TRABAJO DE GRADO
Opción Seminario-Diplomado.

Implementación de arquitectura en la nube con los servicios de AWS para FoodTecno

Corporación Universitaria Remington.
Facultad de Ingenierías
Ingeniería de Sistemas

Jean Carlos Chaverra Caicedo
Jhonier Jose Garzón Romaña
Santiago Borja Charris
Tutor del trabajo de grado
Juan Pablo Berrio López
Trabajo de grado - Seminario
2025

Dedicatoria

A Dios porque sin el nada de esto fuera posible, a mis compañeros de grupo y a nuestras familias, por el apoyo constante durante este proceso de formación profesional.

Tabla de contenido

Resumen

Este trabajo detalla el diseño e implementación de una arquitectura escalable y altamente disponible en la nube, a través de los servicios de Amazon Web Services (AWS), para FoodTecno, un startup que busca posicionarse como pionera en la industria alimentaria mediante la integración se soluciones tecnológicas; conectando a restaurantes con clientes mediante entregas en menor tiempo. Haciendo uso de los principales servicios de AWS como VPC, Amazon EC2, Load Balancer, Auto Scaling y Contenedores Docker.

Inicialmente se realizó la configuración de una red por medio de una VPC las cuales asigna subredes públicas y privadas, que permiten alojar las instancias EC2, en las que se implementaron servidores web con Apache Linux 2023 y Windows Server 2016 Base. Estas máquinas quedaron accesibles desde internet por medio de las direcciones IP públicas. Se implementó ALB para la distribución del tráfico web (HTTP/HTTPS); se habilito el auto escalado garantizando que la arquitectura se adapte a la demanda de manera automática. Por último, se implementó el servicio de Elastic Container Service (ECS) para administrar contenedores Docker facilitando su despliegue y escalabilidad.

Palabras clave

Computación en la nube

Amazon Web Services (AWS)

VPC

Contenedores

Auto Scaling

Marco conceptual y contextual

AWS: Es la plataforma de servicios en la Nube de Amazon donde puedes tener una cuenta con diversos productos, soluciones o aplicaciones. Se basa en la infraestructura de su nube y proporciona una amplia gama de servicios, desde almacenamiento y bases de datos hasta inteligencia artificial y análisis de datos. (Mendieta, 2017).

Computación en la nube: El cloud computing es el acceso bajo demanda a recursos informáticos (servidores físicos o virtuales, almacenamiento de datos, capacidades de red, herramientas de desarrollo de aplicaciones, software, plataformas analíticas con IA y más) a través de Internet con precios de pago por uso. (Susnjara & Smalley, 2025)

VPC: Amazon Virtual Private Cloud (Amazon VPC) es un servicio de AWS que brinda control total sobre el entorno de redes virtuales en la nube. Con Amazon VPC, puedes definir y controlar una sección aislada de la nube de AWS donde puedes desplegar tus recursos y servicios de manera segura. (C, 2022).

Amazon EC2: La herramienta de Amazon EC2, también conocida como Amazon Elastic Compute Cloud, se refiere a un tipo de servicio web que se encarga de ofrecer capacidad informática en la nube, con propiedades como seguridad, un tamaño ajustable y computación escalable en la plataforma de Amazon Web Service (AWS).

Amazon EC2 también permite la creación y ejecución de casi cualquier aplicación del usuario, gracias a sus múltiples opciones y recursos, lo que la hace ideal para su implementación. (Mallón, 2024).

Contenedores: Los contenedores de software, o contendedores en la nube, son herramientas que simplifican la distribución de código. Como lo dijimos anteriormente, la principal misión de un contenedor es el transporte confiable de software de un entorno a otro. (Morales, 2022).

Instantáneas: Amazon Elastic Block Store (Amazon EBS) es un servicio que proporciona almacenamiento persistente a nivel de bloque para instancias de Amazon Elastic Compute Cloud (Amazon EC2). En pocas palabras, este servicio de AWS asigna discos duros fiables (es decir, volúmenes) a servidores basados en la nube. Una de las funciones más útiles de Amazon EBS son las instantáneas de volumen. (NAKIVO, 2024).

Volumen: Un volumen de Amazon EBS es un dispositivo de almacenamiento de nivel de bloque duradero que se puede adjuntar a sus instancias. Después de asociar un volumen a una instancia, puede usarlo como cualquier otro disco duro físico. Los volúmenes de EBS son flexibles. En el caso de los volúmenes de la generación actual adjuntados a los tipos de instancias de la generación actual, puede aumentar el tamaño de forma dinámica, modificar la capacidad de IOPS provisionadas y cambiar el tipo de volumen de los volúmenes de producción activos. (AWS).

Balanceador: Elastic Load Balancing distribuye automáticamente el tráfico entrante entre varios destinos, como EC2 instancias, contenedores y direcciones IP, en una o más zonas de disponibilidad. Monitorea el estado de los destinos registrados y enrutar el tráfico solamente a destinos en buen estado. Elastic Load Balancing escala de forma automática su capacidad de equilibrador de carga en respuesta a los cambios en el tráfico entrante.

AMI: Una imagen de máquina de Amazon, también conocida como AMI, se refiere a un tipo de imagen gestionada y compatible con la plataforma de Amazon Web Service. Su función es la de ofrecer la información que se necesita para lanzar una instancia en específico. Así pues, una misma imagen de máquina de Amazon tiene la capacidad de lanzar múltiples instancias que tengan una misma configuración, de la misma manera que varias AMI pueden lanzar instancias con configuraciones diferentes. (Mallón, 2024).

Escalabilidad: La escalabilidad es un concepto fundamental en el desarrollo de software que se refiere a la capacidad de un sistema para adaptarse a un aumento en la carga de trabajo sin comprometer su rendimiento. En un mundo donde las aplicaciones y plataformas digitales crecen rápidamente, entender y aplicar la escalabilidad se ha vuelto crucial para el éxito a largo plazo de cualquier proyecto tecnológico. (Takami, 2024).

Desarrollo e implementación del aprendizaje

Entrega 1

Diagrama de Arquitectura

- Representación gráfica de la red que muestra:

  EC2 Instancia (Linux/Windows)

  VPC

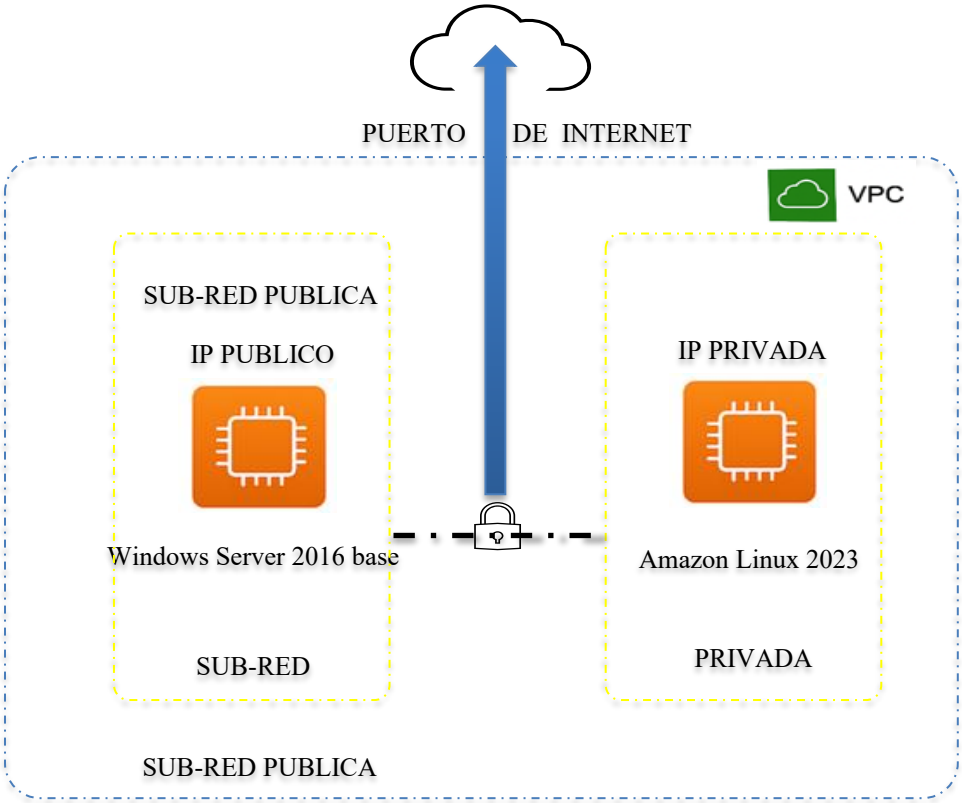
  Subredes públicas y privadas

  Internet Gateway

  Grupos de Seguridad

  IPS públicas y privadas

Figura 1Diseño general de la red en AWS



Descripción de la Arquitectura

Se creó una arquitectura básica en AWS utilizando una VPC personalizada, dividida en subredes públicas y privadas. A cada subred se le asignaron instancias EC2 para distintos propósitos (servidor web Windows y Linux).

Instancias usadas:

- Windows Server 2019 Base (para servidor web con IIS)
- Ubuntu Server 22.04 LTS (para servidor web con Apache)

Justificación:

- VPC personalizada para mayor control del tráfico.
- Subred pública para instancias accesibles desde Internet.
- Internet Gateway para conexión externa.
- Grupos de seguridad para permitir el tráfico necesario(RDP, SSH,HTTP).

Configuraciones Realizadas

- Creación de una VPC con subredes públicas
  - Asociar una tabla de rutas con acceso a IGW.
  - Crear y asociar Internet Gateway a la VPC.
- Creación de Instancias EC2 con Amazon Linux y Windows Server

- Tipo t2.micro (capa gratuita).
- Asignar IPS públicas durante el lanzamiento.
- Crear par de claves (.pem) para acceso RDP y SSH.

- Grupos de Seguridad
  - Windows Server: RDP puerto(3389), puerto HTTP (80): se creó un grupo de seguridad nombrado sgWindowsServer.
  - Amazon Linux: SSH puerto (22), HTTP puerto(80): se creó un grupo de seguridad nombrado launch-wizard-2.
- Asignación de IPS
  - Cada instancia tiene asigna una IP pública, configurada de forma automática.
  - IPS privadas asignadas automáticamente por la subred.

Procedimiento de Acceso

- Windows Server (RDP)
  - Usar cliente RDP (Remote Desktop).
  - Descargar archivo .rdp desde AWS.
  - Obtener la contraseña usando la clave privada del archivo .pem, cargado desde la instancia en AWS para descifrarla.
- Amazon Linux (SSH)
  - Usar consola o cliente SSH.
  - Ingresar las credenciales y acceder.
  - Comando: ssh -i "/ruta/ServerLinux2.pem" ec2-user@

- Consideraciones de seguridad
  - Usar archivos .pem protegidos.
  - No compartir contraseñas ni clave PEM.
  - Limitar rangos de IP en el grupo de seguridad.
  - Asegurarse de que el acceso solo sea desde nuestra IP.

Configuración del Servidor Web

Windows Server (IIS)

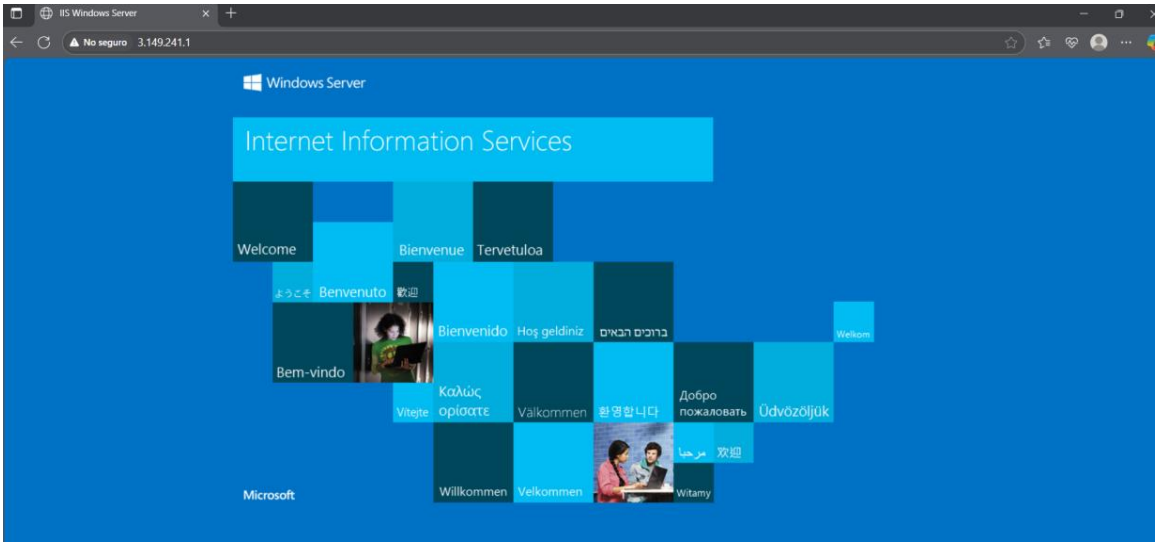
- Acceder por RDP.
- Ir a server Manager – Manage – Add Roles Feactures.
- Seleccionar Web Server (IIS).
- Completar la instalación.
- Verificar accediendo a la IP pública.

Linux Apache

- Acceder vía SSH.
- Ejecutar comandos:
- Para entrar en modo administrador - sudo su
- Para instalar Apache - dnf install httpd
- Para iniciar servicio – systemctl start httpd
- Para verificar el estado del servicio – systemctl status httpd
- Probar en el navegador con la IP pública.

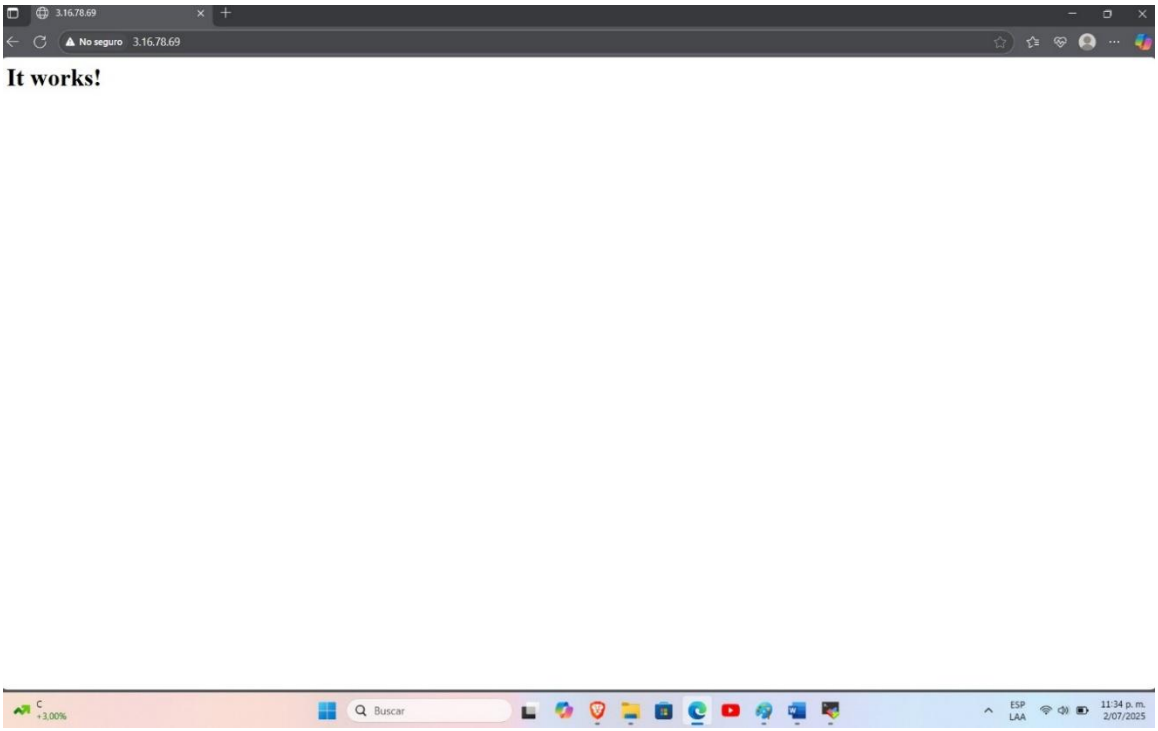
Prueba de acceso al servidor Windows desde el navegador.

Figura 2 Servidor IIS operativo.



Prueba de acceso al servidor de Linux desde el navegador.

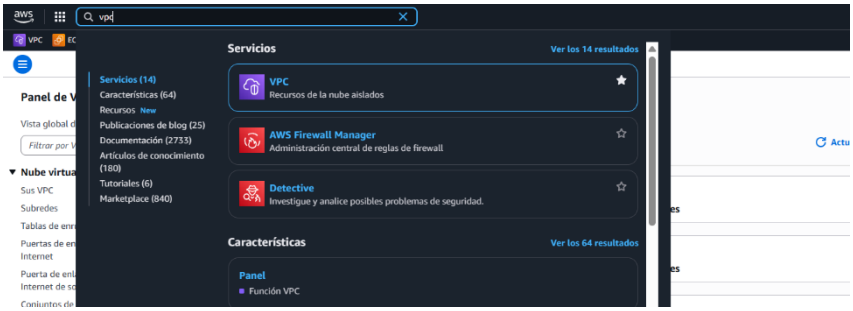
Figura 3 Servidor Apache activo



Creación de VPC y Subredes

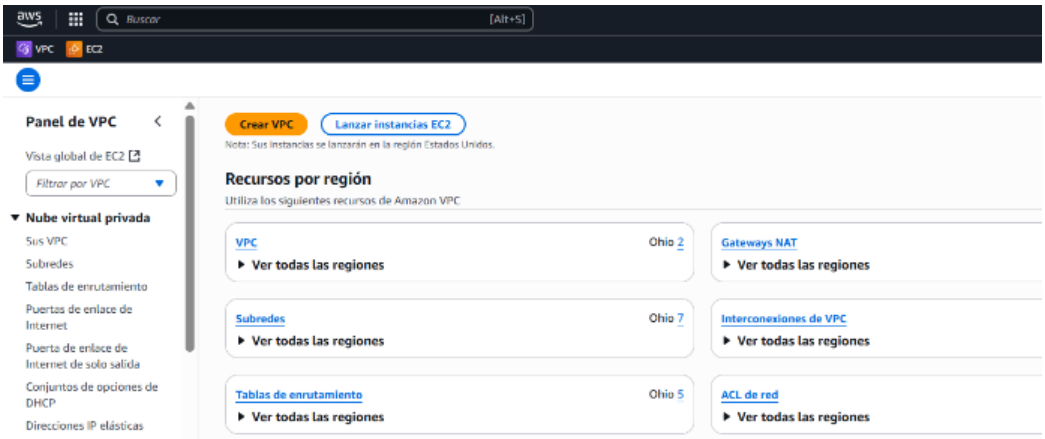
En el buscador escribimos VPC y seleccionamos.

Figura 4 Creación de VPC.



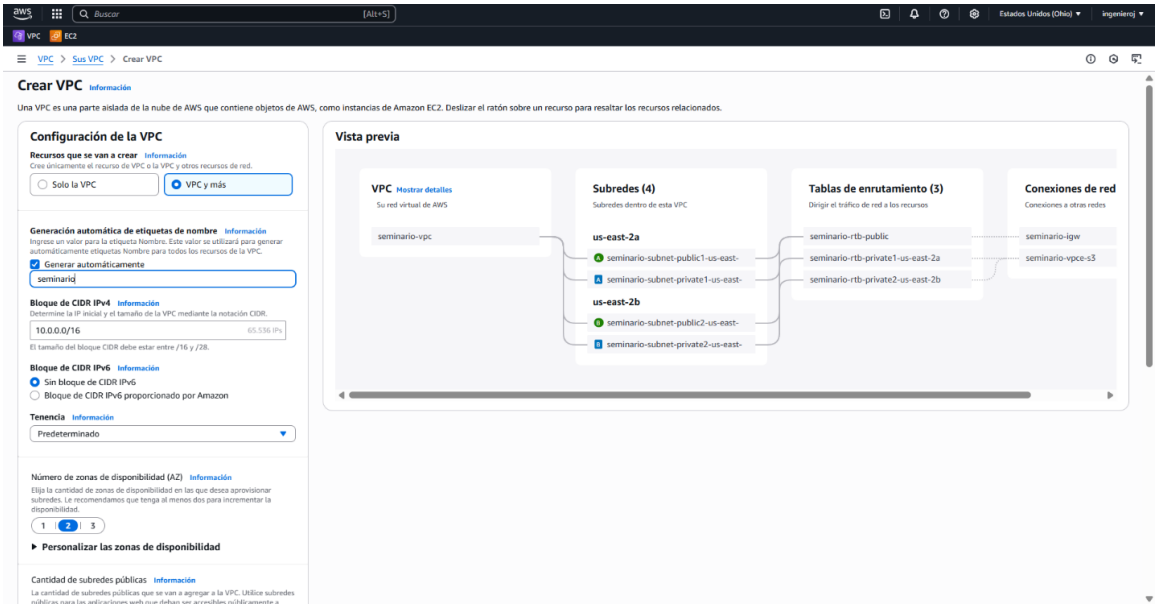
Damos en la opción de crear VPC.

Figura 5 crear VPC.



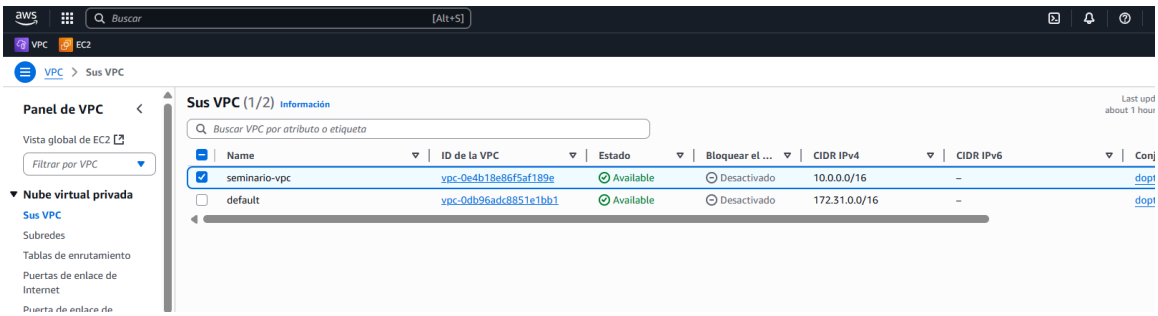
Nombramos nuestra VPC y elegimos la cantidad de subredes que necesitamos, las cuales van a hacer 2 públicas y dos privadas. Le damos en Crear VPC

Figura 6 nombrar VPC.



Evidenciamos que la VPC se creó correctamente

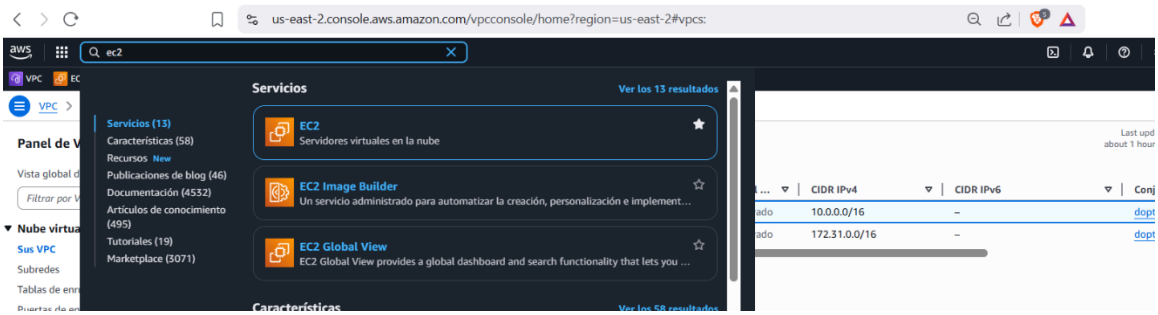
Figura 7 VPC creadas.



Creación de instancia con Windows Server 2016

En el buscador escribimos EC2 y seleccionamos.

Figura 8 Busqueda de ec2.



En el panel izquierdo seleccionamos instancias y luego damos lanzar instancia.

Figura 9 servicio instancias.

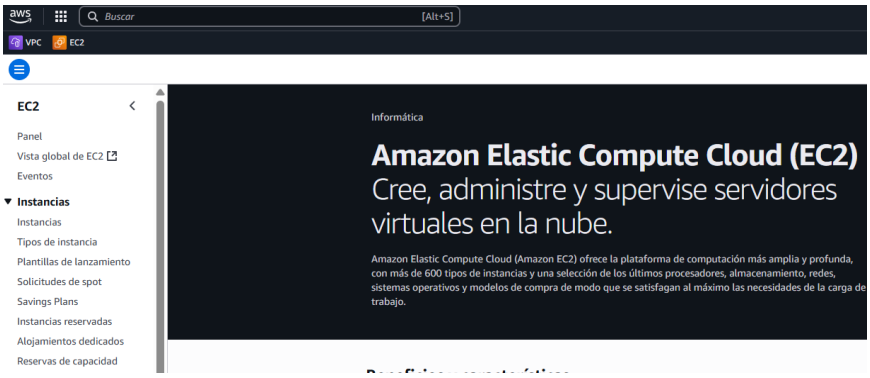
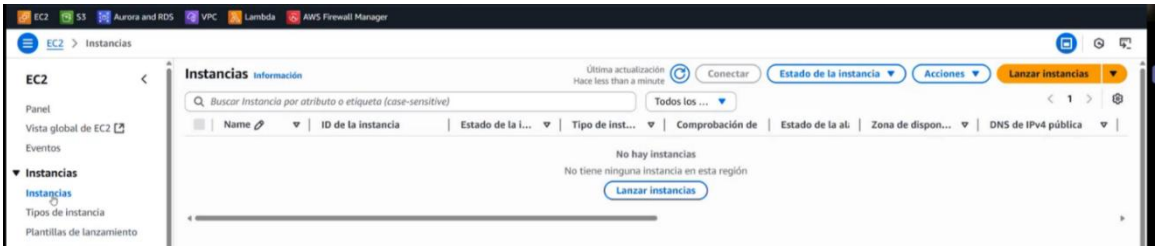
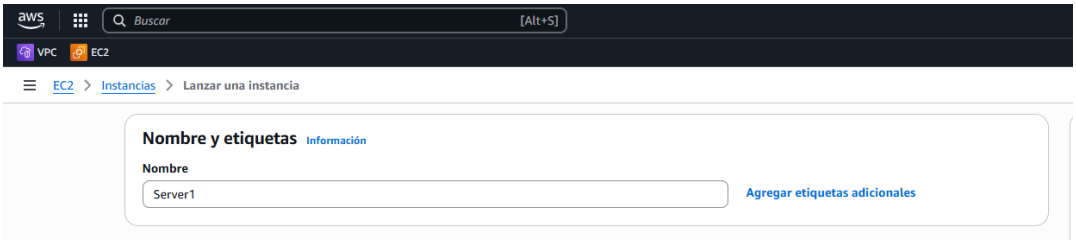


Figura 10 creación de instancia.



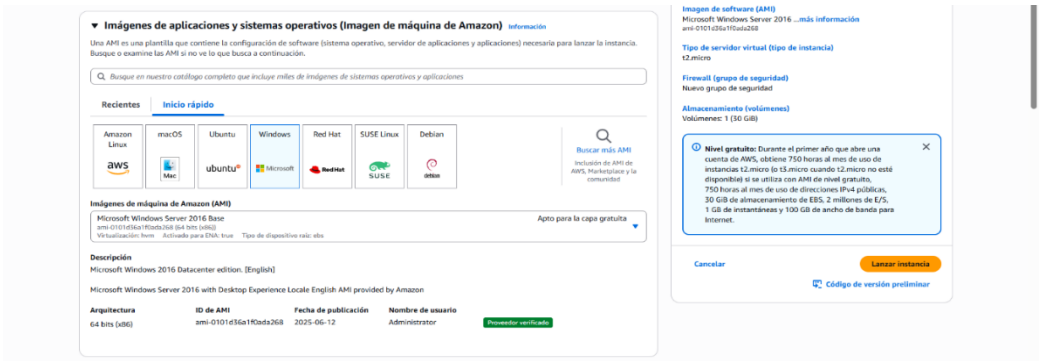
Ingresamos el nombre de nuestra instancia.

