TRABAJO DE GRADO
Opción Seminario-Diplomado.

Aplicación de una red en AWS escalable y sostenible con recursos de alta disponibilidad

Corporación Universitaria Remington.
Facultad de Ingenierías
Ingeniería de Sistemas
Tecnología en Desarrollo de Software

Esteban Javier Causado Sibaja
Marwin Alejandro Perdomo Quintero
Lina Marcela Ríos Ríos

Docente: Juan Pablo Berrio López.
Opción de Trabajo de grado Seminario-Diplomado.
2025

Tabla de Contenidos

Resumen

Para la presentación e implementación de nuestro trabajo se destacan tres aspectos importantes en los que se desarrollaron en nuestro seminario de AWS Amazon plataforma de servicios en la nube:

Se identifican y se desarrollan servicios ofrecidos en AWS, en un lenguaje técnico se conoce diferentes componentes que pueden integrar una solución tecnológica de alta disponibilidad como son las instancias, volúmenes, instantáneas, VPC, Clústeres, configuraciones netscaler, contenedores, Proxis, getways, configuraciones autoescaling, balanceadores de carga, entre otros.

Es importante conocer y entender el funcionamiento de la arquitectura en la nube para disponer de una alta disponibilidad Cumpliendo Con las diferentes configuraciones en los data Center.

Otro ítem a destacar es la implementación de una arquitectura de alta disponibilidad en AWS pueda soportar el funcionamiento de una Startup; para este fin se realizó la configuración de servidores Linux y Windows en los que se definen políticas de seguridad, configuraciones de seguridad y conectividad para disponer de un servicio web en los cuales también se desarrollan y configuran otros componentes como lo son los contenedores utilizando internamente el mismo puerto pero con la configuración de reversos para poder tener la alta disponibilidad sin que afecte el servicio ofrecido configuraciones de auto scaling, servicios de clúster en diferentes data Center.

Durante la ejecución del seminario logramos ampliar el conocimiento de diferentes componentes de arquitectura de una red de datos con el aprovechamiento de los diferentes recursos ofrecidos en una plataforma en la nube, para nuestro ejercicio AWS (Amazon Web Services). Para nuestra practica e implementación logramos configurar las instancias con diferentes sistemas operativos, con IP públicas y privadas, servicios web con alta disponibilidad mediante la configuración de contenedores Clústeres (serverless) y balanceadores de carga entre otros.

Palabras clave

- Diseño de red.
- Arquitectura de red.
- Instancia
- Balanceador de carga
- AWS

Marco conceptual y contextual

Auto Scaling: permite garantizar del número correcto de EC2 (Elastic Compute Cloud), disponibles para gestionar la carga de su aplicación. Se puede especificar el número mínimo o máximo de instancias en cada grupo de Auto Scaling. (Amazon Web Services, s.f.).

AMI: (Amazon machine image) es una imagen que ofrece el software que se necesita para configurar y arrancar una instancia de Amazon EC2. Cada AMI también contiene una asignación de dispositivos de bloques que especifica los dispositivos de bloques que se deben asociar a las instancias que se lancen. Debe especificar una AMI al iniciar una instancia. (Amazon Web Services, s.f.).

AWS: (Amazon Web Service) es un proveedor de servicios en la nube, nos permite disponer de almacenamiento, recursos de computación, aplicaciones móviles, bases de datos en la modalidad de cloud computing. (Amazon Web Services, s.f.).

Balanceador de Carga: Elastic load balancing (ELB) actúa como único punto de contacto para los clientes. Los clientes envían solicitudes al balanceador de cargas y el balanceador de cargas las envía a los destinos, como EC2 las instancias. (Amazon Web Services, s.f.).

Contenedor: paquete de código de software que contienen el código de una aplicación, sus bibliotecas y otras dependencias que necesita para ejecutarse en la nube. Cualquier código de aplicación de software requiere archivos adicionales denominados bibliotecas y dependencias para poder ejecutarse. (Amazon Web Services, s.f.)

EC2: Amazon Elastic Compute Cloud (Amazon EC2) proporciona capacidad de computación escalable bajo demanda en la nube de Amazon Web Services (AWS). Se puede usar Amazon EC2 para lanzar tantos servidores virtuales como se necesite, configurar la seguridad y las redes, y administrar el almacenamiento. (Amazon Web Services, s.f.).

Grupo de seguridad: funciona como un firewall virtual para las instancias de EC2 (Elastic Compute Cloud) para controlar el tráfico entrante y saliente. Las reglas de entrada controlan el tráfico entrante a la instancia y las reglas de salida controlan el tráfico saliente desde la instancia. Al iniciar una instancia puede especificar uno o varios grupos de seguridad. Si no especifica un grupo de seguridad, Amazon EC2 utiliza el grupo de seguridad predeterminado para la VPC. Una vez lanzada la instancia, puede cambiar sus grupos de seguridad. (Amazon Web Services, s.f).

Httpd: más conocido como un servidor web (demonio de protocolo de transferencia de hipercontexto), con un diseño de comunicación en un servicio cliente servidor. (The apache HTTP, s.f)

ICMP: el protocolo de mensajes de control de Internet es un protocolo en la capa de red que utilizan los dispositivos de red para diagnosticar problemas de comunicación en la red. El ICMP se utiliza principalmente para determinar si los datos llegan o no a su destino a su debido tiempo. El protocolo ICMP se suele utilizar en dispositivos de red, como los enrutadores. (CloudFlare, s.f.)

Instancia: es un recurso de servidor que brindan servicios en la nube de terceros. Se puede usar la instancia en la nube para ejecutar cargas de trabajo con uso intensivo de cómputos, como contenedores, bases de datos, microservicios y máquinas virtuales. (Amazon Web Services, s.f.).

Proxy: ubicado frente a la infraestructura del servidor web, un proxy inverso intercepta solicitudes de Internet y las reenvía a servicios internos. A menudo ayuda con el equilibrio de carga, el almacenamiento en memoria caché y la mejora de la mitigación de vulnerabilidades del proxy. (Zcaler, s.f.)

Puerto de red: es un protocolo de comunicación donde se puede configurar servicios entre diferentes dispositivos de una red. (Tivoli Network Manager IP Edition, s. f.)

RDP: Remote desktop protocol, es un servicio de comunicación remota hacia un servidor remoto, equipo de cómputo mediante puerto de conexión común 3389. (Cómo Usar Escritorio Remoto - Soporte Técnico de Microsoft, s. f.)

SSH: es un protocolo de comunicación segura mediante canales de información cifrados principalmente usados para conexión la línea de comandos en servidores Linux, por el puerto 22. (Secure Shell - ArchWiki, s. f.)

VPC: Virtual Private Cloud, puede lanzar recursos de AWS en una red virtual aislada de manera lógica que haya definido. Esta red virtual es muy similar a la red tradicional que usaría en su propio centro de datos, pero con los beneficios que supone utilizar la infraestructura escalable de AWS (Amazon Web Services, s.f.).

Desarrollo e implementación del aprendizaje

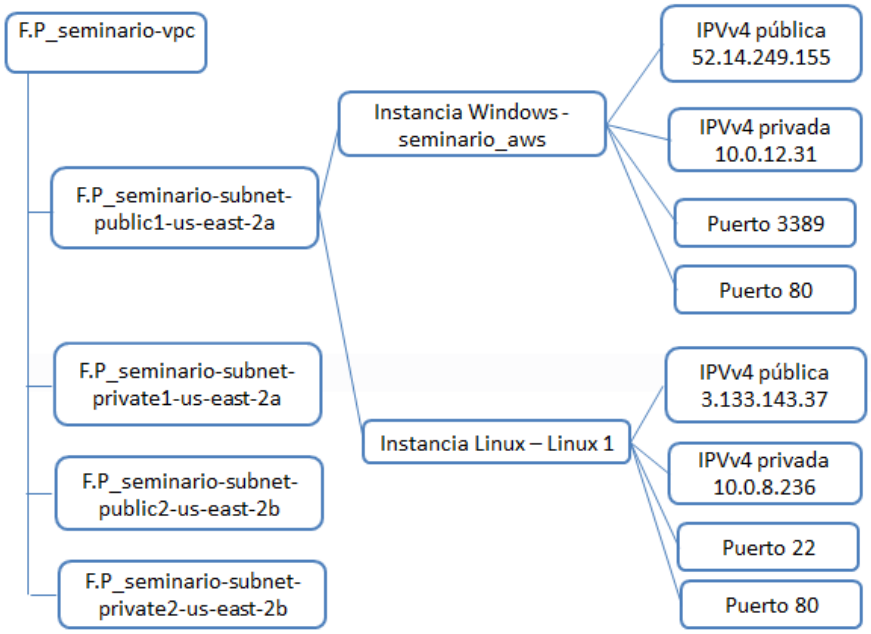
1. Implementación de una red en AWS con servidores Windows y Linux accesibles desde Internet

1.1 Documentación Técnica

Diagrama de arquitectura

- Representación gráfica de la red (EC2s, subredes, IPs públicas/privadas, grupos de seguridad, VPC, etc.)

Figura 1. Representación gráfica de la red

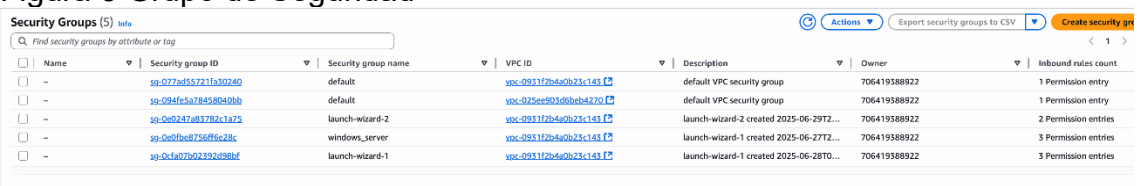


Fuente: Elaboración propia.

Figura 2. VPC

Fuente: Elaboración propia.

Figura 3 Grupo de Seguridad



Fuente: Elaboración propia.

Descripción de la arquitectura

- Breve explicación de la red creada.
  Se creo una red dentro de una VPC denominada F.P_seminario-vpc con cuatro subredes, dos públicas y dos privadas.
  Dentro de la red pública F.P_seminario-subnet-public1-us-east-2a se crearon una instancia de Windows y una de Linux.

- Tipo de instancias usadas (Linux: Amazon Linux, Ubuntu, etc. | Windows: versión de Windows Servidor).
  Para la instancia de Windows se utilizó la versión Windows_Servidor-2016-English-Full-Base-2025.06.11 con un tamaño de disco de 30 GB y se habilitaron los puertos 3389 y 80.

  En cuanto a la instancia de Linux se utilizó la versión al2023-ami-2023.7.20250623.1-kernel-6.1-x86_64 de Amazon Linux con un tamaño de disco de 8 GB y se habilitaron los puertos 22 y 80.

- Justificación de las configuraciones de red (por ejemplo, uso de VPC, subredes públicas, Internet Gateway).
  Se configuro una red con dos subredes públicas ya que nos permitiría tener una mayor disponibilidad si se tiene un alto flujo de usuarios.

Configuraciones realizadas

- Pasos para crear las instancias EC2.

  EC2 para Windows Servidor:
  Para crear la instancia de Windows Servidor, lo primero que debemos hacer es buscar la opción EC2, y lanzamos una nueva instancia, donde seleccionamos la imagen, con la versión Microsoft Windows Servidor 2016 Base por los pocos requisitos que maneja, el tipo de instancia se selección t2.micro que es apta para la capa gratuita de aws, posterior a eso, se crea la clave de inicio de sesión, procedemos con las configuraciones de red donde se selecciona un VPC anteriormente creado, teniendo en cuenta que debe ser elegida una subred publica, habilitamos la asignación de IP publica automáticamente, se crea un grupo de seguridad en donde se habilitan los puertos que se necesitaran para tener acceso, para la configuración del almacenamiento se selecciona gp2 de aproximadamente 30gb o lo que se requiera.

  EC2 para Linux:
  Lazamos una instancia y por recomendación en clase seleccionamos Amazon Linux, apto para la capa gratuita T.2micro con autenticación asimétrica RSA, seleccionamos seminario VPC, en seguridad seleccionamos uno nuevo en la regla fijamos por el puerto 80 para conexión desde cualquier lugar seleccionamos tamaño en disco y lanzamos, SSH Linux solo se conecta por consola.

- Detalles de los Grupos de Seguridad (puertos abiertos: RDP, SSH, HTTP).
  En el grupo de seguridad Windows Servidor se habilitan los protocolos de conexión en los puertos 80, 3389 desde IP publica estudiante ICMP para pruebas de ping
  En el grupo de seguridad Servidor Linux se habilitan los protocolos de conexión en los puertos 80, 22 IMCP para pruebas de ping

- Asignación de IPs públicas y privadas.

La asignación de las IP por default

Servidor	Privada	Pública
Windows Servidor	10.0.12.31	52.14.249.155
Linux	10.0.8.236	3.133.143.37

Procedimiento de acceso

- Cómo acceder a cada servidor (cliente RDP para Windows, SSH para Linux).
  En el apartado de conexión de cada servidor se encuentra la información de conexión, ya habiendo generado la llave PEM descargada en el equipo local y habiendo configurado el remote desktop ingresamos con la IP pública y los datos generados en PEM por el puerto 3389 ya configurado. (ec2-user).
  Esta misma llave PEM nos sirve para la conexión a Linux en Mac desde la terminal o en Windows desde la aplicación mobaxterm por el puerto 22 con el usuario y contraseña indicados en el apartado de conexión (ec2-user)

- Consideraciones de seguridad (por ejemplo, uso de llaves PEM, contraseñas seguras).
  Para nuestro ejercicio y configuración de acceso a los servidores de forma segura SSH con una llave privada para la conexión a la llave publica en la instancia destino.

Configuración del servidor web

- Pasos seguidos para instalar IIS en Windows Servidor.

