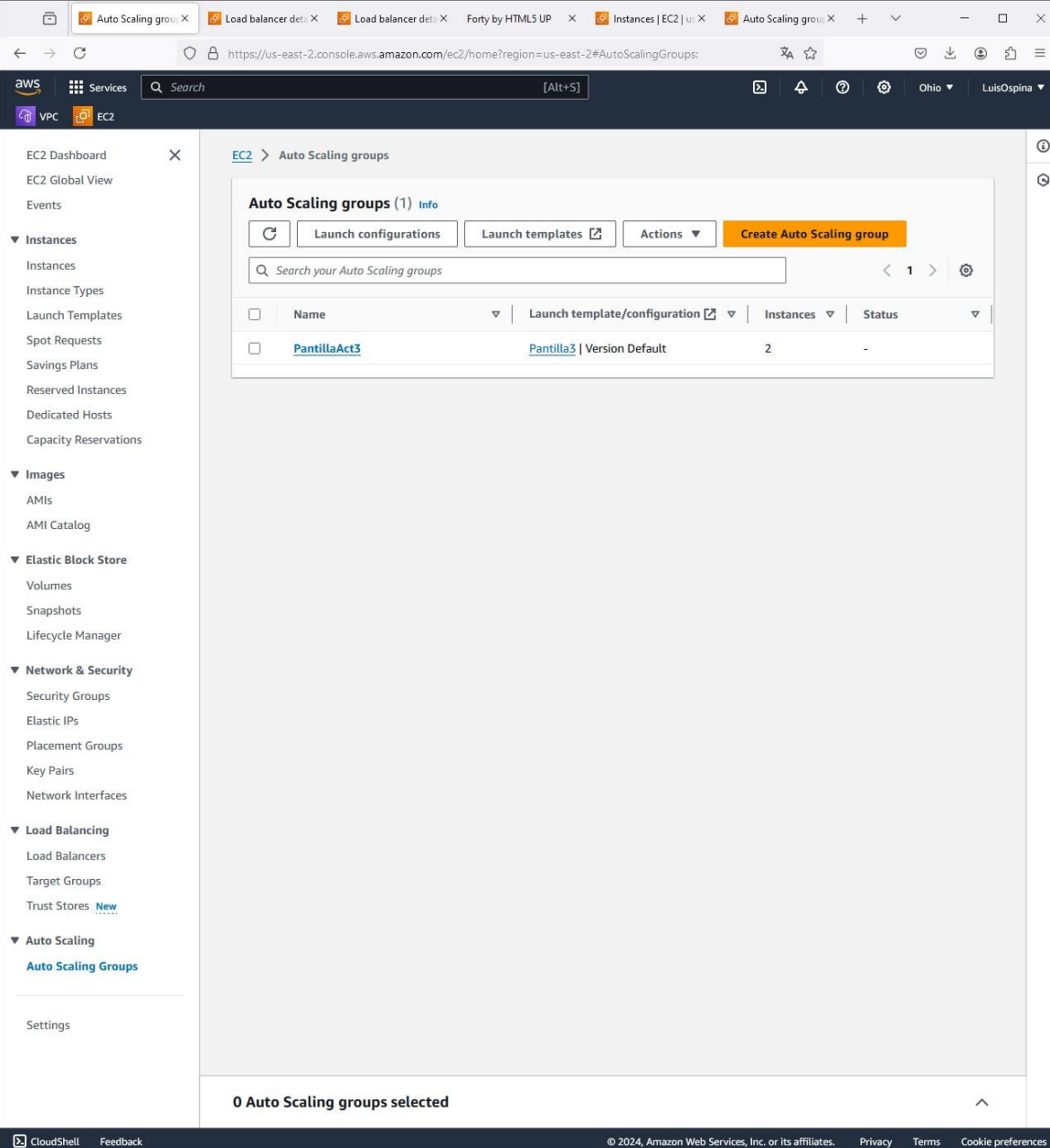
Instance maintenance policy Info
Control your Auto Scaling group's availability during instance replacement events. This includes health checks, instance refreshes, maximum instance lifetime features and events that happen automatically to keep your group balanced, called rebalancing events.

