TRABAJO DE GRADO
Opción Seminario-Diplomado.

Implementación de servicios básicos en la nube con AWS: instancias EC2, balanceador de carga y contenedores Docker

Corporación Universitaria Remington.
Facultad de Ingenierías
Ingeniería de Sistemas

Charly Hernando Avendaño Roa
Sergio Omar Rueda Espinosa

Juan Pablo Berrio López (docente del seminario o diplomado).
Opción de Trabajo de grado Seminario-Diplomado.
2025.

Dedicatoria

Con profunda gratitud, dedico este trabajo a Dios, fuente de sabiduría y fortaleza, por guiarme en cada paso de este proyecto. A Su presencia le atribuyo la paciencia y fe necesarias para superar los desafíos académicos, así como las bendiciones que han iluminado mi camino. También quiero dedicar este logro a mi familia, por su amor incondicional y su apoyo constante. En especial, agradezco a mis padres por su sacrificio, a mis hermanos por su ánimo y comprensión, y a todos mis seres queridos por motivarme a dar lo mejor de mí en todo momento.

Tabla de Contenidos

Resumen

El seminario de grado se centró en la introducción a los servicios básicos en la nube mediante AWS. Para ello, se creó una infraestructura en AWS con una Virtual Private Cloud (VPC) configurada con subredes y grupos de seguridad adecuados. Se desplegaron instancias EC2 con sistemas operativos Linux y Windows, las cuales sirvieron como servidores web y de aplicaciones que ofrecieron los servicios básicos necesarios. En esta infraestructura se configuró un balanceador de carga para distribuir el tráfico entrante y se asignaron direcciones IP públicas para garantizar la accesibilidad de los servidores. Finalmente, se validó el correcto funcionamiento de cada componente antes de avanzar a la siguiente etapa del proyecto.

Posteriormente, se instaló Docker en una de las instancias Linux para habilitar la ejecución de contenedores. Sobre esta plataforma, se desplegaron múltiples aplicaciones encapsuladas en contenedores que representaron diversos servicios web y de servidor. Esta metodología facilitó la escalabilidad y portabilidad de los servicios implementados. Se realizaron pruebas de funcionamiento para asegurar que cada aplicación respondiera correctamente, así como pruebas de estrés para evaluar la estabilidad y rendimiento de los servicios bajo condiciones de alta demanda. Los resultados obtenidos confirman la viabilidad de utilizar AWS y Docker como soluciones eficientes para desarrollar infraestructuras escalables, de alta disponibilidad y confiables en la nube.

Palabras claves

Servidores en la nube, AWS, Balanceador de carga, Docker, Contenedores

Marco conceptual y contextual

A continuación, se presentan los principales conceptos que sustentan el desarrollo del proyecto:

Computación en la Nube (Cloud Computing): Es un modelo para permitir, de manera conveniente, el acceso ubicuo a la red bajo demanda a un conjunto de recursos informáticos configurables (por ejemplo: redes, servidores, almacenamiento, aplicaciones y servicios) que puede ser aprovisionado y liberado rápidamente con un esfuerzo mínimo de gestión o interacción de un proveedor de servicios (Henriquez et al., 2015).

AWS (Amazon Web Services): Es una plataforma en la nube donde se ofrecen más de 175 servicios integrales de centros de datos a nivel mundial y se utiliza para reducir costos, aumentar la agilidad e innovar más rápido en las empresas. Esto permite que niños, jóvenes y adultos tengan la posibilidad de aprender a desarrollar programación sin importar el tipo de herramienta que quieran utilizar (Blanco & Vargas, 2023).

EC2: Es un servicio web de AWS que proporciona capacidad informática en la nube segura y de tamaño modificable. Básicamente es un servicio de despliegue de instancias de máquinas virtuales con opciones de diferentes sistemas operativos, potencias de proceso y cantidades de almacenamiento (Molina Daza & Tardio Lopez, 2018)

Balanceador de carga: El balanceo de carga es la forma en la que las peticiones de Internet se distribuyen en una fila de servidores (Brañes Vílchez, 2019).

Docker: es conocido como un contenedor ejecutable, independiente y ligero ya que

integra todo lo necesario para ejecutar una aplicación, esto incluye sus bibliotecas, herramientas del sistema, código y tiempo de ejecución (Lopez Cendejas, 2008).

Contenedores: Los servicios de contenedores o Containers as a Service(CaaS) son servicios gestionados en la nube que administran el ciclo de vida de los contenedores. Los servicios de contenedores ayudan a orquestar (iniciar, detener, ampliar) el tiempo de ejecución de los contenedores (Lopez Cendejas, 2008).

Este proyecto lo desarrollé como parte del seminario de grado de Ingeniería de Sistemas, con el propósito de llevar a la práctica muchos de los conocimientos adquiridos durante mi formación académica. Quise enfocarme en un entorno moderno y realista, por lo que decidí trabajar con servicios en la nube, específicamente con Amazon Web Services (AWS),

En la primera fase del proyecto, me dediqué a crear y configurar instancias EC2 con sistemas operativos Windows y Linux, dentro de una VPC (nube privada virtual) personalizada. Configuré direcciones IP públicas y apliqué reglas de seguridad mediante grupos de acceso, lo que me permitió habilitar la conexión externa a los servidores. También implementé un balanceador de carga, con el objetivo de distribuir de manera eficiente el tráfico entrante entre las instancias. Una vez completada esta configuración, realicé pruebas básicas de conectividad y funcionamiento para verificar que todo operara correctamente.

En la segunda etapa, instalé y configuré Docker en la instancia Linux. Sobre esta base, desplegué varias aplicaciones usando contenedores, una técnica que permite mejorar la Yo llevaría, escalabilidad y eficiencia del entorno. En total, creé tres contenedores ejecutando una misma aplicación, cada uno expuesto en los puertos 8080, 8082 y 8083. Esto me permitió simular un escenario real de balanceo de tráfico entre contenedores. Para finalizar, realicé

pruebas de funcionamiento y pruebas de estrés utilizando herramientas específicas, con el fin de medir el rendimiento del sistema bajo diferentes niveles de carga.

Esta experiencia me permitió reforzar mis habilidades en áreas como computación en la nube, redes virtuales, balanceo de carga y contenedores. Además, me ayudó a entender mejor cómo se diseñan e implementan soluciones modernas que hoy en día son fundamentales en el mundo de la tecnología.