Figura 11 nombrar instancia.



Seleccionamos la AMI para nuestro servidor, en este caso Windows Server 2016 Base

Figura 12 AMI a instalar.

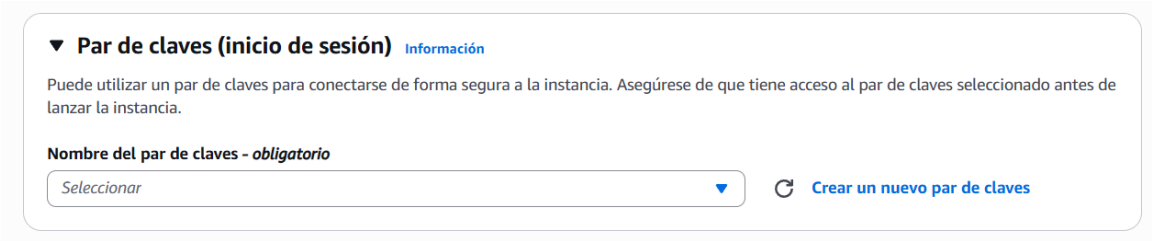


Seleccionamos el tipo de instancia.

Figura 13 tipo de instancia.

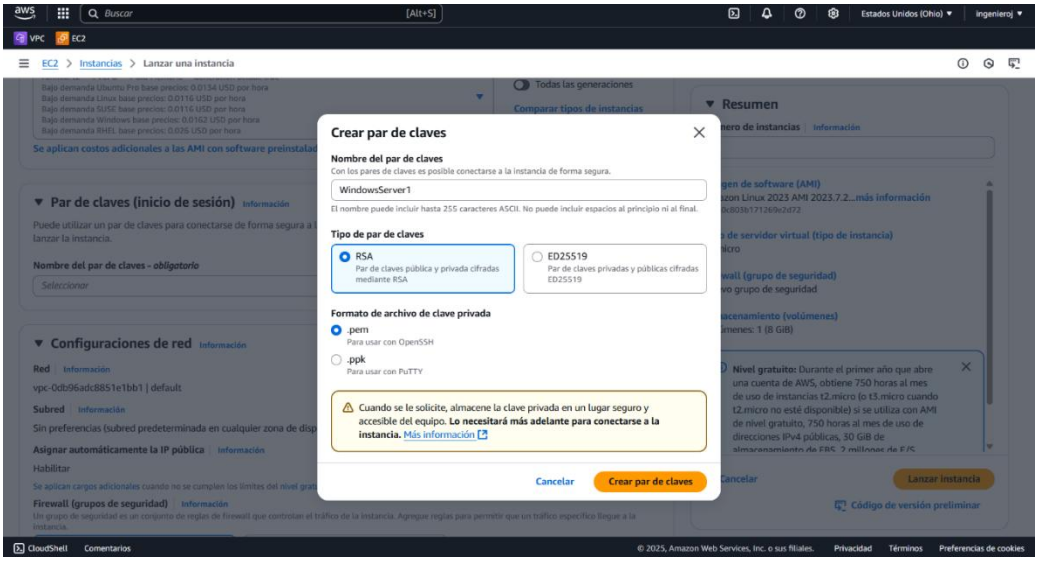
Creamos un par de claves para acceder a nuestra instancia, seleccionamos la opción de crear un nuevo par de claves, para generar un certificado de seguridad.

Figura 14 creación Par de Clave.



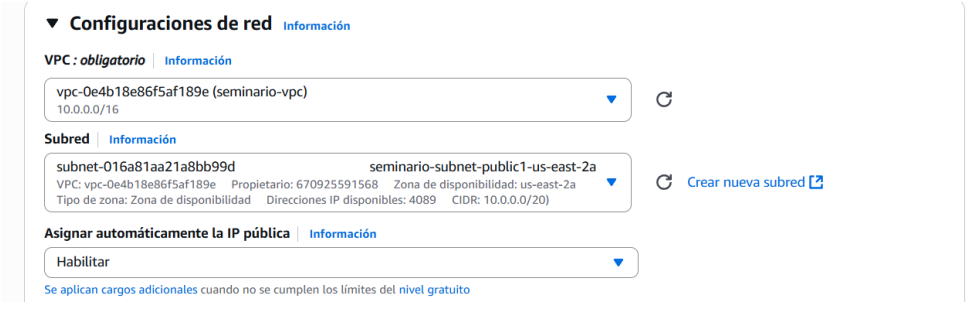
Nombramos el par de claves y le damos crear.

Figura 15 Par de claves.



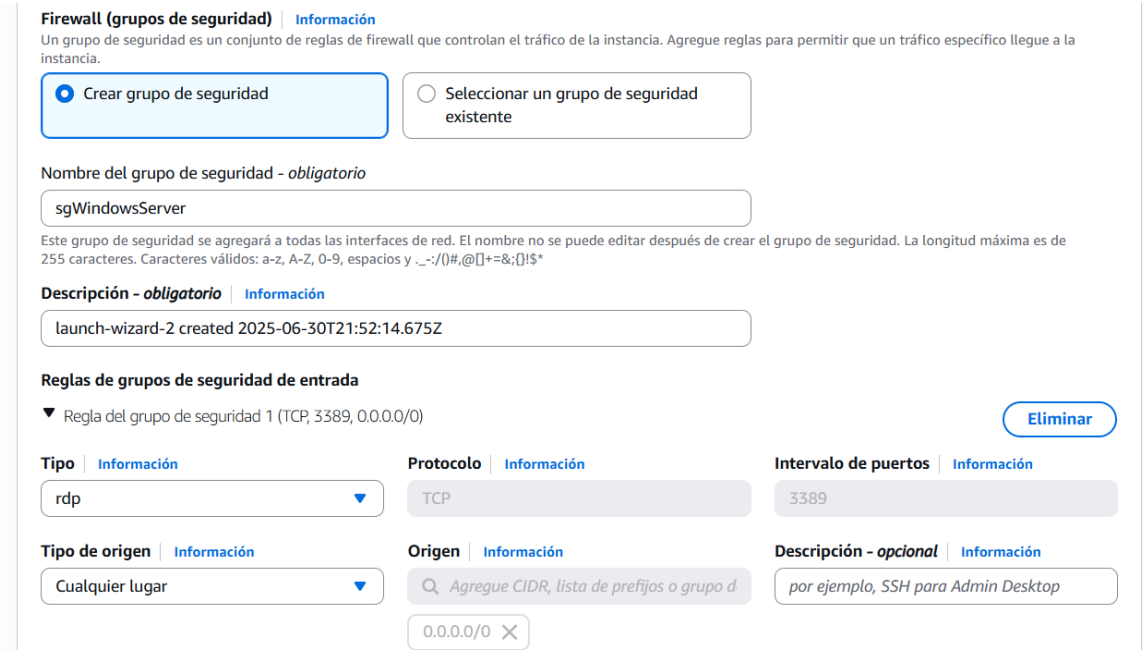
Realizamos la configuración de red: Seleccionamos nuestra VPC y la subred que vamos a utilizar. Ademas habilitamos la opción de asignar IP pública para acceder a la instancia.

Figura 16 configuración de red.



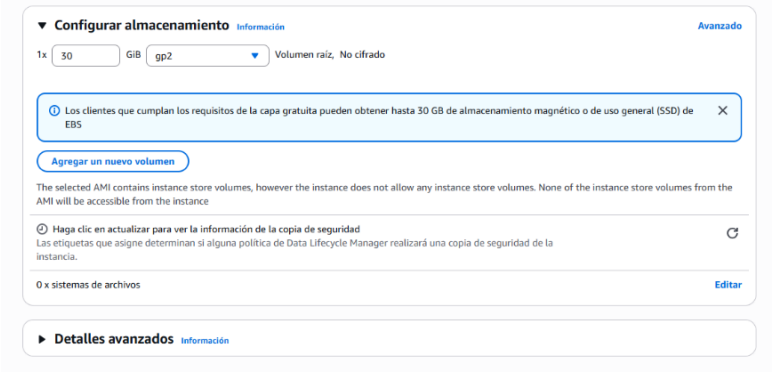
Creamos el grupo de seguridad y revisamos las reglas del grupo de seguridad, que por defecto nos da una regla tipo RDP que nos permite el acceso al protocolo TCP por el puerto 3389. Habilitados el tipo de origen para que se conecte desde cualquier lugar.

Figura 17 grupo de seguridad.



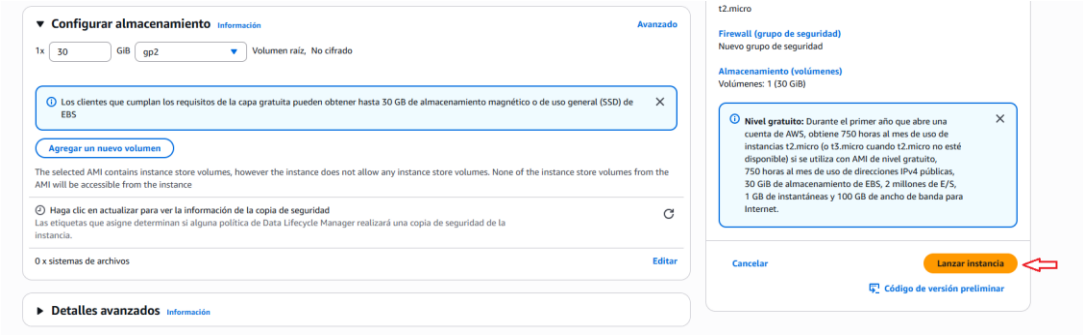
Configuración de almacenamiento, que por defecto nos recomienda unas características.

Figura 18 configuración almacenamiento.



Finalmente le damos Lanzar instancia y esperamos la creación de esta.

Figura 19 lanzar instancia.



Efectivamente observamos que nuestra instancia se creó correctamente.

Figura 20 creación instancia.

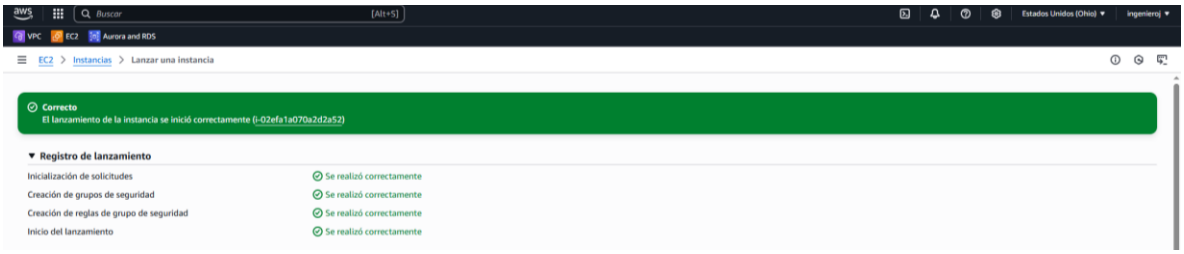
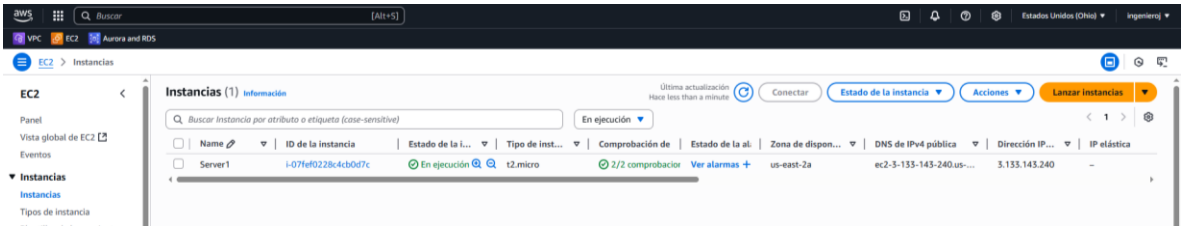
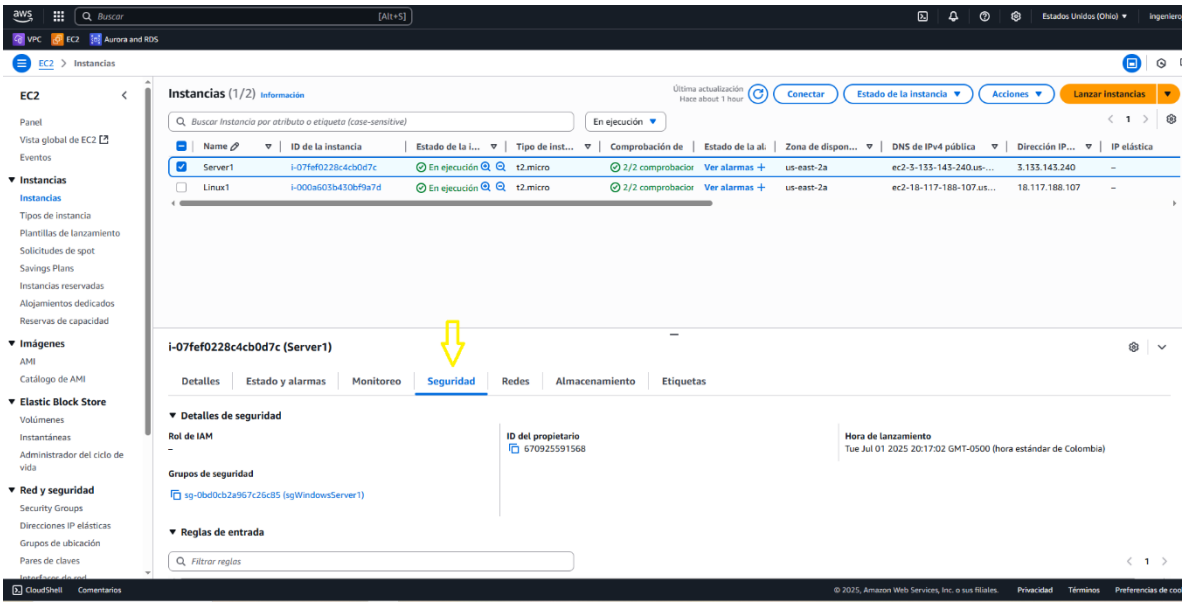


Figura 21 instancia activa.



Vamos a habilitar el puerto 80 para acceder al servidor desde la web, entrando en nuestra instancia en la opción de seguridad.

Figura 22 habilitacion del Puerto 80.



Seleccionamos grupo de seguridad y le damos en editar reglas de entrada.

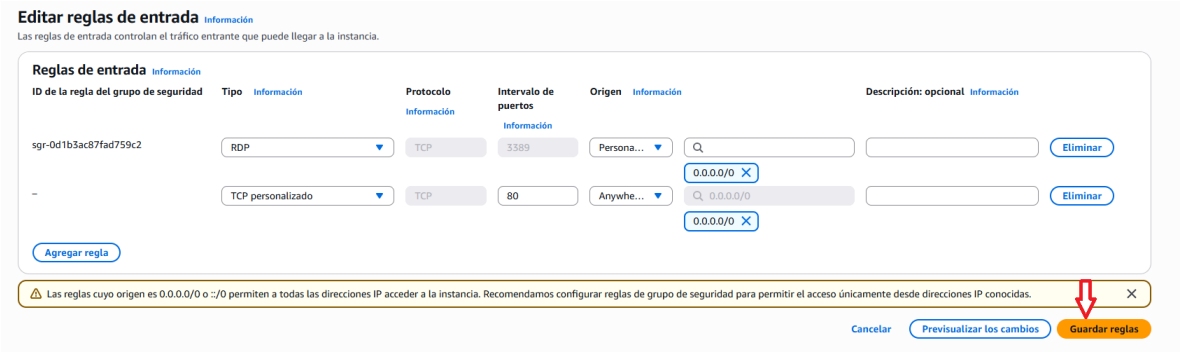
Figura 23 Grupo de seguridad.

Le damos en la opción de agregar reglas.

Figura 24 agregar reglas de entrada.

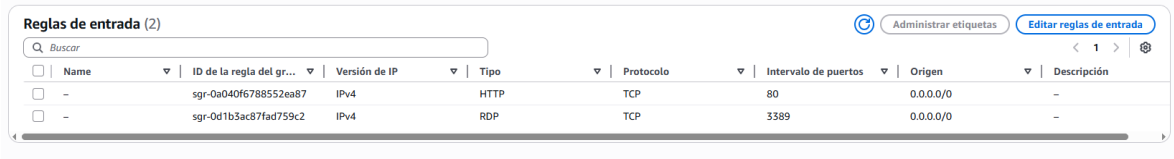
Agregamos el puerto al que se le va a dar acceso en este caso 80, para que se conecte desde cualquier lugar y le damos en guardar regla.

Figura 25 habilitar Puerto 80.



Evidenciamos que ya se agregó y podremos acceder por medio de ese puerto.

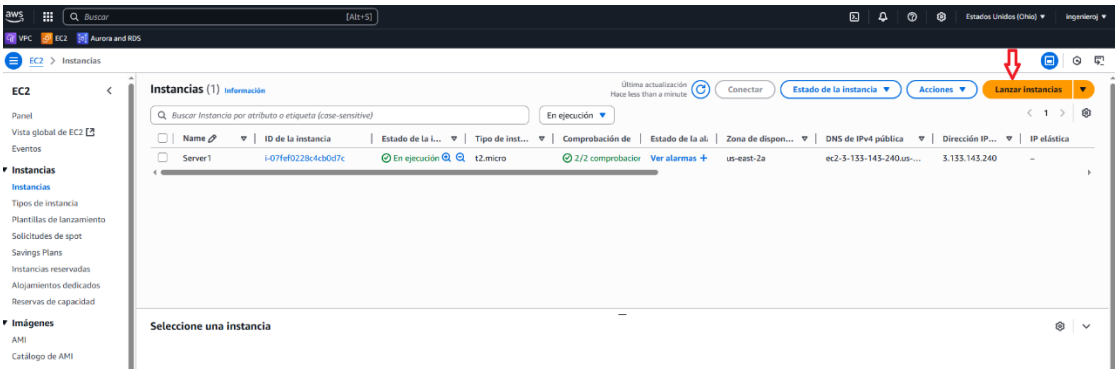
Figura 26 Puerto 80 agregado.



Creación de instancia con Amazon Linux 2023

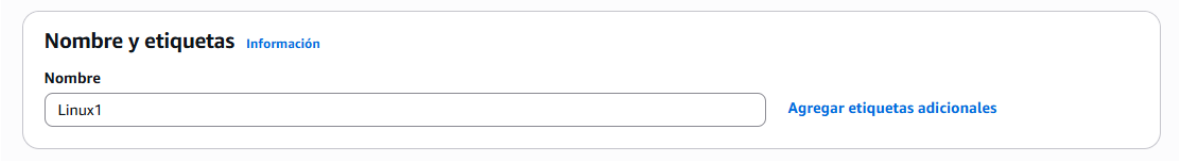
Nos dirigimos al panel de EC2 en la opción de instancias, y seleccionamos lanzar instancia.

Figura 27 creación de instancia Linux.



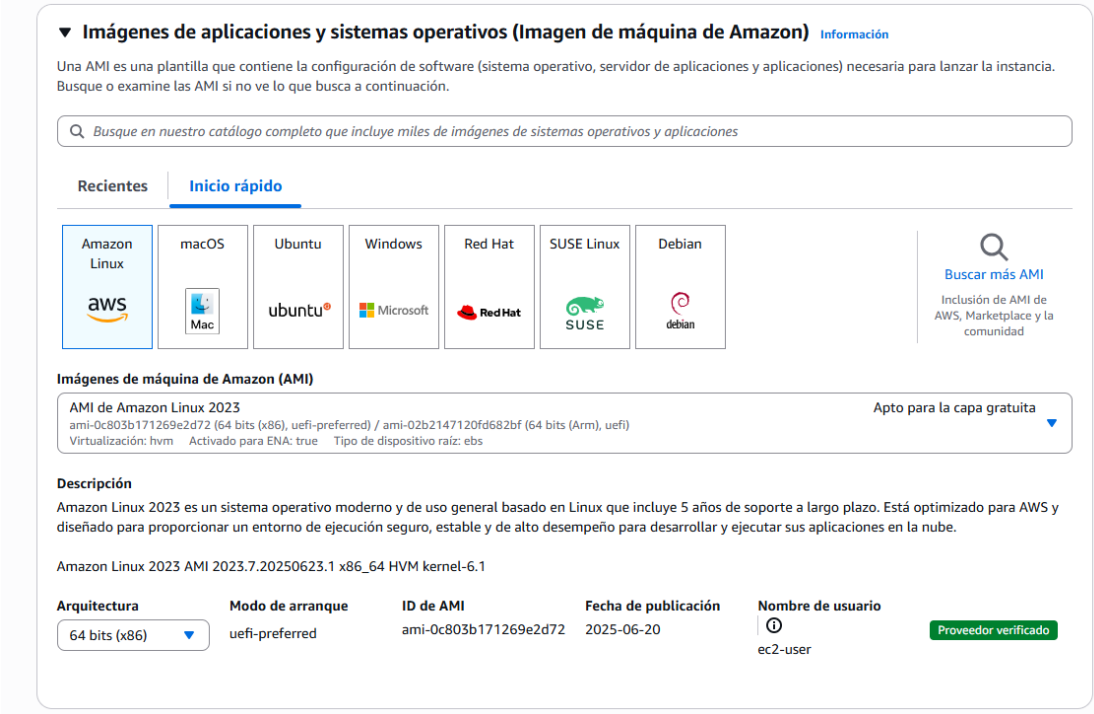
Nombramos nuestra instancia (en este caso Linux1).

Figura 28 nombrar instancia.



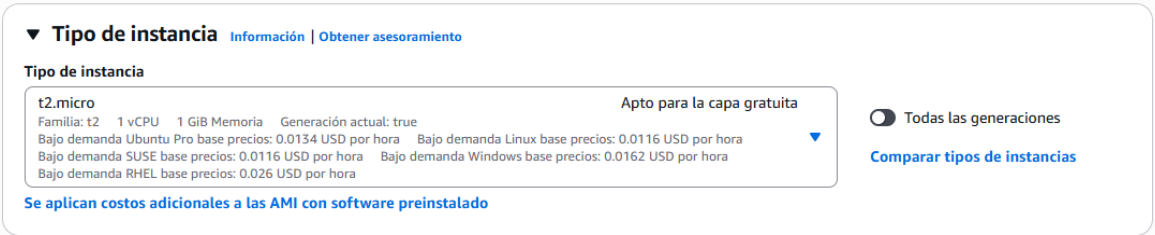
Seleccionamos nuestra AMI para nuestra instancia, en este caso Amazon Linux 2023.

Figura 29 elección de AMI.



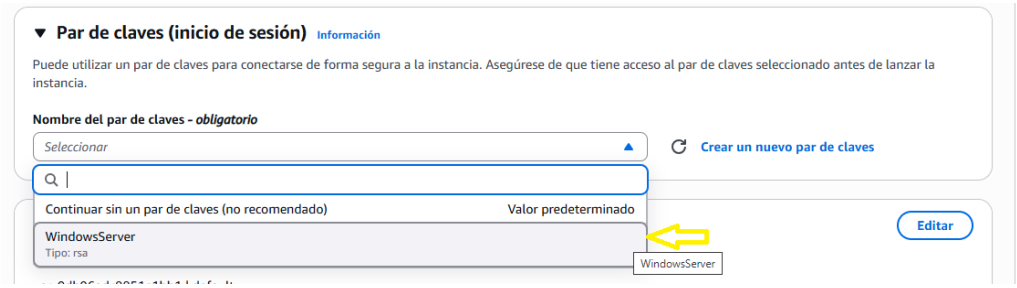
En tipo de instancia lo dejamos por defecto (t2.micro).

Figura 30 tipo de instancia.



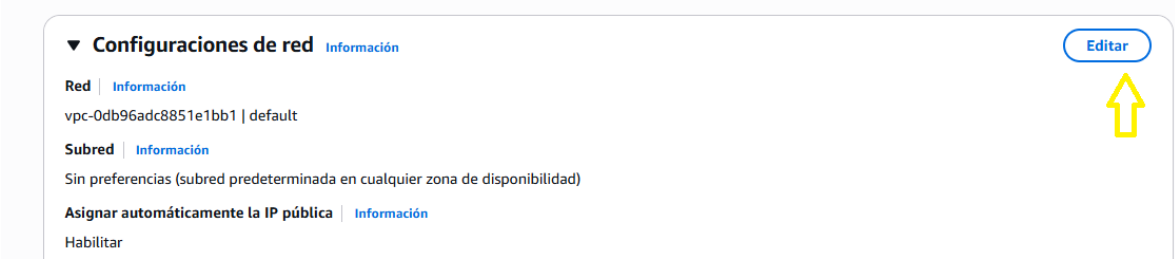
Para el par de claves usamos el mismo que creamos para Windows y lo seleccionamos.

Figura 31 Autenticacion para de calves.



Editamos la configuración de red.

Figura 32 configuración de red.



Seleccionamos la VPC ya creada para que nos asigne la subred pública y habilitamos la opción para que nos cree una IP automáticamente.

Figura 33 eleccion de VPC.

Creamos un nuevo grupo de seguridad y agregamos una regla para acceder por el puerto 80 y que se conecte desde cualquier lugar.

Figura 34 nuevo Grupo de seguridad.

