TRABAJO DE GRADO
Opción Seminario-Diplomado.

Implementación de los servicios AWS en la empresa **RESTAURANTE&BAR AWJ**

Corporación Universitaria Remington.
Nombre de la facultad: Facultad de ingeniería.
Nombre del programa académico: Ingeniería de sistemas.

Jhoan Alberto Torres, Wilmar Ramirez, Andres Felipe Mena.
Juan Pablo Berrio López.
Seminario Amazon Web Services (AWS).
2025.

Tabla de Contenidos

Tabla de ilustraciones

Resumen

Se implementaron los servicios de AWS a la empresa RESTAURANTE&BAR, hemos diseñado una arquitectura preliminar en Amazon Web Services (AWS). Es altamente disponible, escalable y está diseñada para manejar una gran cantidad de tráfico de manera eficiente.

Estos servicios son manejados en la capa gratuita mientras estén disponible por el prestador de este servicio, luego de 1 año que es el plazo máximo, generara cobros por cada uno de los servicios creados, eso no quiere decir que luego de un año la empresa dejara de utilizar los servicios, la solución es invertir un poco económicamente para seguir con esta calidad de servicios prestados por Amazon.

Palabras clave

AWS (Amazon web services), servicios, RESTAURANTE&BAR AWJ, instancias EC2, autoscaling, grupos de servicios, balanceadores de carga, proyecto, web.

Marco conceptual y contextual

Este proyecto nació durante un seminario de Amazon Web Services (AWS), donde aprendimos un montón sobre cómo diseñar y manejar infraestructuras en la nube. La verdad es que fue súper práctico: vimos de todo, desde servicios de cómputo, redes, almacenamiento, seguridad, balanceo de carga, automatización y escalabilidad, cosas clave para cualquier empresa que quiera modernizarse.

AWS, por si no lo sabías, es básicamente como un gigante tecnológico que te presta recursos en la nube cuando los necesitas, lo que ayuda a las empresas a crecer sin complicaciones. En el proyecto usamos cosas como (AWS, s.f.)

- **EC2 (Elastic Compute Cloud)** para la creación de máquinas virtuales,
- **VPC (Virtual Private Cloud)** para la configuración de redes privadas,
- **ELB (Elastic Load Balancing)** para distribuir el tráfico entre instancias,
- **Autoscaling Groups** para ajustar automáticamente la capacidad de cómputo según la demanda,
- **Amazon Machine Images (AMI)** para replicar configuraciones predefinidas.

Trabajamos en un caso real: un RESTAURANTE&BAR AWJ que necesitaba una infraestructura escalable y segura. Aplicamos lo aprendido para diseñar una solución a su medida, enfocándonos en resolver problemas técnicos y trabajar en equipo

Entrega 1

implementación de una red en AWS con servidores Windows y Linux accesibles desde internet

1. Descripción de la arquitectura

Usamos:

- Windows Server 2016 (Capa gratuita) con un tipo de instancia T2.micro
- Amazon Linux con un tipo de instancia T2.micro

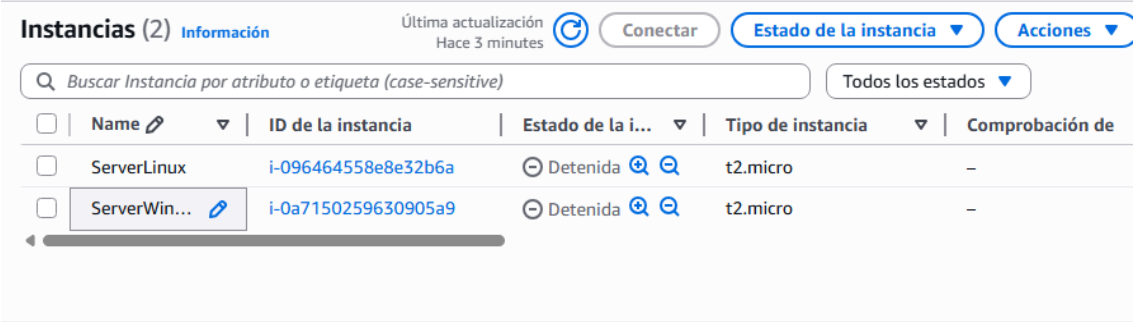

Ilustración 1.Instancias

Justificación de las configuraciones de red:

Se creo un VPC con 4 subredes 2 públicas y 2 privadas, llamado proyecto-vpc.