Desarrollo e implementación del aprendizaje

Introducción

En la actualidad, la computación en la nube representa una solución poderosa para desplegar infraestructura tecnológica escalable, segura y económica. Amazon Web Services (AWS), líder en el sector, proporciona herramientas que permiten crear redes privadas virtuales, instancias de servidores, servicios de almacenamiento y mucho más con alta disponibilidad. Este informe documenta la implementación de una red en AWS que incluye dos instancias EC2 (una con Windows Server y otra con Linux Ubuntu), ambas accesibles desde Internet y con servidores web operativos. Este ejercicio académico busca afianzar competencias clave en infraestructura como servicio (IaaS), redes virtuales y administración de servidores.

Beneficios y utilidad de AWS

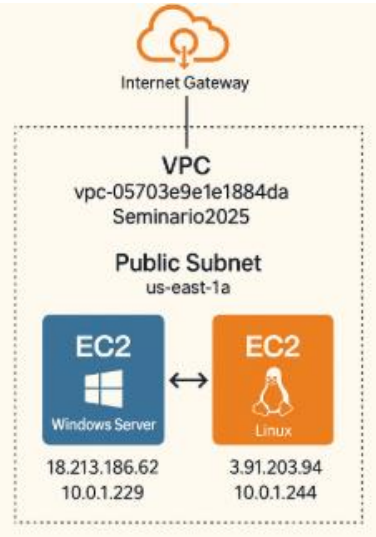
Amazon Web Services ofrece múltiples beneficios que lo posicionan como una de las plataformas preferidas para despliegues en la nube:

- **Escalabilidad:** Permite ajustar los recursos según la demanda de la aplicación.
- **Alta disponibilidad:** Sus zonas de disponibilidad reducen significativamente el riesgo de fallos.
- **Pago por uso:** Se factura solo lo que se consume, optimizando costos.
- **Seguridad:** AWS proporciona cifrado, autenticación y grupos de seguridad altamente configurables.
- **Automatización y agilidad:** Permite desplegar infraestructura en minutos, lo que acelera ciclos de desarrollo y pruebas.

En este proyecto, AWS facilitó la creación de una red aislada, con subredes públicas, acceso seguro y servidores configurados para funcionar como sitios web accesibles desde cualquier navegador.

Diagrama de arquitectura

Figura 1. Arquitectura de *infraestructura*



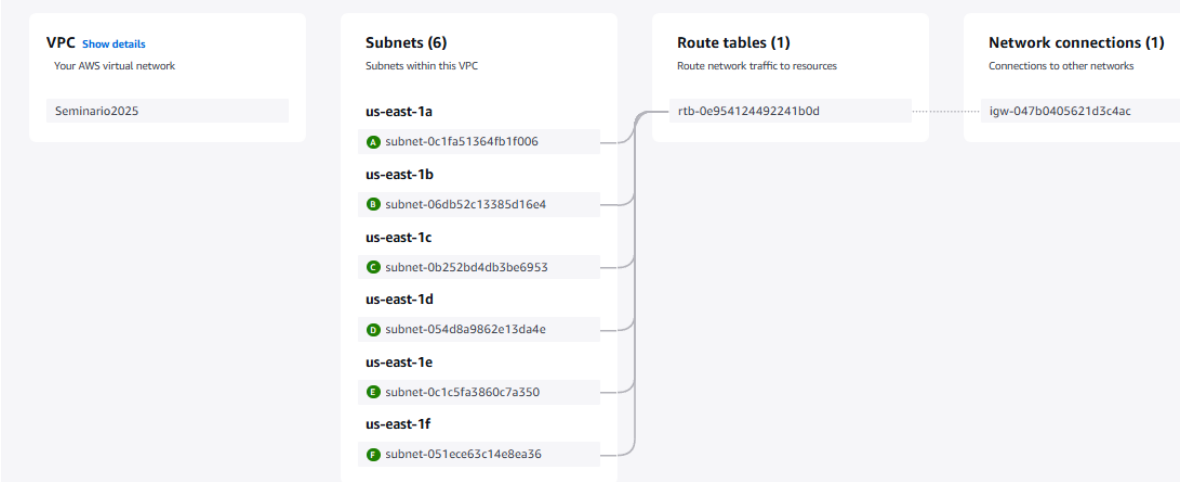
Descripción de la arquitectura

Se diseñó una red en la VPC Seminario2025, utilizando la subred us-east-1a. En ella se desplegaron dos instancias EC2:

- **Windows Server 2022**
- **Linux Server 2016**

Ambas instancias cuentan con una IP pública asignada para su accesibilidad externa y una IP privada para la comunicación interna. La VPC utiliza un Internet Gateway para conectar los recursos a la red global, y se configuró una única tabla de rutas para dirigir el tráfico saliente.

Figura 2. Subnets públicas y privadas



Configuraciones realizadas

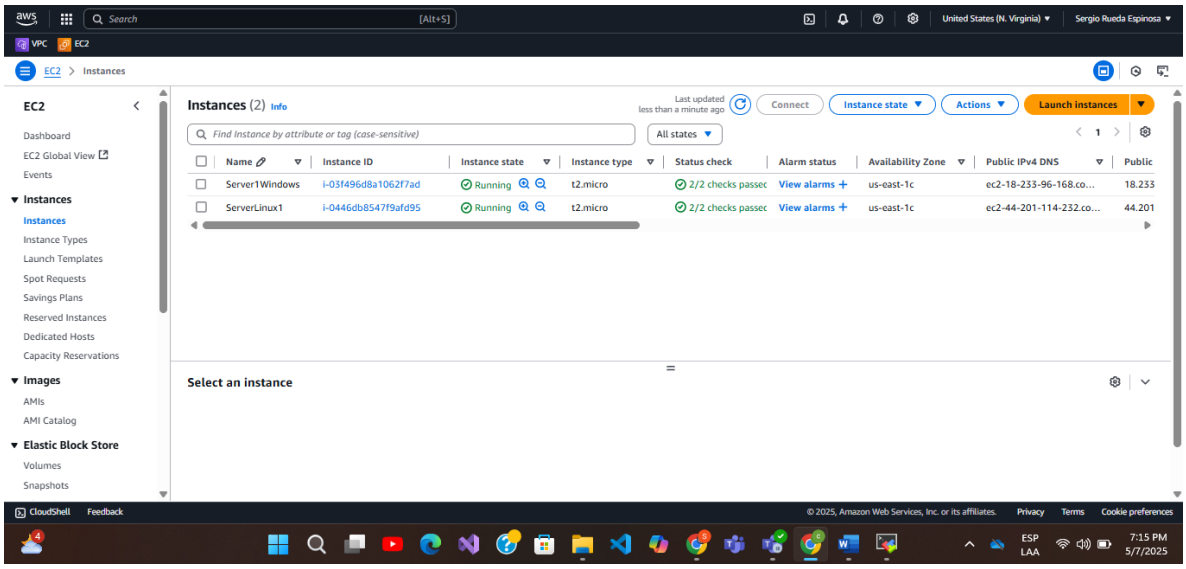
Pasos para crear las instancias EC2:

1. Se eligió la AMI correspondiente (Windows o Ubuntu).
2. Se seleccionó el tipo de instancia t2.micro.
3. Se configuró la red utilizando la subred pública us-east-1a.
4. Se habilitó la asignación automática de IP pública.
5. Se creó un par de llaves para la instancia Linux y se definió una contraseña segura para la instancia Windows.