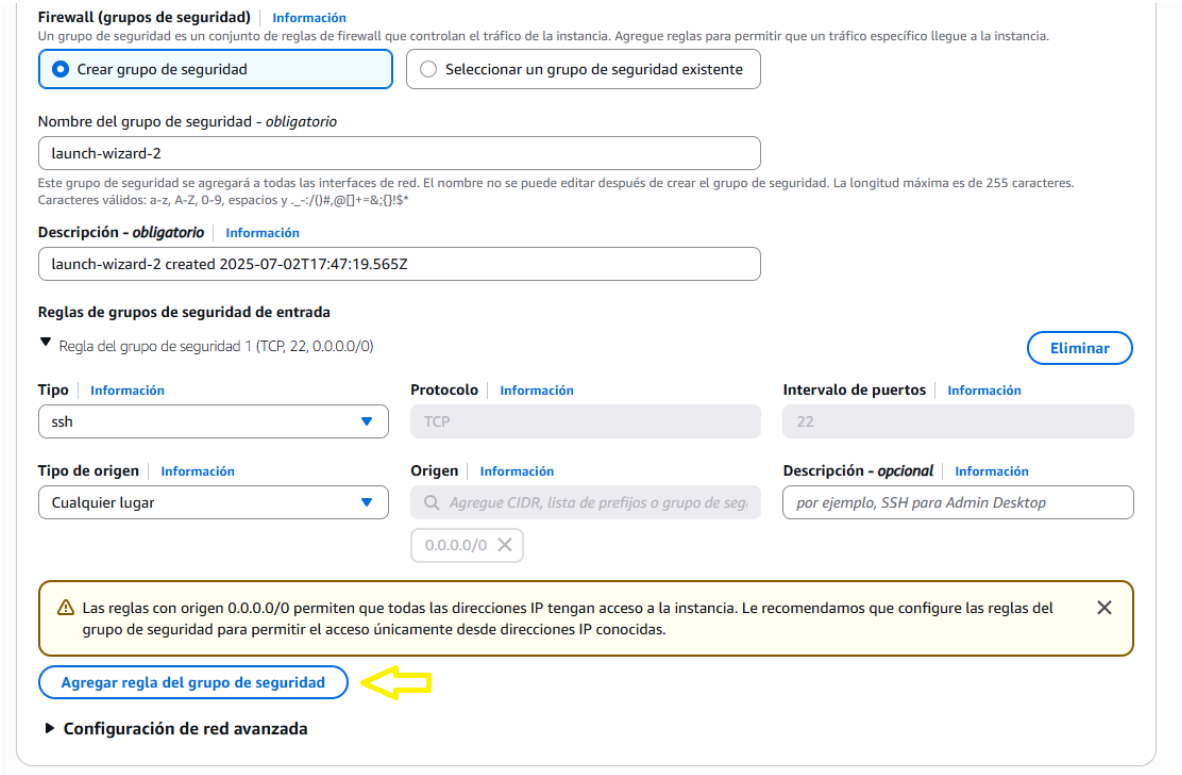
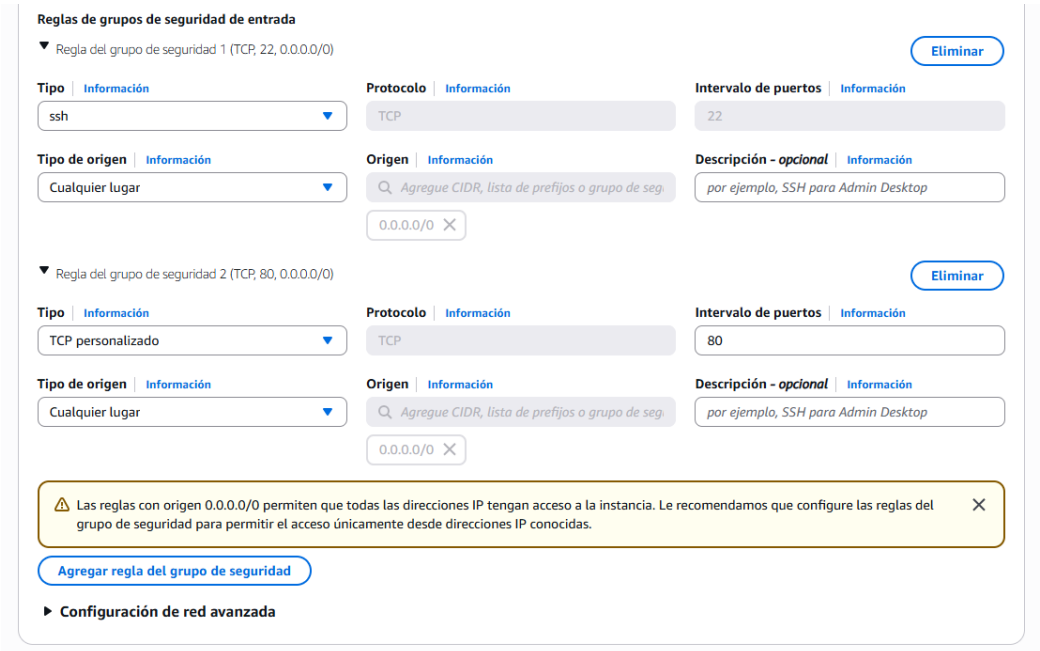
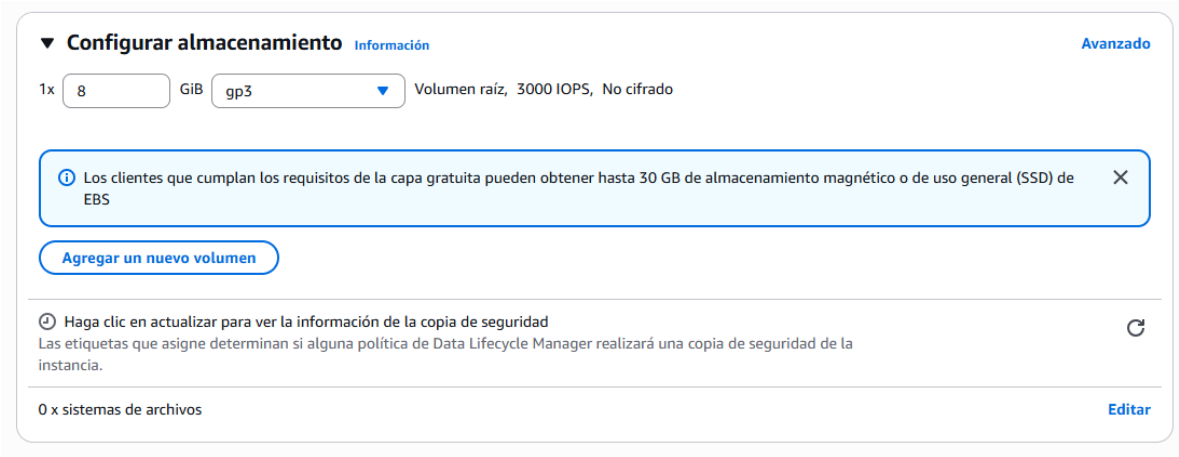


Figura 35 habilitacion Puerto 80.



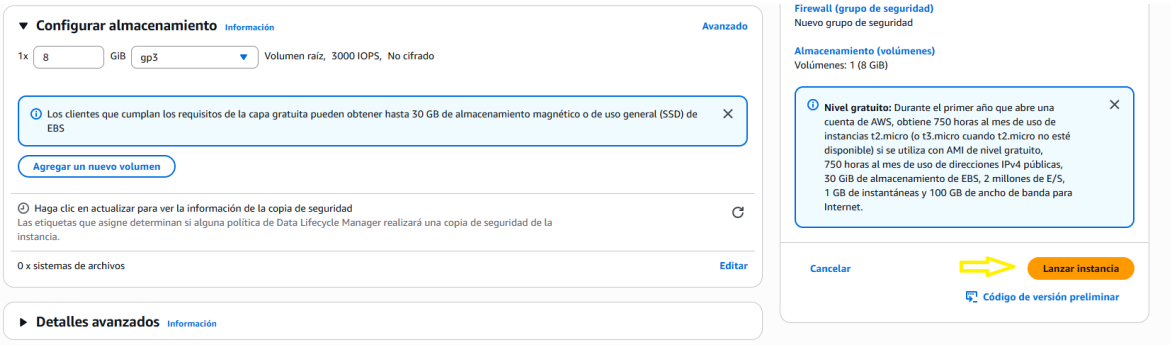
Configuramos el almacenamiento, por defecto nos sugiere unas características.

Figura 36 Almacenamiento instancia.



Presionamos finalmente en lanzar instancia para que se cree.

Figura 37 lanzamiento instancia.



Observamos que nuestra instancia se creó.

Figura 38 instancia creada.

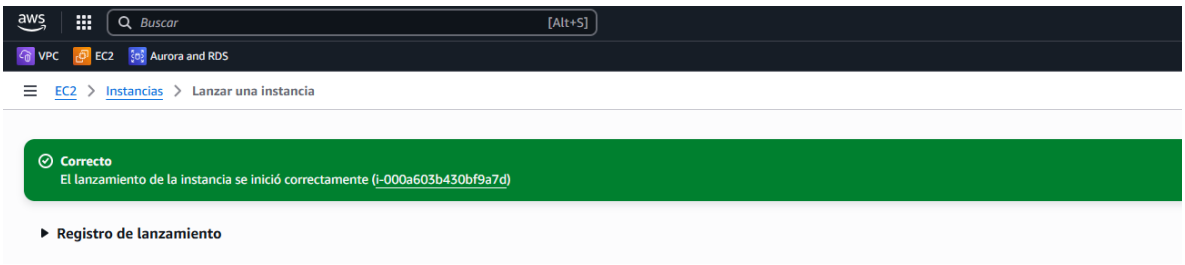
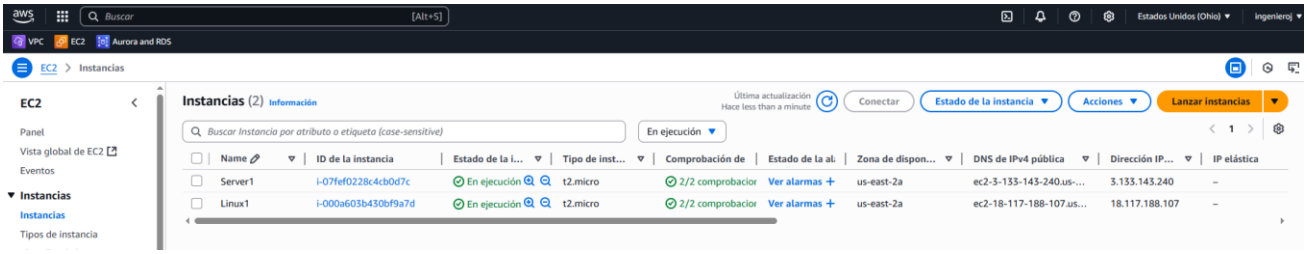


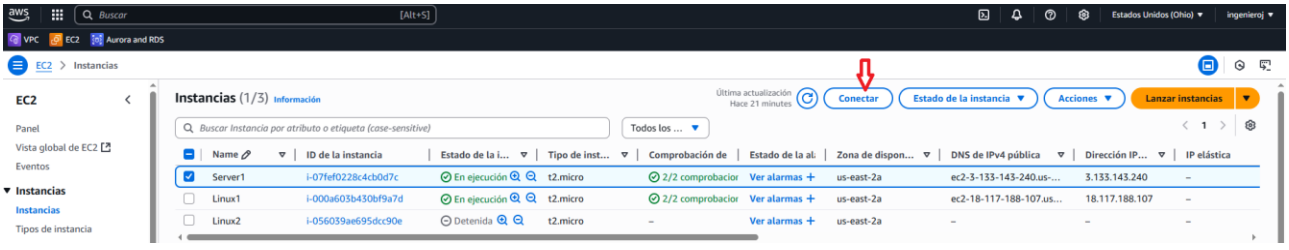
Figura 39 visualización instancia.



Acceso vía RDP a la instancia Windows

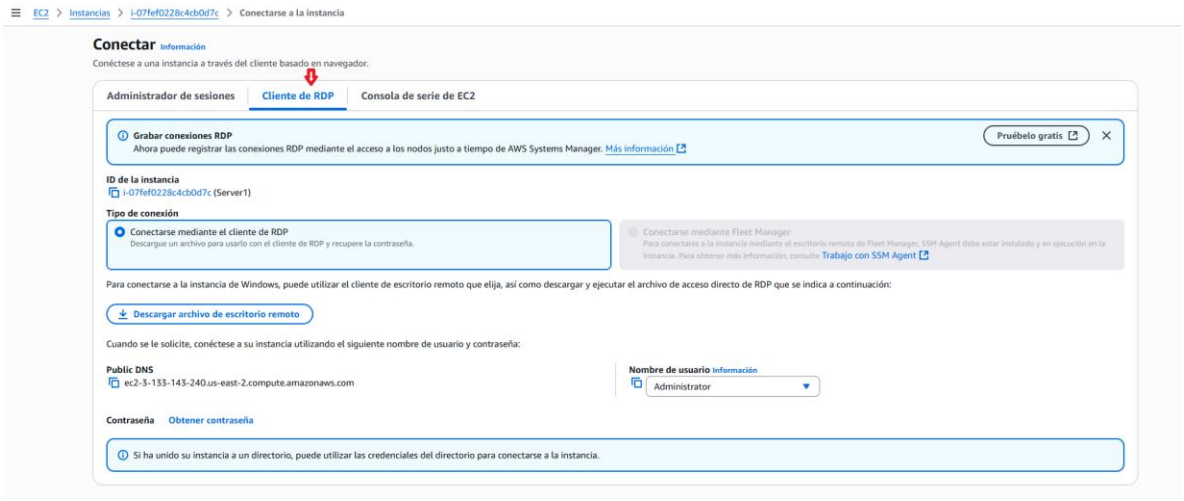
Seleccionamos la instancia y damos en conectar.

Figura 40 acceso por RDP a windows.



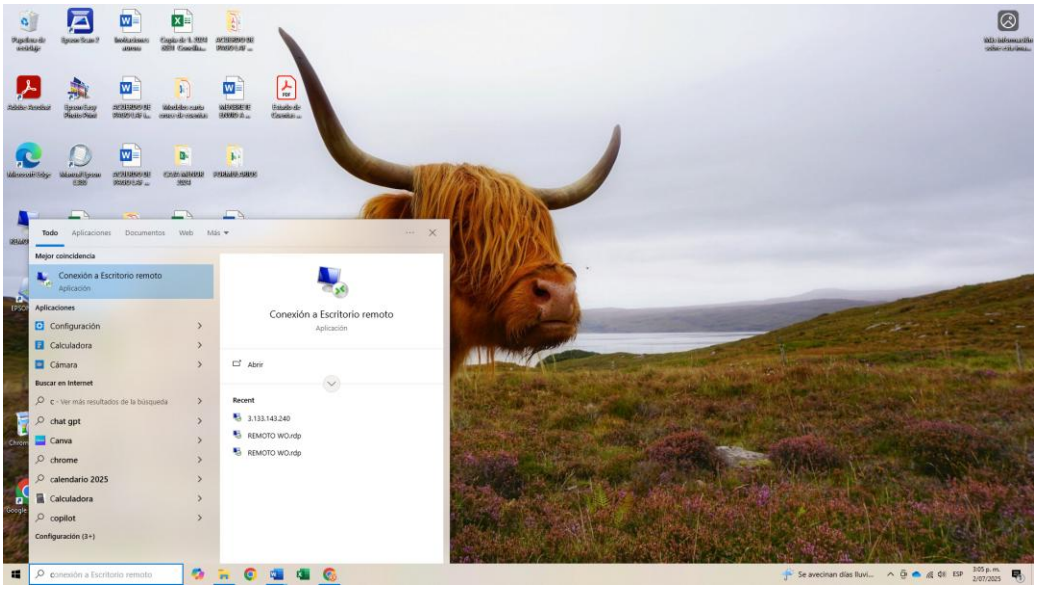
Seleccionamos cliente RDP.

Figura 41 conexión RDP.



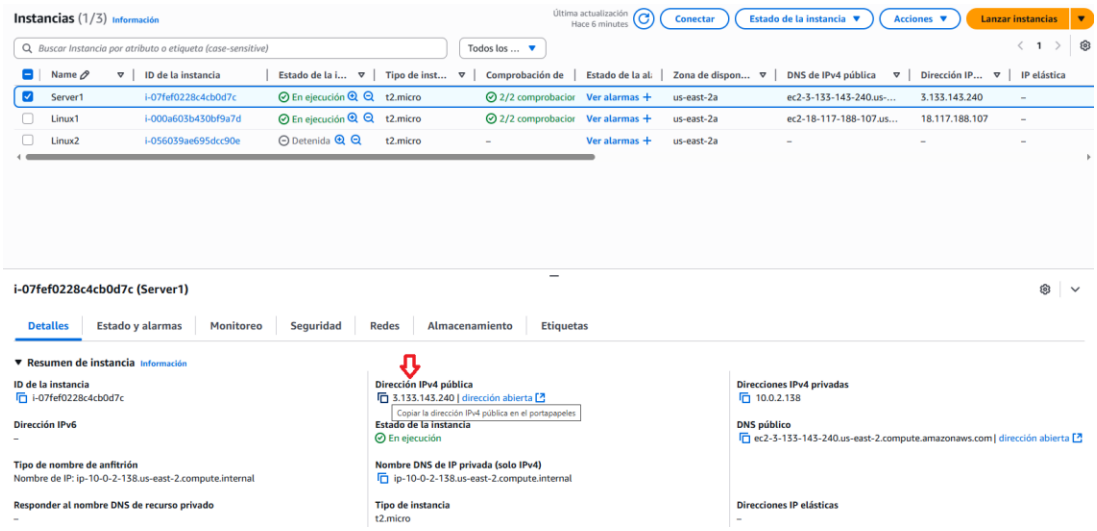
Buscamos en nuestro equipo conexión a escritorio remoto.

Figura 42 conexion por escritorio remoto.



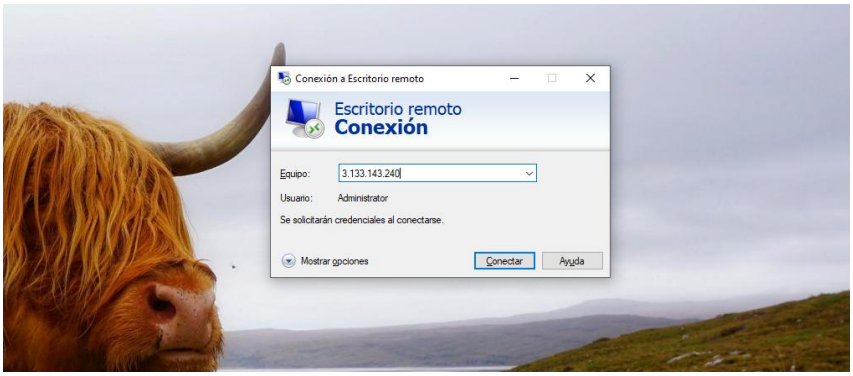
Copiamos la dirección IP pública de nuestra instancia.

Figura 43 seleccion IP publica.



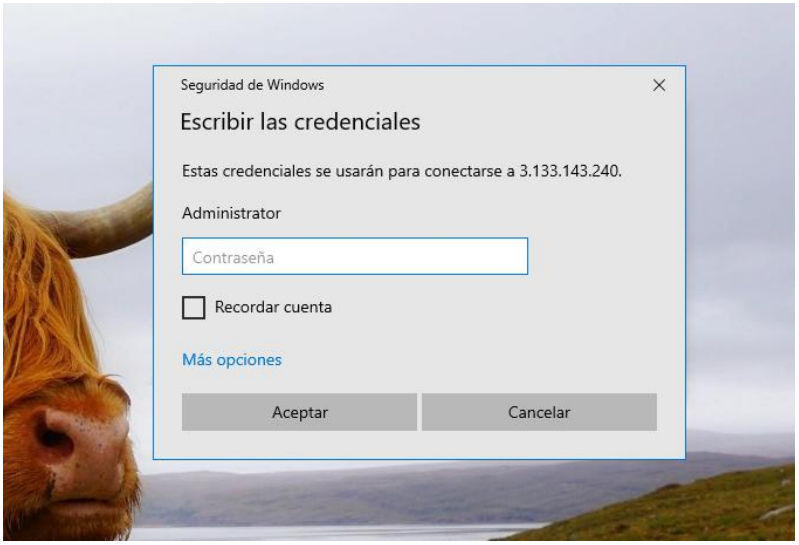
La pegamos en la venta de conexión a escritorio remoto y damos conectar.

Figura 44 conexion a escritorio remoto.



Nos pide la contraseña de acceso. Para esto nos devolvemos a la conexión de la instancia en cliente RDP.

Figura 45 contraseña de acceso.

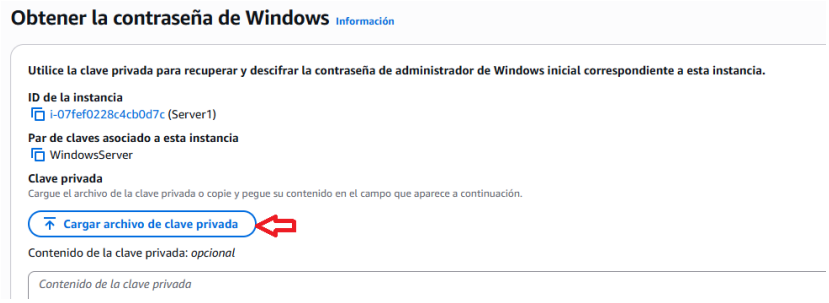


Damos en la opción de obtener contraseña.

Figura 46 obtención de contraseña.

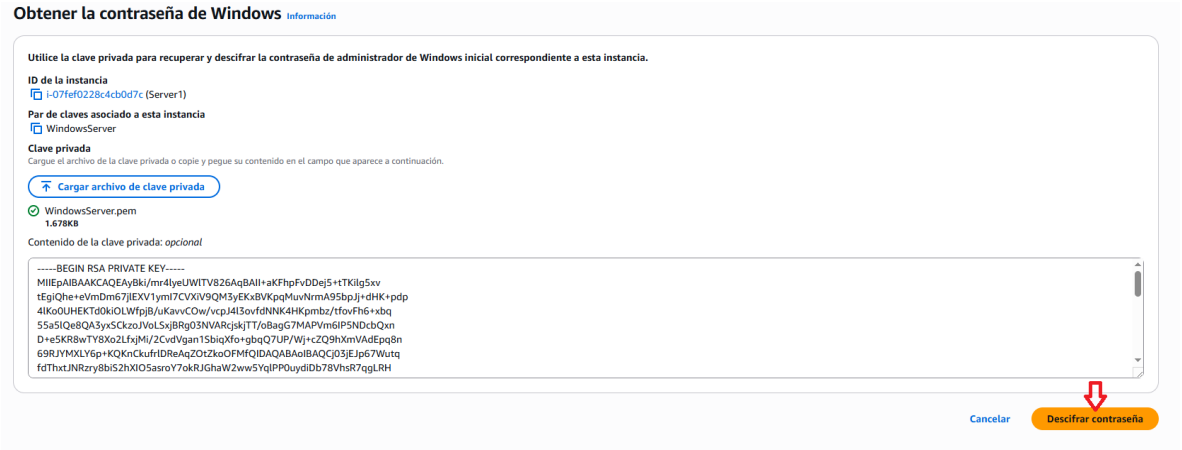
Cargamos nuestro archivo del par de claves para que nos proporcione la contraseña.

Figura 47 cargue de par de clave.



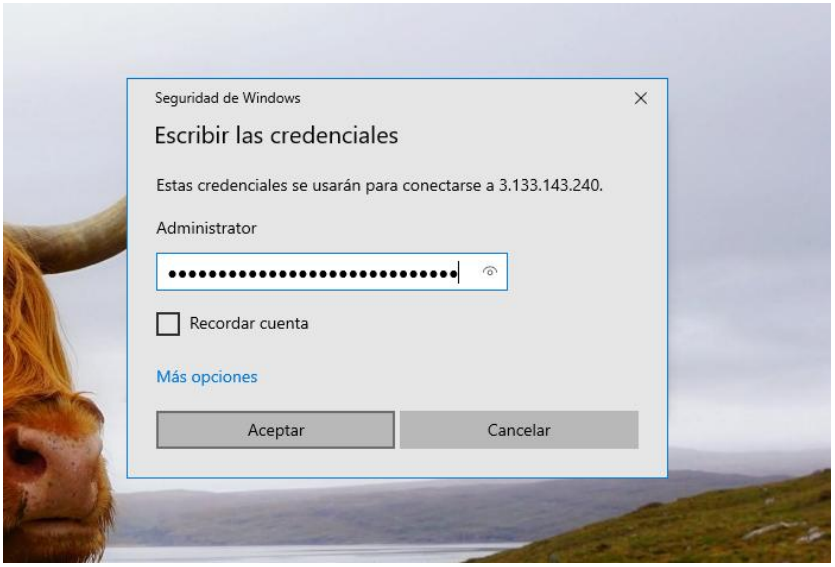
Una vez cargado, le damos descifrar contraseña.

Figura 48 cifrado de contraseña.



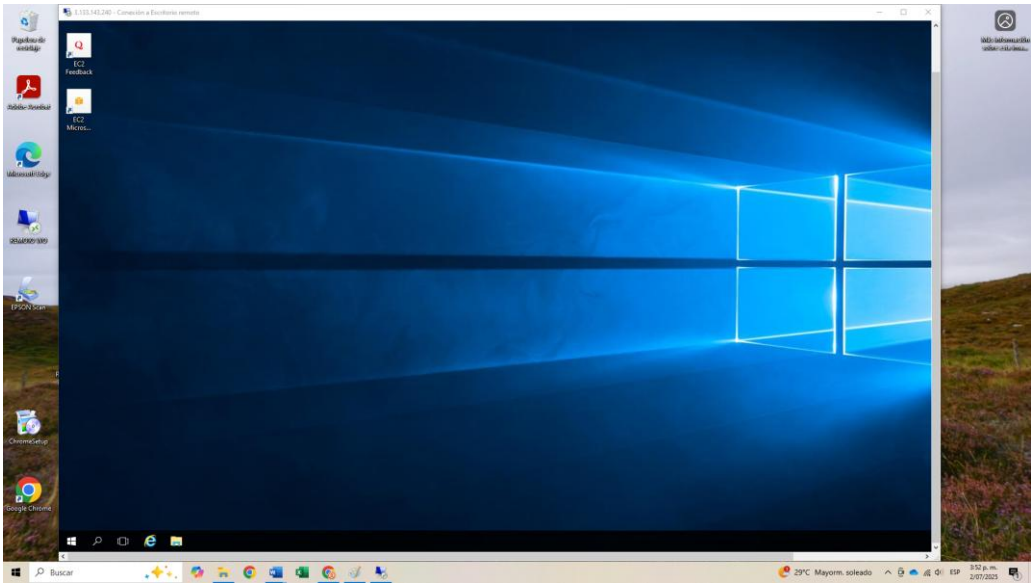
Ingresamos las credenciales y damos aceptar.

Figura 49 credenciales de acceso.



Efectivamente se conecta al servidor.

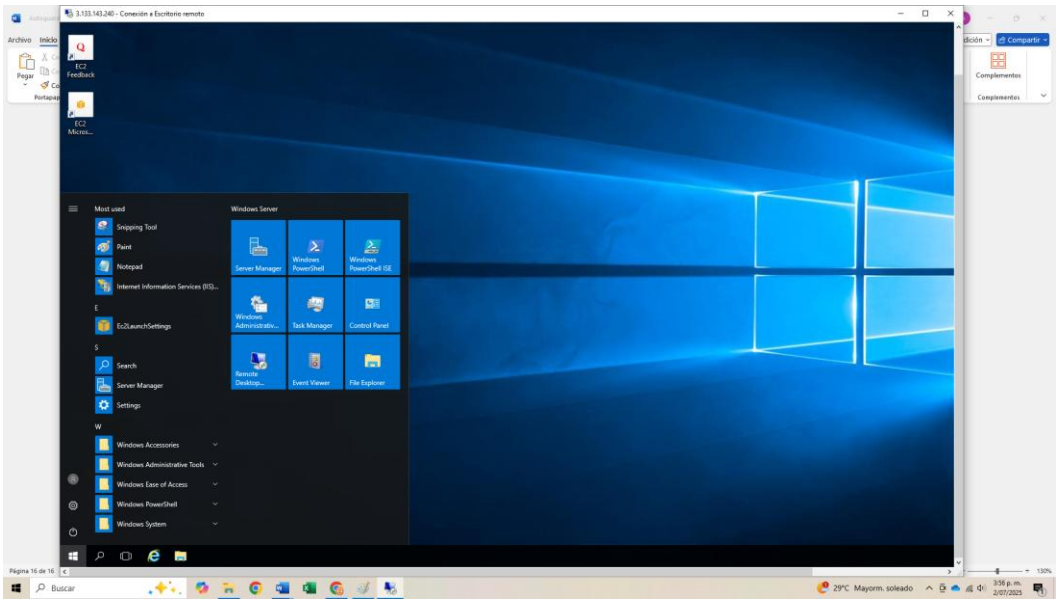
Figura 50 conexion servidor.



Instalación del rol IIS

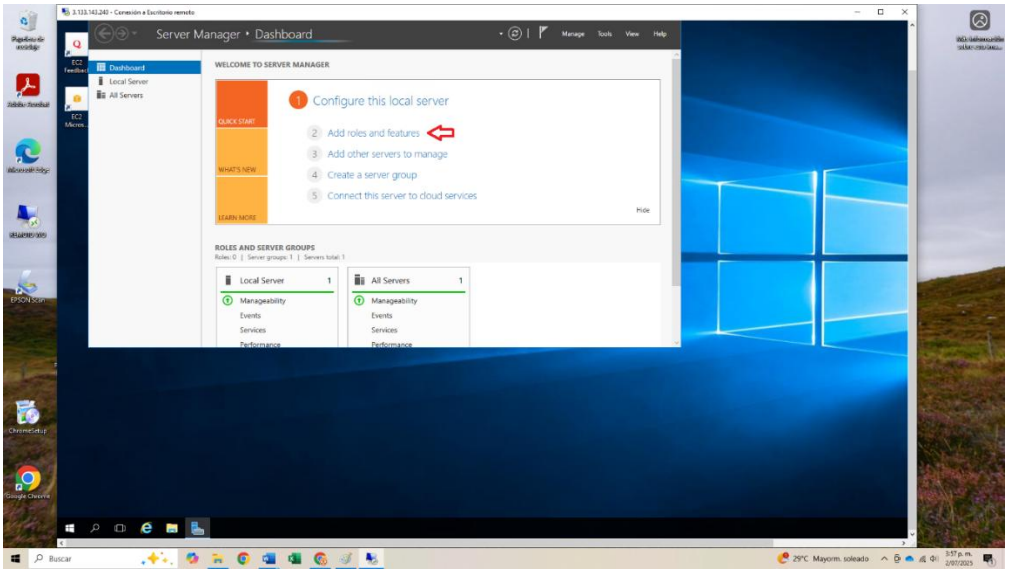
Dentro de nuestro servidor buscamos server manager y damos clic.

Figura 51 Busqueda server manager.



Para instalar el rol damos clic en añadir roles y características.

Figura 52 instalación de rol.



Iniciamos la instalación y damos siguiente.

Figura 53 proceso instalación.