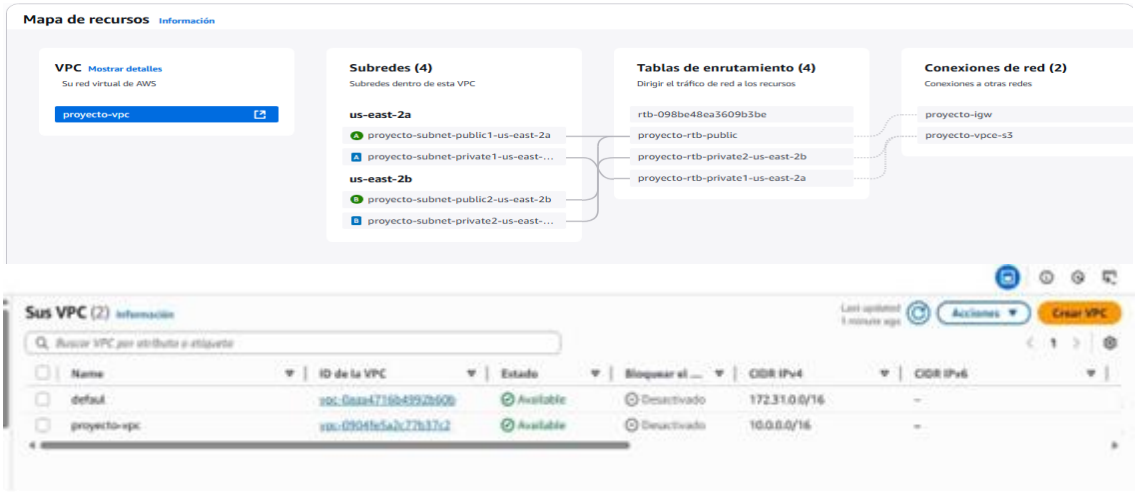

Ilustración 2.VPC

2. Configuraciones realizadas

Pasos para crear las instancias EC2. Para crearlas instancias EC2 en Amazon Server, debemos dirigirnos a la configuración EC2 luego seguimos con los siguientes pasos:

- **Nombre y etiqueta:** Aquí creamos un nombre para esta instancia puede llamarlo de la manera que prefiera.
- **Elegir la imagen del sistema operativo:** Nos aparecen la principales AMI o las más utilizadas en este servicio debemos elegir la que vamos a utilizar en este caso las de capa gratuita como Amazon Linux y Windows Server 2016 Beta.
- **Tipo de instancia:** Debemos elegir el tipo de instancia teniendo en cuenta las características que necesitamos según nuestro servicio que queremos utilizar en esta actividad elegimos la T2.micro por ser un tipo dentro de la capa gratuita.
- **Par de claves:** Este paso es importante creamos las claves para el inicio de sesión de nuestras instancias, se crea al darle click en la opción “crear un nuevo par de claves”, se creará un archivo .pem este debe ser guardado en una carpeta o un sitio seguro para que no sufra de pérdida o robo de información.
- **Configuraciones de red:** En este paso debemos agregar la red pública del VPC que creamos para poder tener acceso desde internet.
- **Configuración de almacenamiento:** Se recomienda dejarlo por defecto.
- Como último paso debemos dar click en lanzar instancia.

Detalles de los Grupos de Seguridad: Cada instancia maneja diferentes puertos.

- En Windows, se utiliza el puerto RDP para el acceso al escritorio el puerto RDP es el 3389. El puerto HTTP por defecto es el 80. Este puerto se utiliza para la comunicación no segura entre navegadores web y servidores.

- En Linux, el puerto predeterminado para las conexiones SSH es el 22. Este puerto se utiliza para establecer conexiones seguras entre un cliente y un servidor