Asignación de IPs:

- Windows:
  - IP Pública: 18.213.186.62
  - IP Privada: 10.0.1.229
- Linux:
  - IP Pública: 3.91.203.94
  - IP Privada: 10.0.1.244

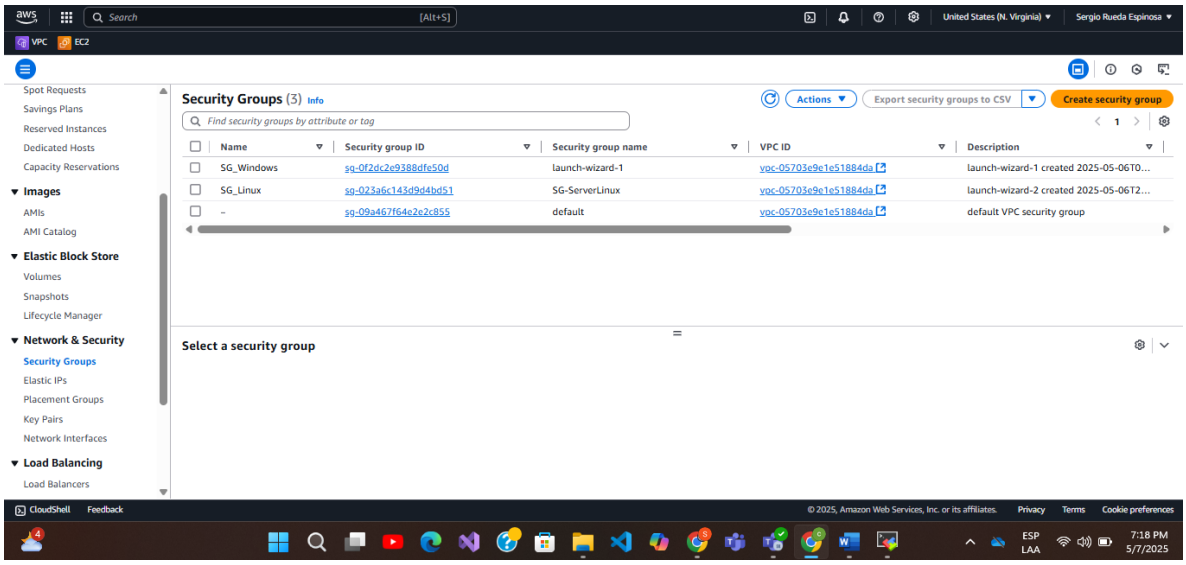
Figura 3. Creación de instancias windows y Linux



Grupos de Seguridad:

- **Instancia Windows:**
  - Puerto 3389 (RDP) habilitado desde la IP del usuario.
  - Puerto 80 (HTTP) abierto a todo Internet.
- **Instancia Linux:**
  - Puerto 22 (SSH) habilitado desde la IP del usuario.
  - Puerto 80 (HTTP) abierto a todo Internet.

Figura 4. Reglas de Seguridad de entradas



Procedimiento de acceso

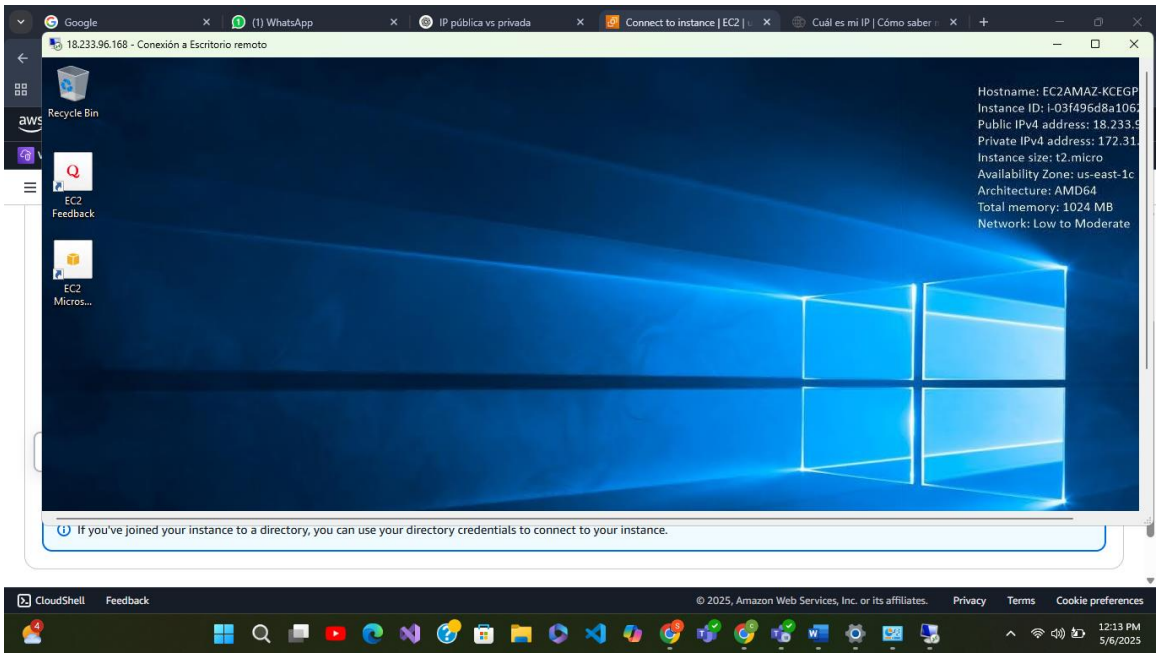
- **Windows:** Se accedió mediante Remote Desktop (RDP), utilizando la IP pública y las credenciales generadas al momento de crear la instancia.
- **Linux:** Se accedió mediante una terminal SSH con el archivo PEM proporcionado por AWS.

Seguridad aplicada:

- Uso de grupos de seguridad con IP de origen restringida.
- Llaves privadas (.pem) para autenticación segura en SSH.
- Contraseña aleatoria para la cuenta de administrador de Windows, descargada una sola vez.