  Para la instalación de IIS, una vez que establecemos conexión con la instancia Windows Servidor, debemos abrir Inicio, y seleccionar "Servidor Manager", una vez dentro, en la pantalla principal, aparecerán diferentes opciones, debemos elegir la que dice, "Add roles and features", nos mostrara un cuadro donde debemos presionar siguiente 3 veces, dejando todo por defecto, hasta llegar a la opción de "Servidor Roles", entre las opciones que se muestran debemos activar la que dice, "Web Servidor (IIS), presionamos siguiente y posterior a eso se instalará. Para ejecutarla solo debemos Abrir inicio, y buscamos la carpeta llamada, "Windows Administrative Tools", dentro la carpeta estará el ejecutable, "Internet Information Service(IIS).

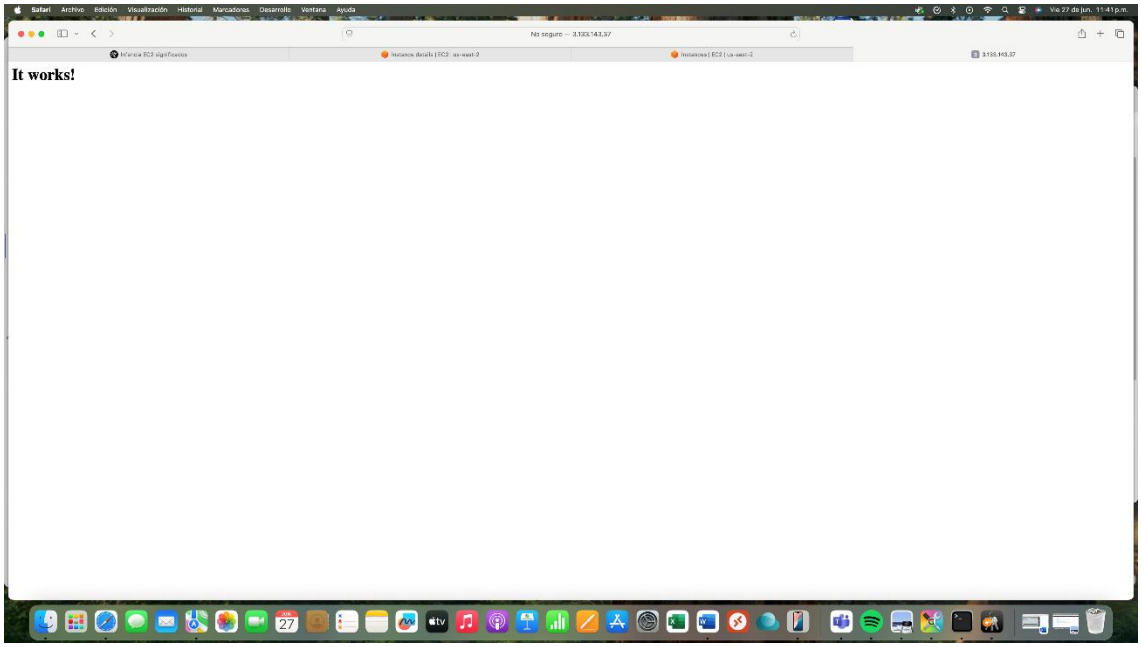
- Pasos para instalar Apache o Nginx en Linux.

  Ya conectado a Linux por consola configuramos el usuario administrador *sudo su #* y con el gestor de paquete el comando *dnf install httpd* para la validación *systemctl status httpd* si esta inactivo iniciamos s*ystemcl start httpd* y validamos nuevamente con el comando *systemctl status httpd* y debe estar activo (listening en puerto 80). Y al cargar IP pública en el navegador debe estar el mensaje **it works**

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

Linux

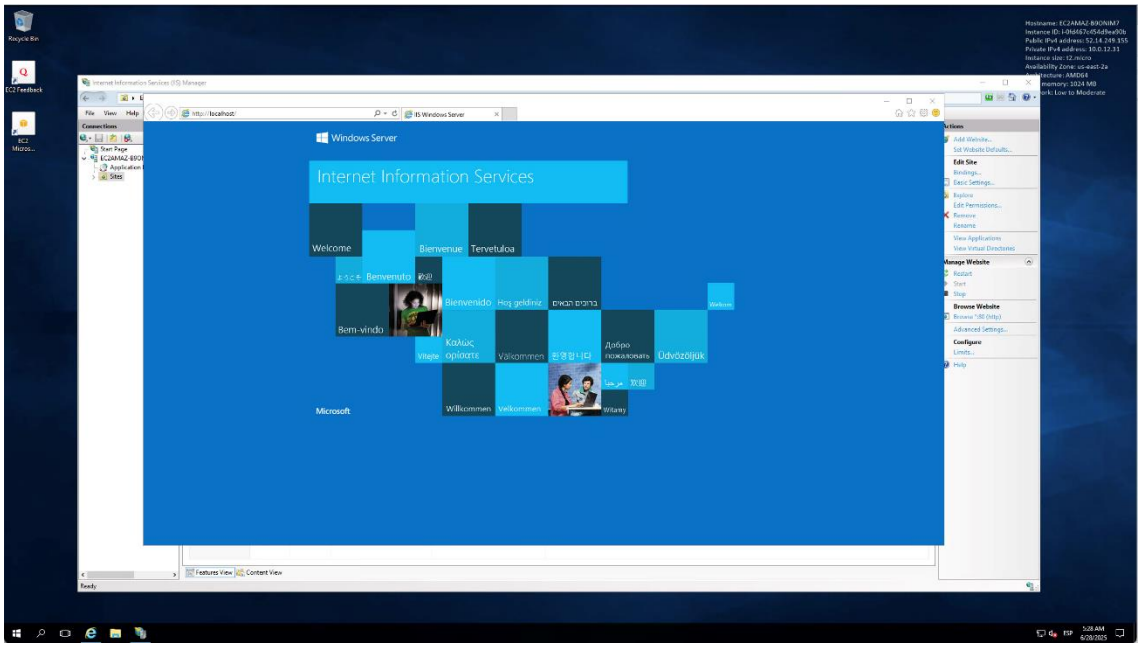
Figura 4. Página con Servidor Apache



Fuente: Elaboración propia.

Windows

Figura 5. Pagina con Servidor Windows



Fuente: Elaboración propia.

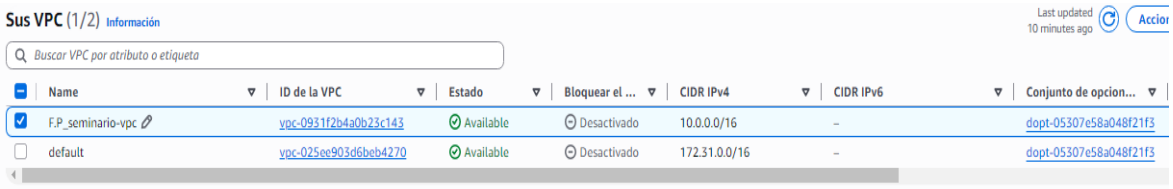
1.2 Implementación y pruebas

Actividades prácticas:

Crear la infraestructura

- VPC y subredes (pueden usar la default VPC para simplificar si es el primer proyecto).

Figura 6. Red VPC



Fuente: Elaboración propia.

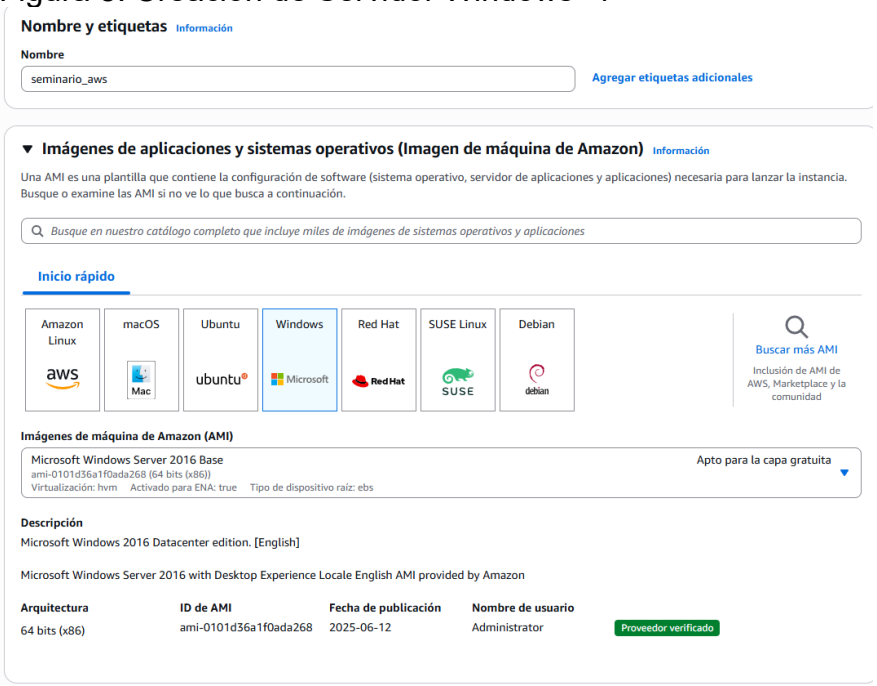
Figura 7. VPC y subredes

Fuente: Elaboración propia.

- Lanzar dos instancias EC2:

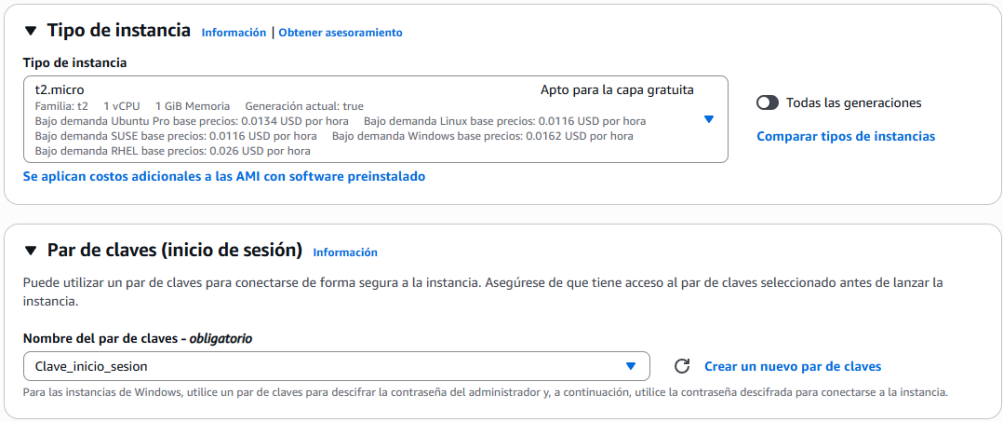
  Una instancia Windows Servidor.
  Para crear la instancia de Windows Servidor se realizó la siguiente configuración:

Figura 8. Creación de Servidor Windows -1



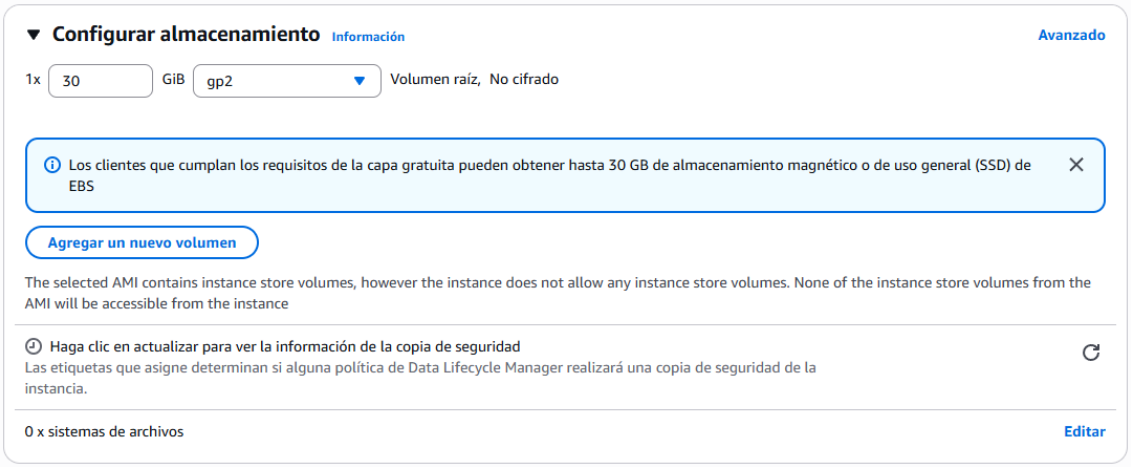
Fuente: Elaboración propia.

Figura 9. Creación de Servidor Windows -2



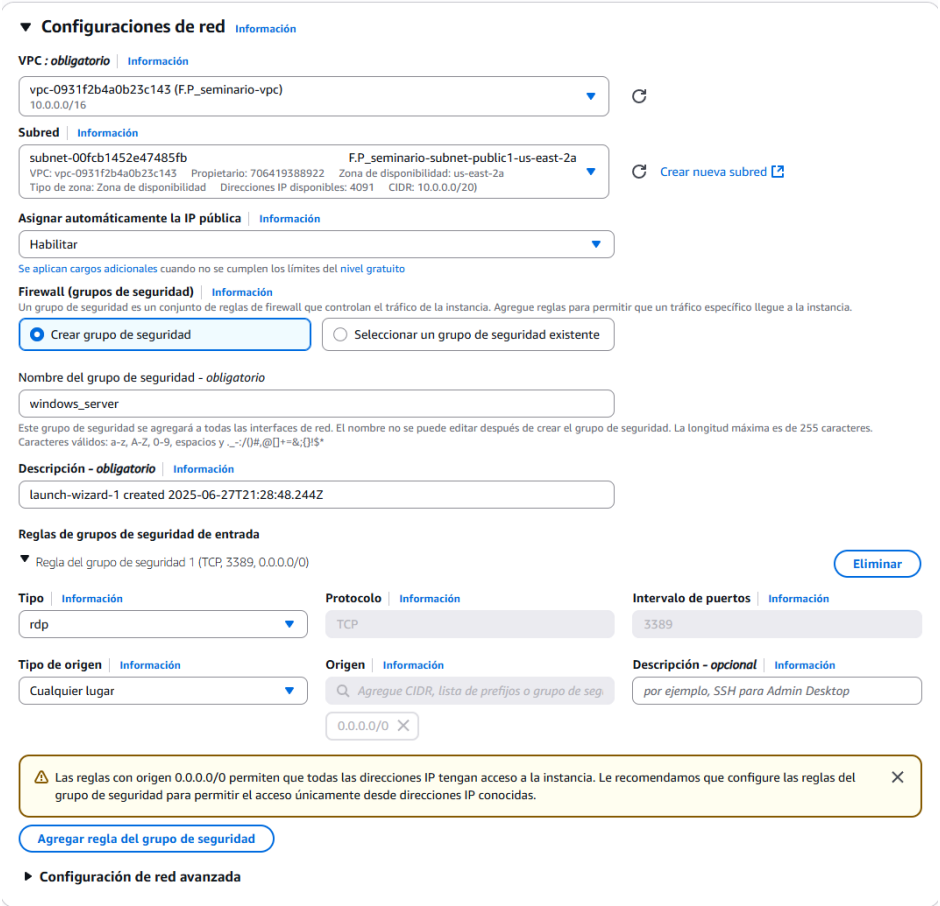
Fuente: Elaboración propia.