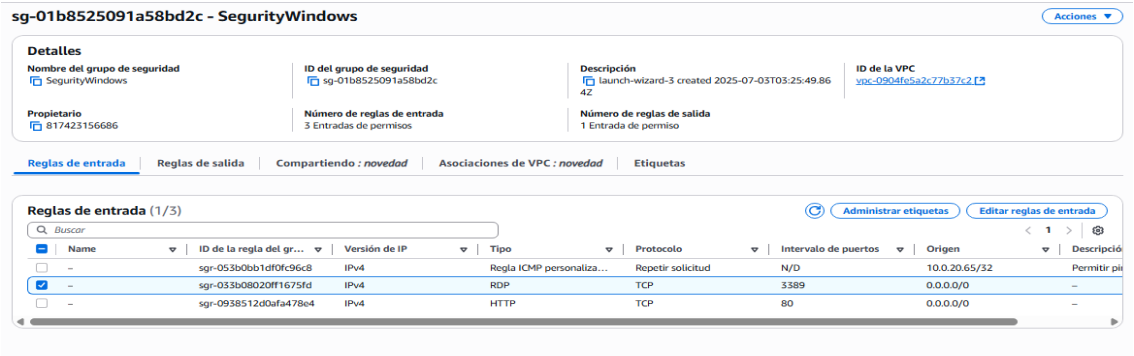

Ilustración 3.Grupo de seguridad

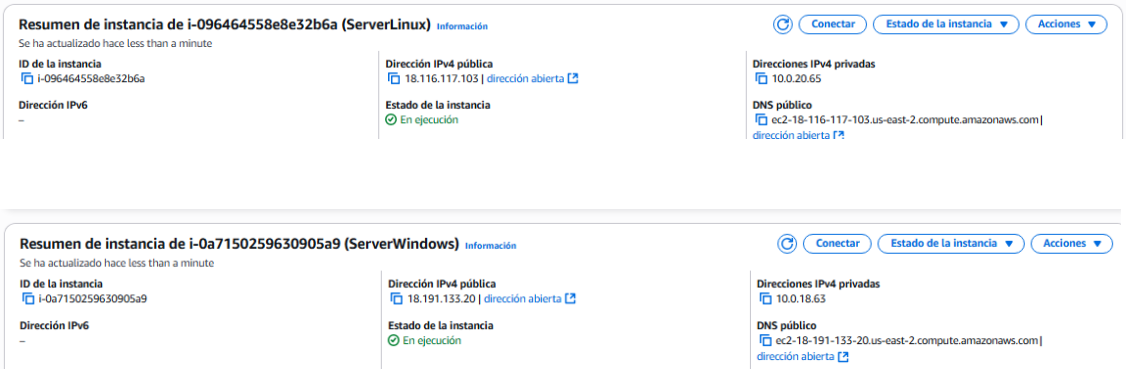

Ilustración 4.Resumen de instancias

Las IP’s públicas y privadas son generadas automáticamente por AWS quien las crea según el servidor que asigna en la región elegida, estas IP’s también se pueden crear manualmente.

3. **Procedimiento de acceso**

Cómo acceder a cada servidor:

- Para acceder al servidor de Windows lo hicimos desde el escritorio remoto, poniendo la ip publica o el DNS público para conectarse al servidor y luego entrar con el administrador predeterminado por Amazon Server y la clave generada en el cliente RDP donde utilizaremos el archivo .pem.

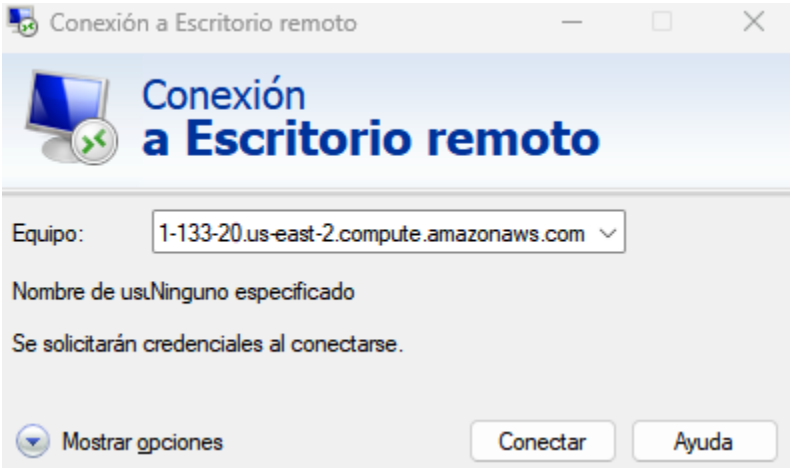

Ilustración 5.Conexion RDH

- En Linux utilizamos un cliente SSH llamado MobaXterm donde debemos copiar nuestra ip publica o DNS público y generar la clave con el archivo .pem y el siguiente paso es escribir el usuario que viene predeterminado por el cliente SSH.