Choose a replacement behavior depending on your availability requirements

Mixed behavior	Prioritize availability	Control costs	Flexible
<input checked="" type="radio"/> No policy For rebalancing events, new instances will be launched before terminating existing instances.	<input type="radio"/> Launch before terminating Launch new instances before terminating existing instances.	<input type="radio"/> Terminate and launch Terminate existing instances before launching new instances.	<input type="radio"/> Custom behavior Set custom values for the minimum and maximum number of instances.

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Figura 58

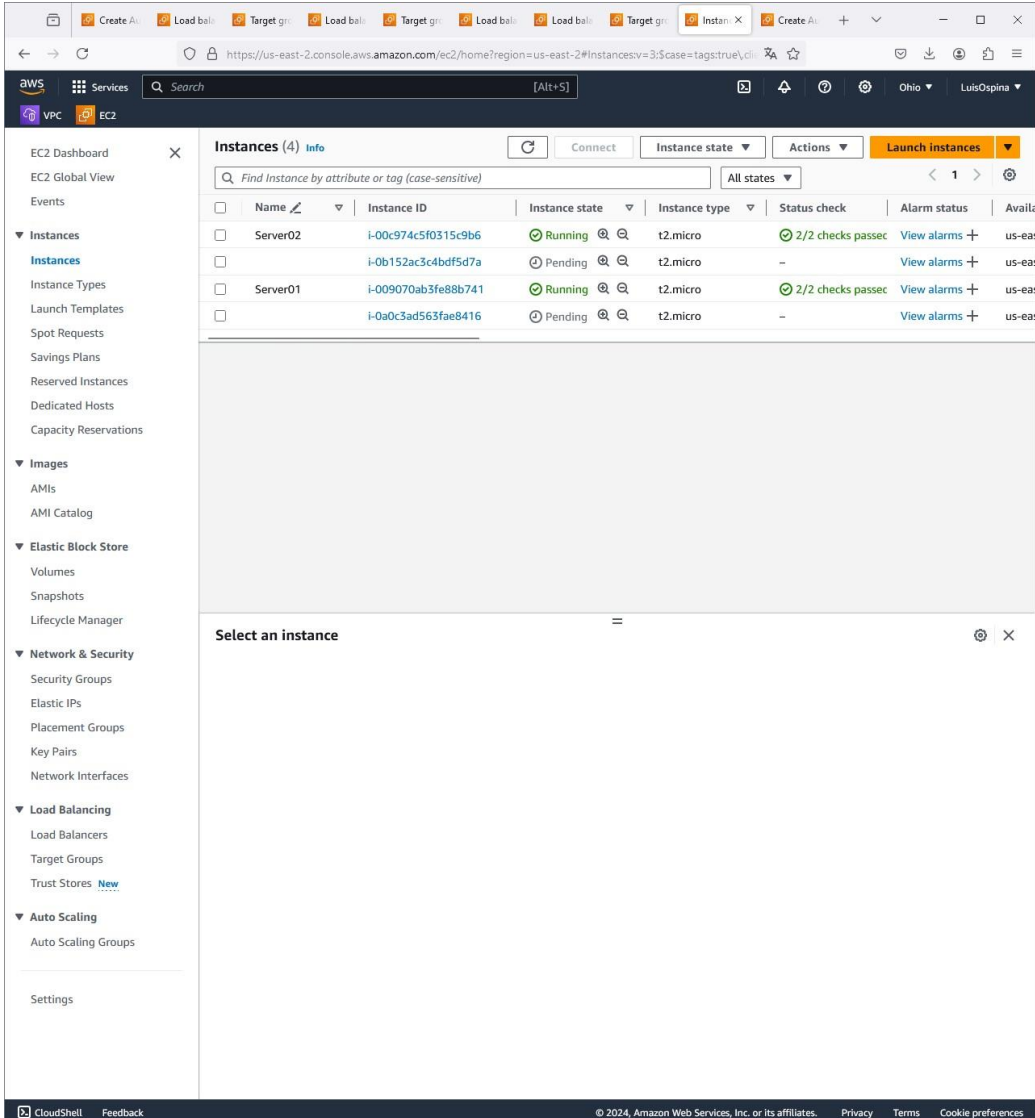
auto scaling

The screenshot shows the AWS Management Console interface for Auto Scaling groups. The browser address bar indicates the URL: `https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#AutoScalingGroups:`. The console header includes the AWS logo, a search bar, and navigation icons. The left sidebar contains a navigation menu with categories: EC2 Dashboard, EC2 Global View, Events, Instances (with sub-items: Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (with sub-items: AMIs, AMI Catalog), Elastic Block Store (with sub-items: Volumes, Snapshots, Lifecycle Manager), Network & Security (with sub-items: Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (with sub-items: Load Balancers, Target Groups, Trust Stores), and Auto Scaling (with sub-item: Auto Scaling Groups). The main content area is titled "Auto Scaling groups (1) info" and features a search bar, a table, and a "Create Auto Scaling group" button. The table lists one group: "PantillaAct3" with a launch template of "Pantilla3 | Version Default", 2 instances, and a status of "-". At the bottom of the main content area, it states "0 Auto Scaling groups selected". The footer includes "CloudShell", "Feedback", and copyright information: "© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