Windows

Figura 5. Servidor windows ejecutandose



Configuración del servidor web

Windows (IIS):

1. Abrir el Administrador del servidor.
2. Añadir roles y características > Servidor Web (IIS).
3. Confirmar instalación.
4. Verificar acceso desde navegador: http://18.213.186.62

Figura 6. Configurando servidor windows

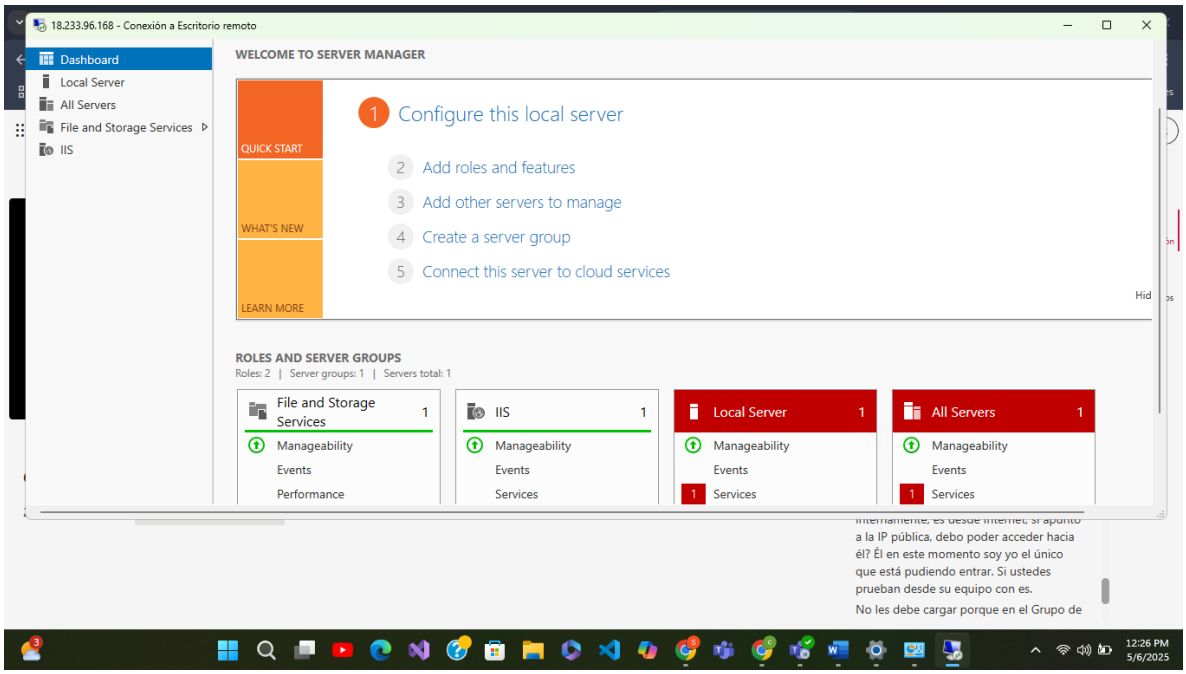


Figura 7. Prueba de servidor utilizando la IP

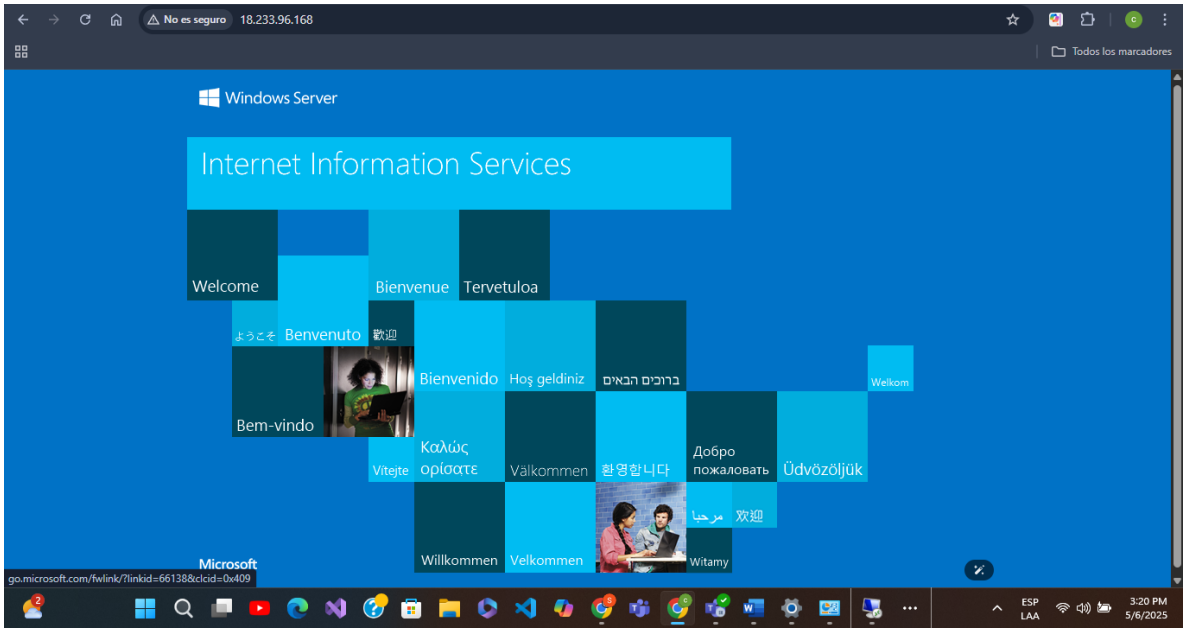
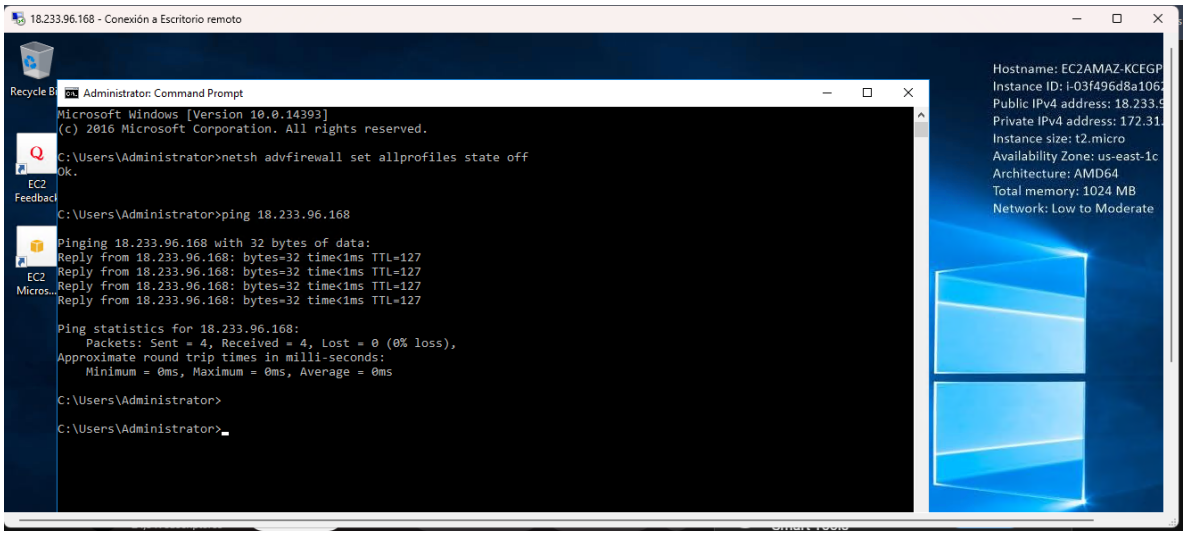