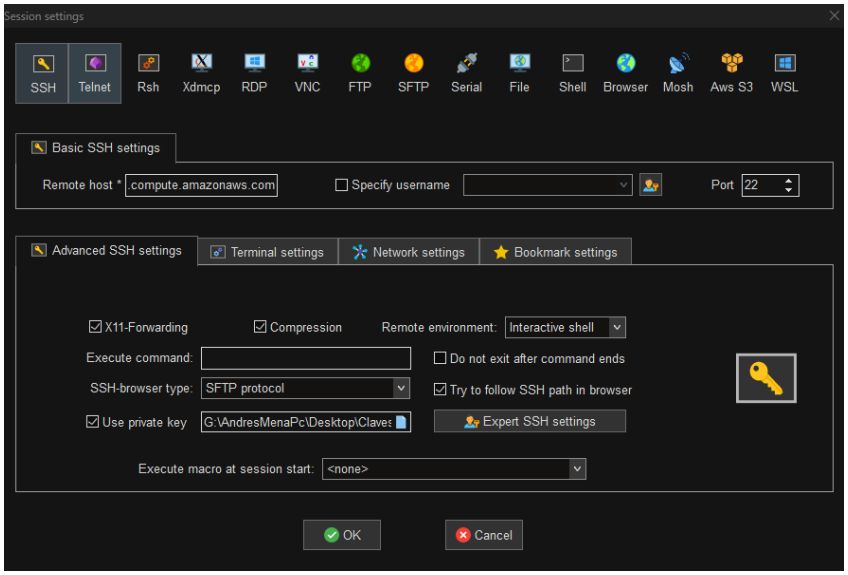

Ilustración 6.Cliente SSH MobaXterm

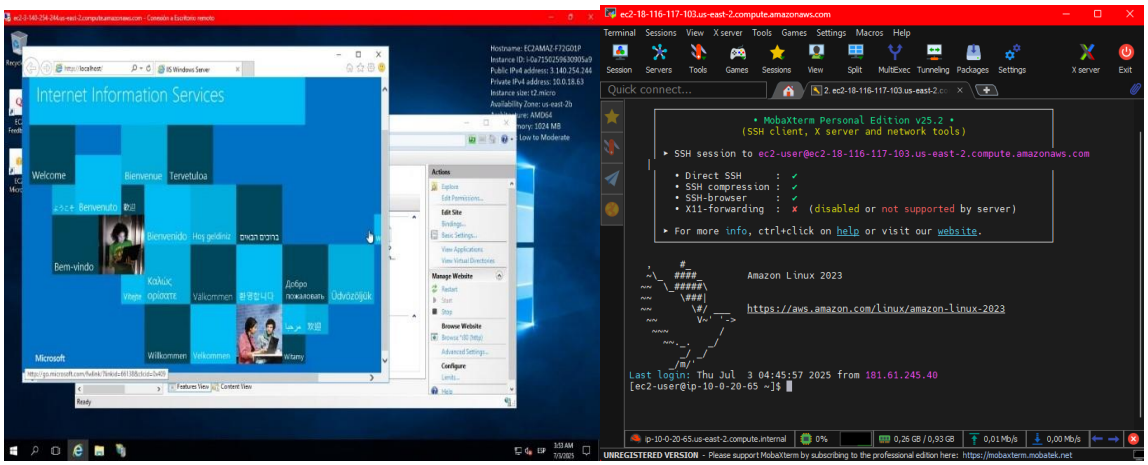

Ilustración 7.Conexiones de servidores

Consideraciones de seguridad: Consideramos utilizar el archivo .pem para ambas instancias, este archivo está en un lugar seguro para evitar pérdidas.

4. **Configuración del servidor web**

Pasos seguidos para instalar IIS en Windows Server:

- Abre el Administrador del Servidor.
- Haz clic en "Agregar roles y características".

- Selecciona "Instalación basada en roles o características".
- Elige el servidor donde deseas instalar IIS.
- Selecciona "Servidor Web (IIS)".
- Elige los servicios de rol que necesitas, como ASP.NET.
- Revisa la configuración y haz clic en "Instalar".

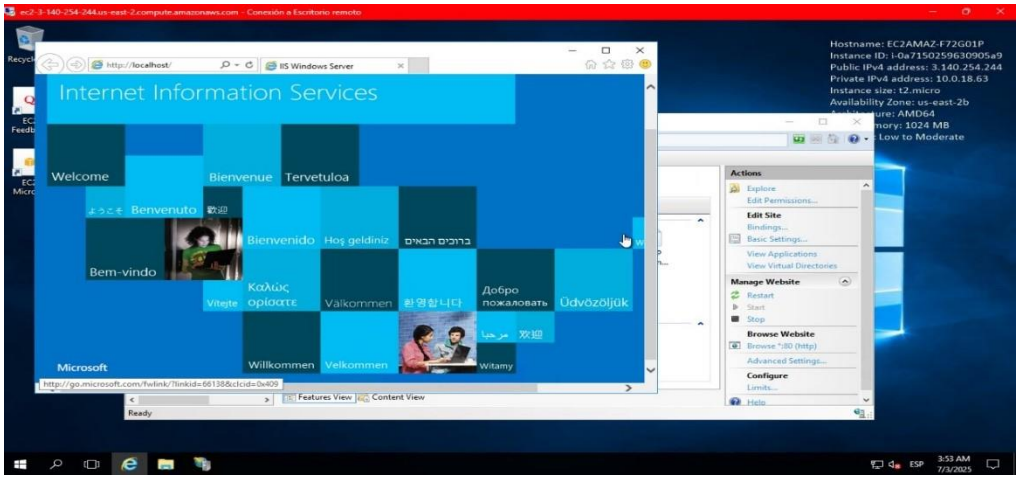

Ilustración 8.Server Windows

Pasos para instalar Apache o Nginx en Linux: Utilizamos los siguientes comandos en el cliente SSH