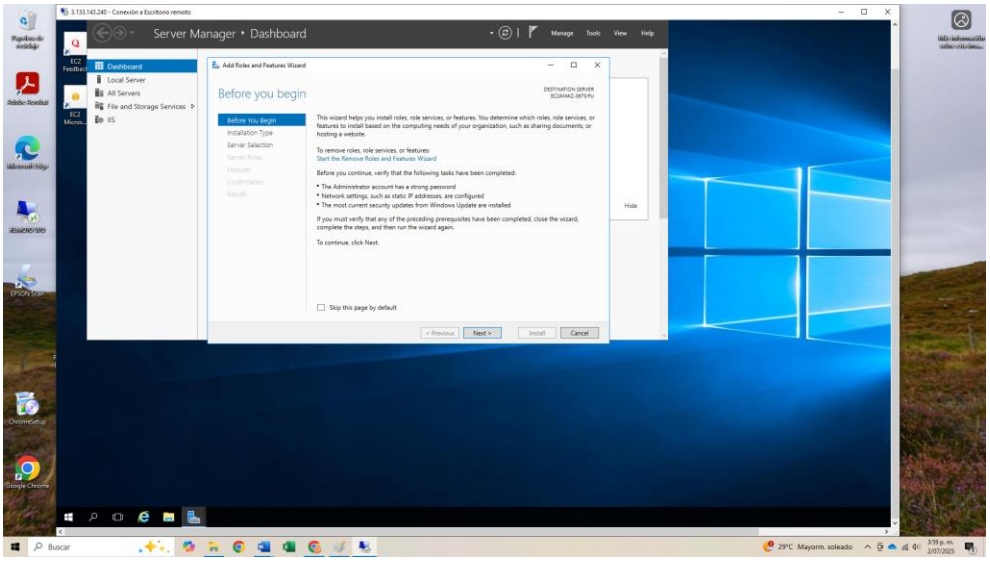


Figura 54 proceso de instalación.

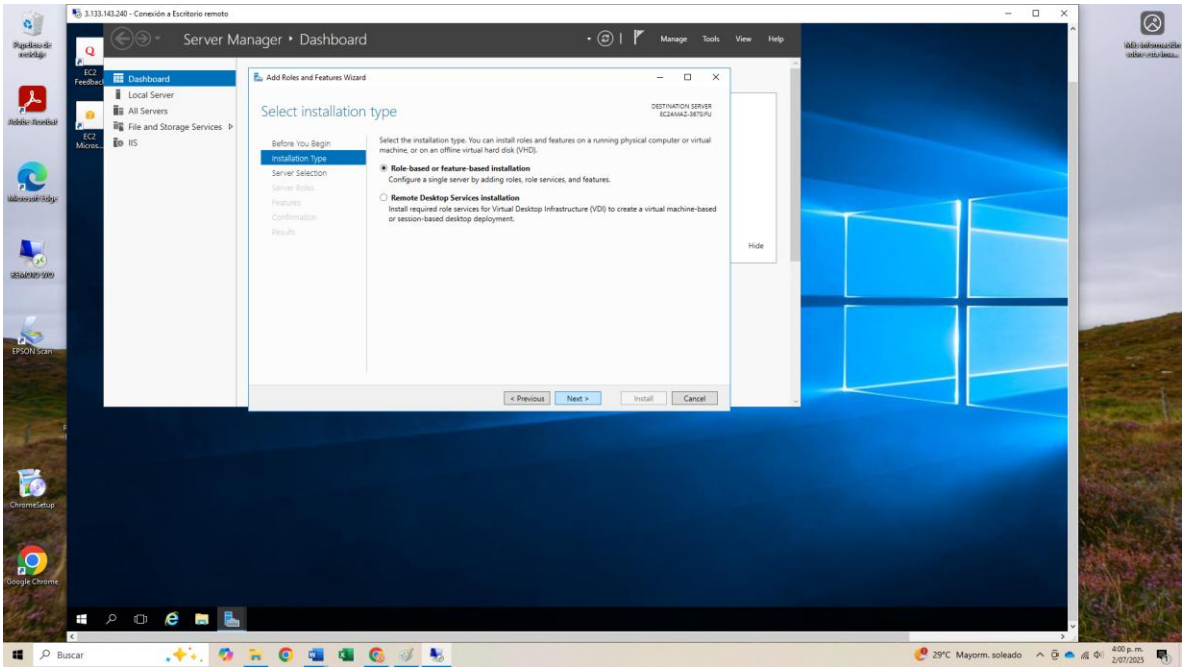
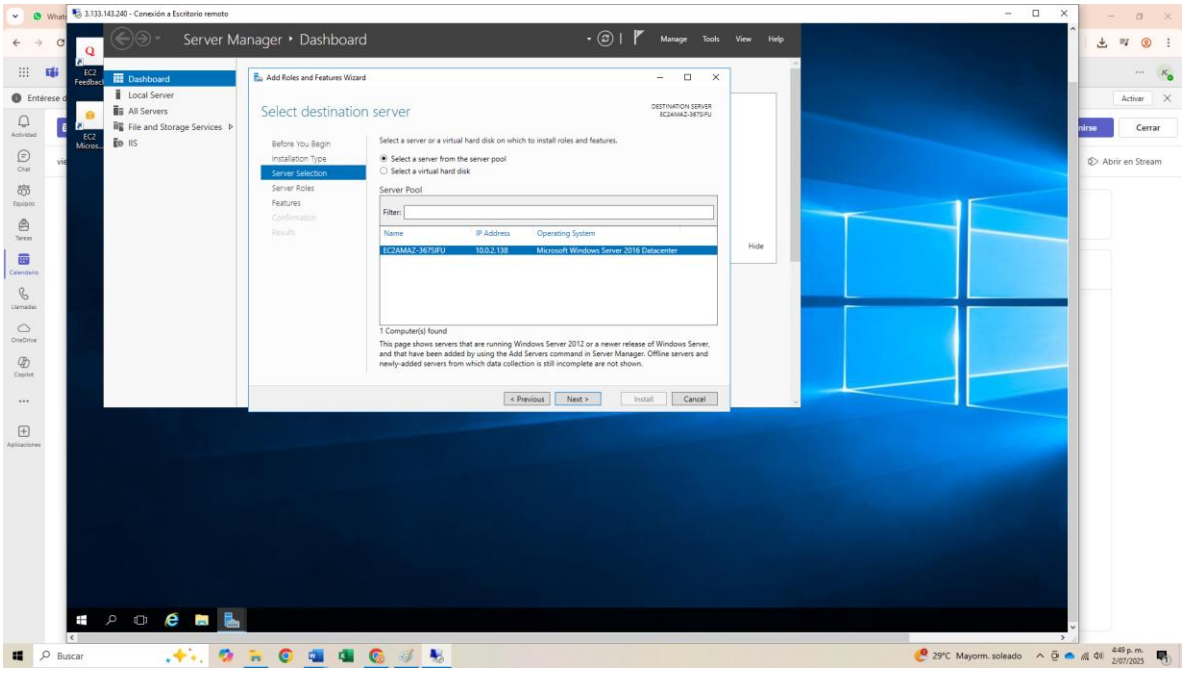


Figura 55 instalación rol.



Instalamos el servicio de web server.

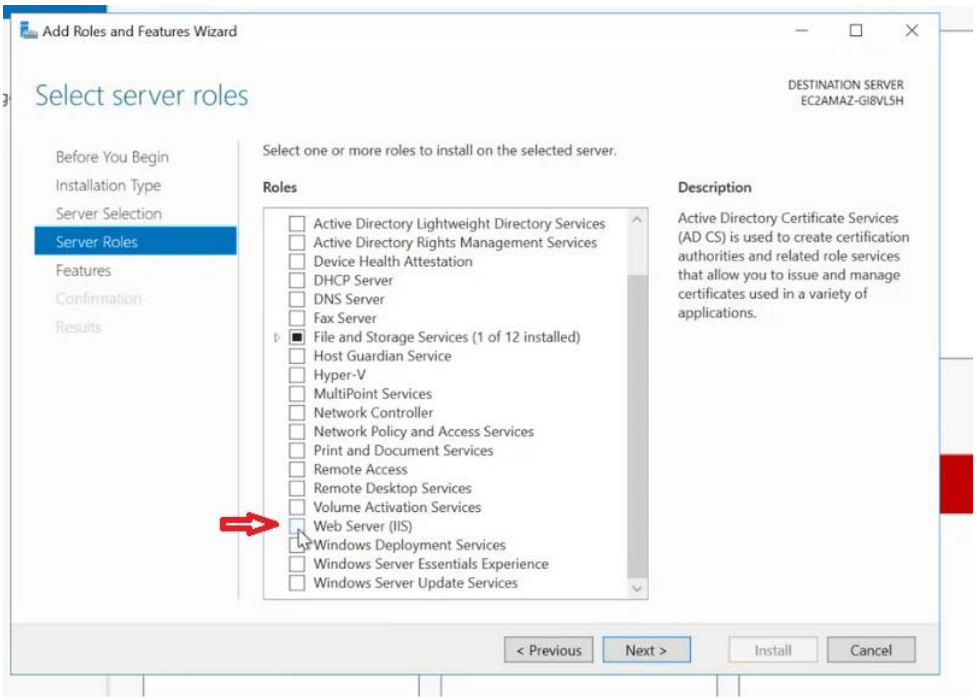


Figura 56 instalación web server.

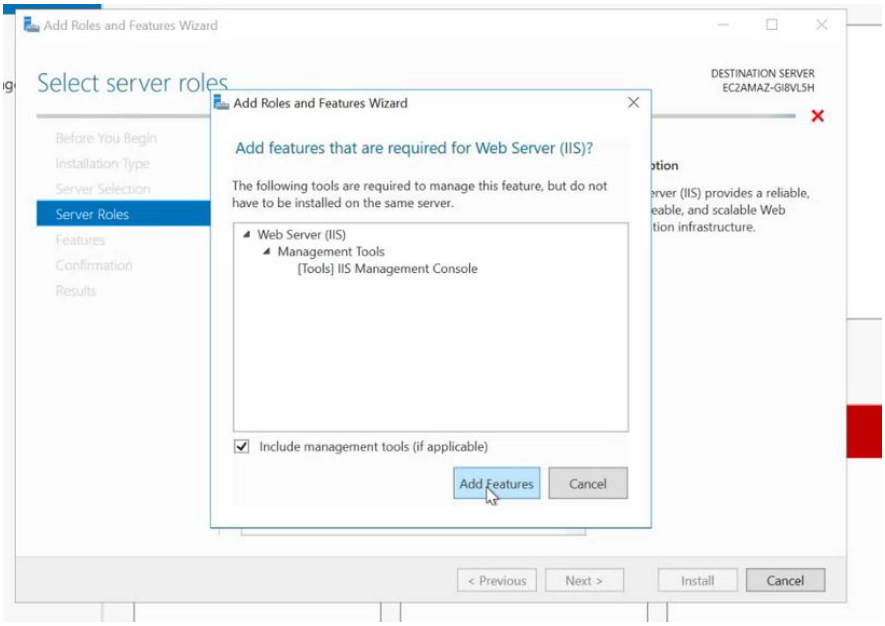
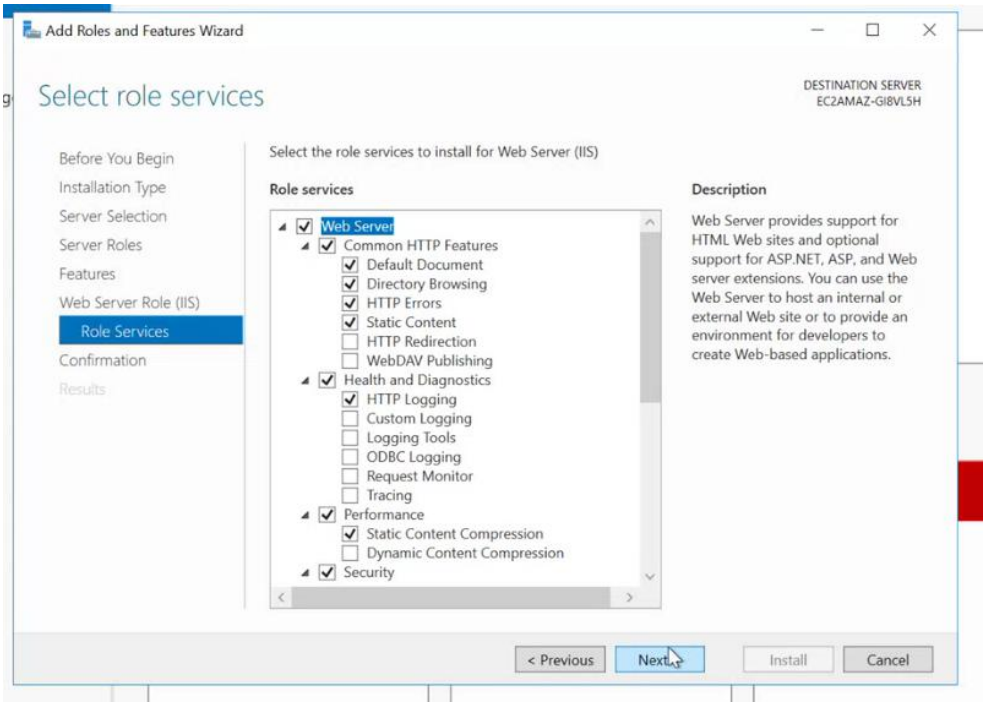


Figura 57 instalación rol.



Instalamos y esperamos que se ejecute la instalación.

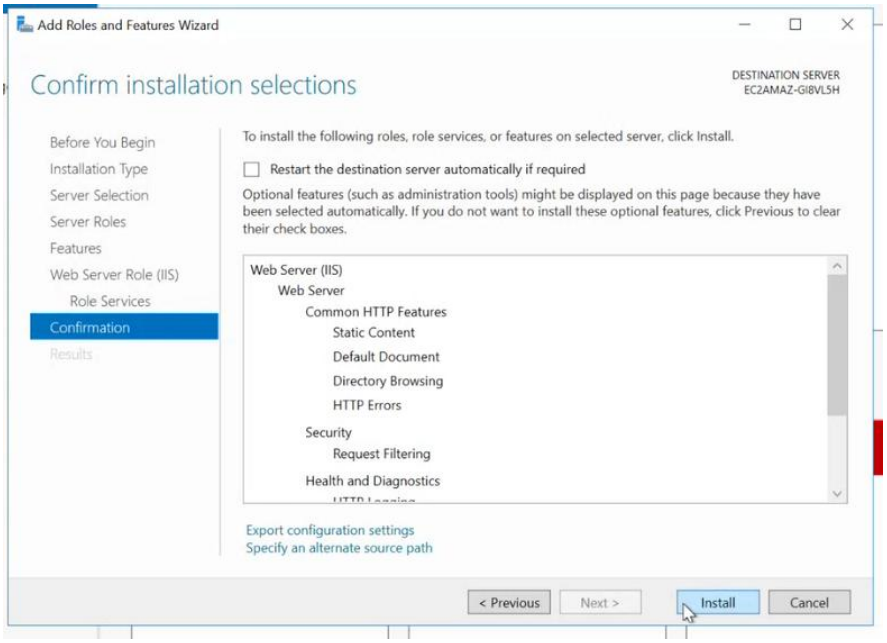
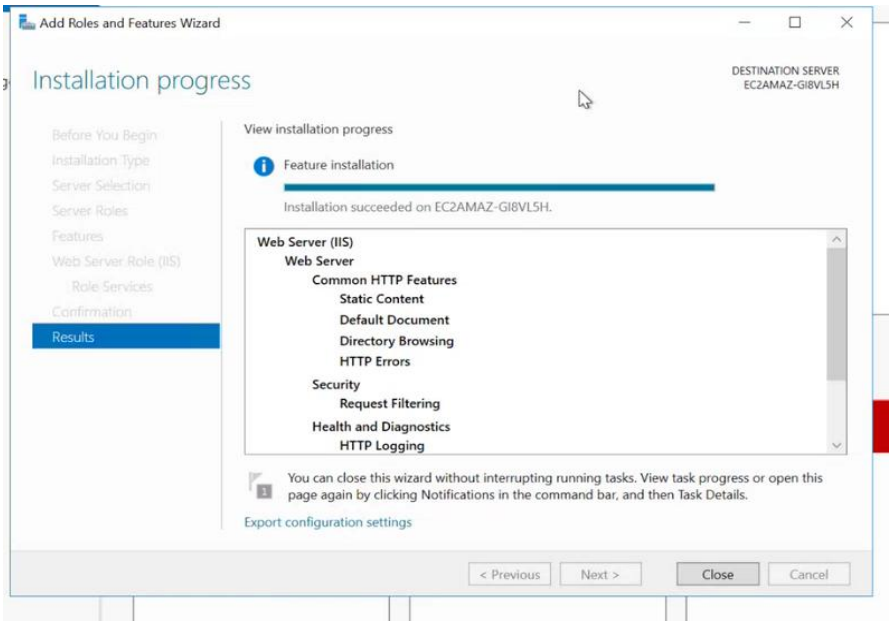
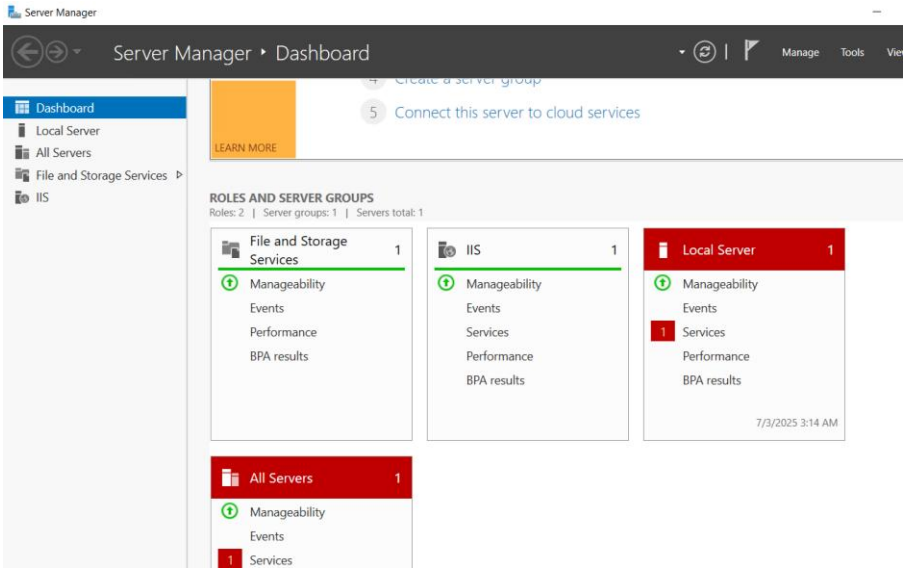


Figura 58 finalización de instalación.



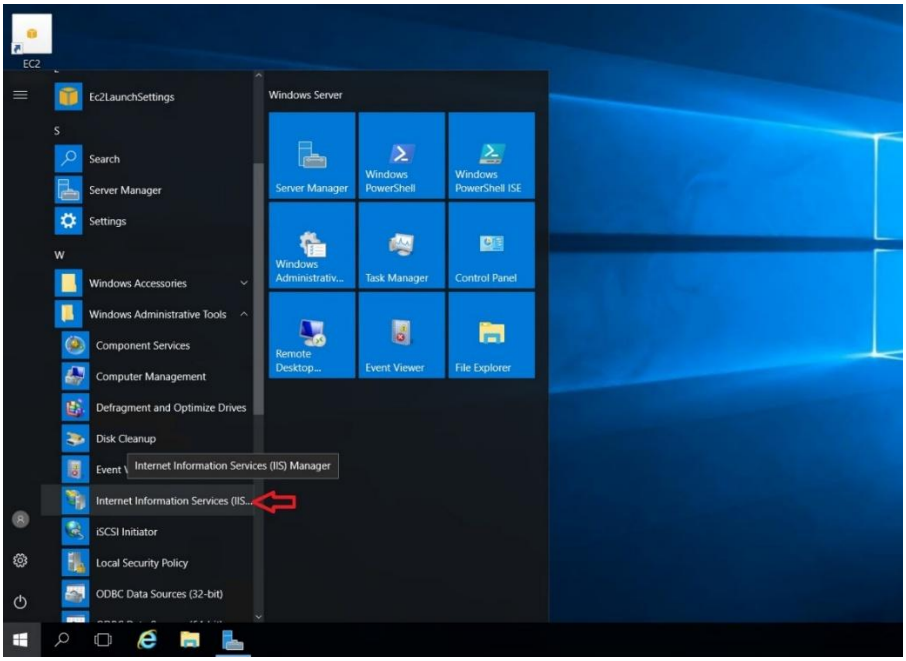
Evidenciamos que ya se instaló correctamente.

Figura 59 visualización instalación.



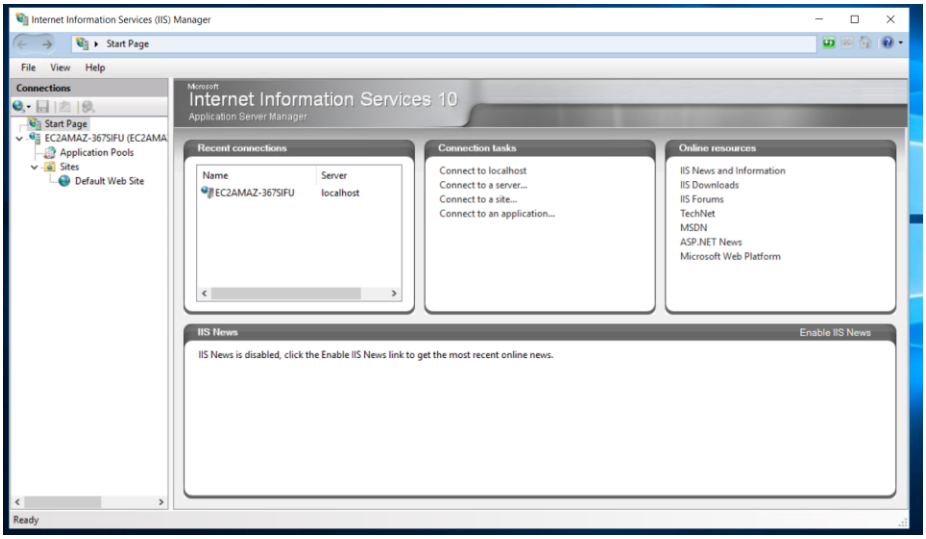
Buscamos el servicio que se instaló y seleccionamos internet information services.

Figura 60 Busqueda servicio instalado.



Observamos nuestro servidor web creado.

Figura 61 servidor creado.



Probamos nuestro sitio desde el servidor, para comprobar que funciona.

Figura 62 prueba servidor.

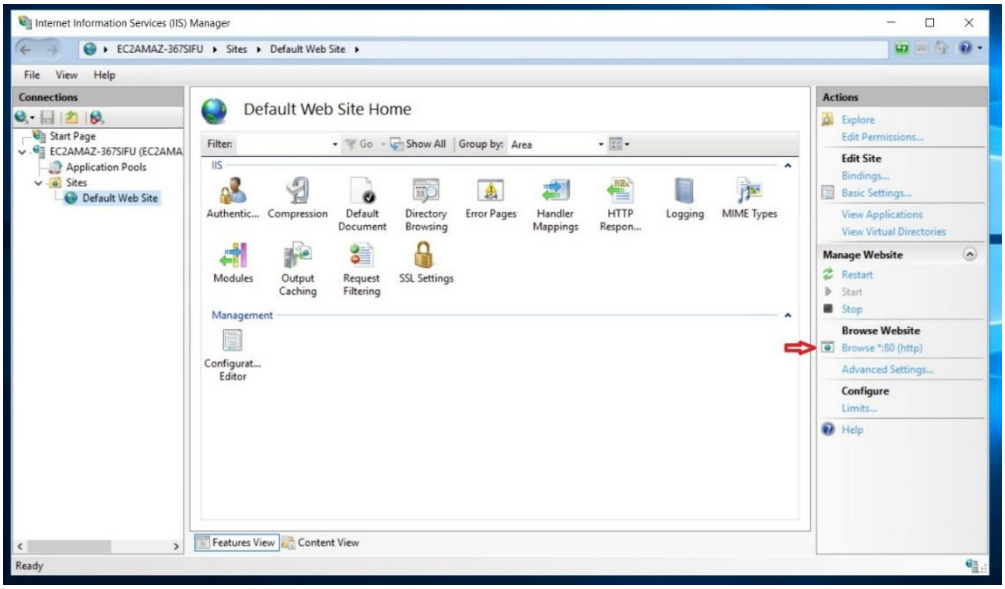
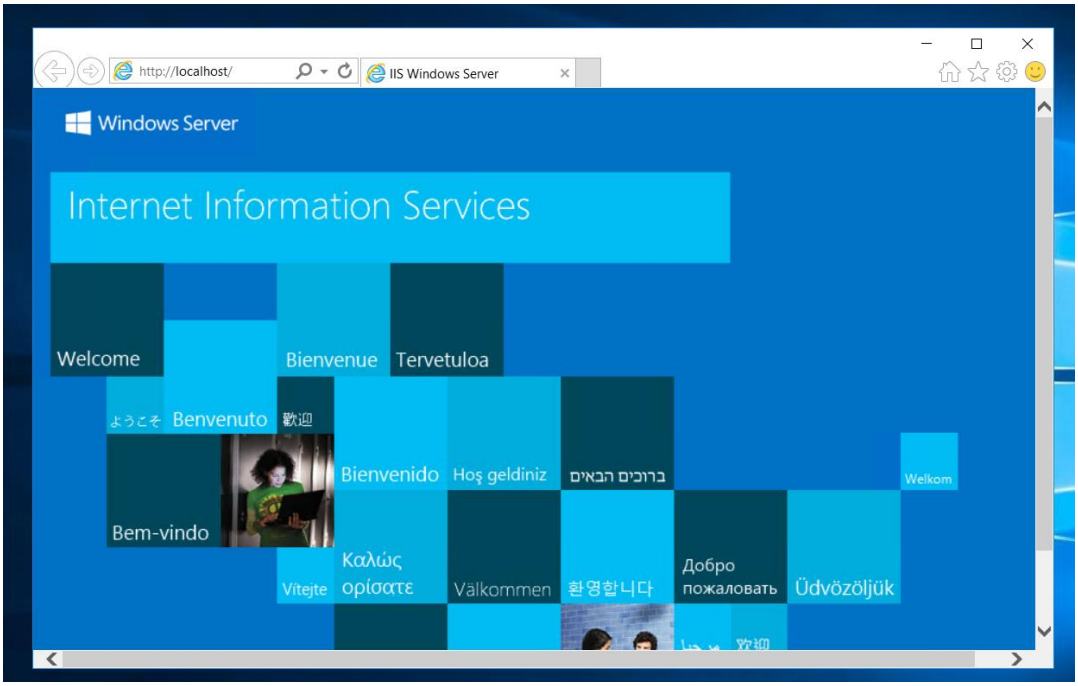
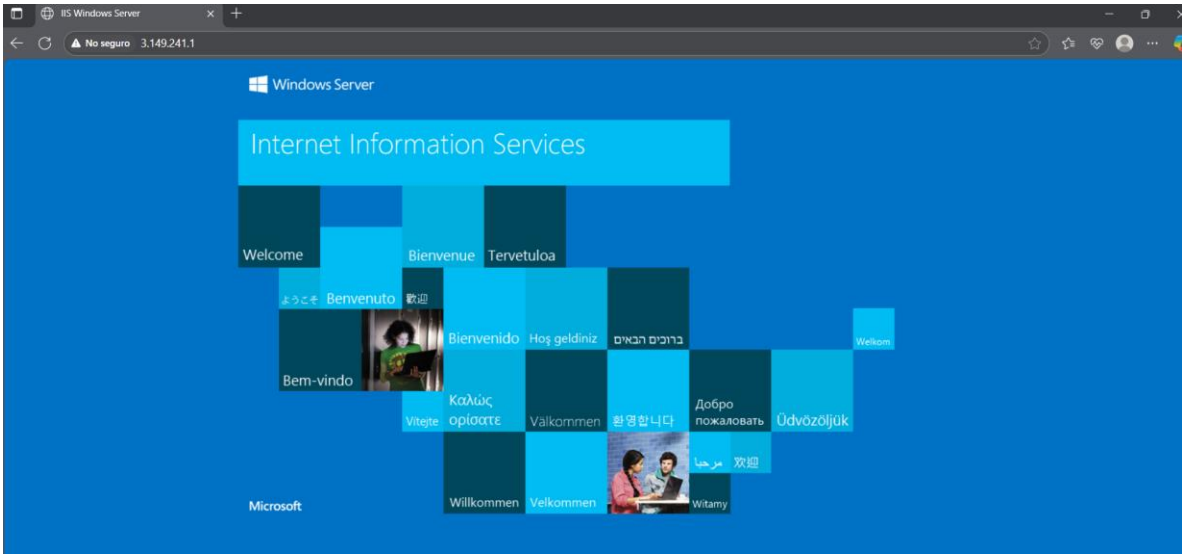


Figura 63 servidor funcionando.