Figura 7. Hacemos Ping con la IP en windows



Linux

Figura 8. Descargamos la aplicación de Mobaxterm para la conexión del puerto SSH

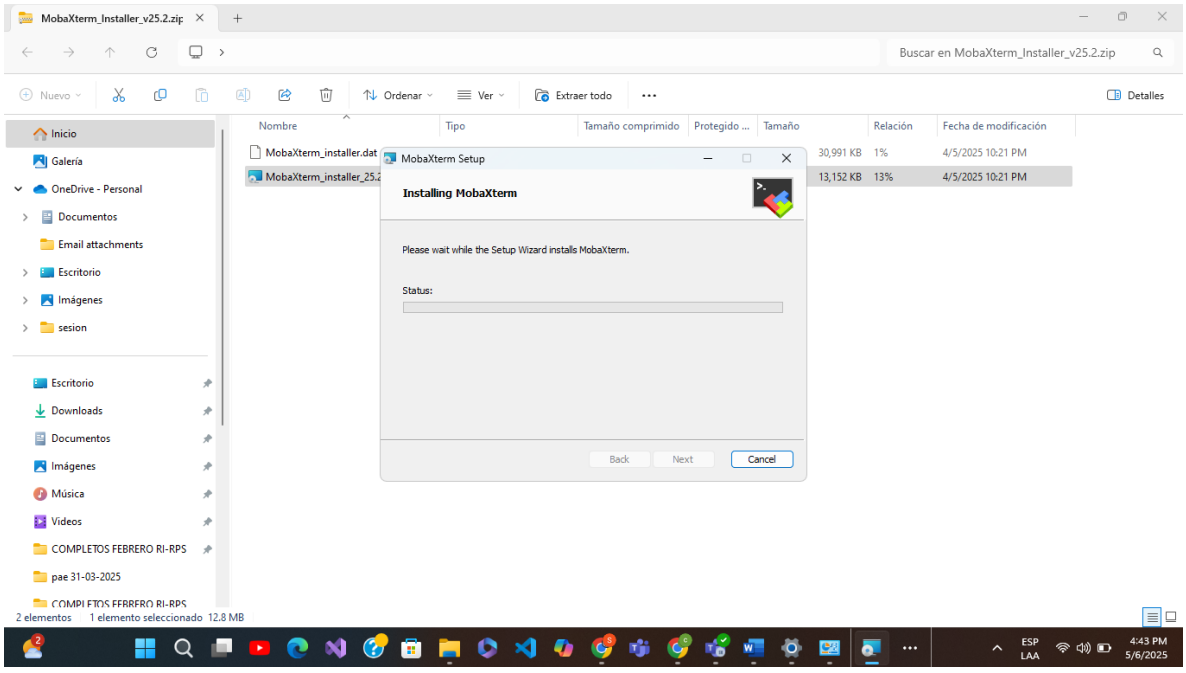
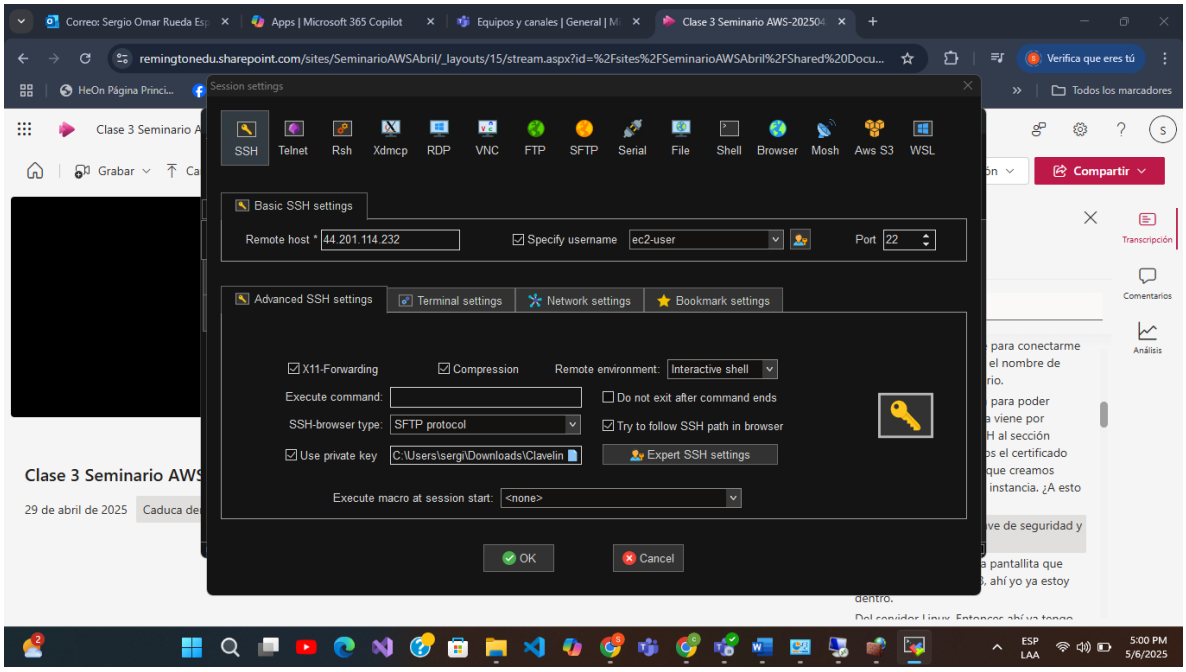


Figura 9. Nos conectamos al servicio de SSH a través de Mobaxterm



Linux (Apache):

1. Conectarse por SSH.
2. Ejecutar:
   sudo su
   dnf install httpd
3. Verificar servicio:
   systemctl status httpd
4. Iniciar systemctl start httpd
5. Acceder desde navegador: http://3.91.203.94