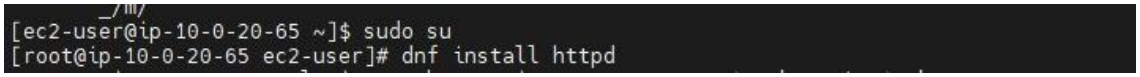

Ilustración 9.Instalacion Apache

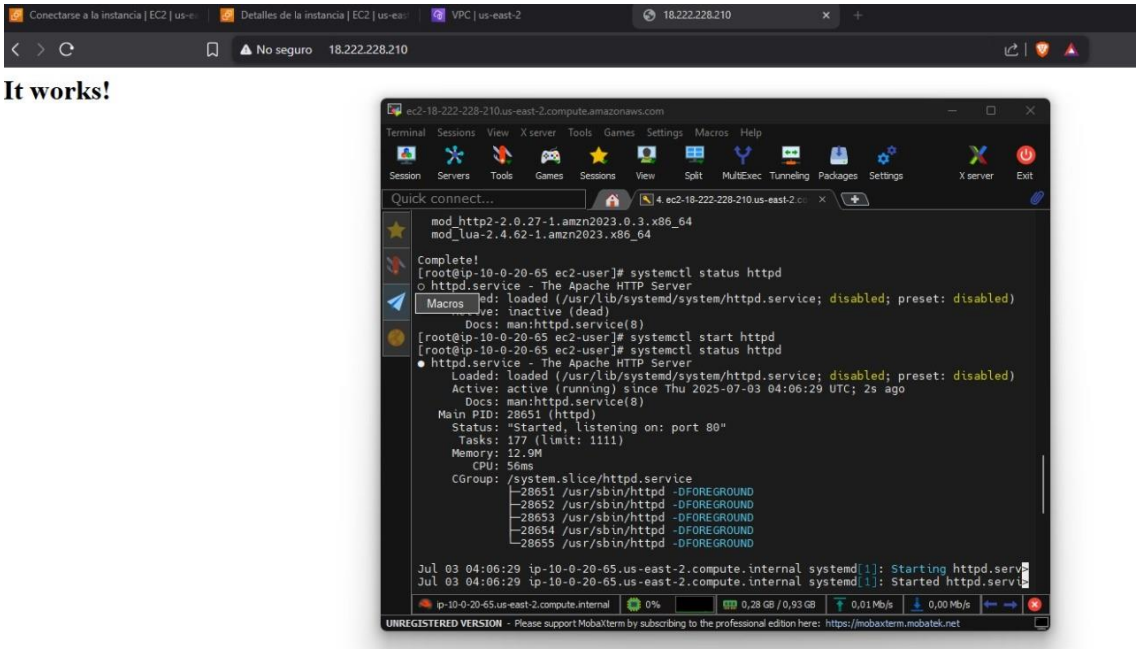

Ilustración 10.Server Linux

Pruebas básicas para verificar que los servidores web son accesibles desde Internet (captura de pantallas del navegador).