Una vez creado el auto escalado y comprobado que se creen las instancias automáticamente en caso de que una falle y suba el servicio el IIS, haciendo el llamado desde el nombre dns del balanceador “BalanceadorAct3-629517115.us-east-2.elb.amazonaws.com” damos por completada la actividad.

Figura 59 dns

balanceador



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input type="checkbox"/>	Server02	i-00c974c5f0315c9b6	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2
<input type="checkbox"/>		i-0b152ac3c4bdf5d7a	Pending	t2.micro	-	View alarms +	us-east-2
<input type="checkbox"/>	Server01	i-009070ab3fe88b741	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2
<input type="checkbox"/>		i-0a0c3ad563fae8416	Pending	t2.micro	-	View alarms +	us-east-2

Select an instance

Figura 60 selected

The screenshot displays the AWS Management Console interface for EC2 instances. At the top, a green notification banner states "Successfully initiated stopping of i-009070ab3fe88b741, i-00c974c5f0315c9b6". Below this, the "Instances (2/4)" section is visible, featuring a table with columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability zone. Two instances are selected: "Server02" (ID: i-00c974c5f0315c9b6) and "Server01" (ID: i-009070ab3fe88b741), both in a "Stopping" state. The remaining two instances are in a "Running" state. Below the table, a "Monitoring" section is active, showing "2 instances selected" and "Alarm recommendations" (disabled). The monitoring dashboards include: CPU utilization (%), Network in (bytes), Network out (bytes), Network packets in (count), Network packets out (count), CPU credit usage (count), and CPU credit balance (count). Each dashboard shows a line graph with a peak around 01:20. The footer of the console includes "CloudShell", "Feedback", and copyright information for Amazon Web Services, Inc. or its affiliates.