Figura 10. Creación de Servidor Windows -3



Fuente: Elaboración propia.

Figura 11. Creación de Servidor Windows -4

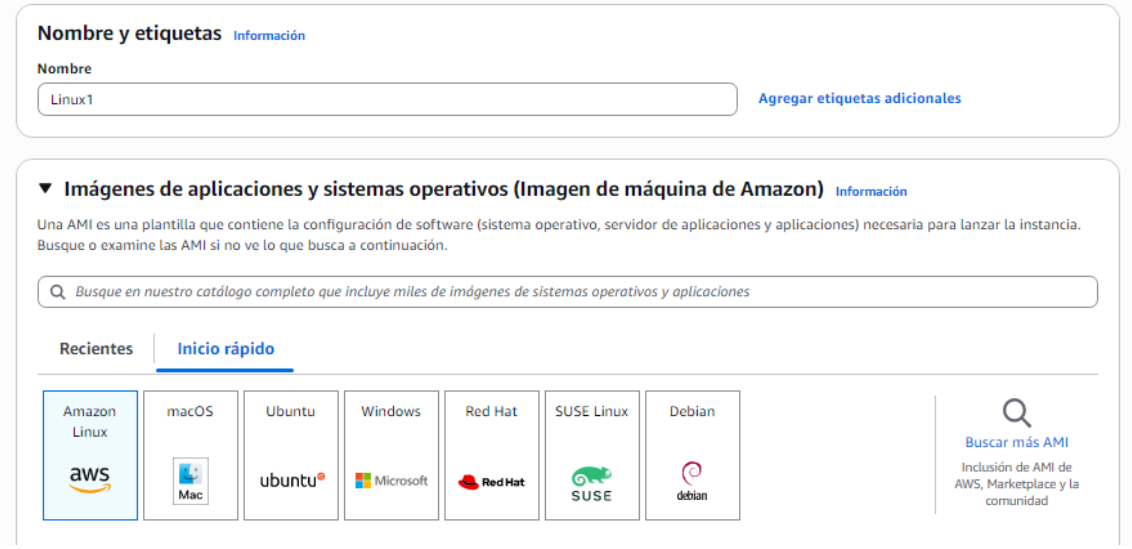


Fuente: Elaboración propia.

Una instancia Linux (por ejemplo, Ubuntu 22.04 o Amazon Linux 2):

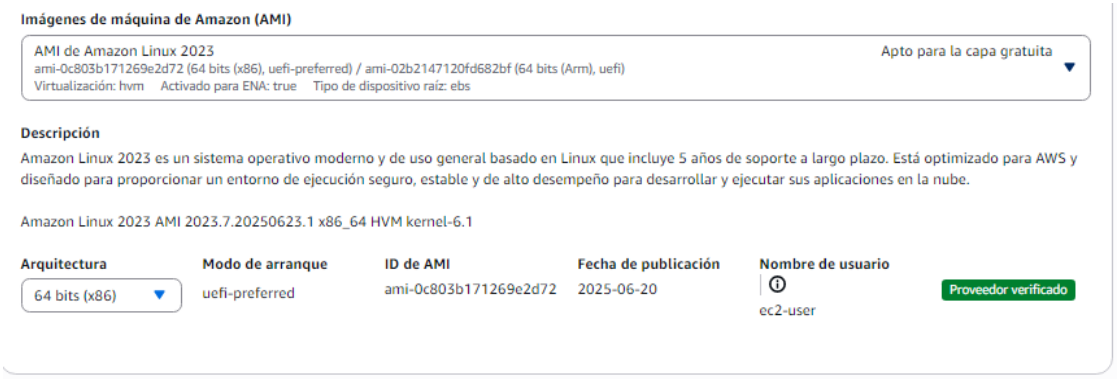
Para crear la instancia de Amazon Linux se realizó la siguiente configuración:

Figura 12. Creación de Servidor Linux -1



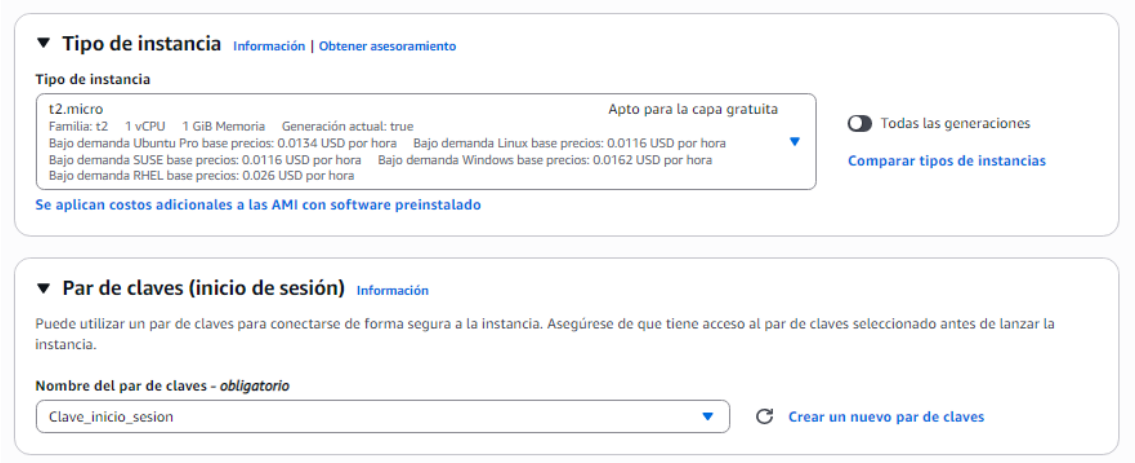
Fuente: Elaboración propia.

Figura 13. Creación de Servidor Linux -2



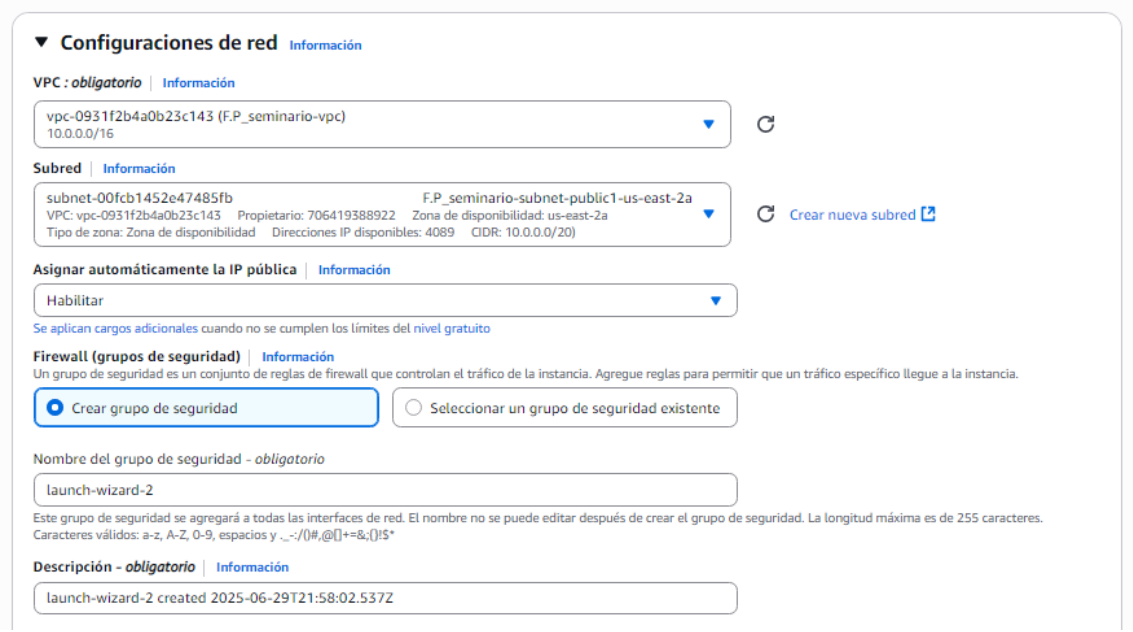
Fuente: Elaboración propia.

Figura 14. Creación de Servidor Linux -2



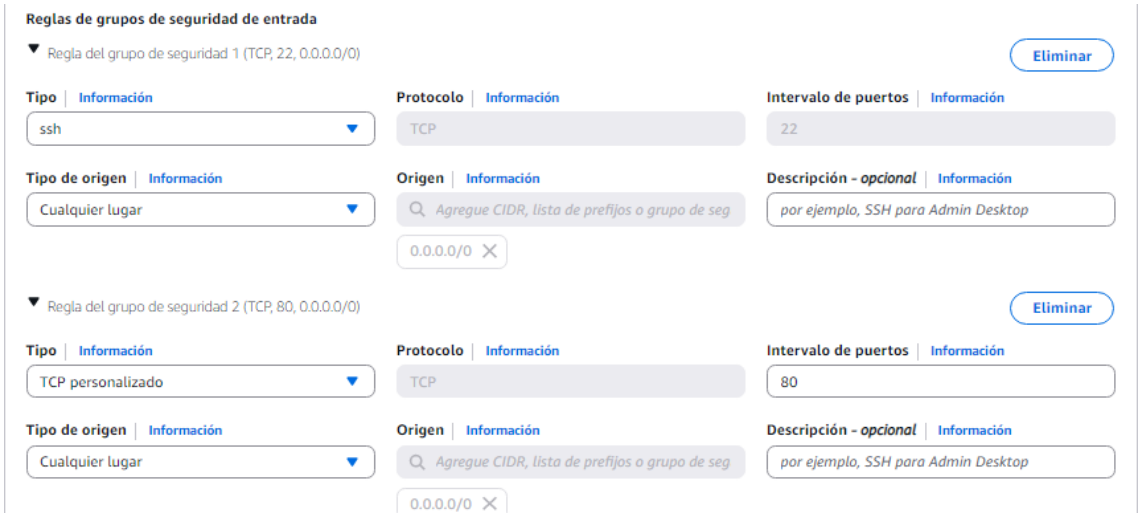
Fuente: Elaboración propia.

Figura 15. Creación de Servidor Linux -3



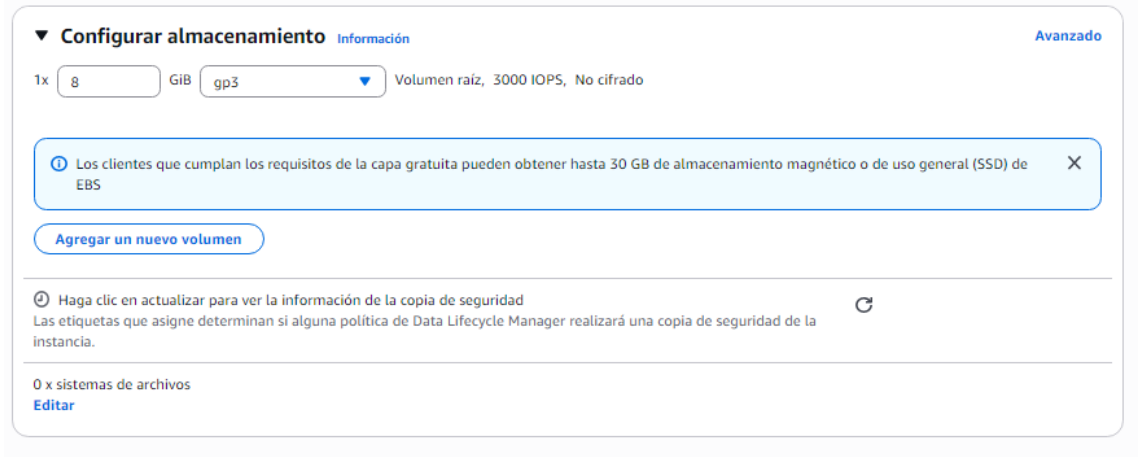
Fuente: Elaboración propia.

Figura 16. Creación de Servidor Linux –4



Fuente: Elaboración propia.

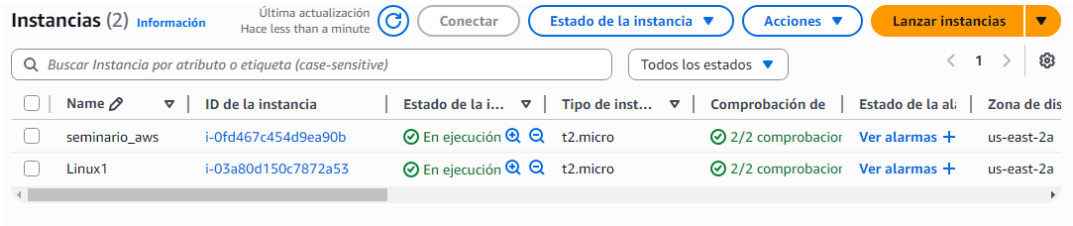
Figura 17. Creación de Servidor Linux -5



Fuente: Elaboración propia.