Probamos que nuestro sitio web se vea desde la dirección publica en nuestro navegador.

Figura 64 prueba en navegar web.

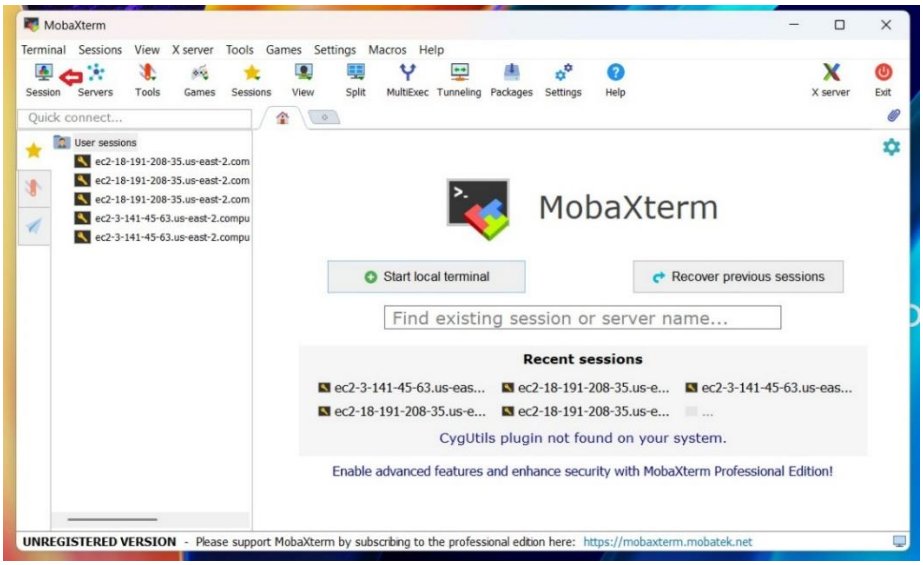


Acceso vía SSH a la instancia Linux

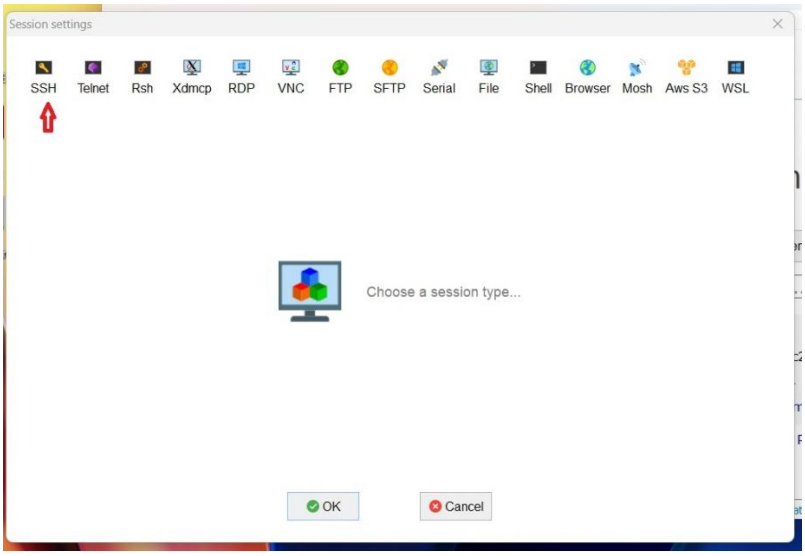
Para conectarnos a la instancia de Linux necesitamos MobaXterm

Abrimos nuestro programa y damos clic en la opción Session.

Figura 65 acceso SSH.



Seleccionamos SSH, colocamos la información que nos pide de nuestra instancia.

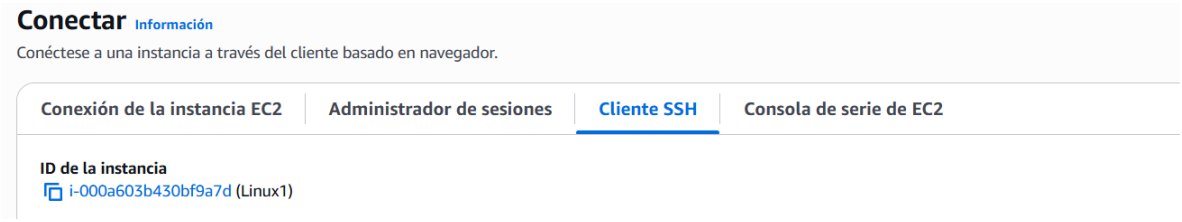


Seleccionamos nuestra instancia y le damos clic en conectar.

Figura 66 conexión.

Seleccionamos cliente SSH.

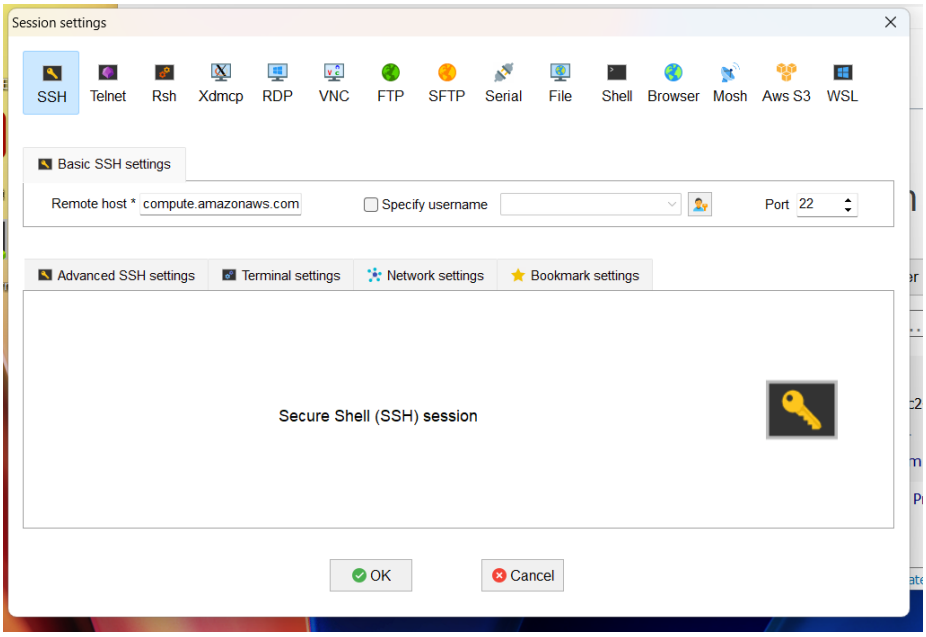
Figura 67 sección cliente SSH.



Copiamos la IP Pública y la pegamos en MobaXterm en la opción de Remote Host.

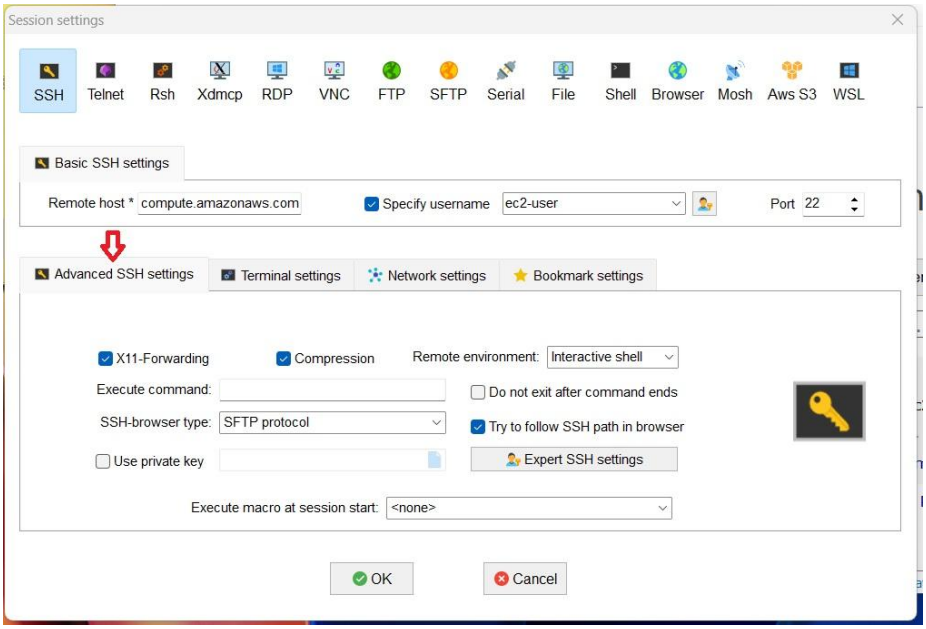
Figura 68 conexión por DNS.

Figura 69 acceso servidor.



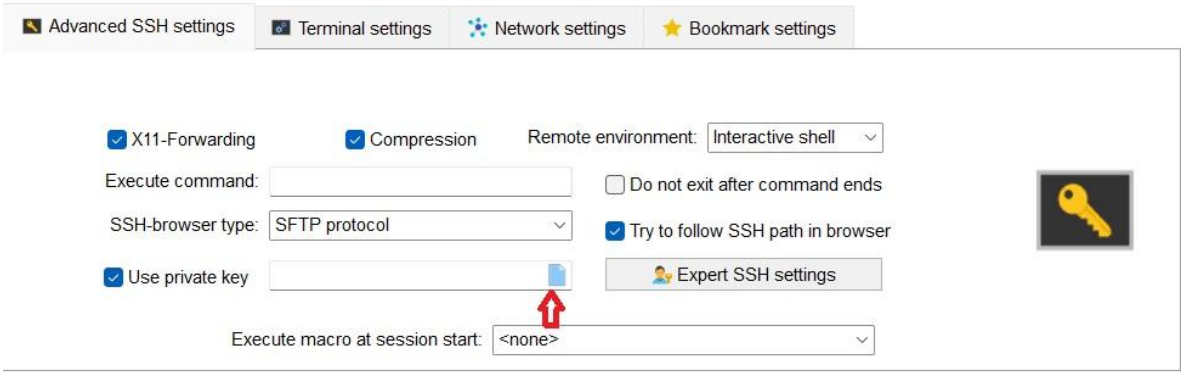
Ingresamos nuestro usuario, que en este caso es (ec2-user) y presionamos la opción Advanced SSH settings.

Figura 70 acceso SSH.

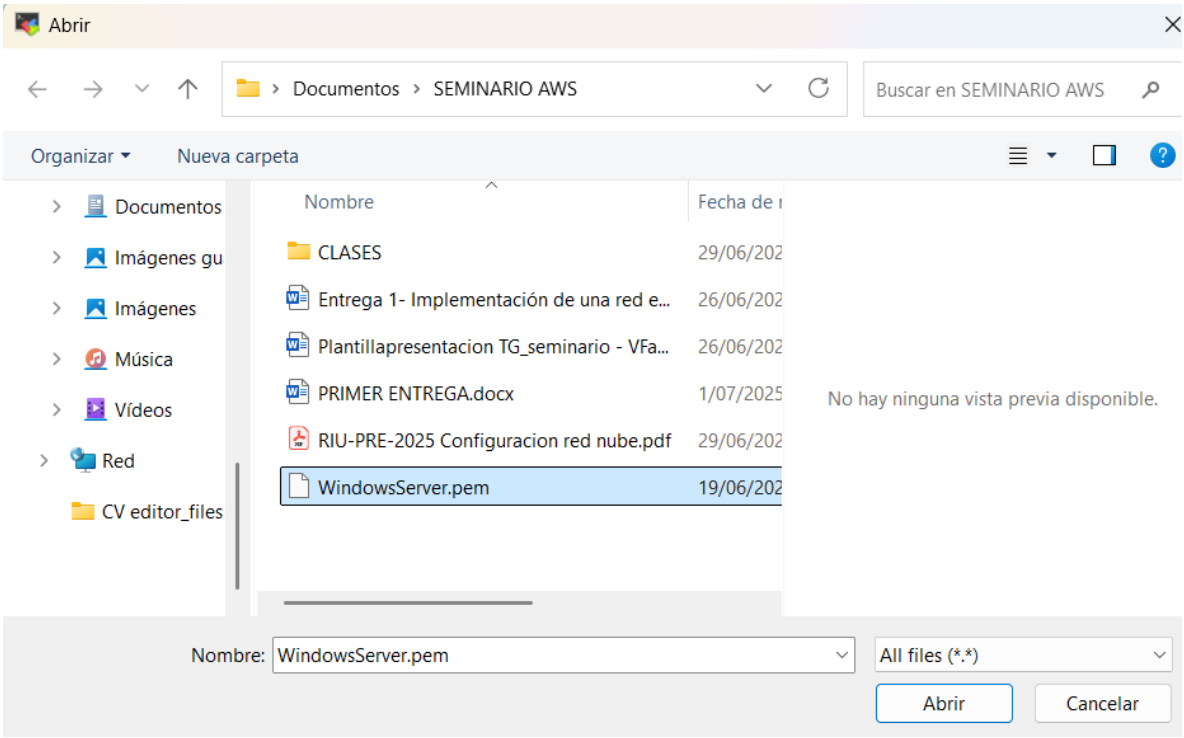


Seleccionamos la casilla para usar la clave privada y buscamos nuestro archivo WindowsServer.pem.

Figura 71 ingreso de archivo .pem.

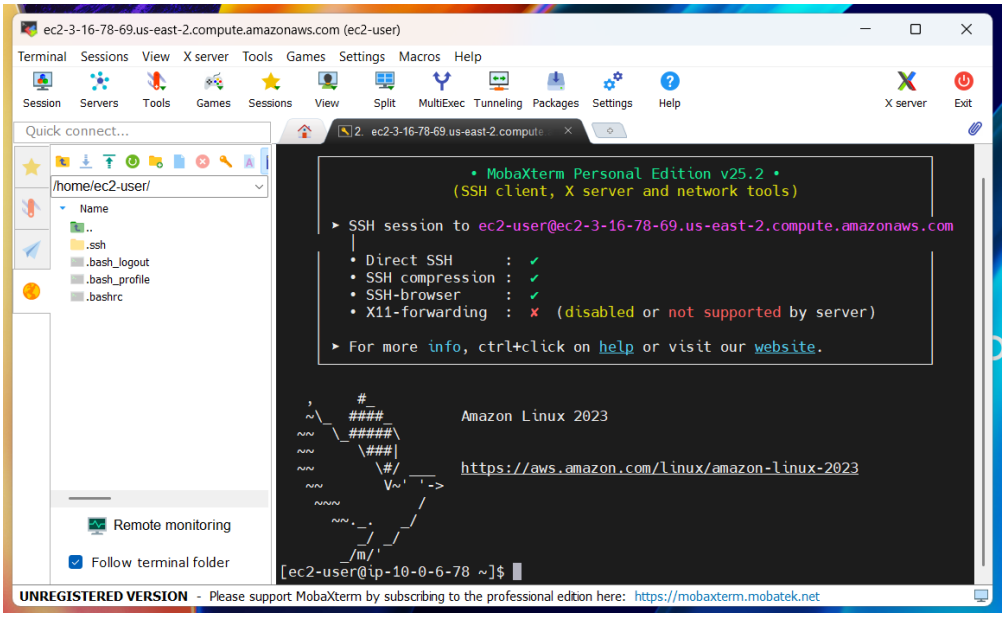


Lo seleccionamos y damos aceptar.



Presionamos OK para acceder.

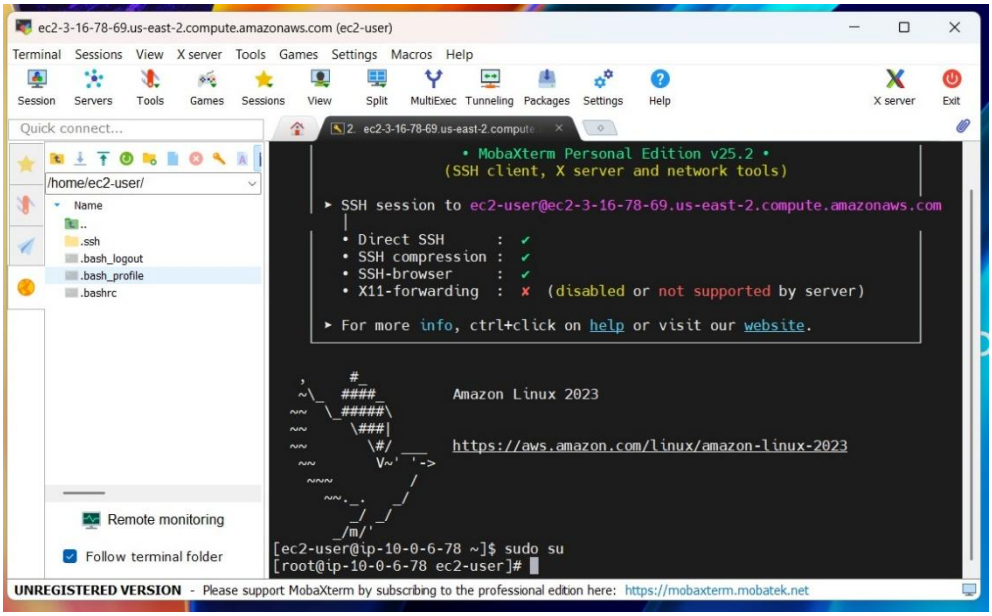
Figura 72 acceso servidor Linux.



Instalación del servidor Apache en Linux.

Ingresamos en comando (sudo su), para volvernos administrador.

Figura 73 modo administración.



Ejecutamos el comando (dnf install httpd) para instalar el Apache.

Figura 74 instalación Apache.

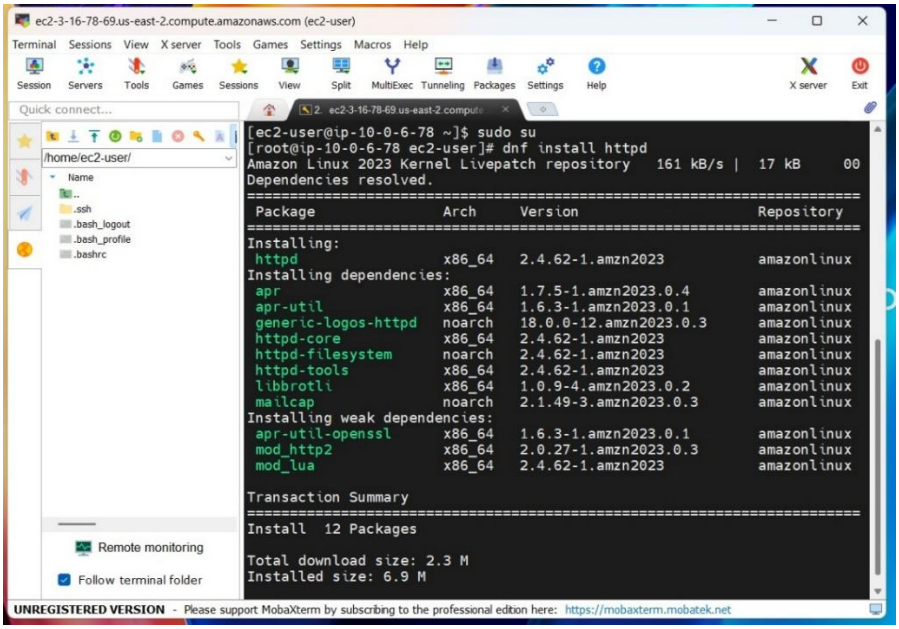
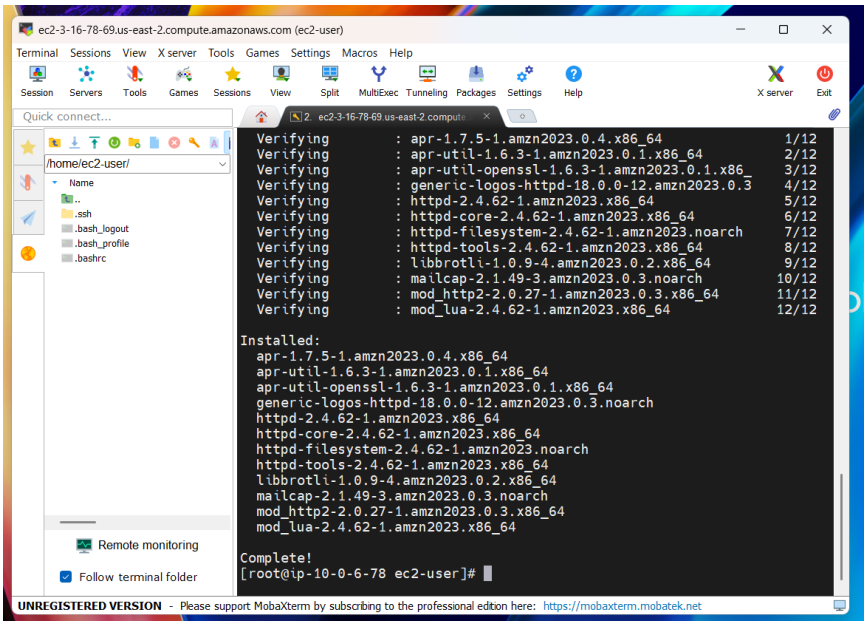
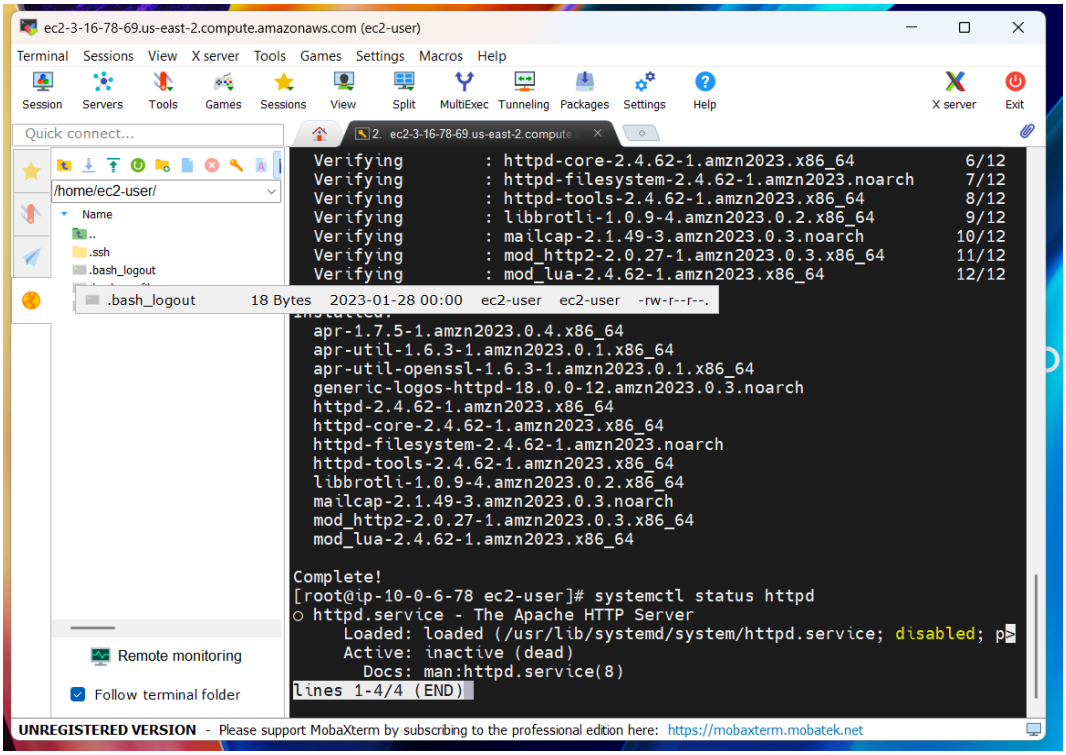


Figura 75 instalación completa.



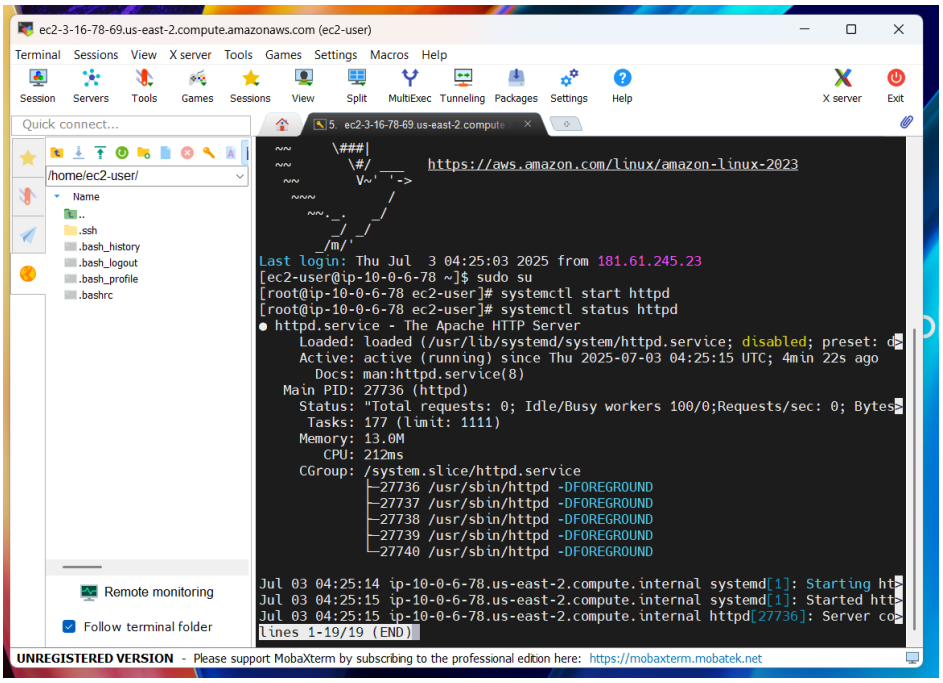
Ingresamos el siguiente comando para verificar que el servidor este corriendo (systemctl status httpd).

Figura 76 verificación estado.



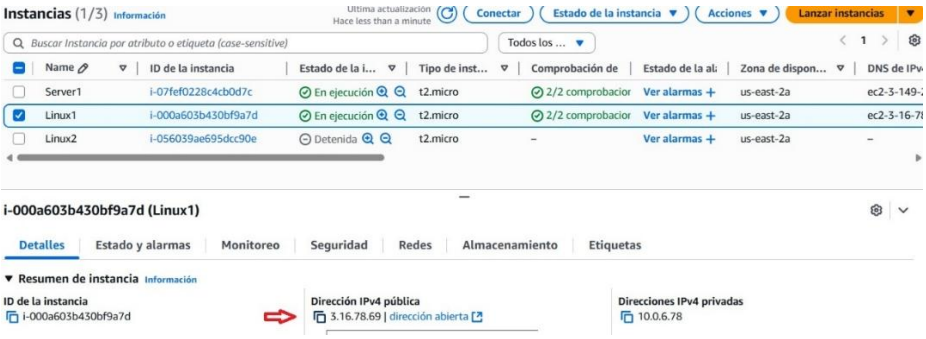
Podemos verificar que esta inactiva. Para activarla usamos el comando(systemctl start httpd). Verificamos que la activación con el comando Systemctl status httpd.

Figura 77 activación.