Figura 61

selected instance

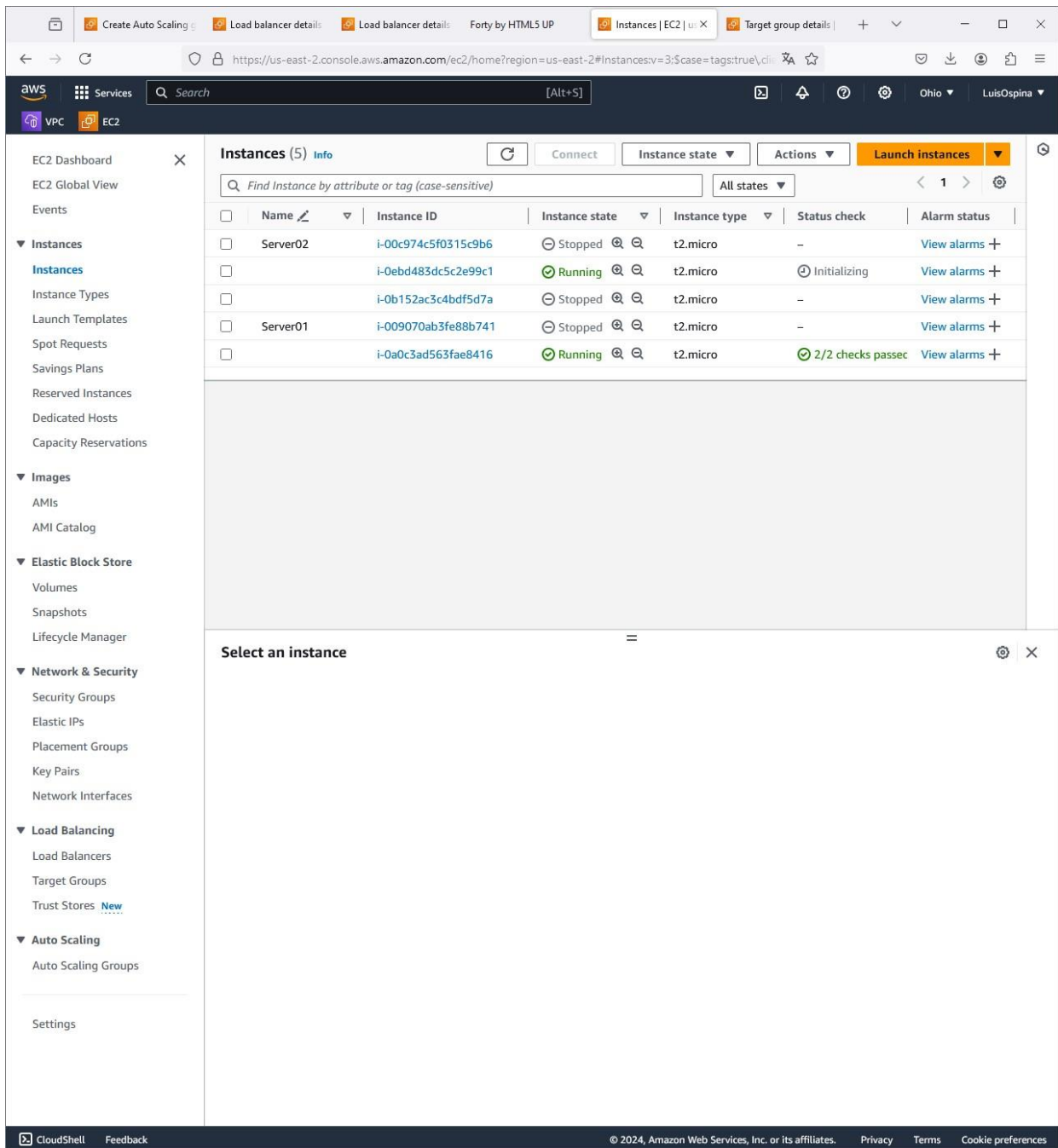


Figura
plantilla Act3

The screenshot shows the AWS Management Console interface for an Auto Scaling Group named 'PantillaAct3'. The 'Activity' tab is selected, showing a summary of activity notifications and a detailed activity history table.

Activity notifications (0)

Filter notifications:

Send to: **On instance action**

No notifications are currently specified

Activity history (2)

Status	Description	Cause
Successful	Launching a new EC2 instance: i-0b152ac3c4bd5d7a	At 2024-07-18T01:22:14Z a user request created an AutoScalingGroup changing the desired from 0 to 2. At 2024-07-18T01:22:27Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.
Successful	Launching a new EC2 instance: i-0a0c3ad563fae8416	At 2024-07-18T01:22:14Z a user request created an AutoScalingGroup changing the desired from 0 to 2. At 2024-07-18T01:22:27Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.

Figura
Detalles act3

The screenshot shows the AWS Management Console interface for an Amazon EC2 Target Group named 'Act3'. The breadcrumb navigation is 'EC2 > Target groups > Act3'. The main content area displays the following details:

- Details:** Target type: Instance; Protocol: Port: HTTP: 80; Protocol version: HTTP1; VPC: vpc-00c2eba35422edfb.
- IP address type:** Load balancer: BalanceadorAct3; IP address type: IPv4.
- Target Status Summary:** 5 Total targets, 2 Healthy, 0 Unhealthy, 2 Unused, 0 Initial, 1 Draining, 0 Anomalous.
- Distribution of targets by Availability Zone (AZ):** A section with a note: 'Select values in this table to see corresponding filters applied to the Registered targets table below.'
- Registered targets (5):** A table with columns: Instance ID, Name, Port, Zone, Health status, Health state.

Instance ID	Name	Port	Zone	Health status	Health state
i-0ebd483dc5c2e99c1		80	us-east-2b	Healthy	-
i-0a0c3ad563fae8416		80	us-east-2a	Healthy	-
i-0b152ac3c4bdf5d7a		80	us-east-2b	Draining	Target der...
i-00c974c5f0315c9b6	Server02	80	us-east-2b	Unused	Target is i...
i-009070ab3fe88b741	Server01	80	us-east-2a	Unused	Target is i...