Se crearon las siguientes instancias:

Figura 18. Instancias de Windows y Linux



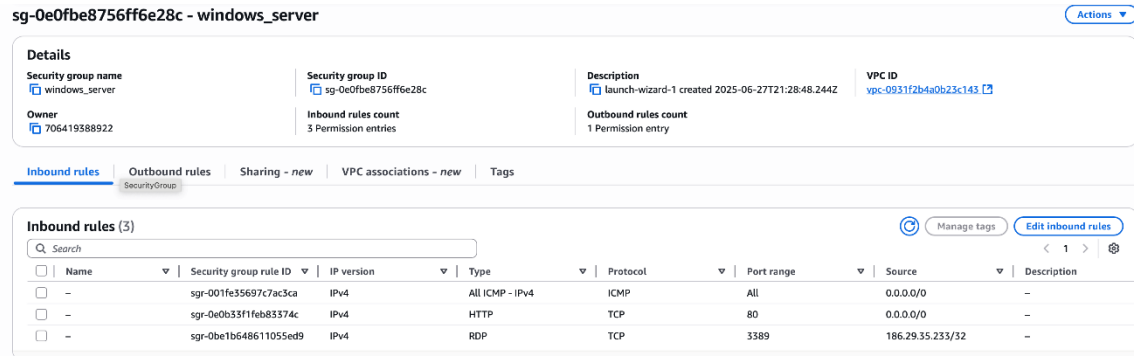
Fuente: Elaboración propia.

Configurar Grupos de Seguridad

- Permitir:

  - RDP (puerto 3389) para Windows desde la IP pública del alumno.
  - HTTP (puerto 80) abierto a todo Internet (0.0.0.0/0) para ambos.

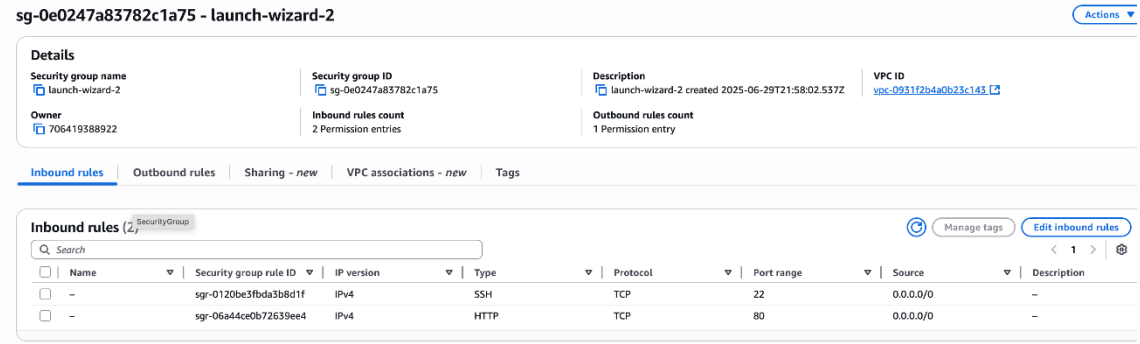
Figura 19. Grupo de Seguridad de Windows



Fuente: Elaboración propia.

  - SSH (puerto 22) para Linux desde la IP pública del alumno.

Figura 20. Grupo de Seguridad de Linux



Fuente: Elaboración propia.

Acceder a las instancias

- Acceder vía RDP a la instancia Windows.

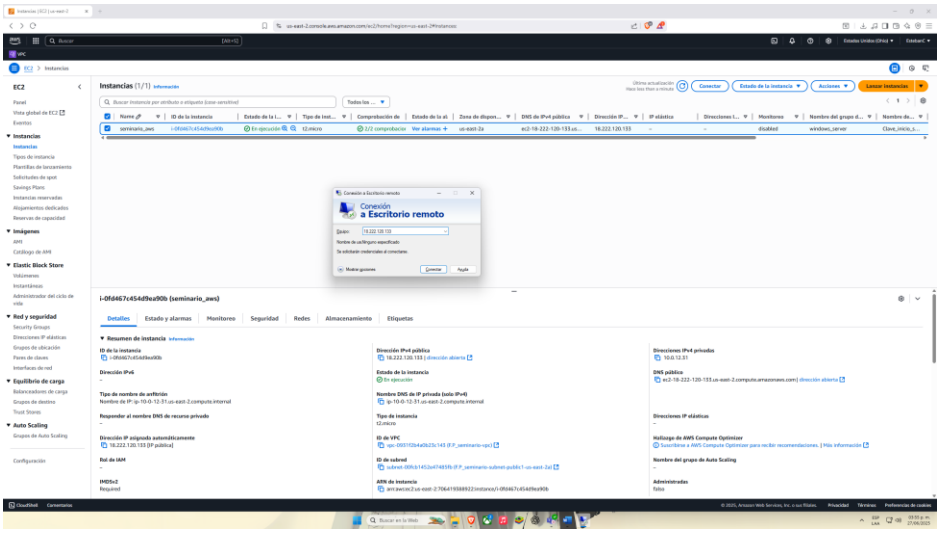
  Para acceder a la instancia de Windows se descifro la clave y posteriormente utilizando la dirección IPv4 52.14.249.155 y se ingresó a la instancia.

Figura 21. Conexión a Servidor de Windows

Fuente: Elaboración propia.

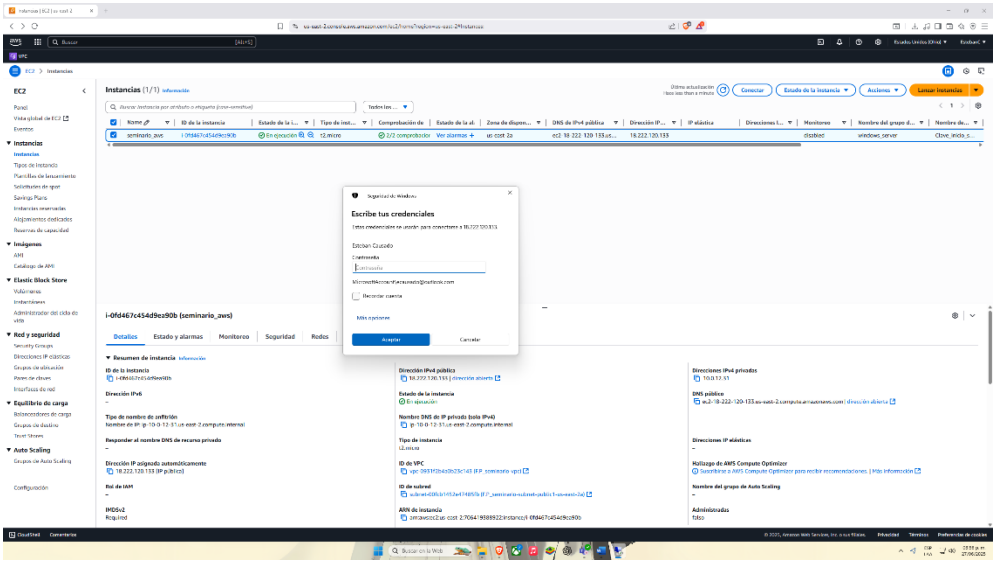
Con la llave PEM y configuración de protocolo RDP 3389 podemos acceder desde escritorio remoto

Figura 22. Conexión a escritorio remoto en Windows



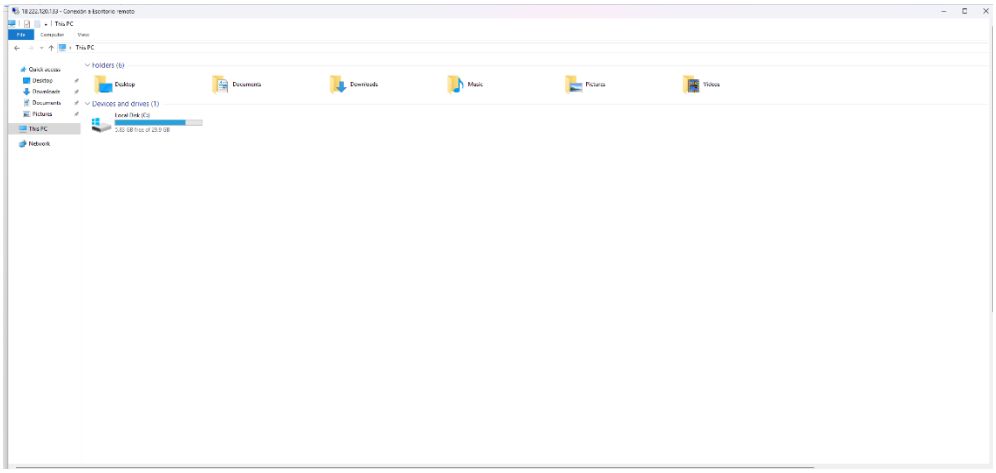
Fuente: Elaboración propia.

Figura 23. Ingreso a conexión remota



Fuente: Elaboración propia.

Figura 24. Ingreso al Servidor de Windows



Fuente: Elaboración propia.

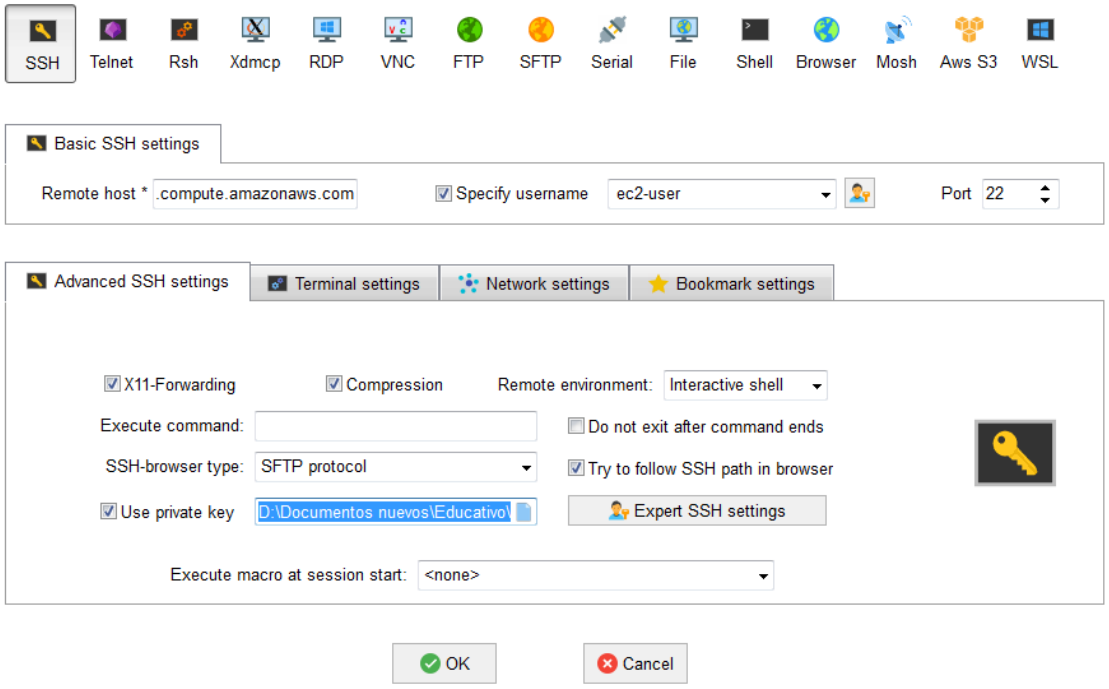
- Acceder vía SSH a la instancia Linux.

Figura 25. Conexión a Servidor de Linux

Fuente; Elaboración propia.

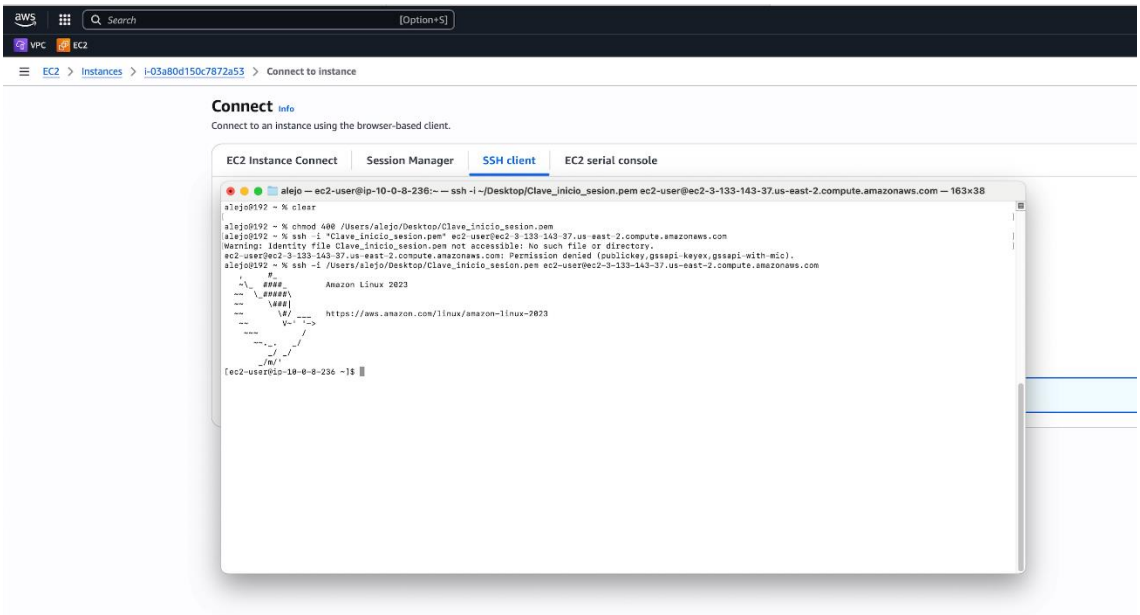
Con los datos de información en SSH puerto 22 colocamos la clave PEM que fue configurada para Windows nos sirve para el acceso a Linux, descargamos en el escritorio y desde una terminal Mac hacemos el ingreso ya habiendo configurado el puerto de conexión en la política de seguridad

Figura 26. Registro para ingresar a Servidor Linux con SSH



Elaboración propia.