Figura 10. Configuramos todo desde la terminal y arrancamos el servicio

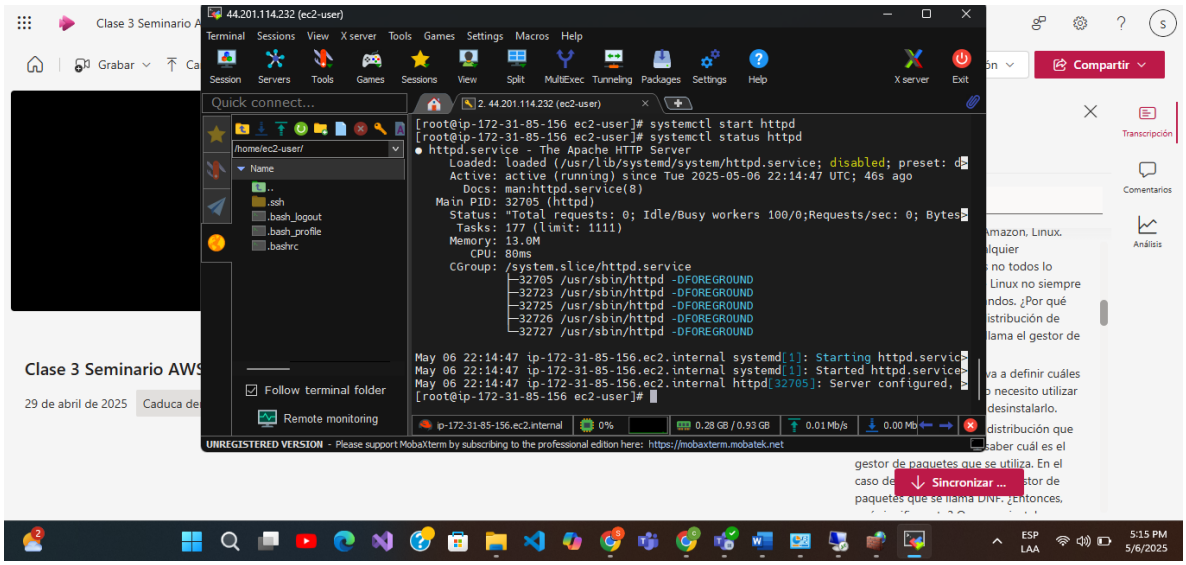
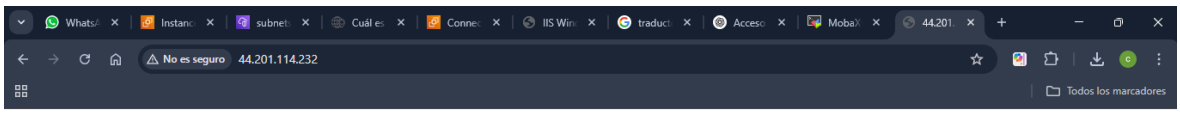


Figura 11. Arrancamos el servicio desde la web



It works!

Figura 12. Subimos una plantilla de prueba al servidor

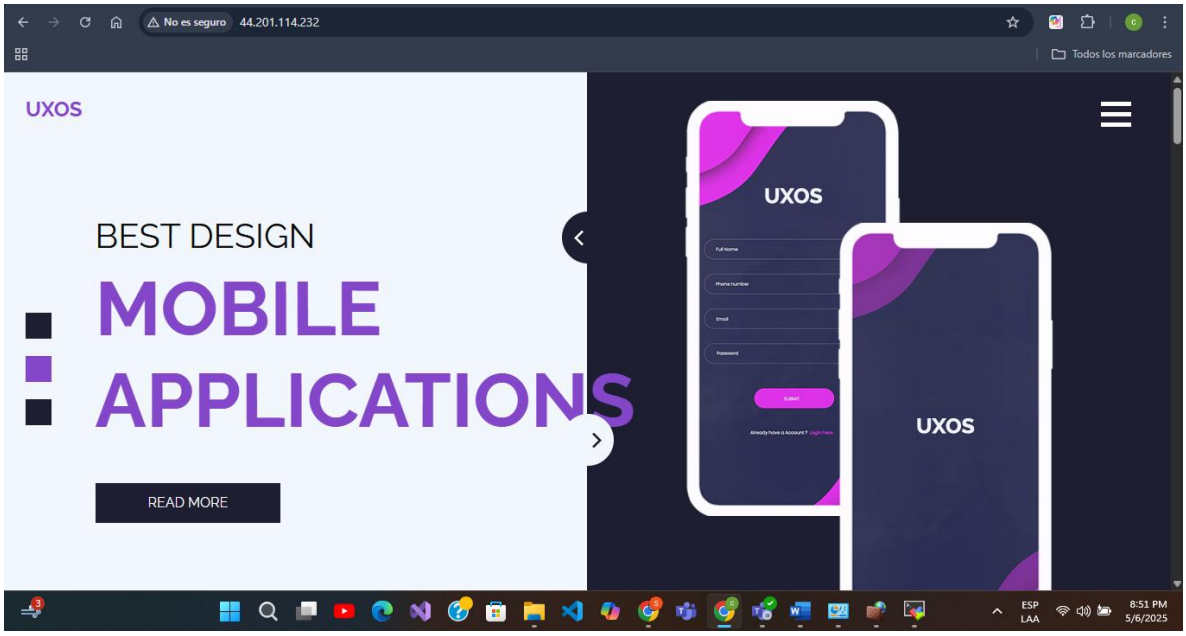
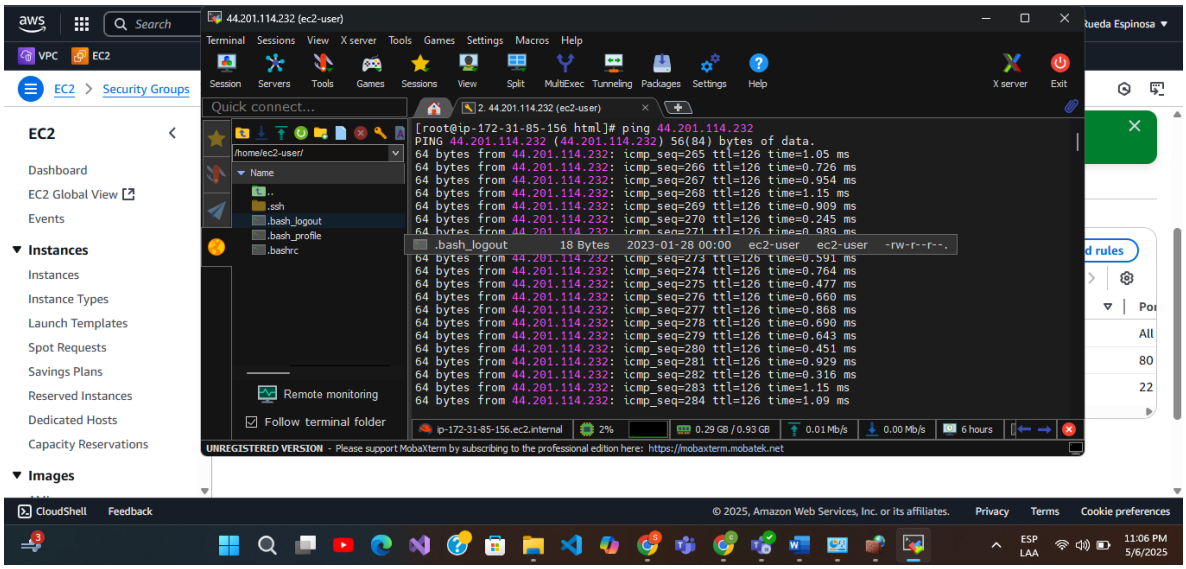