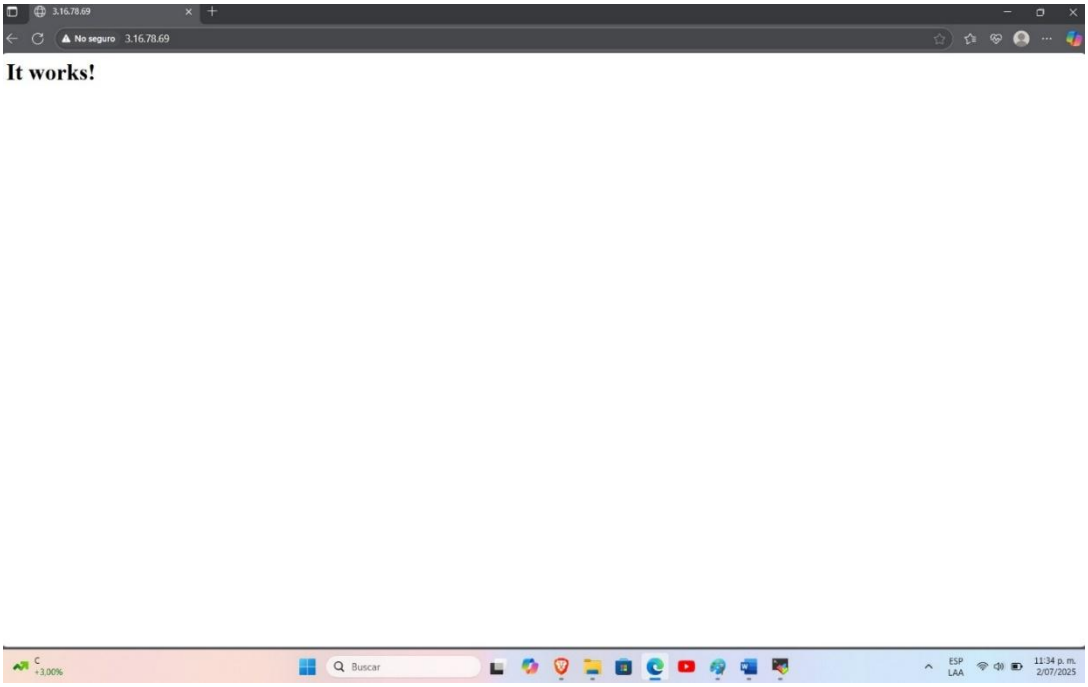
Verificamos el acceso al servidor desde el navegador, copiando nuestra IP pública.

Figura 78 prueba de acceso.



Observamos en nuestro navegador la página que crea por defecto.

Figura 79 prueba en navegador web.

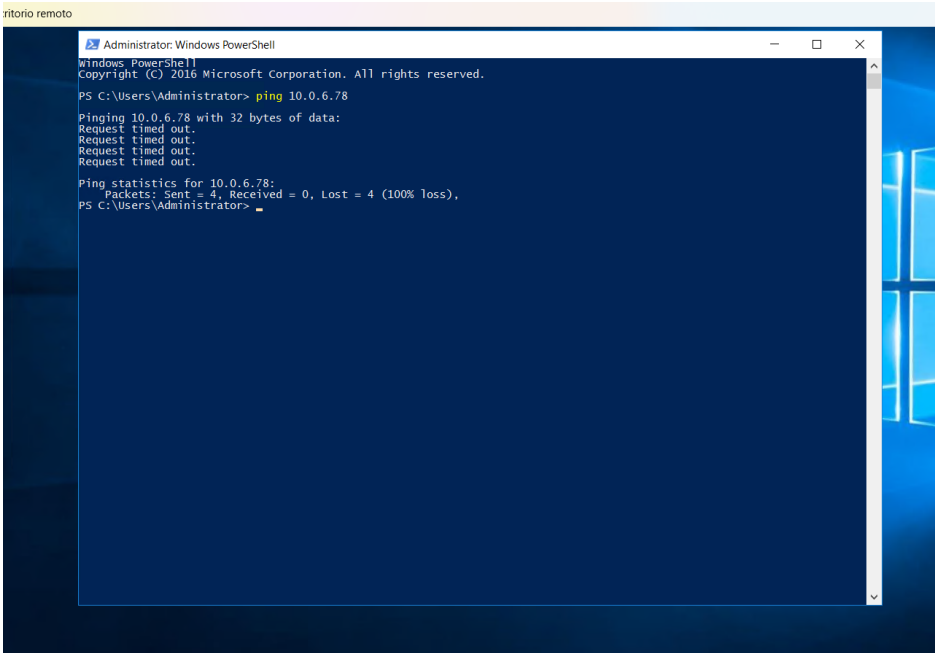


Prueba de conectividad de Windows a Linux.

Para que la conectividad entre ambas instancias por la IP privada sea exitosa, es necesario crear una nueva regla de seguridad con el protocolo ICMP. Sin este no pueden acceder entre ambas.

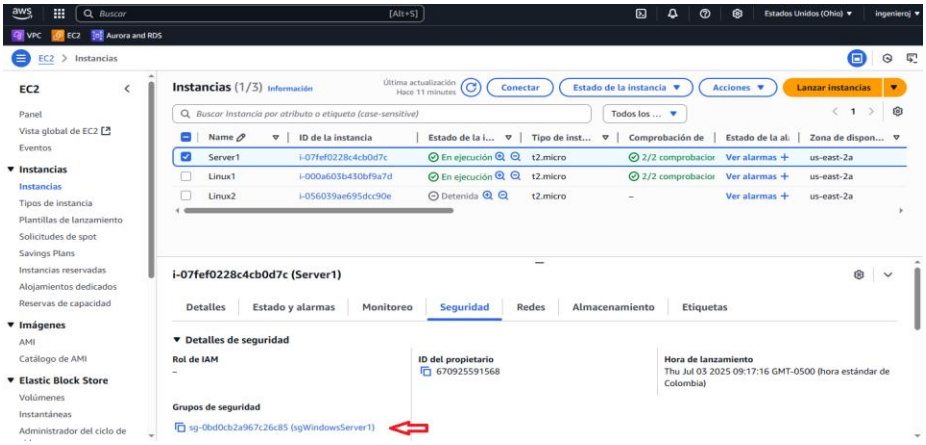
En esta imagen observamos que se ejecuta el ping desde Windows hacia Linux, pero no está establecido el protocolo ICMP y no es exitosa la conexión.

Figura 80 ping sin protocol ICMP.



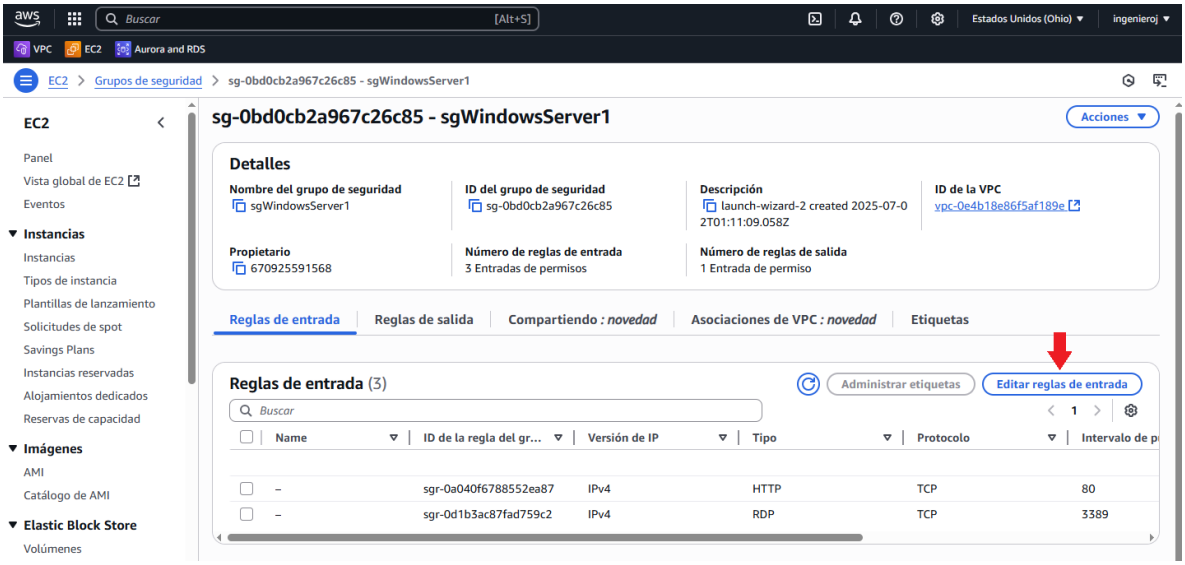
Para habilitar el protocolo ICMP nos dirigimos a nuestras instancias, en la sección de seguridad y seleccionamos grupo de seguridad. Este proceso es el mismo para ambas instancias.

Figura 81 seleccion grupo de seguridad.

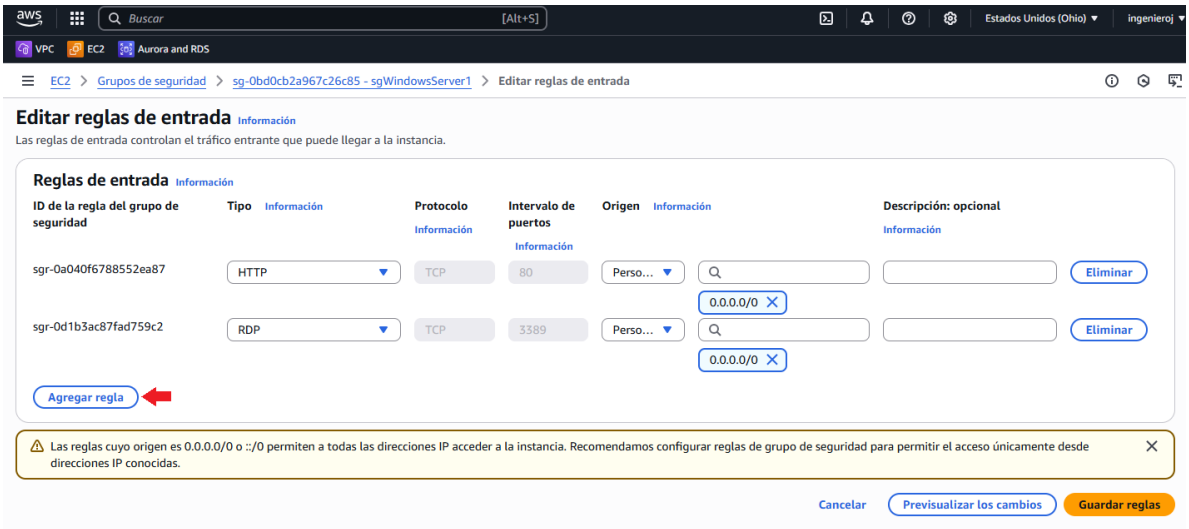


Damos clic en editar reglas de entrada.

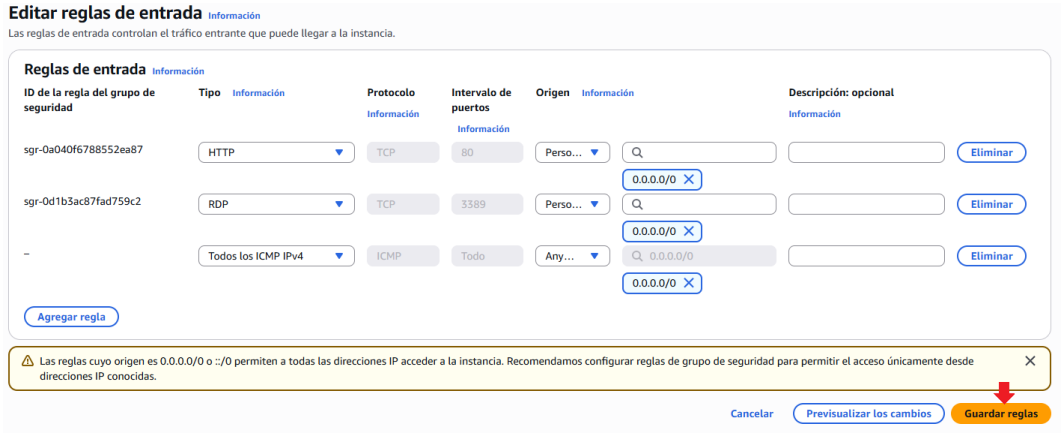
Figura 82 regla de entrada.



Seleccionamos agregar regla e ingresamos la información para que nos permite realizar la conexión por la IP privada.

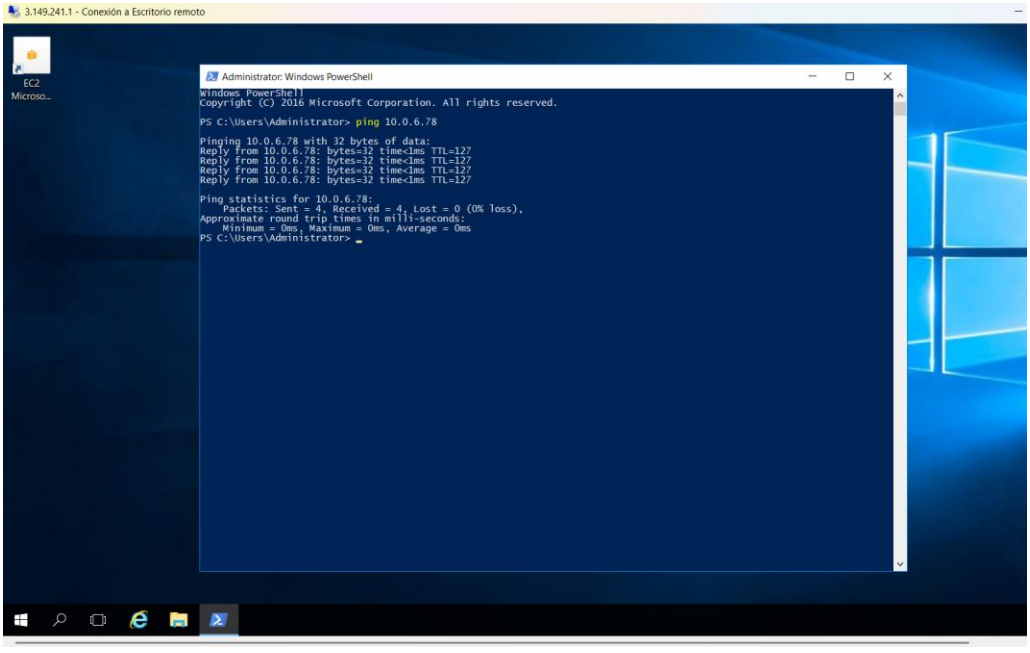


Damos guardar reglas y ejecutamos ping a la IP para revisar si hay conexión.

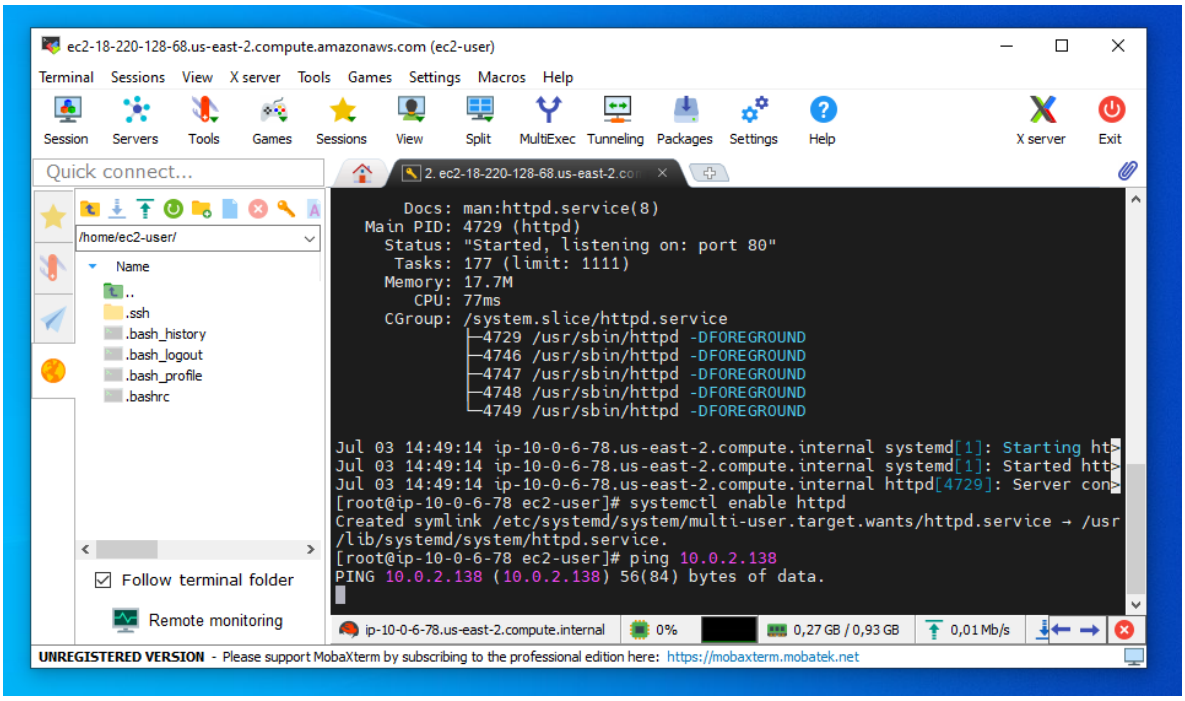


Efectivamente la instancia de Windows pudo comunicarse correctamente con la instancia de Linux usando su IP privada.

Figura 83 prueba de conexión entre instancias.

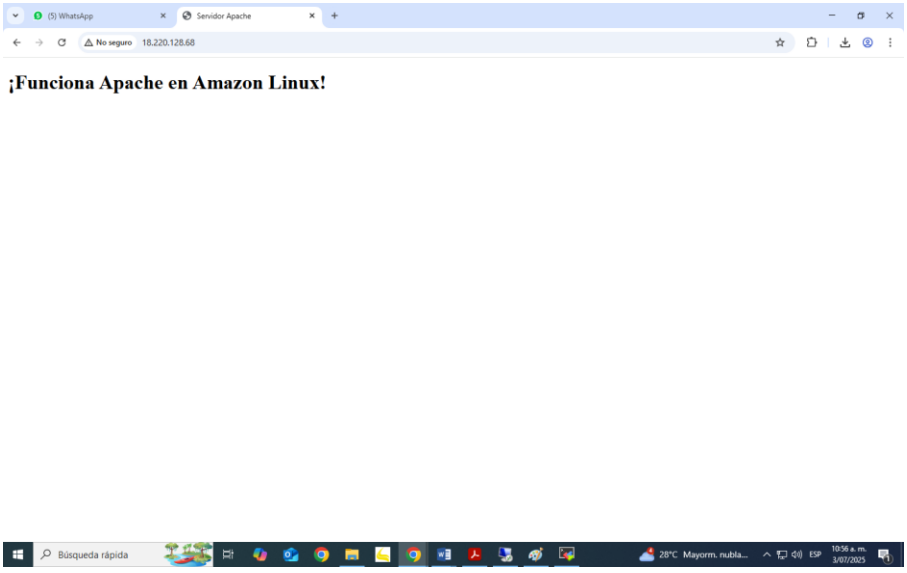


Comprobamos que también hay conexión exitosa de Linux a Windows.



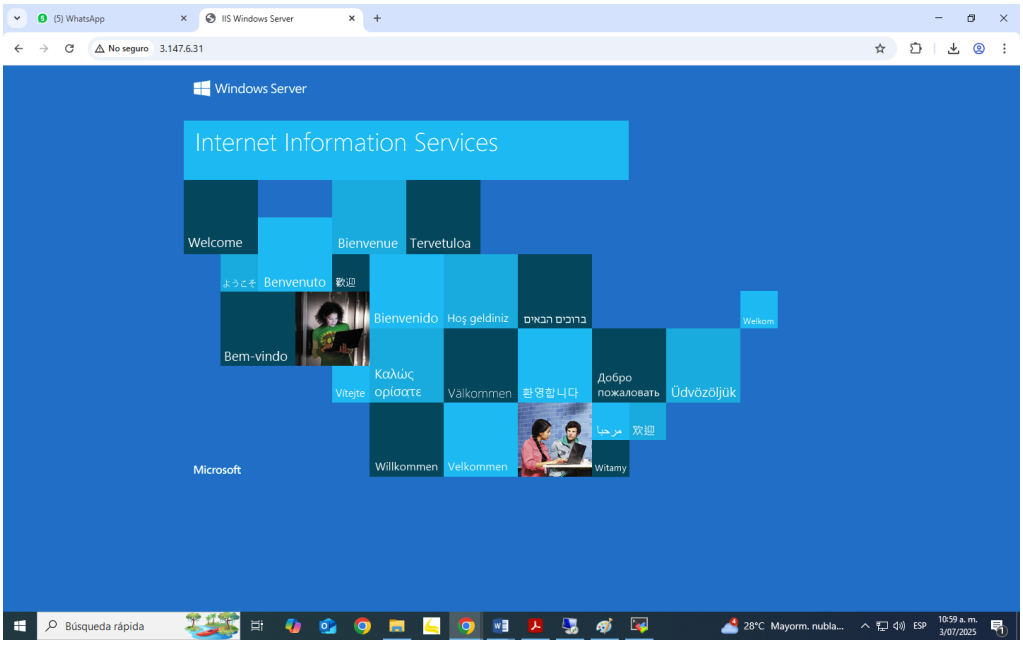
Validación de acceso web desde Linux.

Figura 84 servidor funcionando.



Validación de acceso web desde Windows.

Figura 85 servidor funcionando.



Entrega 2

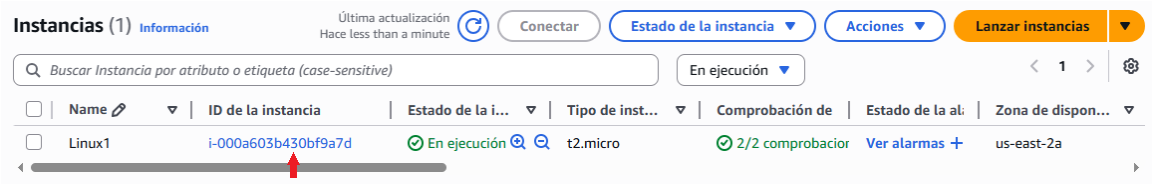
Para la implementación de Balanceador de Carga (ALB), es necesario contar con al menos 2 instancias EC2 configuradas de manera idéntica, para garantizar que el tráfico se distribuya de manera uniforme sin generar errores; manteniendo la alta disponibilidad y tolerancia a fallos.

Creación de Instancias EC2

Teniendo en cuenta que ya tenemos creado una instancia con Amazon Linux y Apache instalado, vamos a crear una imagen (AMI) de esa instancia, lo que permitirá lanzar una segunda instancia con la misma configuración.

Seleccionamos nuestra ID de la instancia, nos dirigimos a almacenamiento y procedemos a seleccionar nuestro volumen.

Figura 86 selección ID.



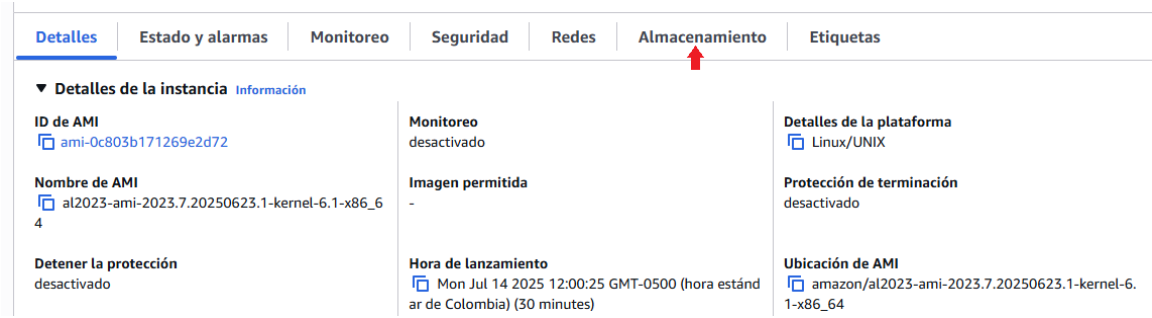
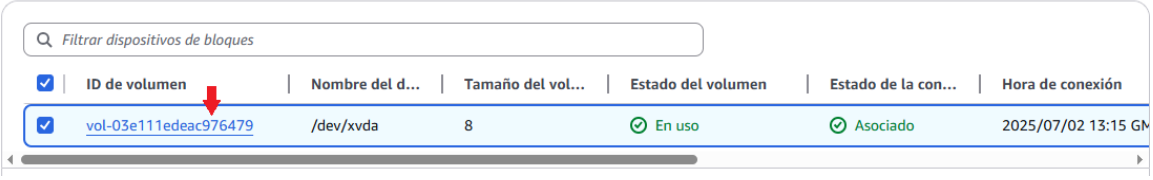
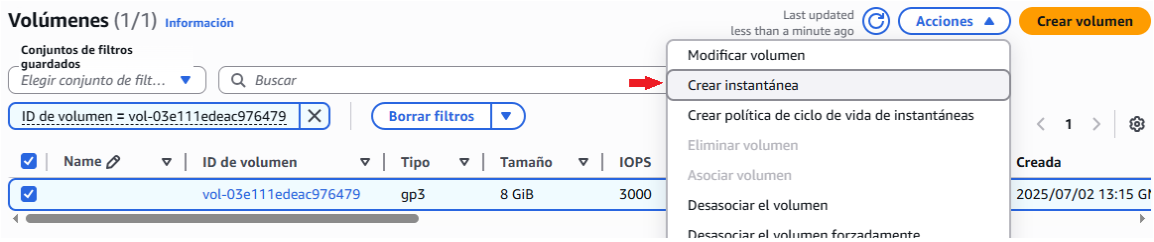


Figura 87 selección de volumen.

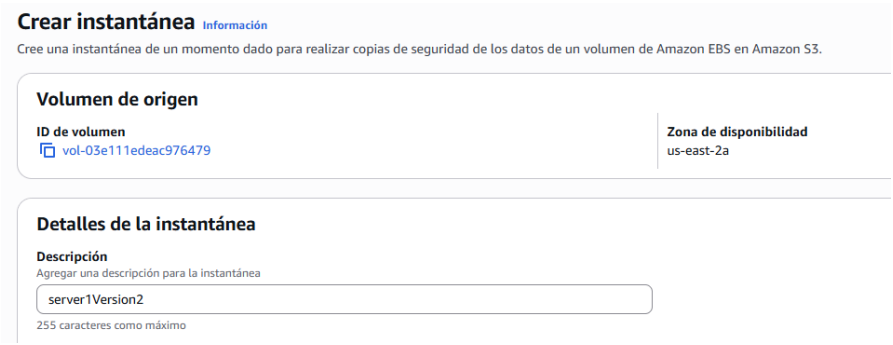


Seleccionamos el volumen, damos en acciones y creamos nuestra instantánea.



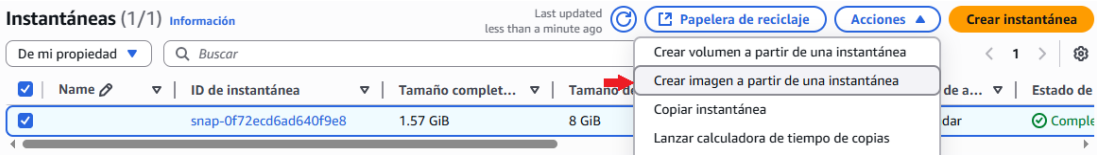
Asignamos el nombre a nuestra instantánea y damos crear.

Figura 88 creación instantánea.



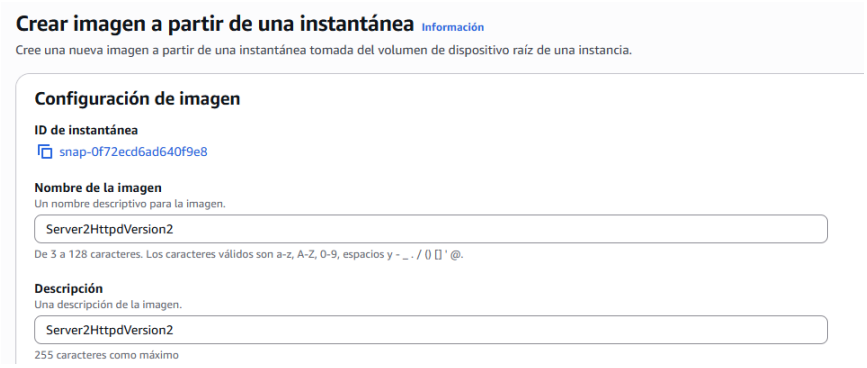
Una vez creada damos en acciones para seleccionar la opción de crear imagen a partir de una instantánea.

Figura 89 creación de imagen.



Le damos un nombre y descripción, revisamos el nombre de dispositivo de raíz que sea el mismo nombre con el que crea los volúmenes, lo copiamos y pegamos.

Figura 90 creación de AMI.