Figura 27. Ingreso a Linux

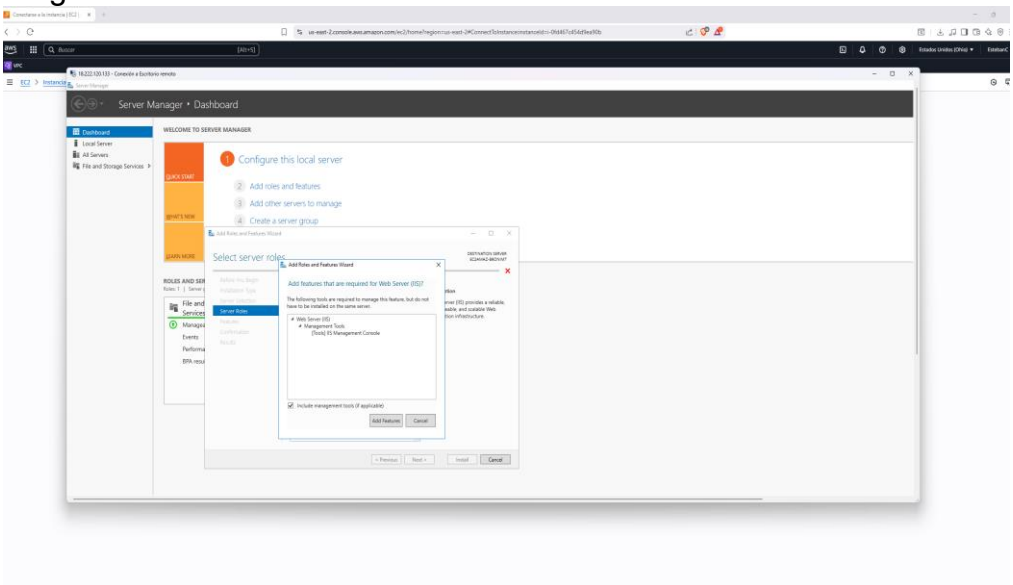


Fuente: Elaboración propia.

Instalar y configurar los servidores web

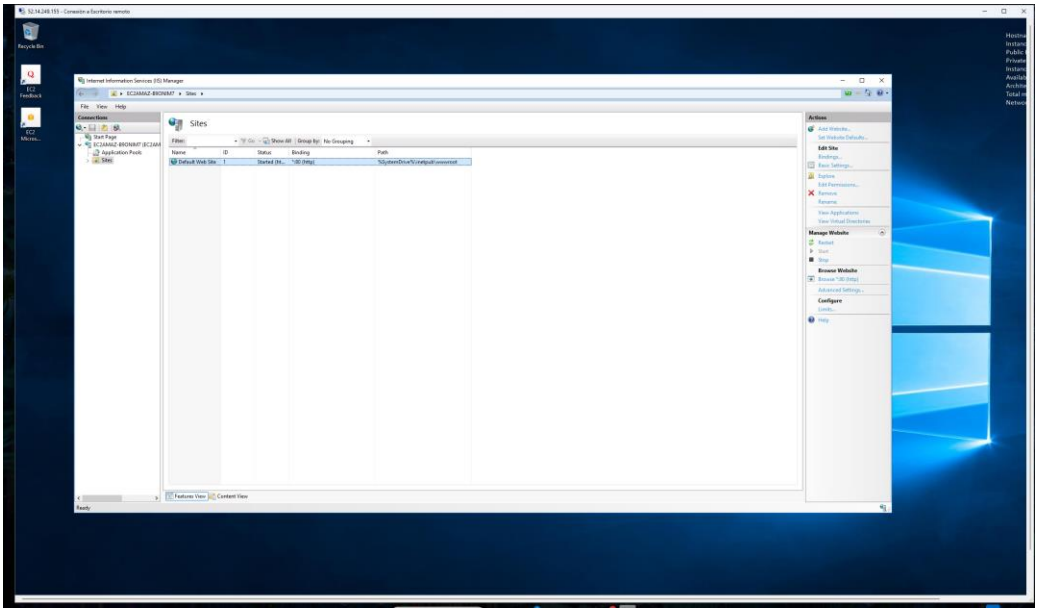
- **Windows**: Instalar el rol de IIS y levantar el sitio por defecto.

Figura 29. Instalación de Servidor Web



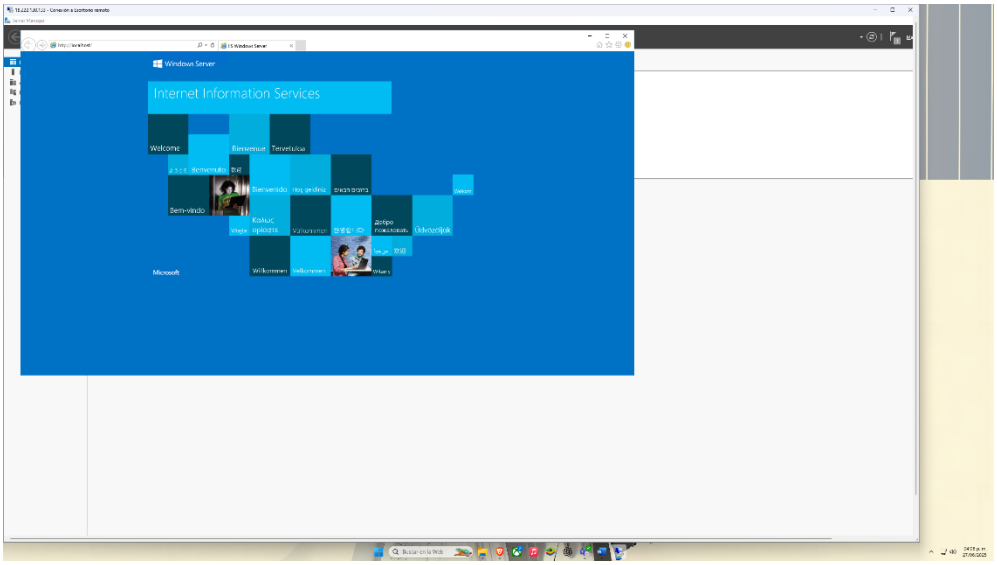
Fuente: Elaboración propia.

Figura 30. Acceso a Servidor Web ISS



Fuente: Elaboración propia.

Figura 31. Pagina inicio del Servidor Web



Fuente: Elaboración propia.

- **Linux**: Instalar Apache o Nginx y levantar el sitio por defecto.

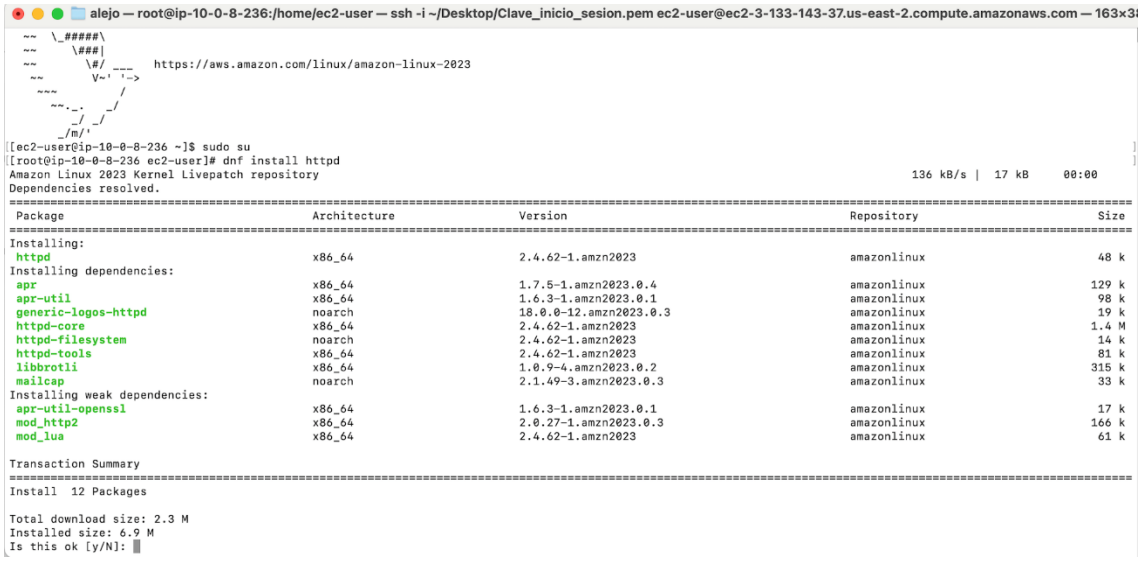
Con el acceso ya al servidor ingresamos en comando sudo su para entrar como administrador

Figura 32. Ingreso a Linux para instalar Servidor Web

Fuente: Elaboración propia.

Ingresamos el comando dnf install httpd – este es el apache

Figura 33. Instalación del Servidor Web



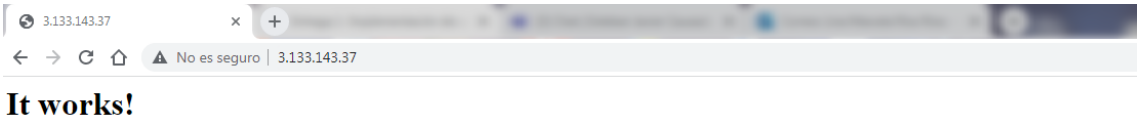
Fuente: Elaboración propia.

Figura 34. Servidor Web activo

Fuente: Elaboración propia.

Y verificamos que quede en escucha por el puerto 80

Figura 35. Página inicio Servidor Web



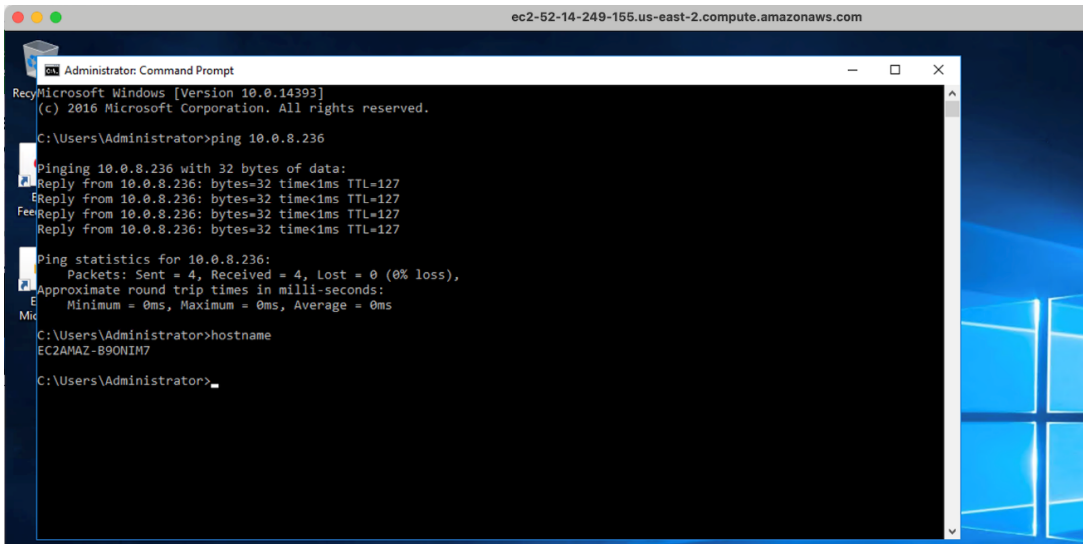
Fuente: Elaboración propia.

Pruebas de conectividad

- Desde la instancia Windows hacer *ping* a la IP privada de la instancia Linux y viceversa.

  Ping IP privada desde Windows a Linux

Figura 36. Prueba desde la instancia de Windows



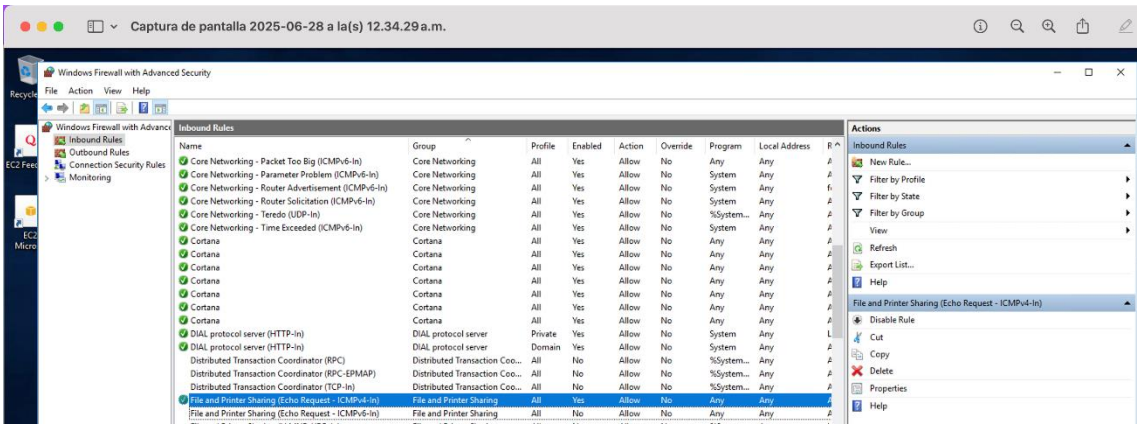
Fuente: Elaboración propia.
Figura 37. Prueba desde Linux a Windows

Fuente: Elaboración propia.

- Documentar si hay necesidad de habilitar ICMP en los Grupos de Seguridad para permitir ping.