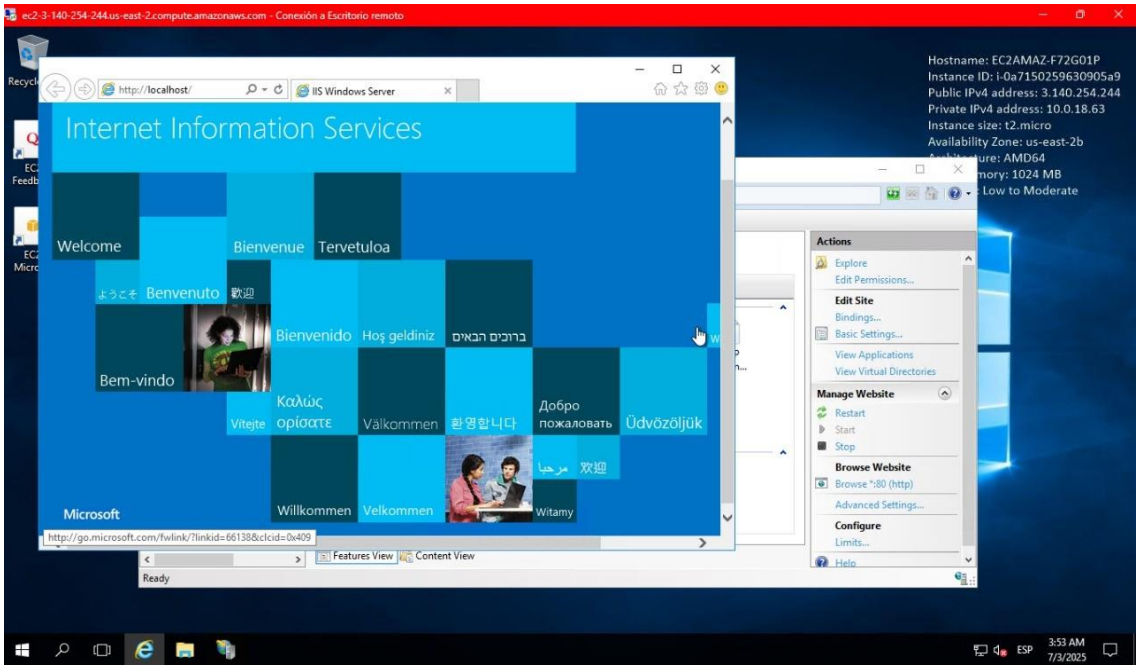

Ilustración 11.Navegador server

Entregable 2

Implementación de arquitectura en AWS con balanceadores de carga y contenedores

Asumimos el rol de arquitectos en la nube para un startup llamada RESTAURANTE&BAR AWJ, una plataforma innovadora que conecta a restaurantes con clientes mediante entregas rápidas. La empresa ha experimentado un rápido crecimiento en los últimos meses y ahora necesita escalar su infraestructura tecnológica para manejar una mayor demanda, garantizar la disponibilidad y mejorar los tiempos de respuesta.

- **Balanceador de Carga**

Fueron creados los balanceadores de carga que son muy importante para la efectividad del servicio, estos balanceadores ayudan a sostener un equilibrio entre los servidores de esta empresa.

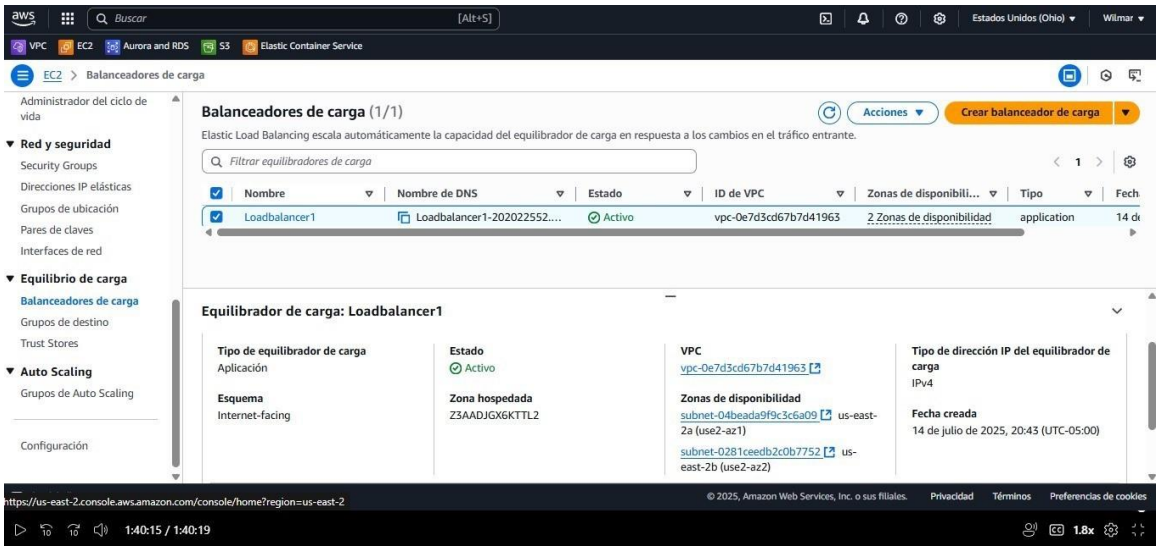

Ilustración 12.Balanceador de carga

- **Instancias EC2.**

Fueron creadas 2 instancias una publica y otra privada, cabe recalcar que de una instancia (la publica) en este caso “Server Linux” nace una AMI para la creación del servidos “ServerLinux2”, en la gráfica se muestran estas instancias y la implementación de la regla autoscaling.