Figura 13. Hacemos Ping con la IP de la instancia de linux



Pruebas realizadas:

Se accedió desde un navegador externo a ambas IPs públicas, mostrando correctamente la página predeterminada de cada servidor web.

Aprendizajes adquiridos

Esta actividad permitió comprender de manera práctica la creación y administración de infraestructura en la nube, incluyendo:

- Configuración de redes privadas virtuales (VPC) y subredes.
- Despliegue y administración de instancias EC2.
- Implementación de servidores web en diferentes sistemas operativos.
- Gestión de accesos y reglas de firewall con grupos de seguridad.
- Verificación de conectividad y seguridad entre recursos.

Implementación de Docker

Docker es una herramienta que me permitió ejecutar aplicaciones de forma más práctica, rápida y ordenada. Lo que hace básicamente es empaquetar una aplicación con todo lo que necesita para funcionar (como sus archivos, librerías y configuraciones), y lo guarda dentro de algo llamado contenedor.

Este contenedor es como una caja independiente que puede moverse y ejecutarse en cualquier lugar sin preocuparse por si el sistema donde se va a usar es diferente al original. Lo mejor es que, a diferencia de una máquina virtual, no necesita todo un sistema operativo completo, por lo que es mucho más ligero y eficiente.

Figura 14. Instalamos docker

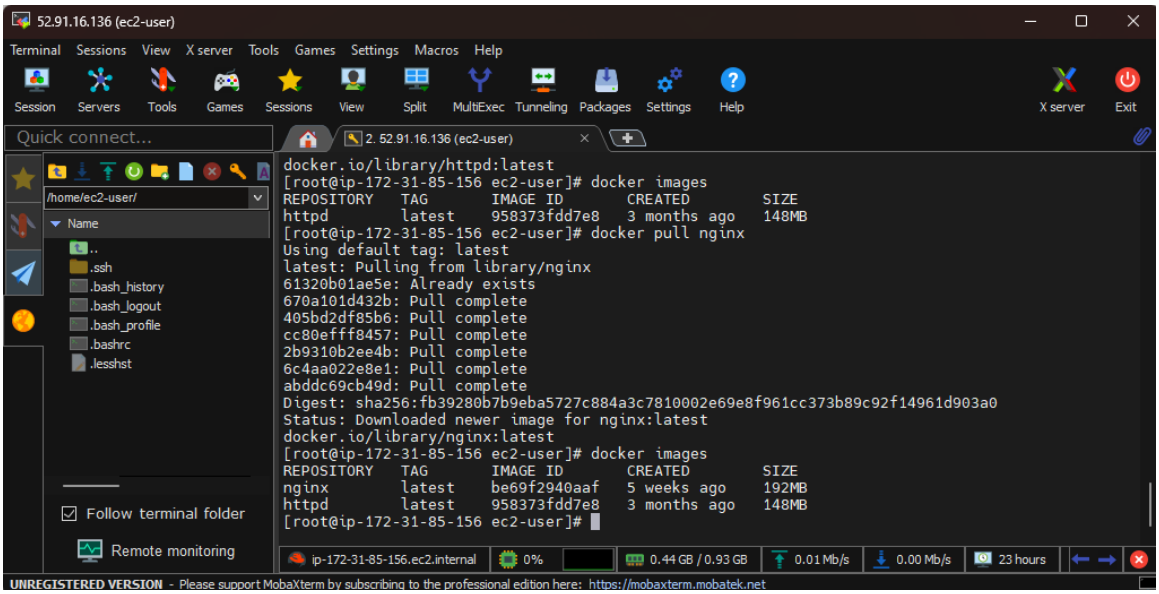


Figura 15. Creamos los contenedores

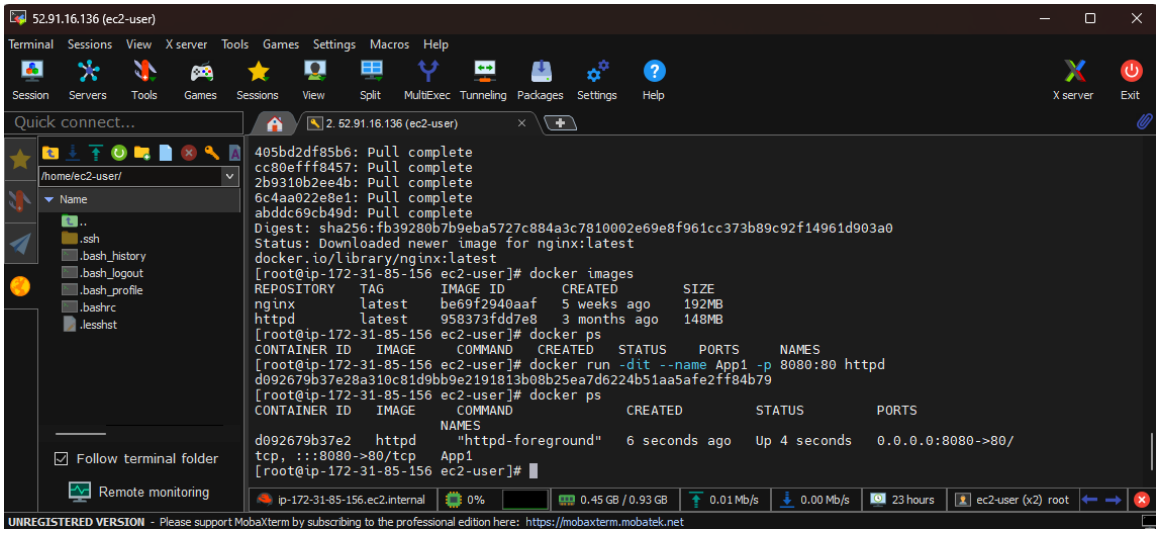


Figura 16. Los contenedores apuntan hacia un proxy el cual redirecciona el sitio web