Por defecto el protocolo en el Servidor Windows se encuentra deshabilitado por lo que se activa desde el firewall del servidor ICMPV4

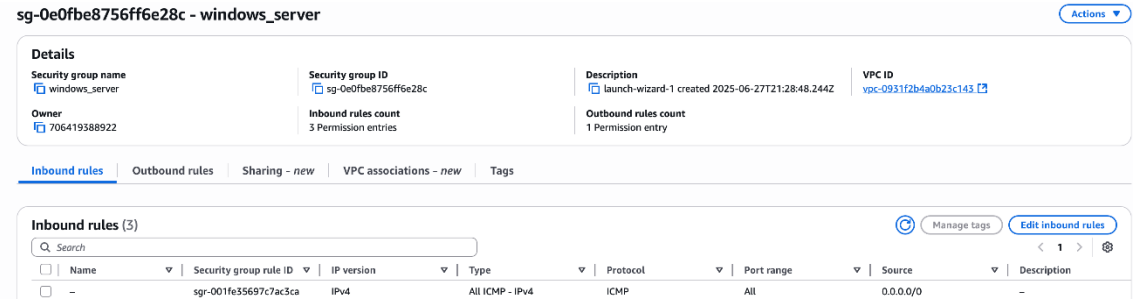
Figura 38 Configuración Protocolo ICMP Windows



Fuente: Elaboración propia.

Y se ingresa la regla ICMP en la política de seguridad Servidor Windows

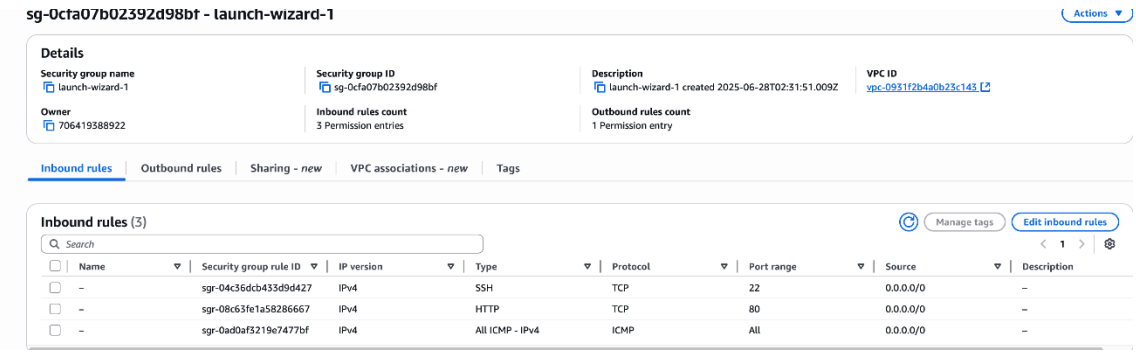
Figura 39. Ingreso a la configuración de ICMP en Windows



Fuente: Elaboración propia.

De igual forma se ingresa la regla para el Servidor Linux

Figura 40. Ingreso a la configuración de ICMP Linux

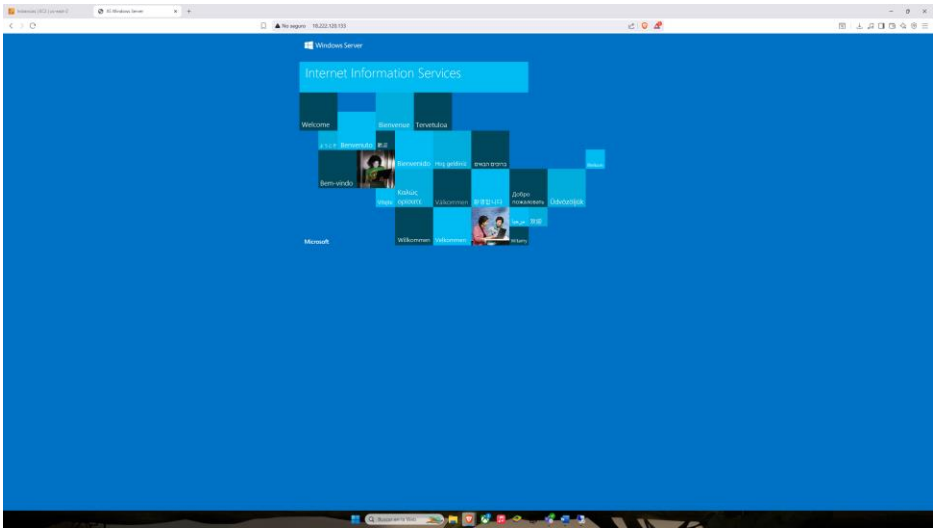


Fuente: Elaboración propia.

Validación de acceso web

- Acceder desde el navegador local al sitio web de la instancia Windows (http://<IP_Pública_Windows>).

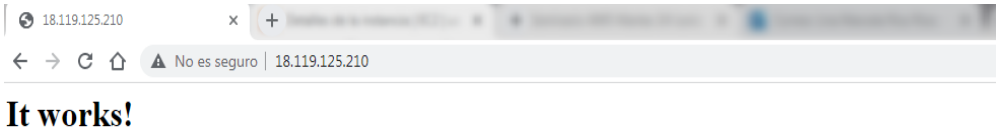
Figura 41. Página de inicio de Windows



Fuente: Elaboración propia.

- Acceder desde el navegador local al sitio web de la instancia Linux (http://<IP_Pública_Linux>).

Figura 42. Página de Inicio de Linux



Fuente: Elaboración propia.

2. Implementación de Arquitectura en AWS con Balanceador de Carga y Contenedores

Introducción: La startup " Dimorphos"

Bienvenidos al proyecto de implementación. En esta tarea, ustedes asumirán el rol de arquitectos en la nube para una startup llamada **Dimorphos**, 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.
El CTO de **Dimorphos** ha diseñado una arquitectura preliminar y necesita de su ayuda para implementarla en **Amazon Web Services (AWS)**. La solución debe ser altamente disponible, escalable y estar diseñada para manejar una gran cantidad de tráfico de manera eficiente.

Objetivo del Trabajo

Implementar una arquitectura en AWS que cumpla con los siguientes requisitos:

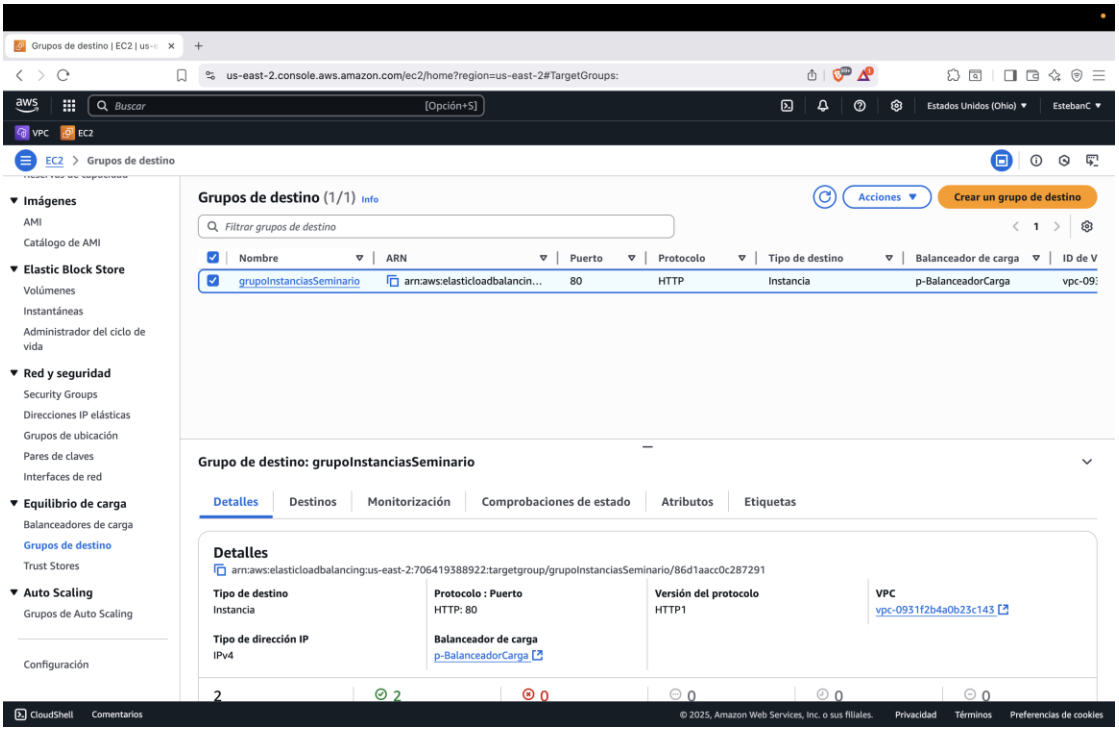
2.1 Balanceador de Carga

Configure un Application Load Balancer (ALB) para distribuir el tráfico entrante a múltiples instancias EC2.

Para la creación de un balanceador de carga es necesario crearlo desde cero, lo primero es que este expuesto a internet, esto nos permite que los usuarios puedan acceder y lo usen como un puente hacia la página de destino, para el mapeo de red se usa el creado con anterioridad, donde tiene, dos redes públicas y dos privadas, para las zonas de disponibilidad se seleccionan dos públicas, en el grupo de seguridad se crea uno que en este caso nos habilite el puerto 80, con esto la primera parte estaría completa.

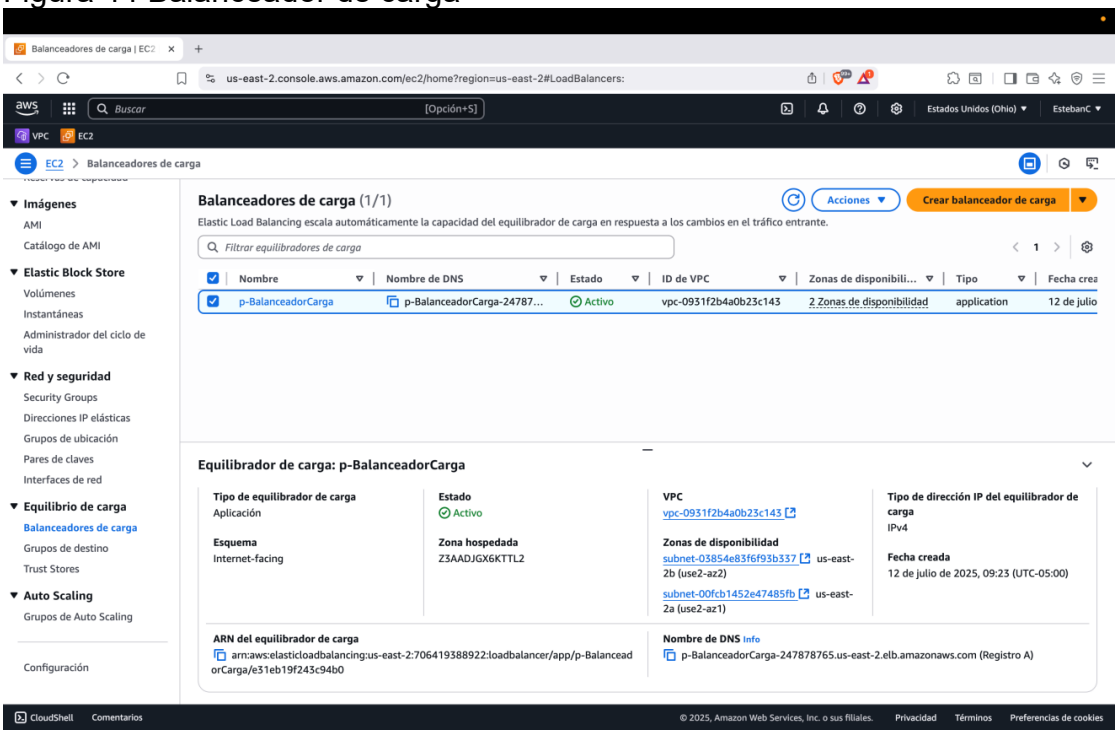
Luego de crear el balanceador, debemos crear un grupo de destino, es donde va a ir alojado el servidor o página que se desea aplicar, para este caso, es necesario crear al menos 2 instancias para que el balanceador de carga pueda enviar a un lado dependiendo la carga de cada uno, se registran las instancias creadas previamente y nos aparecerán como disponible si no hay ningún error.

Figura 43 Grupo de Destinos



Fuente: Elaboración Propia

Figura 44 Balanceador de carga



Fuente: Elaboración propia

Figura 45 Prueba con balanceador de carga en Linux

Fuente: Elaboración propia

2.2 Instancias EC2

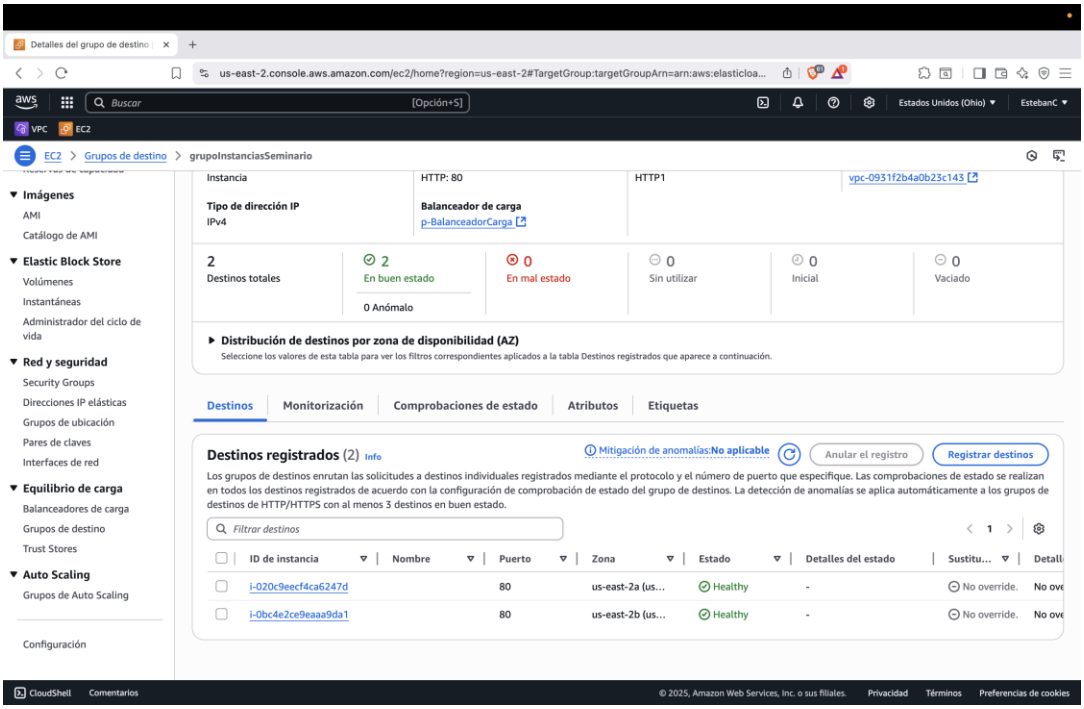
Implemente al menos dos instancias EC2 en una configuración multizona para garantizar alta disponibilidad.