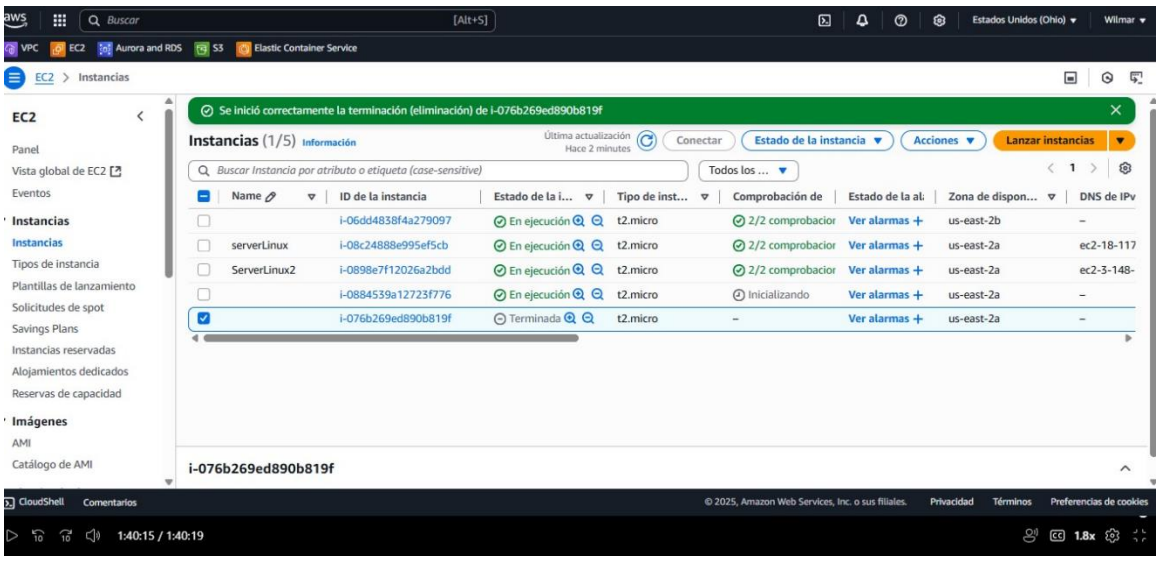

Ilustración 13.Instancias con Autoscaling

- **Instancias con Proxy Reverso**:

implementemos el proxy reverso en nuestras instancias EC2 para dirigir las solicitudes a servicios internos.

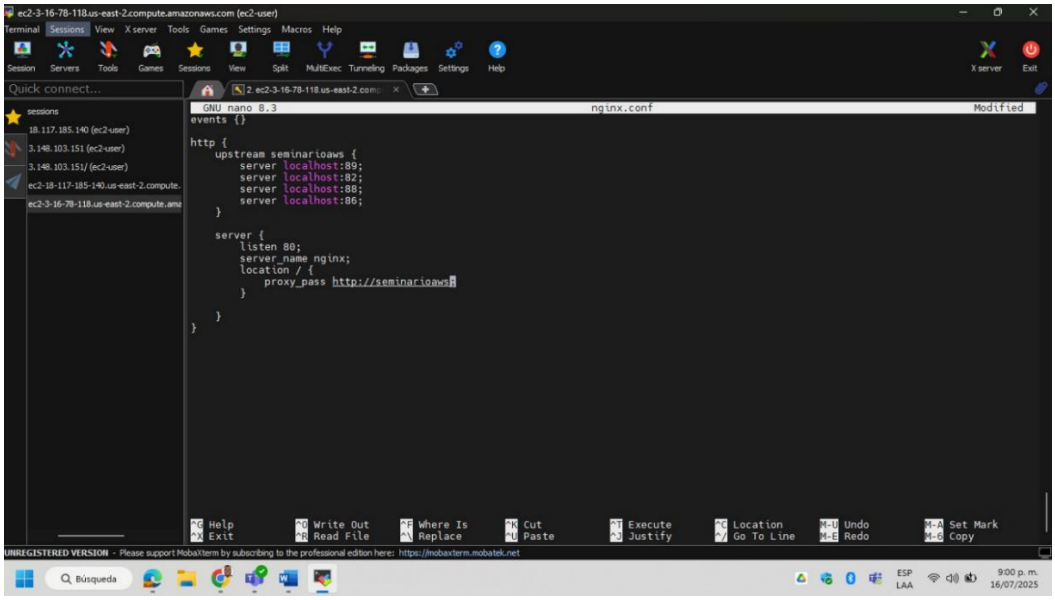

Ilustración 14.Proxy Reverso

- **Servicio de Docker de forma manual, con una aplicación de prueba**.

La implementación de los Docker de forma manual se vuelve un poco extenso, pero es una manera practica de realizarlo, utilizamos una plantilla HTML como aplicación de prueba.