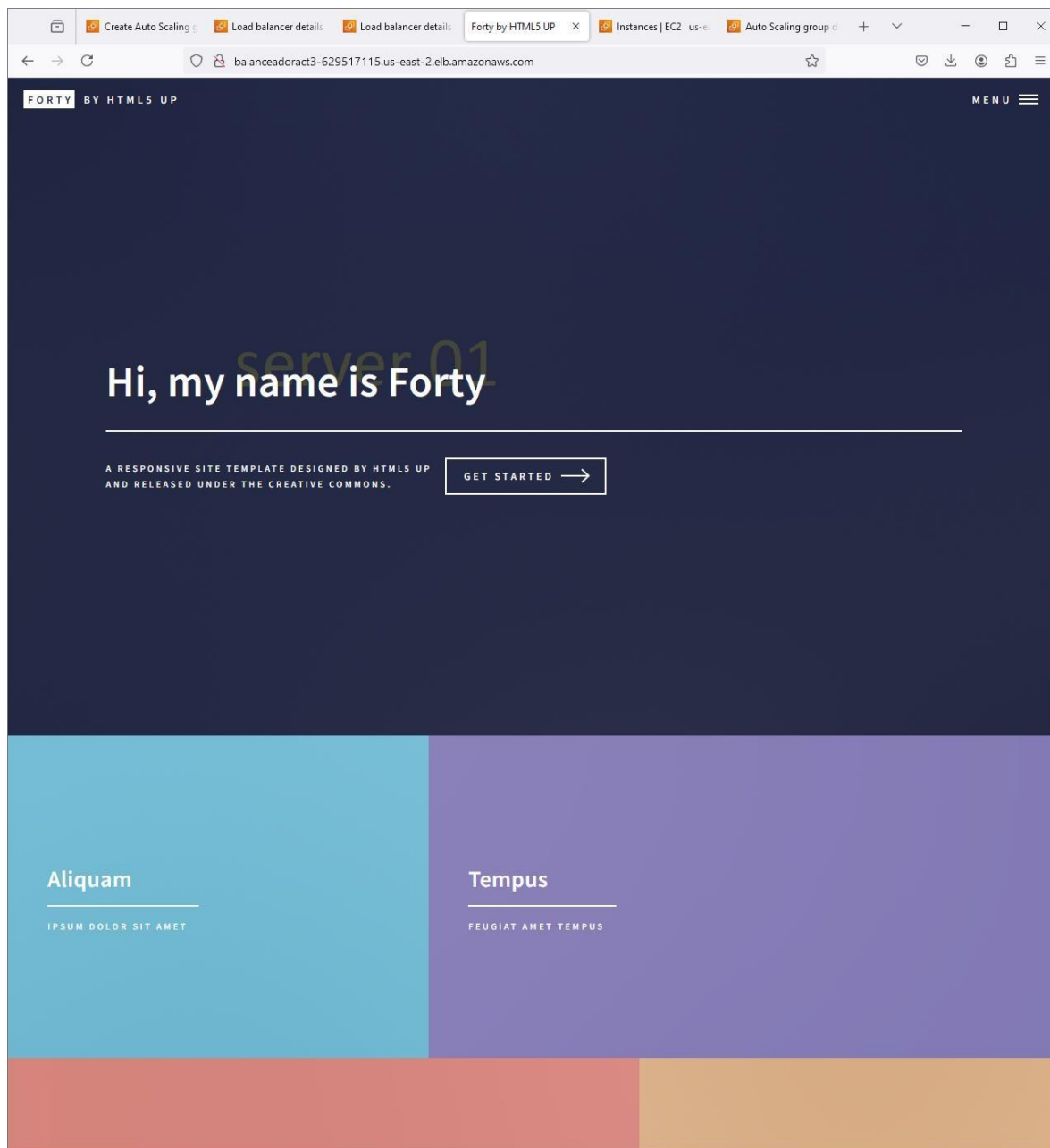
Figura
Activity act3

The screenshot displays the AWS Management Console interface for the Auto Scaling group 'PantillaAct3'. The left sidebar shows navigation options for EC2, including Dashboard, Global View, Events, and various instance and launch options. The main content area is titled 'PantillaAct3' and includes tabs for Details, Activity, Automatic scaling, Instance management, Monitoring, and Instance refresh. The 'Activity' tab is active, showing 'Activity notifications (0)' and 'Activity history (4)'. The activity history table contains four entries:

Status	Description	Cause
Successful	Launching a new EC2 instance: i-0ebd483dc5c2e99c1	At 2024-07-18T01:30:35Z an instance was launched in response to an unhealthy instance replaced.
Connection draining in progress	Terminating EC2 instance: i-0b152ac3c4bdf5d7a - Waiting For ELB Connection Draining.	At 2024-07-18T01:30:34Z an instance was taken out of service in response to an EC2 health indicating it has been terminated or stopped.
Successful	Launching a new EC2 instance: i-0b152ac3c4bdf5d7a	At 2024-07-18T01:22:14Z a user request created an AutoScalingGroup changing the desired from 0 to 2. At 2024-07-18T01:22:27Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.
Successful	Launching a new EC2 instance: i-0a0c3ad563fae8416	At 2024-07-18T01:22:14Z a user request created an AutoScalingGroup changing the desired from 0 to 2. At 2024-07-18T01:22:27Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.

The footer of the console shows 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates, along with links for Privacy, Terms, and Cookie preferences.

Instalancia en servicio nuevamente.



Conclusiones

El seminario de AWS nos proporcionó una comprensión muy clara sobre cómo la nube de Amazon permite beneficiarse de las economías de escala masivas, deja de adivinar la capacidad aumenta la velocidad y la agilidad, nos enseñó lo que es la computación en la nube que solamente con un pago tenemos el servicio en la nube. Nube también son servidores mucho más ideales que tenerlos de forma local en la empresa. Son datacenters especializados que tienen todas las garantías de seguridad de red.

Sin embargo, la infraestructura global de AWS se compone de regiones, es la forma como se divide los servidores de AWS, si se crean servicios en otra región y saber jugar con esas regiones es lo que va a permitir en algún momento tener una alta disponibilidad. La zona de disponibilidad básicamente es un datacenter que tiene las condiciones ideales y brinda la opción de elegir en cual zona de disponibilidad quiere instalar los servicios.

Finalmente se aprendió de los servicios principales de AWS son los servicios administrativos donde el usuario se olvida del hardware, son los de más bajo nivel pero el usuario si tiene acceso al sistema operativo y aplica parches de seguridad, están los servicios completamente administrativos AWS se encarga de los servicios parches de seguridad, servicios con tecnologías en este se olvida completamente del hardware y AWS se encarga de esa parte y ahí es donde cobra factura.

Referencias

Bastidas, L.R. (2007). *El inicio del siglo XXI*. Planeta. Sitio web:

<http://www.rbastidasl.com/libro-inicio-del-sigloxxi>.

Confirmación del registro. (s. f.-a). Amazon Web Services, Inc.

<https://aws.amazon.com/es/registration-confirmation/>

Equipo editorial de IONOS. (2023, 1 marzo). ¿Qué es un servidor? IONOS Digital Guide.

<https://www.ionos.com/es-us/digitalguide/servidores/know-how/que-es-un-servidor-unconcepto-dos-definiciones/>

Servidores Cloud VPS Dongee, mayor uptime y rendimiento - Dongee. (2024, 5 febrero).

Dongee.

https://www.dongee.com/hosting/cloud/colombia/?utm_source=google&utm_medium=pc-keyword&utm_campaign=vps&as_clid=a64c4acf-23a9-476c-8a57-92f2ba4f983b:servidores%20de%20cloud%20computing:e:g.&tm=tt&ap=gads&aaid=adaFnZvkSdpvp&gad_source=1&gclid=CjwKCAjw1920BhA3EiwAJT3lSYK_xwZCU7sR4dWrnb75bNELR6WtWLinXKIIC42dNjNIHVWXiP-FJBoC8_0QAvD_BwE

Equipo editorial de IONOS. (2023b, marzo 1). ¿Qué es un servidor? IONOS Digital Guide.

<https://www.ionos.com/es-us/digitalguide/servidores/know-how/que-es-un-servidor-unconcepto-dos-definiciones/>

Usar Amazon S3 con Amazon EC2 - Amazon Elastic Compute Cloud. (s. f.).

https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/AmazonS3.html