Se configuro el auto escalado para que creara las instancias de forma automática, a partir de una plantilla previamente definida, cada instancia tiene las siguientes características:

- Amazon Linux
- 8gb de disco duro
- t2.micro
- Grupo de seguridad, puertos 80,22

Además, se configuro para que las instancias se crearan en zonas diferentes mejorar la disponibilidad.

Figura 46 Instancias registradas



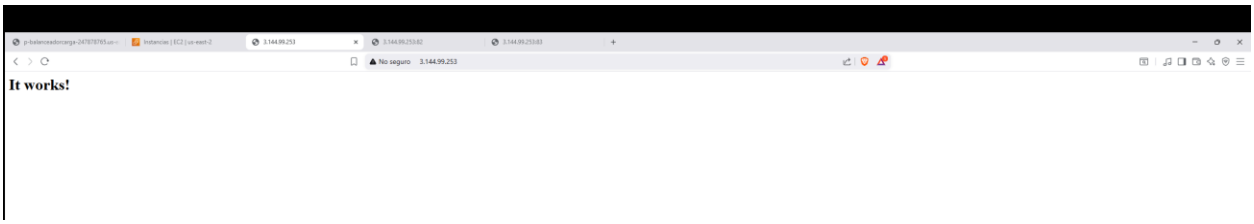
Fuente: Elaboración propia

2.3 Instancias con Proxy Reverso

Dentro de cada instancia EC2, deben implementar un **proxy reverso** (por ejemplo, Nginx) para redirigir solicitudes a servicios internos.

Configuramos el archivo nginx.conf, lo que hará es redireccionar lo que llega al puerto que tengamos, en este caso el 82.83.84 hacia http://backend;, esto nos debe funcionar como el balanceador de carga, para que solo con la dirección IP 3.144.99.253 sin necesidad de especificar si es puerto :82,83,84, podamos acceder a cada uno de los contenedores, donde se encuentra la página o servicio

Figura 47 Pagina web redireccionada desde un proxy

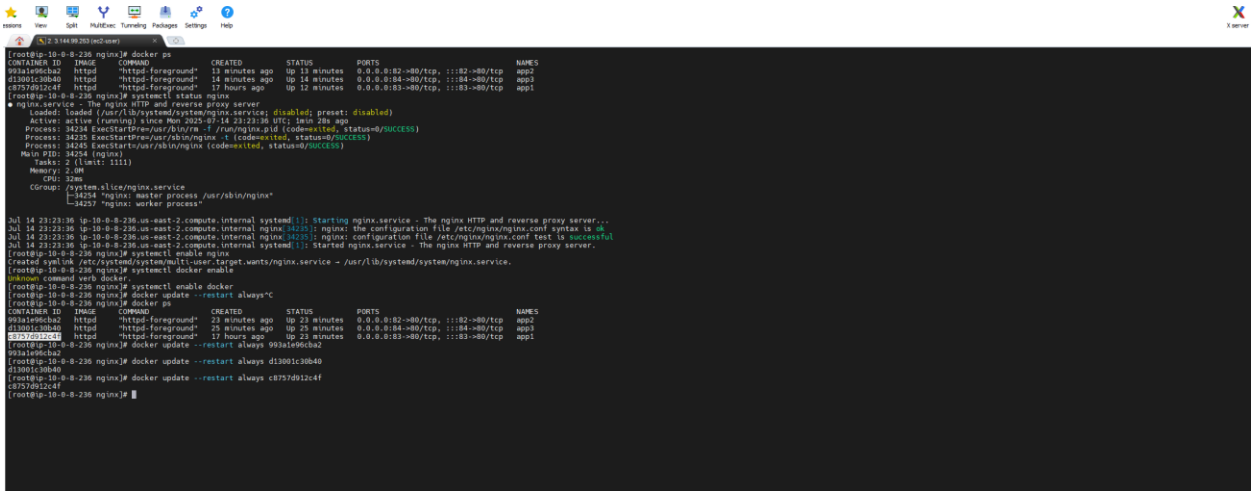


Fuente: Elaboración propia

Eso es la primera parte, para luego hacer que tengamos diferentes instancias con diferentes contenedores, es necesario, configurar nginx y Docker para que inicien

automáticamente una vez se inicie la instancia, además, configurar los contenedores para que inicien al igual que Docker

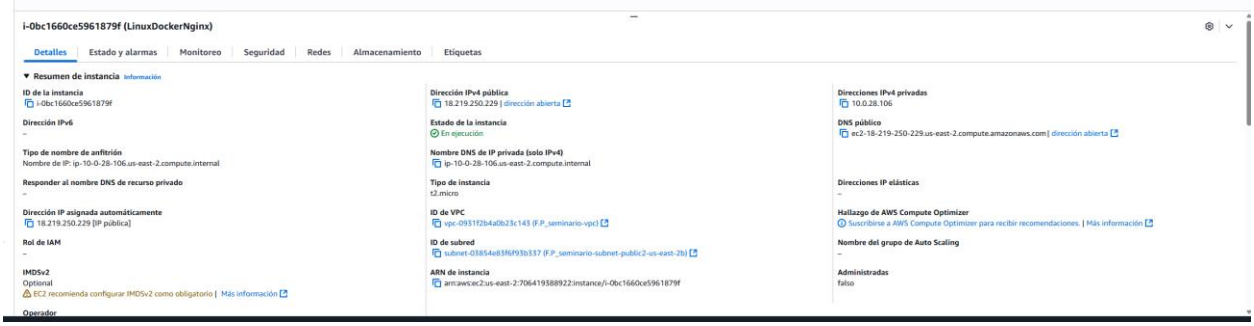
Figura 48 Configuración NGINX



Fuente: Elaboración propia

Creamos una nueva instancia con la AMI creada, donde se encuentra toda la configuración.

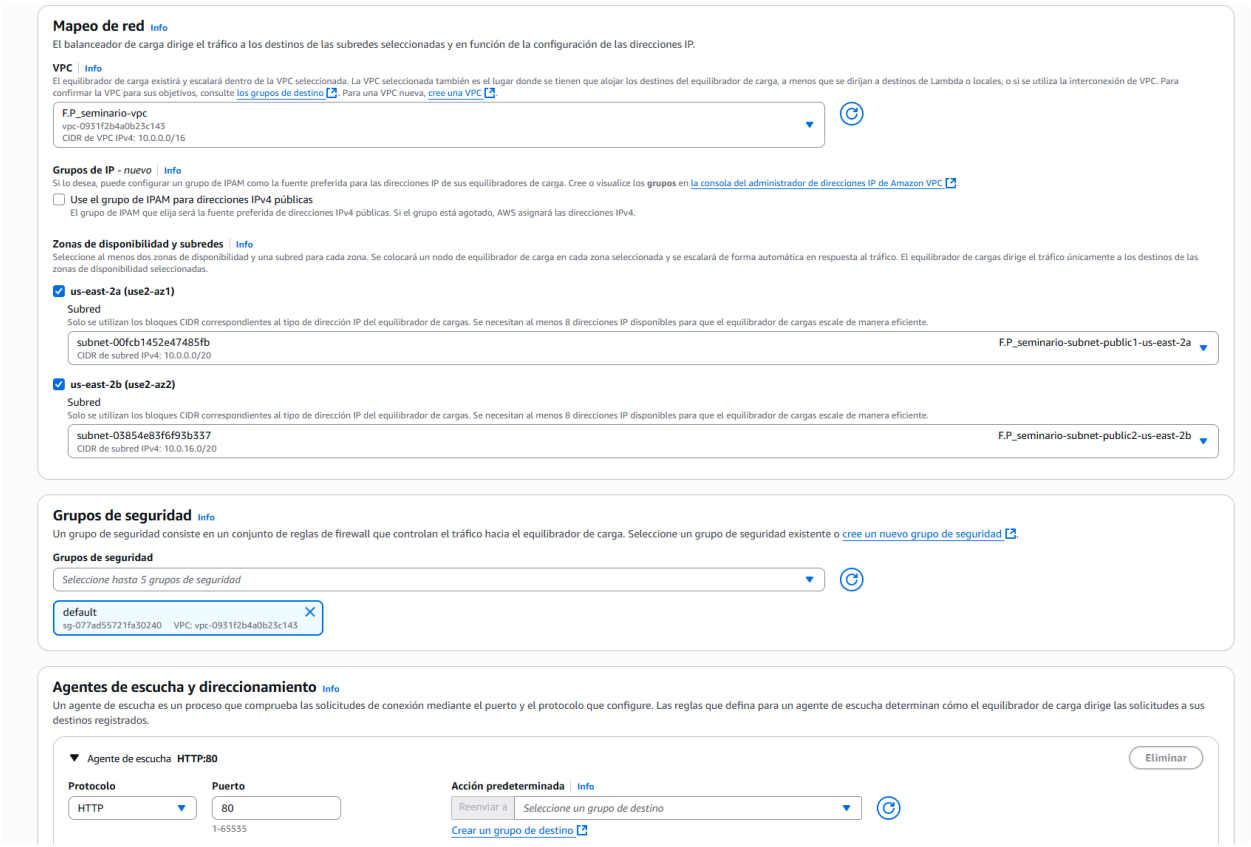
Figura 49 AMI configurada con proxy



Fuente: Elaboración propia

Luego creamos un nuevo balanceador de carga de aplicaciones, configurando 2 zonas de disponibilidad

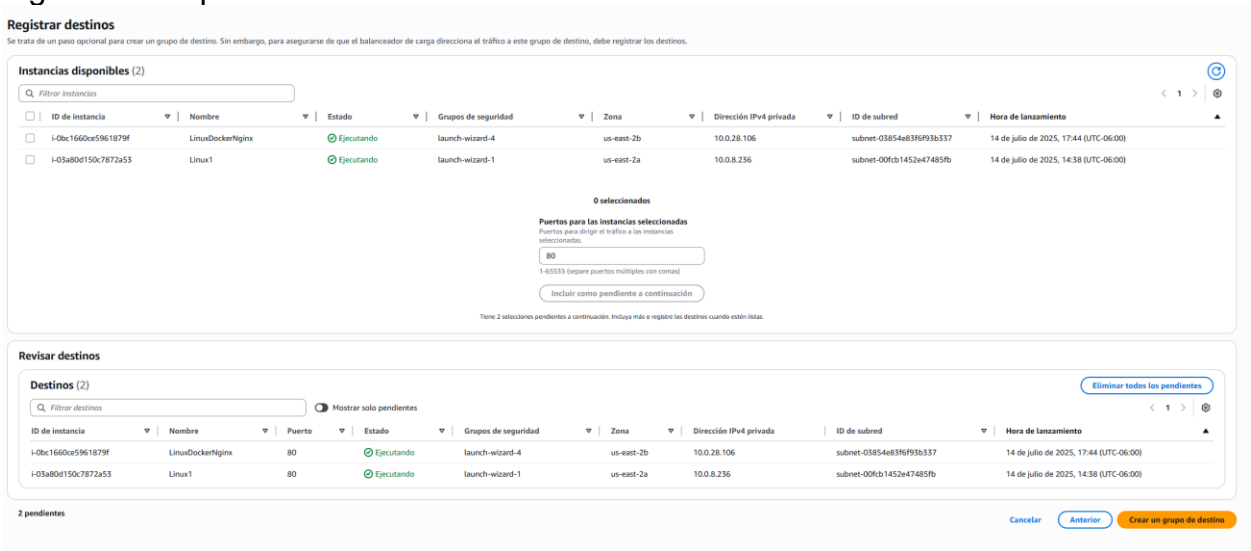
Figura 50 Balanceador de Carga con instancias con proxy



Fuente: Elaboración propia

Se crea un nuevo grupo de destino de instancias

Figura 51 Grupo de destino con las nuevas instancias



Fuente: Elaboración propia

Con la configuración aplicada quedaría representado de la siguiente manera

Fuente: Elaboración propia

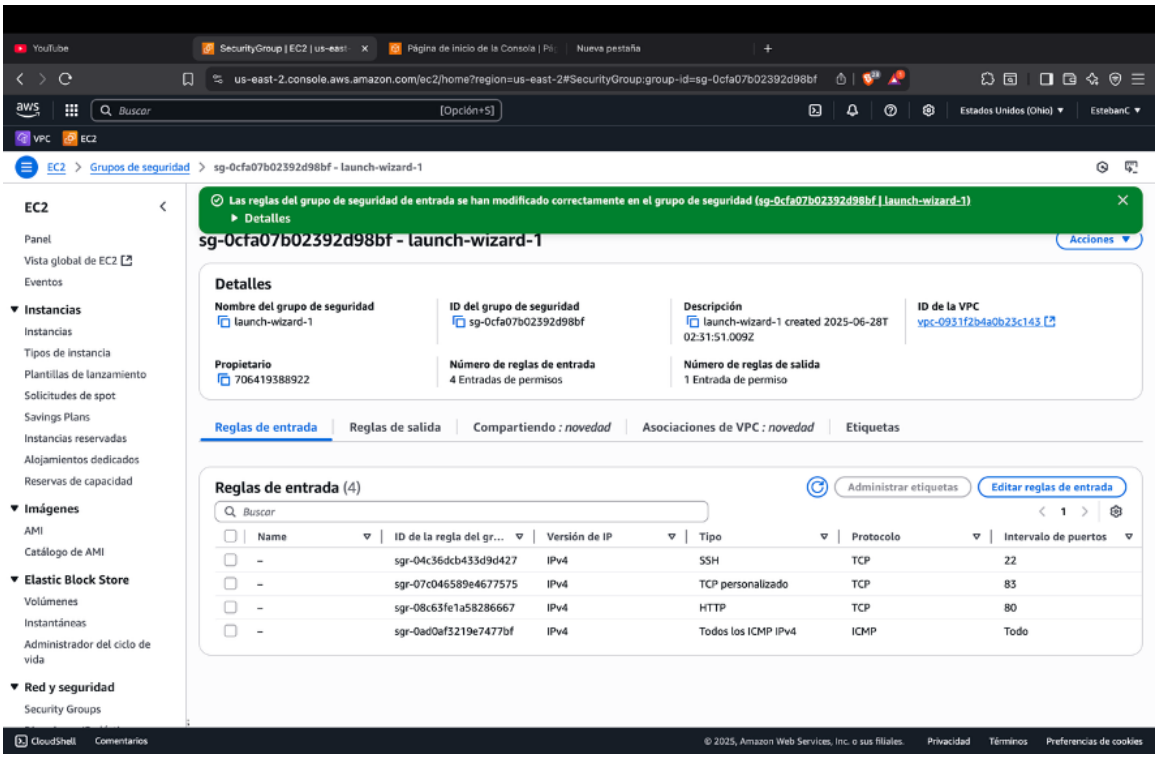
2.4 Implemente el servicio de Docker de forma manual, con una aplicación de prueba.