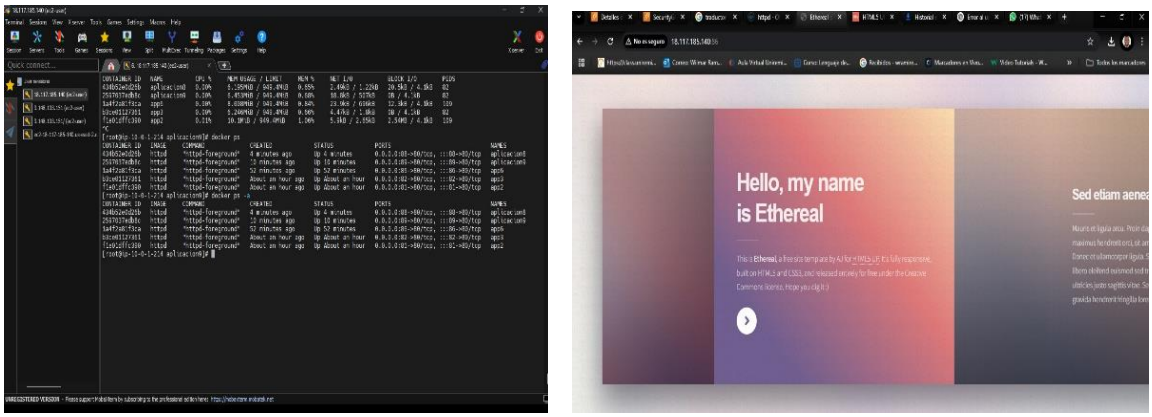

Ilustración 15.Docker manuales y pagina de prueba

- **Autoescalado.**

Para la creación del autoscaling, hay políticas para que el servicio tenga reglas que cumplir siempre y cuando haya un fallo en alguna de las instancias.

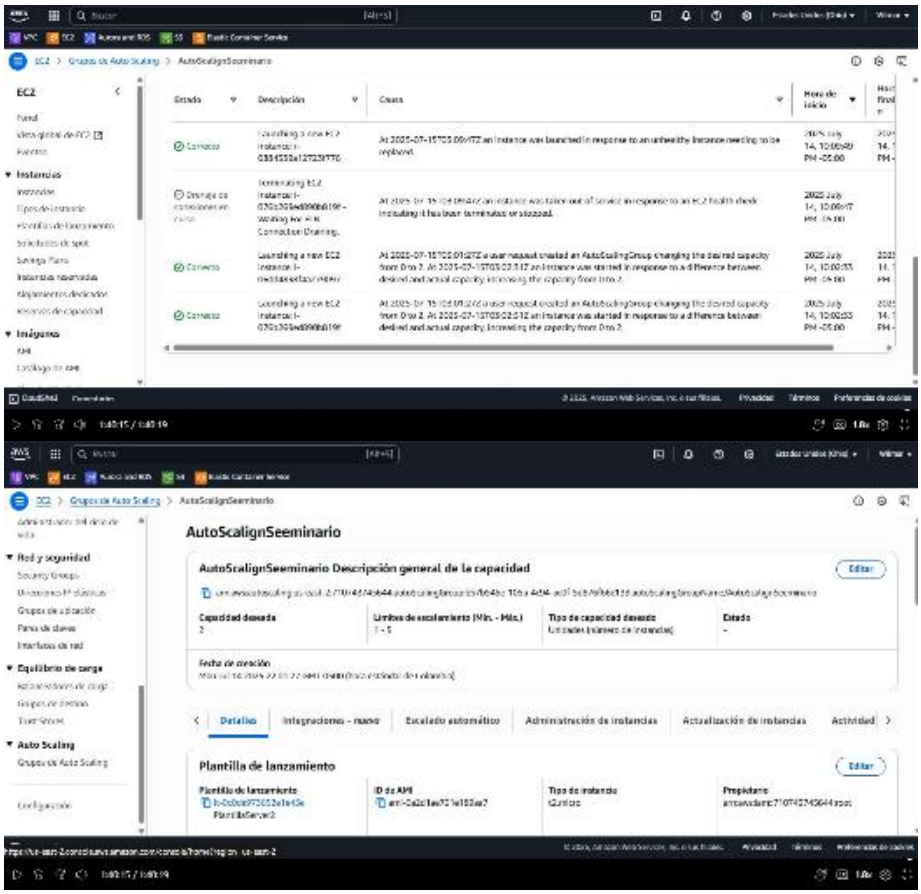

Ilustración 16.AutoScailing

Conclusiones

En conclusión, AWS es un servicio muy completo para la creación y gestión de proyectos empresariales como lo es **RESTAURANTE&BAR AWJ**, a lo largo de este proyecto implementamos diferentes servicios como equipo de trabajo, donde se unieron las ideas y soluciones para así poder resolver los desafíos, errores o mejoras para la eficiencia del proyecto de este restaurante, bar.

Utilizamos Servicios fundamentales como la creación de instancias en EC2, balanceadores de carga, autoescalables o Autoscaling, vpc's, grupos de destino, y los diferentes servicios enseñados en el desarrollo del documento, cada uno de estos servicios son importante ya que uno complementa al otro en muchos casos, también se utilizaron AMI que son imágenes de las instancias EC2, que sirvieron para así crear unas copias de las principales instancias.

Gracias a esta implementación se logró el objetivo esperado en el proyecto para la empresa RESTAURANTE&BAR AWJ.

Referencias

AWS. (s.f.). Obtenido de Amazon Web Services: https://aws.amazon.com