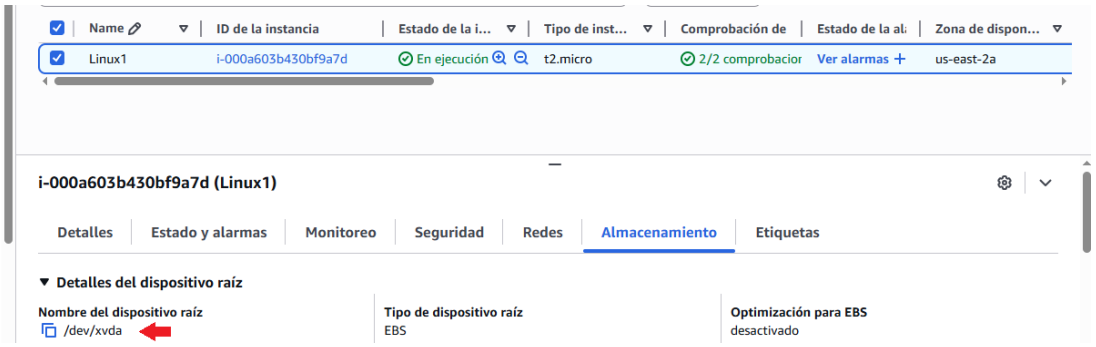
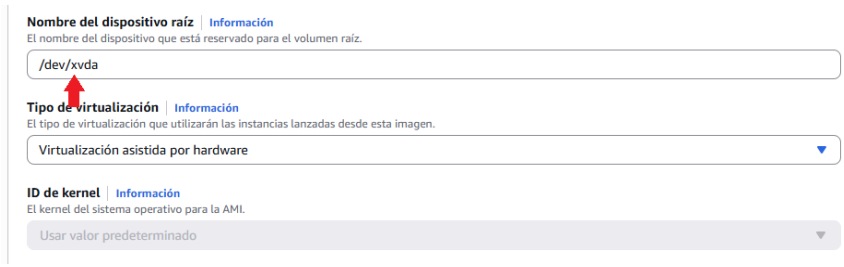
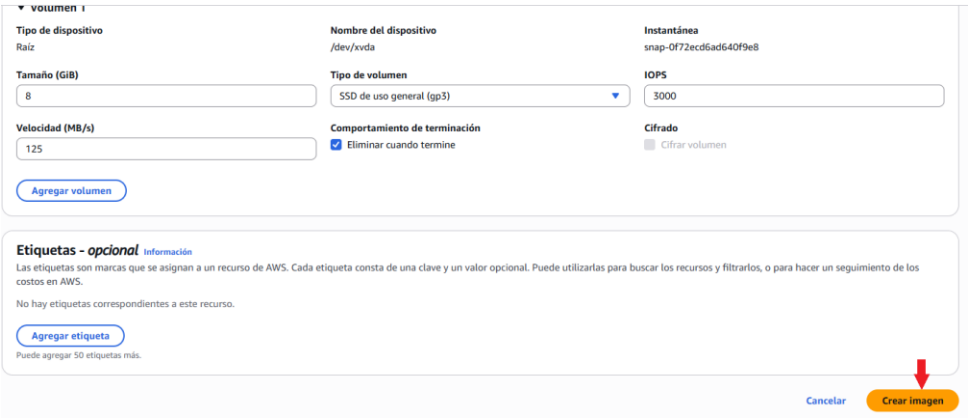


Figura 91 identificación dispositivo raíz .

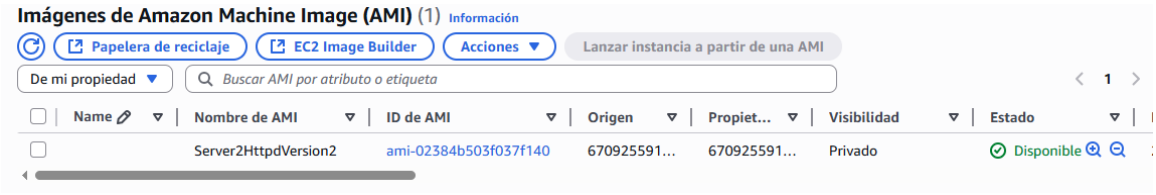


Dejamos por defecto la demás información y damos crear imagen.



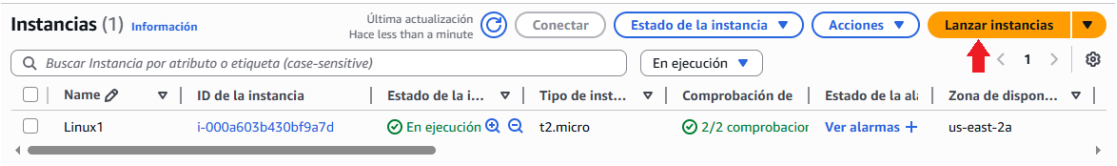
Verificamos que nuestra AMI se creó correctamente.

Figura 92 Figura 90 creación de AMI.



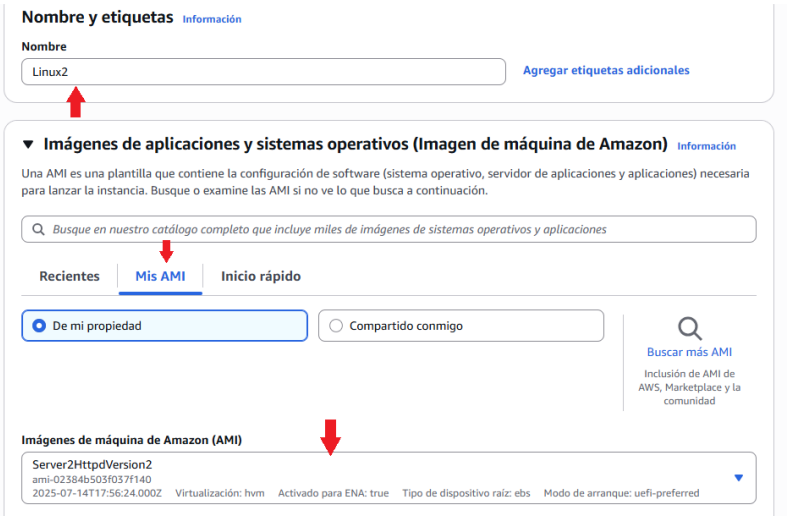
Procedemos a crear nuestra segunda instancia con nuestra AMI personalizada.

Figura 93 creación instancia.

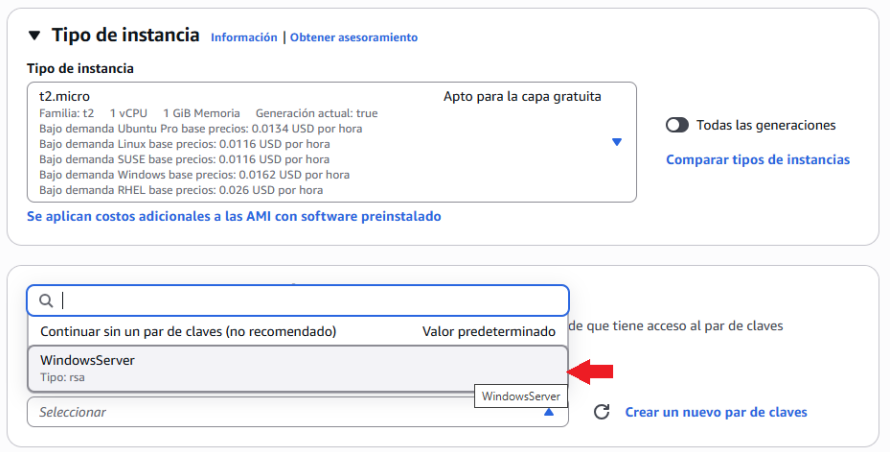


Asignamos el nombre, seleccionamos la opción de mis AMI y automáticamente nos aparece a AMI que creamos anteriormente.

Figura 94 seleccion de AMI personalizada.



En el tipo de instancia lo dejamos en t2.micro y elegimos nuestro par de claves que ya tenemos creado.

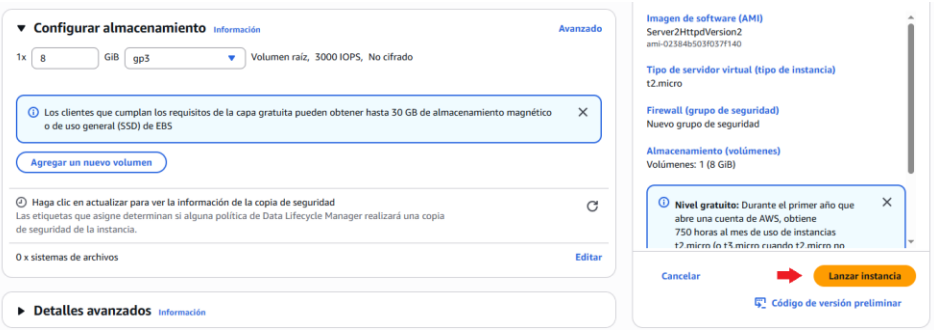


Configuramos la red, para que tenga acceso a nuestra VPC, asignamos la subred publica y damos habilitar para que se asigne la IP publica automáticamente.

Agregamos una nueva regla de seguridad para que podamos acceder al servidor por el puerto 80 desde cualquier lugar.

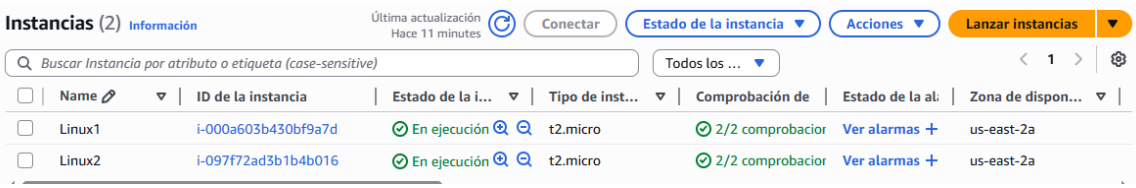
Figura 95 regla de entrada Puerto 80.

Dejamos lo demás por defecto y presionamos en lanzar instancia.



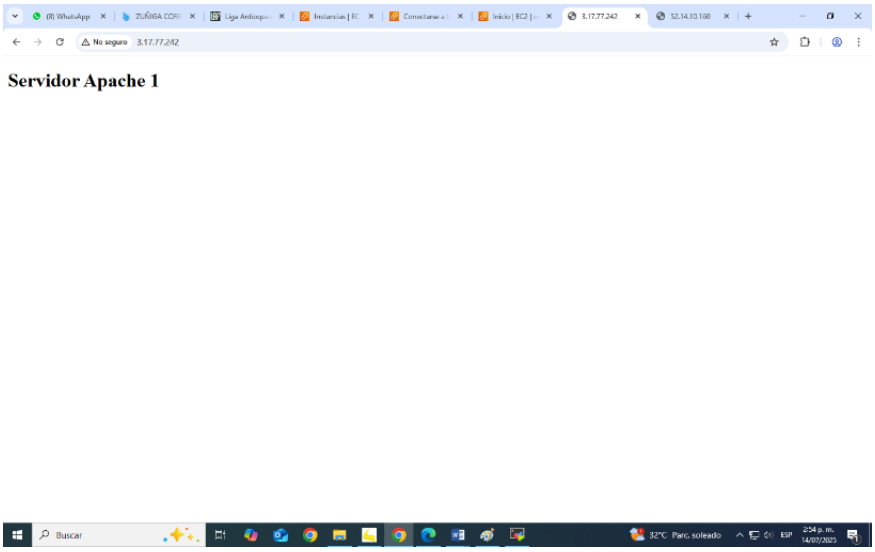
Efectivamente se creó nuestra instancia Linux 2.

Figura 96 Segunda instancia creada de la AMI.



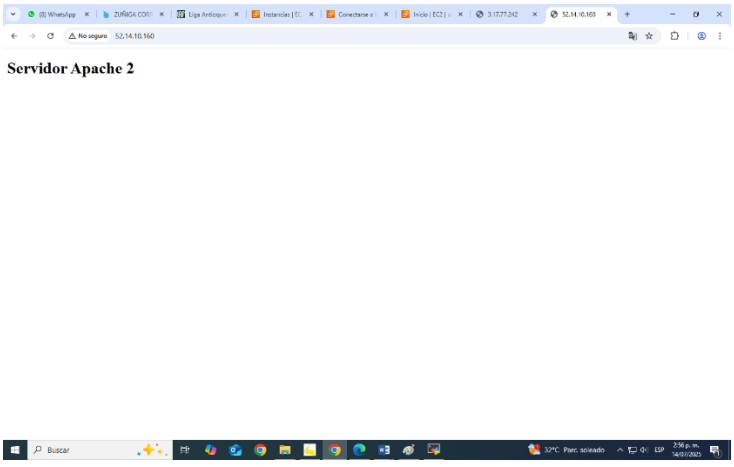
Servidor 1 en correcto funcionamiento.

Figura 97 prueba en navegar web.



Servidor 2 en correcto funcionamiento.

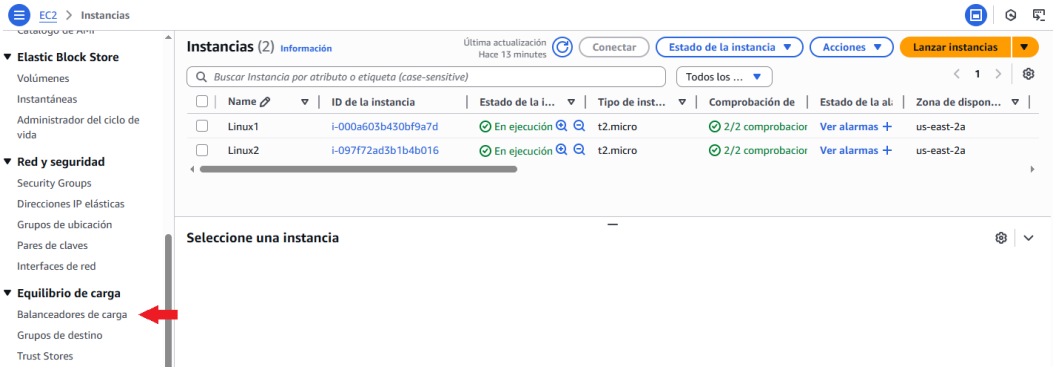
Figura 98 prueba en navegar web.



Con las instancias funcionando de manera correcta, continuamos con la creación del servicio de balanceador de carga(ALB), el cual permitirá balancear el tráfico HTTP a través de múltiples zonas de disponibilidad.

Dentro de EC2 nos dirigimos a equilibrio de carga y seleccionamos Balanceadores de carga.

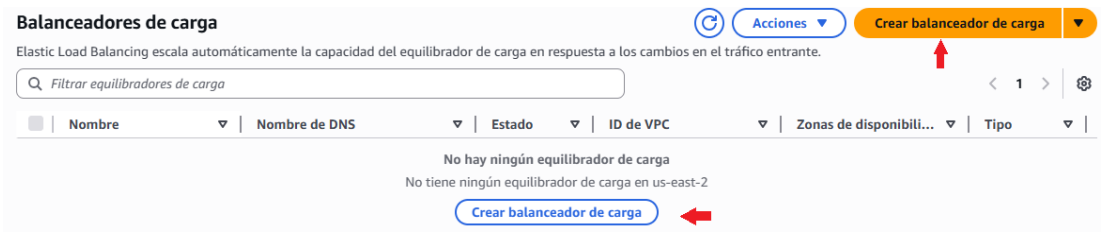
Figura 99 selección balanceadores de carga.



Creación de Balanceador de Carga

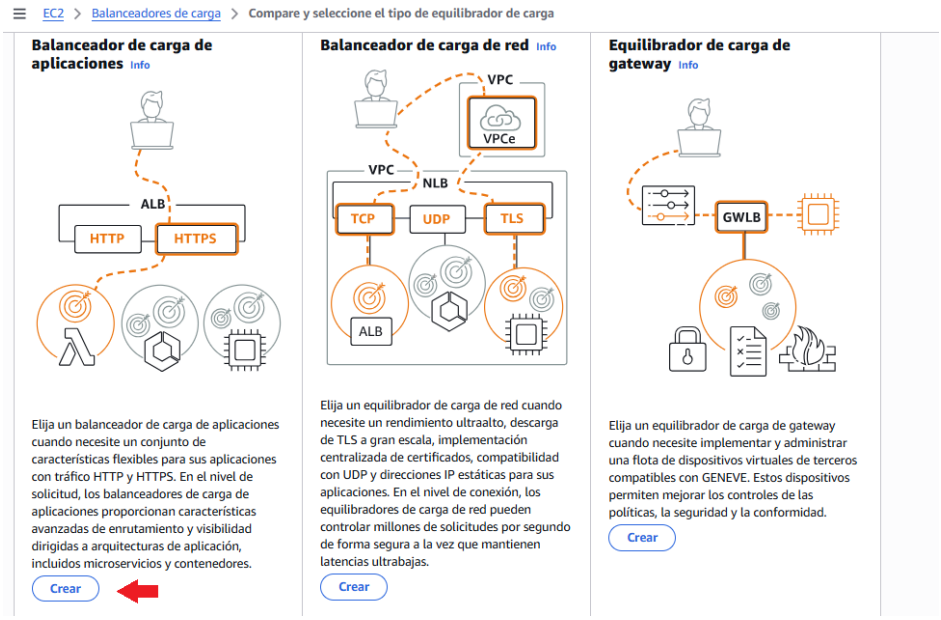
Damos en crear Balanceador de carga.

Figura 100 creación balanceador.



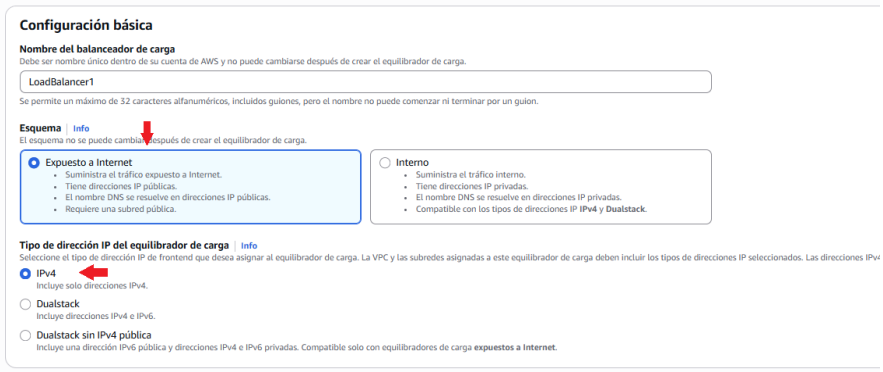
Seleccionamos el tipo de balanceador a crear, que en nuestro caso es balanceador de carga de aplicaciones.

Figura 101 selección tipo balanceador.



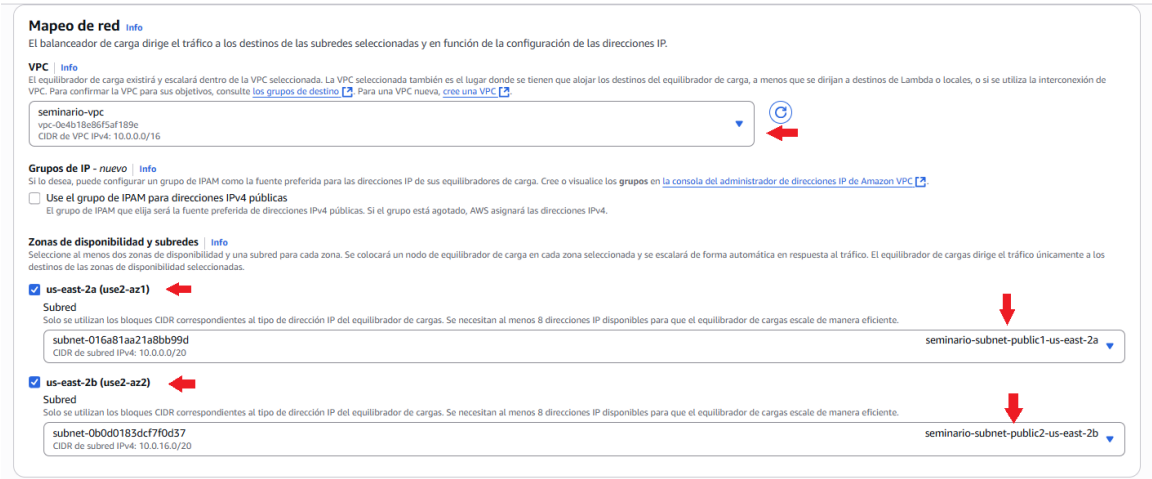
Iniciamos con la configuración de nuestro balanceador, asignamos un nombre, elegimos que será expuesto a internet y que el tipo de dirección IP del balanceador será versión 4.

Figura 102 configuración balanceador.



Para la configuración de red seleccionamos nuestro VPC creado anteriormente y seleccionamos las dos zonas de disponibilidad (data centers) dentro de la región, para distribuir el tráfico entre ambas instancias. Además, elegimos las IP públicas para tener acceso a ellas.

Figura 103 seleccion de VPC y zonas de disponibilidad.



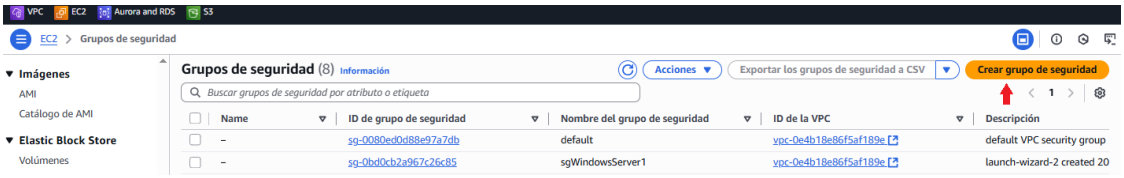
Por seguridad se recomienda crearle su propio grupo de seguridad al balanceador, que permita el tráfico HTTP (puerto 80) desde cualquier IP.

Para esto nos dirigimos en grupos de seguridad.

Figura 104 grupos de seguridad.

Damos en crear grupo de seguridad.

Figura 105 creación grupo de seguridad.

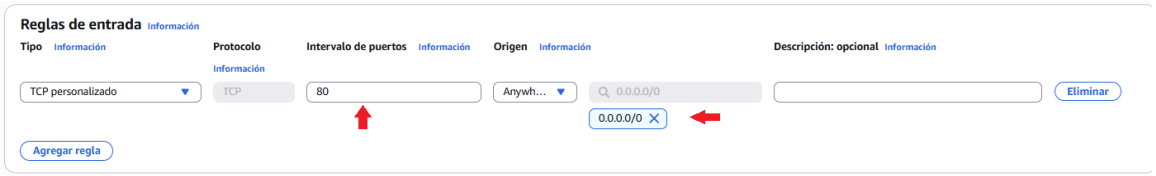


Le asignamos un nombre a nuestro grupo de seguridad y elegimos nuestra VPC ya creada.

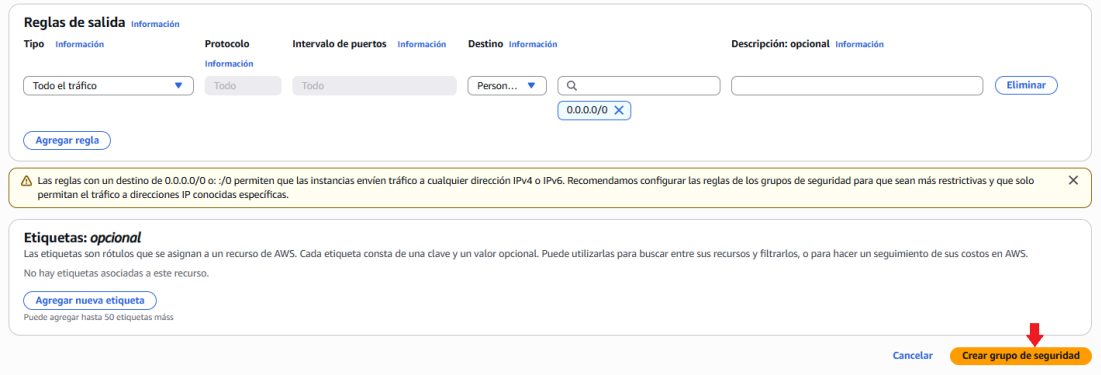
Figura 106 configuración grupo de seguridad.

Agregamos la regla de entrada para que nos permita el acceso por el puerto 80 desde cualquier lugar.

Figura 107

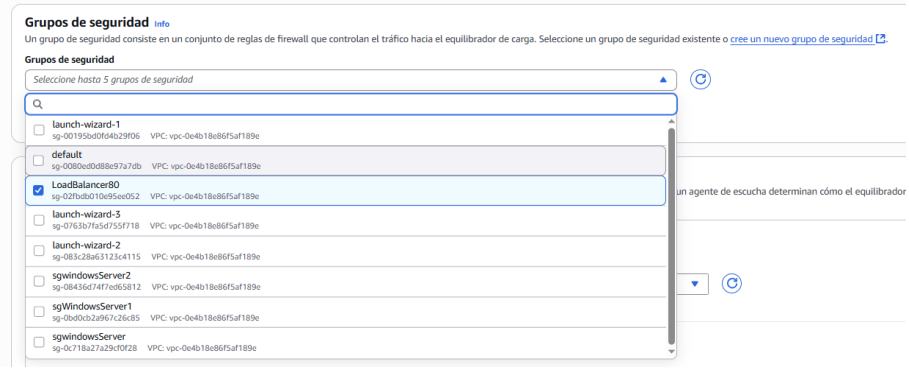


Dejamos por defecto la regla de salida y damos en crear grupo de seguridad.



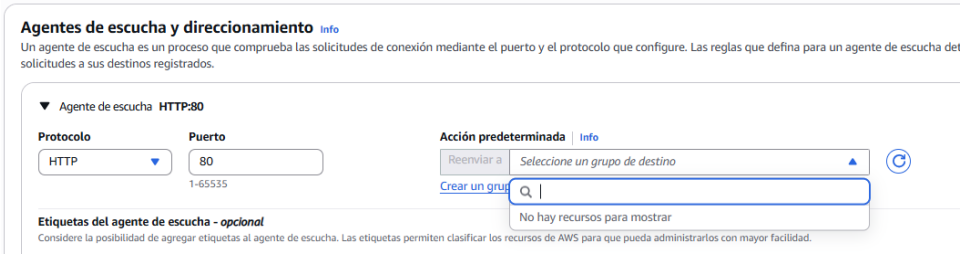
Una vez creado el grupo de seguridad, continuamos con la elaboración de nuestro balanceador eligiendo el grupo de seguridad que acabamos de realizar.

Figura 108 creación balanceador.



En esta etapa se habilita el acceso por el puerto 80 de forma predeterminada. Además, como aún no existe, se procede a crear un grupo de destino que permitirá al balanceador enrutar el tráfico correctamente hacia las instancias EC2 disponibles.

Figura 109 puerto 80 habilitado.



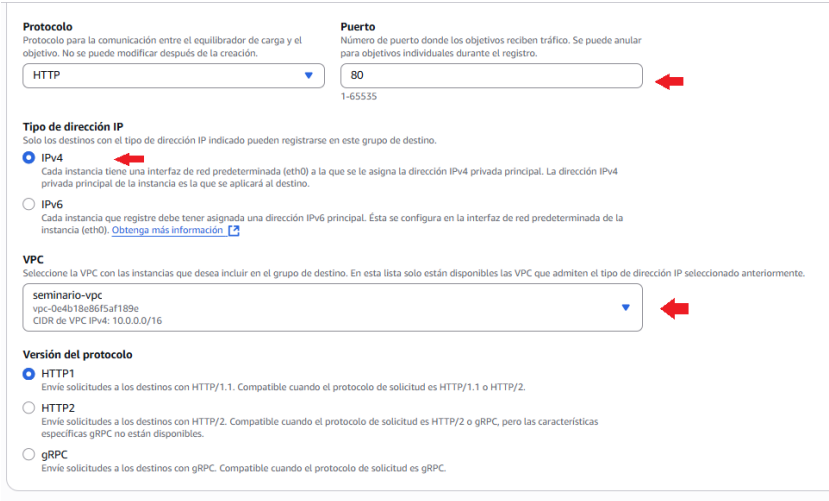
Seleccionamos crear un grupo de destino.

Elegimos nuestro grupo de destino, que en nuestro caso serán nuestras dos instancias creadas y asignamos un nombre.