Para implementar el servicio de Docker es necesario instalarlo, con el comando (yum install Docker), se verifica si se instaló correctamente para posteriormente ejecutarlo, usamos el comando (systemctl enable Docker) para que se inicie automáticamente cuando arranque el sistema.

Buscamos en el repositorio de Docker hub httpd, este es el Apache Web Server, está la aplicación que se instalará dentro de los contenedores que posteriormente se definirán.

Para crear nuestro primer contenedor es necesario el comando (Docker run -dit –name app1 -p 81:80 httpd, app1 representa el nombre del contenedor, 81:80, 81 equivale al puerto de la instancia, que debemos editar reglas de seguridad para habilitar ese puerto.

Figura 52 Configuración grupos de seguridad



Fuente: Elaboración propia

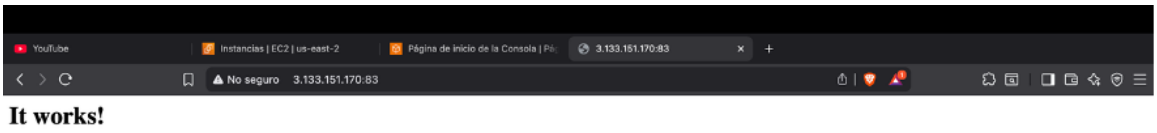
80 es el puerto que se encuentra dentro del contenedor.

Figura 53 puerto de contenedor

Fuente: Elaboración propia

Una vez creado, usamos el comando (docker ps -a) para mostrar todos los contenedores que existen, -a nos mostrara los contenedores creados pero que no están activos, una vez todo está listo, comprobamos el funcionamiento del servidor, con la ip 3.133.151.170:83, en este caso usamos :83 para ejecutar el servidor desde el puerto 83 comprobando el correcto funcionamiento, ya que, si solo colocamos la IP, nos redireccionará al puerto 80 por defecto.

Figura 54 ingreso al contenedor puerto 83



Fuente: Elaboración propia

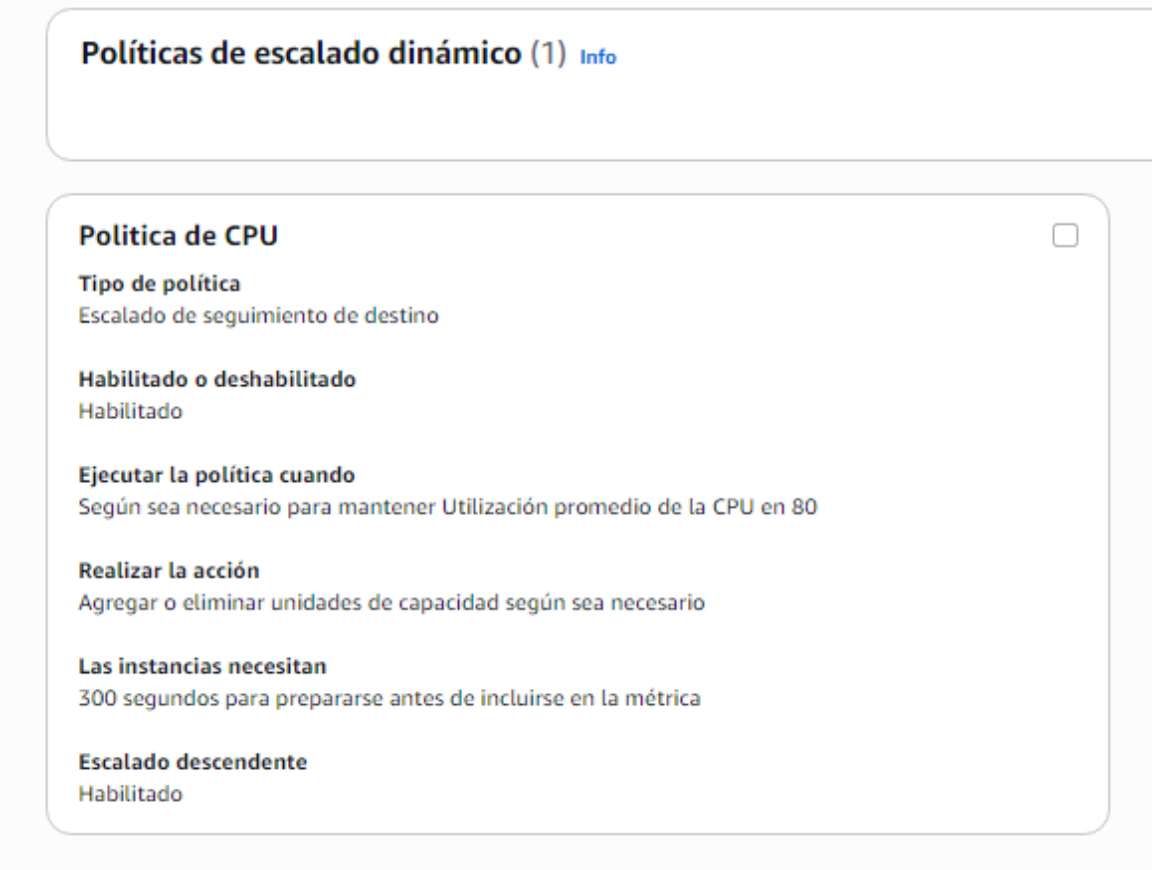
2.5 Autoescalado

Configure políticas de autoescalado para aumentar o reducir las instancias EC2 según la carga.

Si queremos garantizar que la aplicación siga funcionando sin ninguna interrupción si hay mucha demanda o algún fallo, se configura el servicio de auto escalado que incluye AWS. Para eso, se prepara una instancia en EC2, que se encuentre funcionando correctamente y que ya tenga las configuraciones necesarias, para crear una instantánea en EBS y generar una imagen que se utilizará en la plantilla de lanzamiento.

Se creo un grupo de auto escaldo con esa plantilla donde se establecieron límites de mínimo 1 instancias máximo 5 instancias, con esta configuración el sistema puede elegir crear o eliminar instancias de acuerdo con sus necesidades, todo esto sin ayuda externa.
Se estableció una política de escalado dinámico donde cuando se alcance un uso promedio de CPU del 80% se crean nuevas instancias para aumentar la capacidad.

Figura 55 Política de escalado dinámico



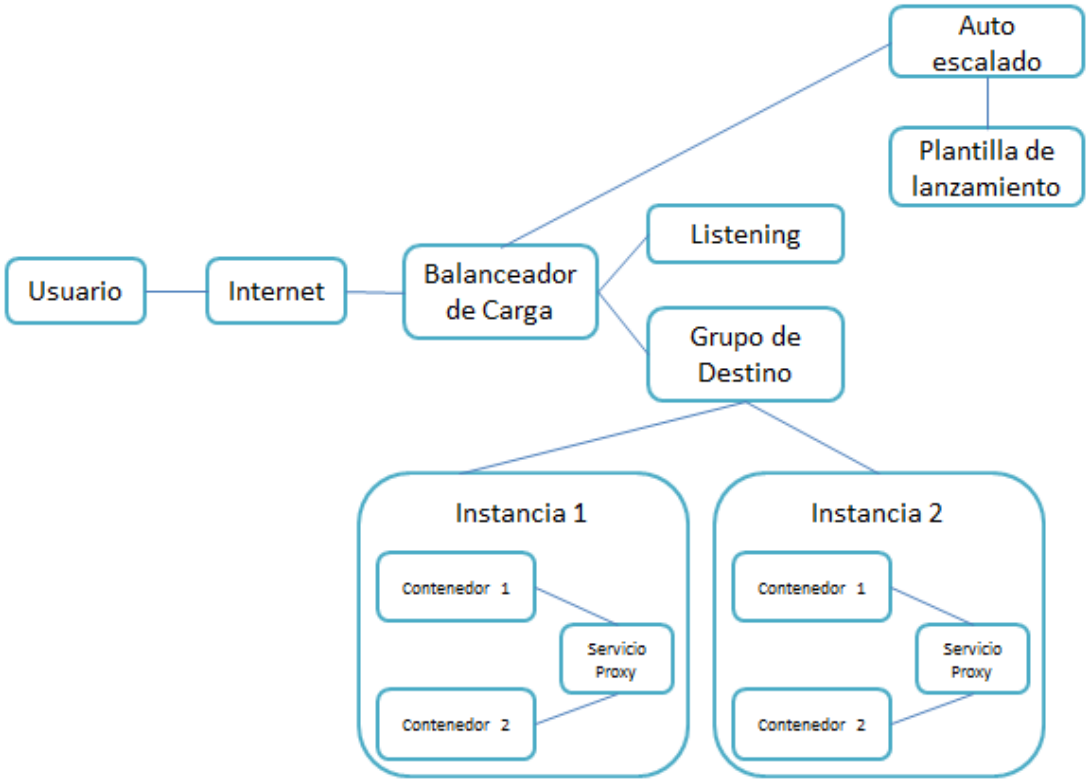
Fuente: Elaboración propia

2.6 Documentación

Toda la documentación debe estar en el formato de trabajo de grado proporcionado.

a. **Incluya un diagrama de los recursos usados y cómo se comunican entre ellos**.

Figura 56 Diagrama de la red creada



Fuente: Elaboración propia

Conclusiones

Uno de los objetivos principales para nuestro practica de nuestro seminario es la implementación y configuración de servicio web con instancias EC2 de alta disponibilidad para una startup, desarrollamos nuestra implementación en la plataforma AWS Amazon de computación en la nube, de manera fácil con todos los temas vistos en clases donde se contaba con amplia información detallada del paso a paso para la configuración de los diferentes servicios que hacen parte de los servicios dispuestos para el cumplimiento, del objetivo planteado.

Una de las ventajas obtenidas es poder contar con los servicios dispuestos por la plataforma de AWS, con un mínimo de conocimientos para un ingeniero de sistemas comprendiendo diferentes temas del área de la tecnología como es la configuración de servidores, instancias, protocolos de comunicación, servicios web, políticas de seguridad entre otros. Se logro indagar que grandes compañías sus servicios hacen uso de la plataforma AWS, como lo son JP Morgan, Amazon.com, Netflix, Airbnb entre muchas más.

En nuestras pruebas de conectividad, autoescalado y alta disponibilidad, los servicios respondieron de forma exitosa y tiempos de respuesta bastante aceptables, por lo que podemos concluir que está es una de las plataformas más seguras, confiables y con una robustes significativa, para la Implementación de startups, unicornios, o empresas con solidez y trayectoria en los diferentes mercados pequeños, medianos o grandes.

La plataforma AWS Amazon puede soportar servicios de gran concurrencia y alta disponibilidad para la Startup **Dimorphos** en las prestaciones de los restaurantes que podían ser localizados en diferentes áreas o regiones soportados por los cluster; configurados en los diferentes datacenter y con conexión de múltiples usuarios(clientes) por medios de los puertos de conexión a los servicios dispuestos, que son configurados en las políticas de seguridad de red y puede contar con el amplio crecimiento y/o delimitación de las instancias dispuestas en la configuraciones de auto scalling.

Referencias

Amazon Web Services [AWS].(2025). ¿Qué es Amazon EC2? https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/concepts.html

Apache (s.f.). The Apache HTTP Server Project. The Apache Http

https://httpd.apache.org/

CloudFlare.(s.f.).¿Qué es el Protocolo de control de mensajes de Internet (ICMP)? https://www.cloudflare.com/es-es/learning/ddos/glossary/internet-control-message-protocol-icmp/

CloudFlare (s.f.). ¿Qué es el protocolo Secure Shell (SSH)? https://www.cloudflare.com/es-es/learning/access-management/what-is-ssh/

Espinosa, O. (2025). Qué son los puertos TCP y UDP y para qué sirven cada uno de ellos. RedesZone. https://www.redeszone.net/tutoriales/configuracion-puertos/puertos-tcp-udp/

Giménez, Mónica (2020). Amazon Web Services (AWS): ¿qué es y qué ofrece? https://www.hiberus.com/crecemos-contigo/amazon-web-services-aws-que-es-y-que-ofrece/

IBM (s.f.) .Tivoli Network Manager IP Edition. https://www.ibm.com/docs/es/networkmanager/4.2.0?topic=ports-defining-fixed-tcp-port

Microsoft (s.f.). Cómo usar Escritorio remoto - Soporte técnico de Microsoft. https://support.microsoft.com/es-es/windows/c%C3%B3mo-usar-escritorio-remoto-5fe128d5-8fb1-7a23-3b8a-41e636865e8c

Zcaler (2025) ¿Qué es un servidor proxy? https://www.zscaler.com/es/zpedia/what-is-a-proxy-server

Zhang, Q., Cheng, L., & Boutaba, R. (s. f.). Cloud computing: state-of-the-art and research challenges. Journal Of Internet Services And Applications. https://doi.org/10.1007/s13174-010-0007-6