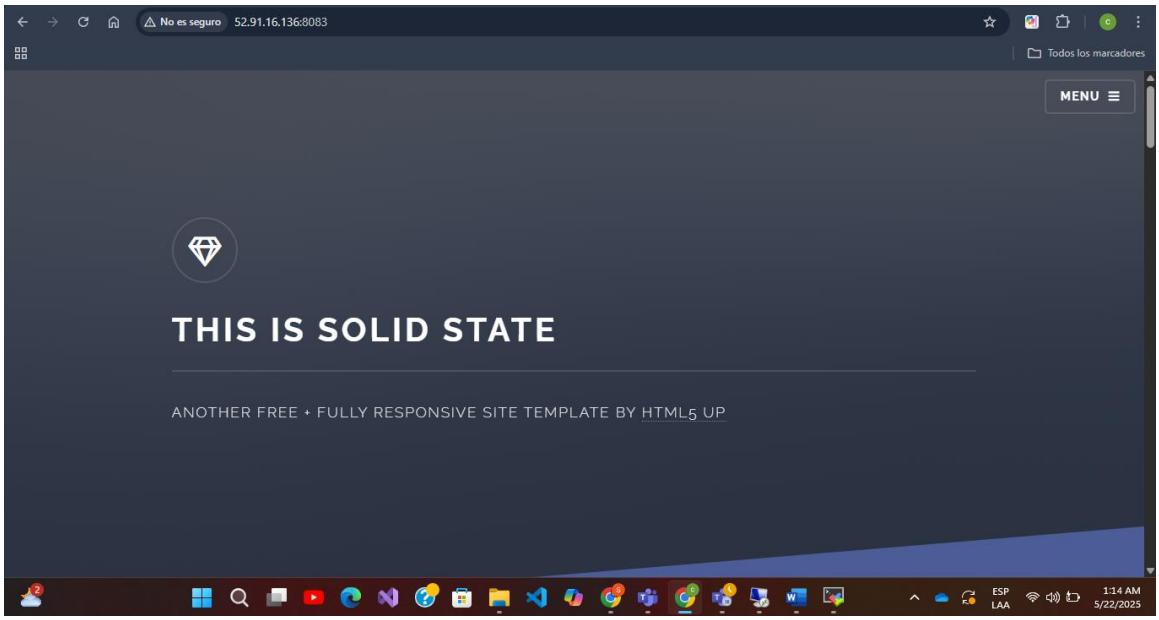
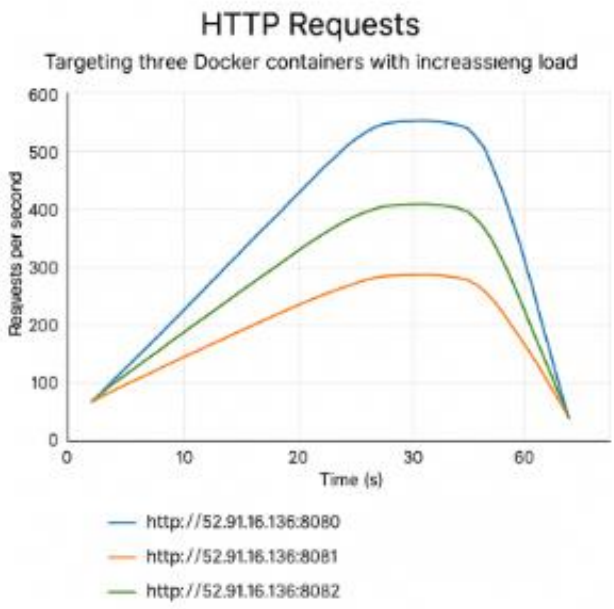


Figura 17. Prueba de estrés



Usar contenedores en lugar de máquinas virtuales tradicionales tiene varias ventajas que pude notar al trabajar con Docker.

Lo primero es que los contenedores son mucho más livianos. No necesitan instalar todo un sistema operativo como lo hacen las máquinas virtuales. Solo llevan lo justo y necesario para que la aplicación funcione, y eso hace que se inicien muy rápido y ocupen menos espacio y memoria.

Otra cosa que me gustó es su portabilidad. Como todo está dentro del contenedor (el código, las configuraciones, las librerías), puedo mover mi aplicación de un servidor a otro o incluso entre diferentes entornos sin que falle por diferencias del sistema. Es como tener una maleta con todo listo para funcionar en cualquier lugar.

También son muy útiles cuando se necesita escalar o crecer. Por ejemplo, si una aplicación empieza a tener muchos usuarios, puedo abrir más contenedores de esa misma app en segundos para que todos puedan usarla sin problemas. Y como cada contenedor es independiente, no se estorban entre sí ni generan conflictos.

En conclusión, trabajar con contenedores me permitió ver una forma más rápida, flexible y ordenada de desplegar aplicaciones, especialmente en entornos en la nube. Son una gran herramienta para los que buscamos eficiencia sin complicarnos demasiado.

Conclusiones

La implementación de esta red en AWS evidenció la eficiencia y flexibilidad que ofrece la computación en la nube. La actividad no solo permitió el despliegue de servidores con acceso público, sino también afianzó conocimientos sobre administración de recursos, configuración de servicios web y aplicación de buenas prácticas de seguridad. Estas habilidades son fundamentales para la gestión moderna de infraestructuras tecnológicas en entornos profesionales y académicos.

Además gracias a la implementación de Docker en este proyecto, pude comprender y aplicar de forma práctica los beneficios de la contenerización. Aprendí a crear, configurar y ejecutar contenedores, lo que me permitió desplegar aplicaciones de manera rápida, eficiente y portátil. Además, entendí cómo Docker facilita la escalabilidad del sistema y simplifica el manejo de entornos complejos, consolidándose como una herramienta fundamental en la administración de infraestructuras modernas en la nube.

Referencias

Blanco, G., & Vargas, G. (2023). Corredor profundo Amazon AWS. *CITAS*, *9*(1). https://doi.org/10.15332/24224529.8834

Brañes Vílchez, R. E. (2019). *Arquitectura De Back End Con Amazon Web Services (AWS) Para Sistemas Escolares*.

Henriquez, C., Del Vecchio, J. F., & Paternina, F. J. (2015). La computación en la nube: un modelo para el desarrollo de las empresas. *Prospectiva*, *13*(2), 81. https://doi.org/10.15665/rp.v13i2.490

Lopez Cendejas, K. (2008). *Estimación de los residuos biológico infecciosos en los hospitales de la ciudad de Chetumal, Quintana Roo*. https://risisbi.uqroo.mx/handle/20.500.12249/450

Molina Daza, R., & Tardio Lopez, J. (2018). *Introducción al Cloud Computing y comparativa de plataformas*. Universitat Politècnica de Catalunya.