Figura 110 creación grupo destino.

Dejamos por defecto para que las peticiones se envíen por el puerto 80, nuestro tipo de dirección IP es IPv4, nuestro VPC es seminariovpc.

Figura 111 configuración VPC, puerto y tipo dirección.

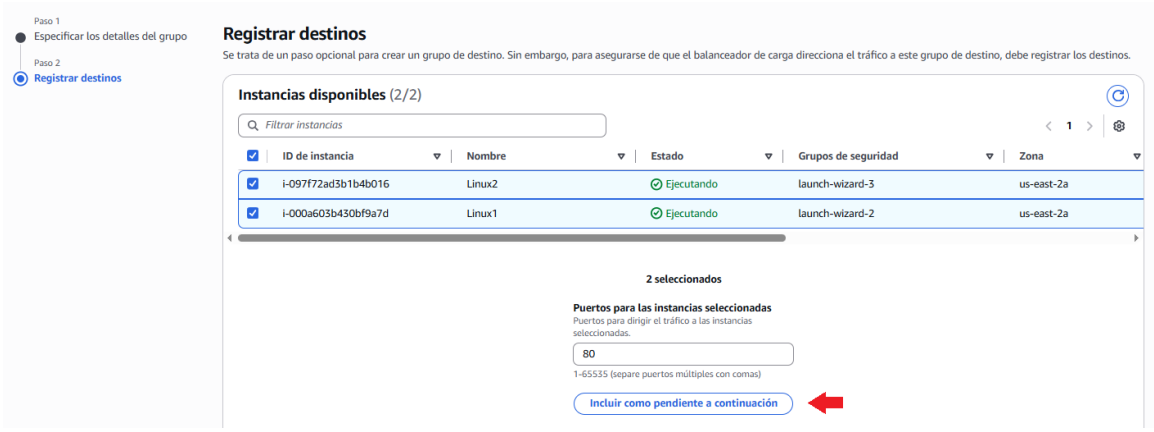


Configuramos los parámetros de comprobacion de estado para monitorear la disponibilidad de las instancias. Se utiliza el puerto de tráfico (80), la ruta / y el código de éxito 200, lo que permite al balanceador determinar si una instancia está saludable antes de enviarle tráfico.

Figura 112 parametros de comprobacion de estado.

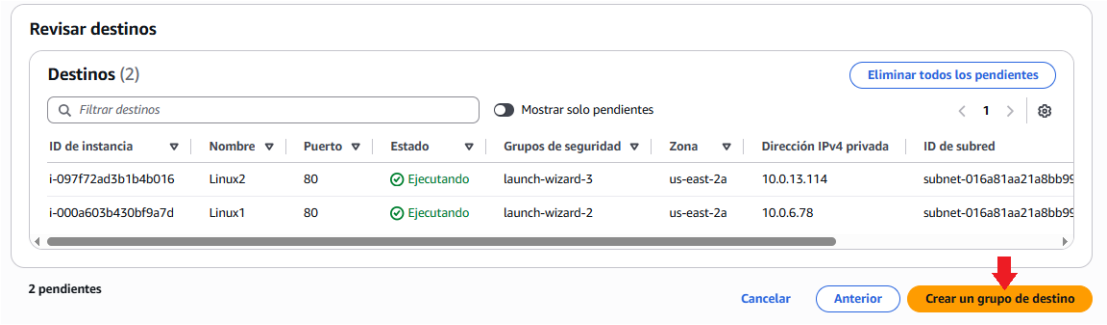
Registramos nuestras dos instancias que estan dentro de la VPC y seleccionamos incluir como pendiente a continuación.

Figura 113 registro de destinos.



Seleccionamos crear grupo de destino.

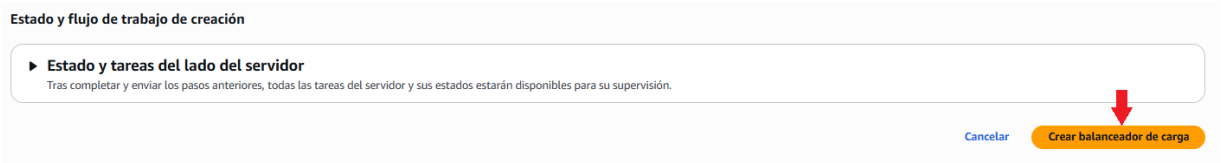
Figura 114 creación grupo de destino.



De esa manera se crea nuestro grupo de destino y podemos continuar con la creación del balanceador de carga y seleccionar el grupo de destino creado.

Damos crear balanceador de carga y esperamos a que se ejecute la creación.

Figura 115 creación balanceador.



Se creó correctamente nuestro balanceador y grupo de destino.

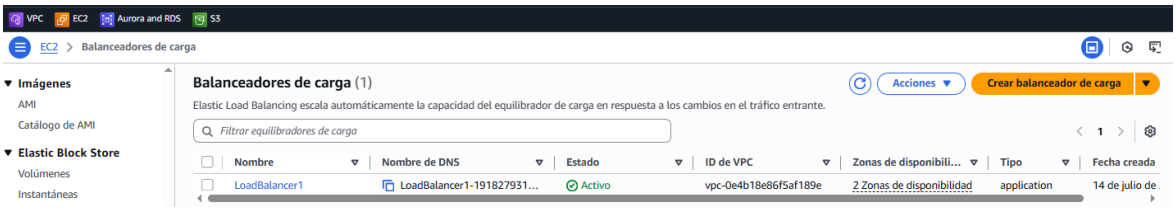
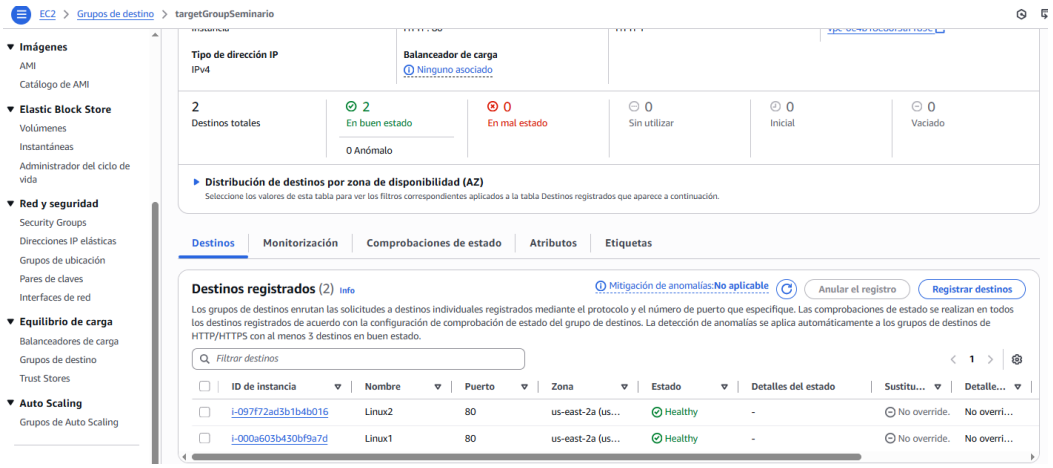


Figura 116 visualización de balanceador.



Para verificar que nuestro balanceador funciona correctamente, desde nuestro navegador web accedemos con el nombre DNS que crea nuestro balanceador y evidenciamos que desde ese mismo DNS se puede acceder a los dos servidores cargando la página con contrl+r.

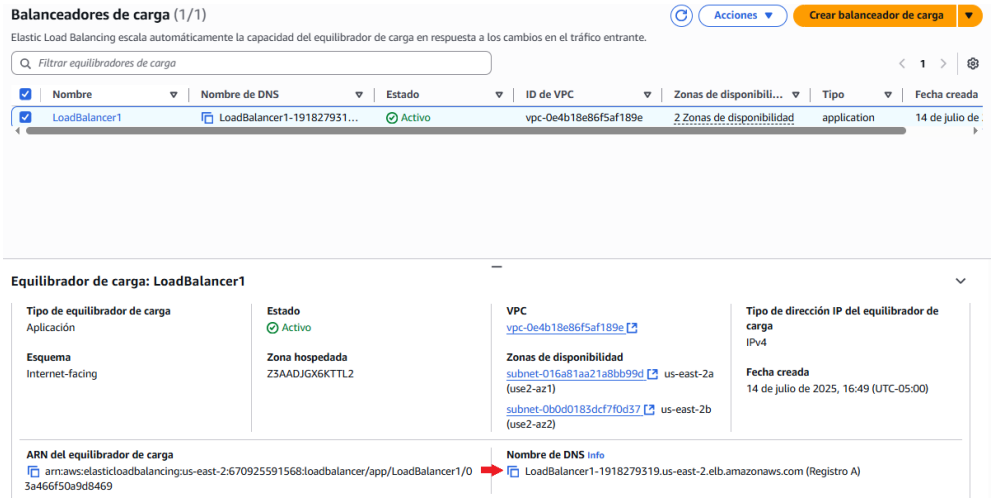


Figura 117 prueba de balanceador en navegador web.

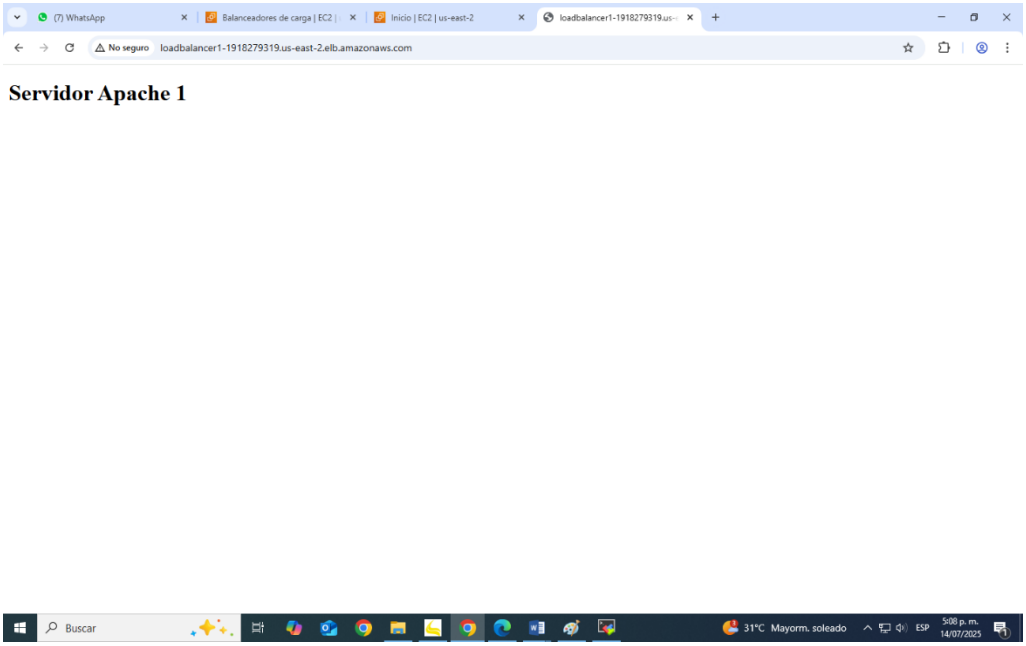
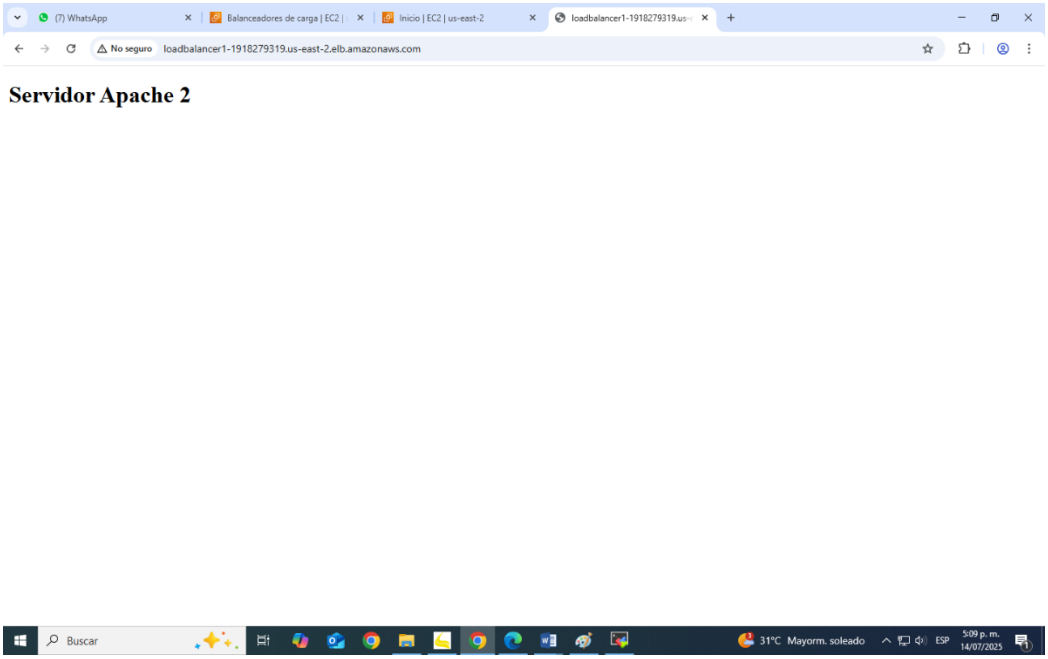


Figura 118 prueba balanceador en navegador web.

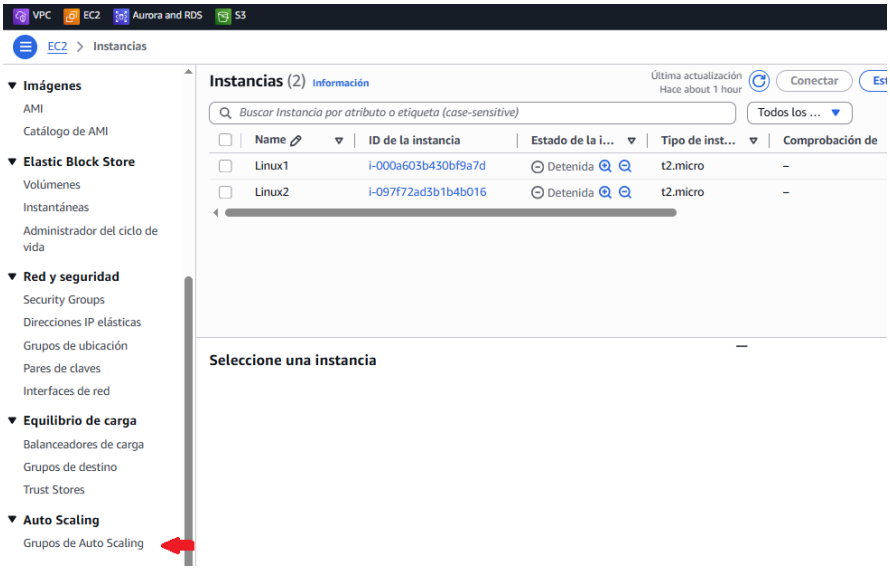


Creación del Auto Scaling

Procedemos con la configuración del Auto Scaling, teniendo en cuenta que este servicio permite ajustar la cantidad de instancias según la demanda, asegurando la disponibilidad.

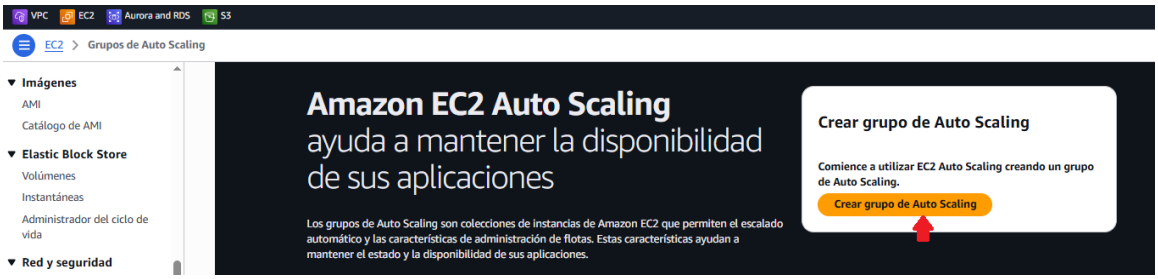
Dentro del servicio EC2 seleccionamos la opción de grupo de Auto Scaling.

Figura 119 servicio auto scaling.



Damos clic en crear grupo de auto scaling e iniciamos la configuración.

Figura 120 creación de servicio auto scaling.

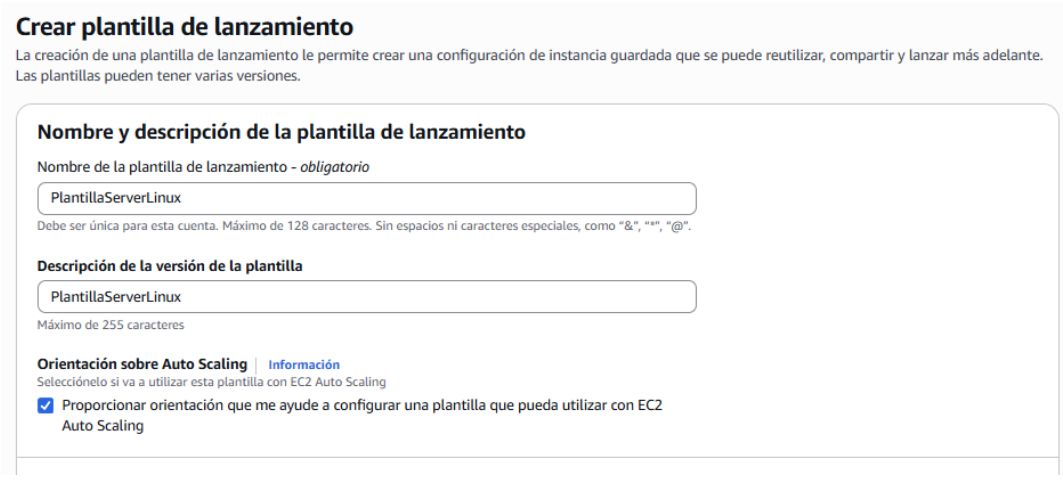


Asignamos un nombre a nuestro auto scaling y como no hemos creado una plantilla de lanzamiento donde va a estar la configuración de las nuevas instancias que se van a crear, damos clic en crear configuración de lanzamiento.

Figura 121 nombrar auto scaling.

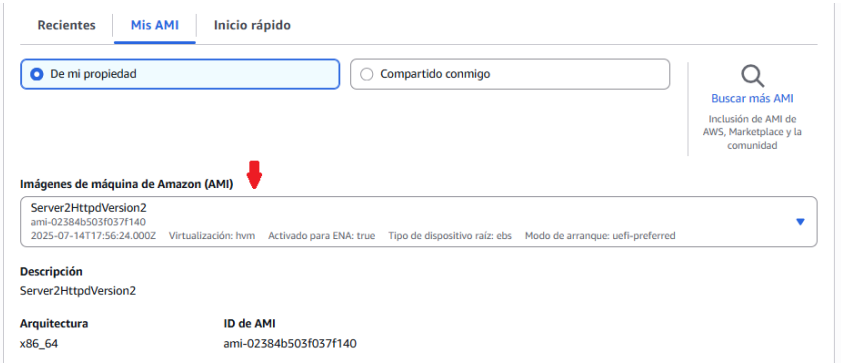
Ingresamos nombre y descripción a nuestra plantilla.

Figura 122 creación Plantilla de lanzamiento.

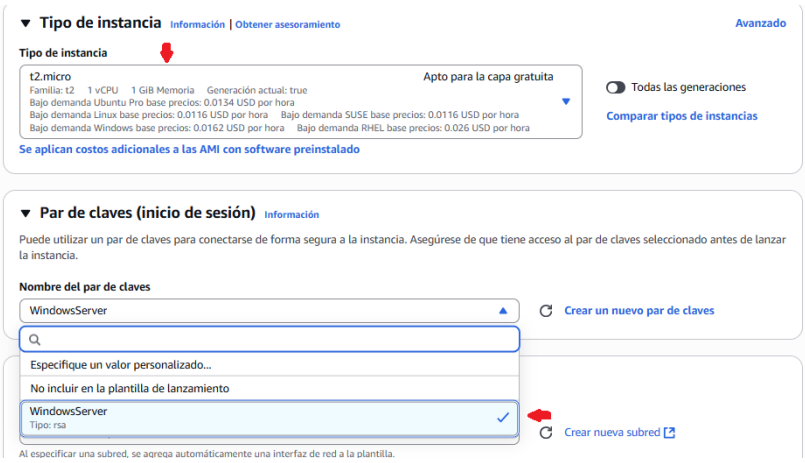


Seleccionamos nuestra AMI creada anteriormente.

Figura 123 selección de AMI.



En tipo de instancia seleccionamos la capa gratuita t2.micro y elegimos nuestro par de claves creado anteriormente.



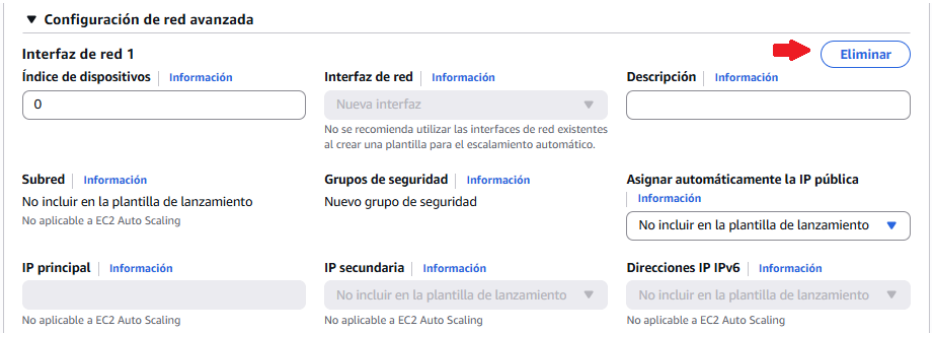
En la configuración de la red elegimos no incluir en la plantilla de lanzamiento, porque ya está incluido en el balanceador, creamos el grupo de seguridad, asignamos un nombre y seleccionamos nuestra VPC.

Figura 124 configuración de la red.

En la regla del grupo de seguridad agregamos el puerto 80 para que permita el acceso desde cualquier lugar.

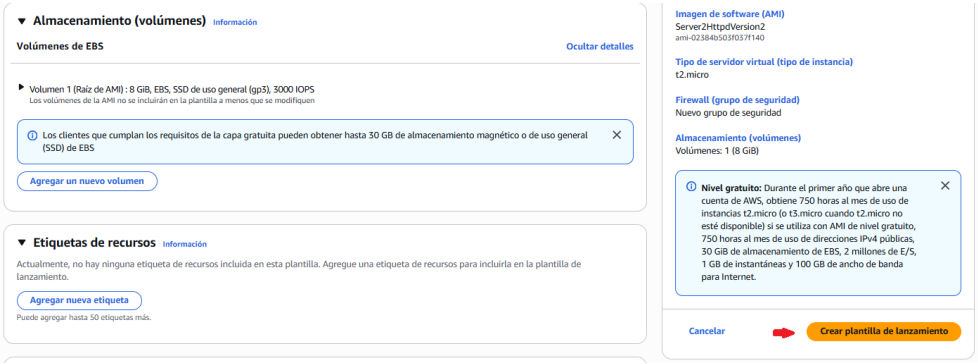
Figura 125 puerto 80 agregado.

En la sección de configuración de red, damos clic en eliminar porque no vamos a usar segunda tarjeta de red.



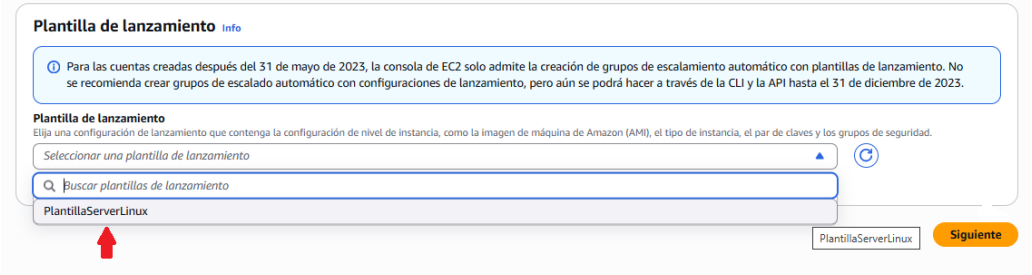
Dejamos la información del almacenamiento que nos sugiere, la cual es la misma que tiene la AMI y damos clic en crear plantilla de lanzamiento.

Figura 126 almacenamiento por defecto y creación.



Una vez creada nos aparece efectivamente la plantilla para continuar con la creación del auto scaling.

Figura 127 plantilla disponible.



Dejamos la versión que nos muestra por defecto y damos siguiente.

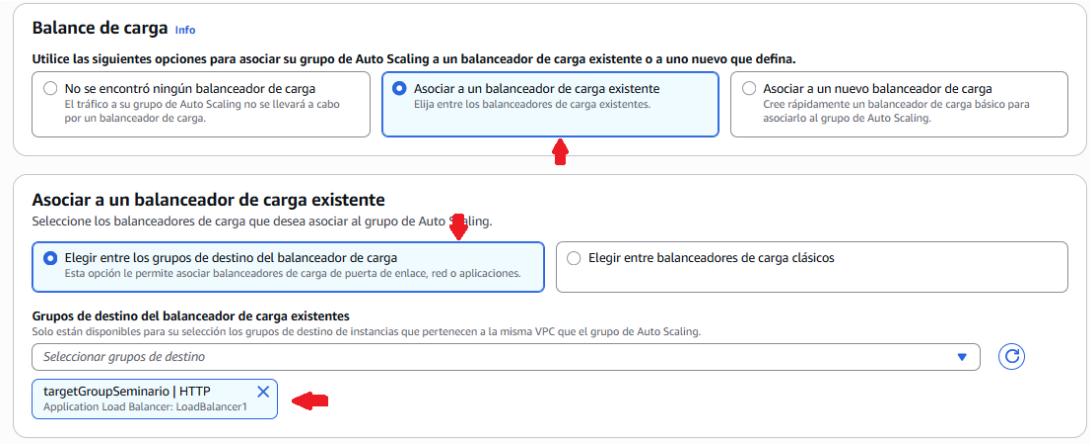
En la configuración de red seleccionamos nuestra VPC y las subredes privadas que se recomienda como buena práctica para que las instancias no tengan acceso desde internet. Además, seleccionamos en mejor esfuerzo equilibrado para que las instancias que se van a crear se distribuyan entre las dos data centers.

Figura 128 configuración de red.

Continuamos con la configuración asociando nuestro auto scaling al balanceador previamente creado.

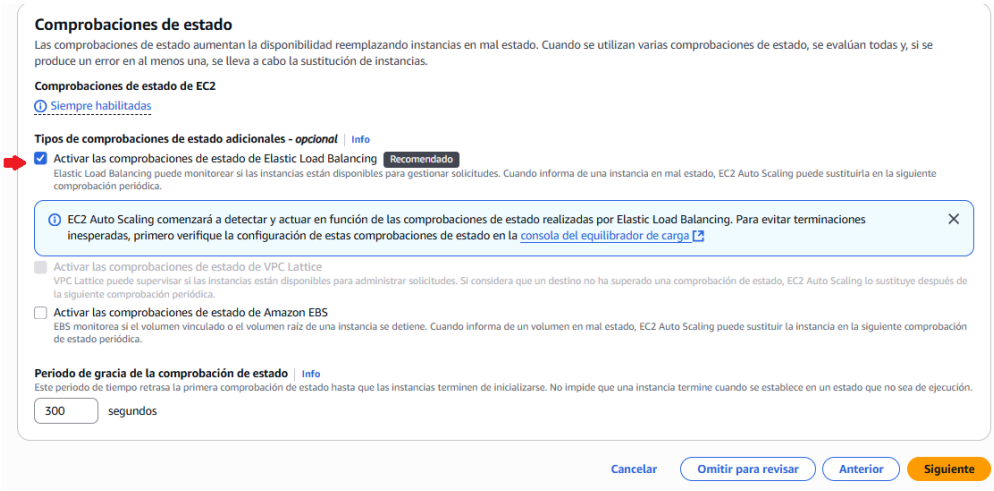
Seleccionamos asociar balanceador existente y elegimos nuestro grupo de destino de nuestro balanceador.

Figura 129 asociar balanceador y grupo de destino.



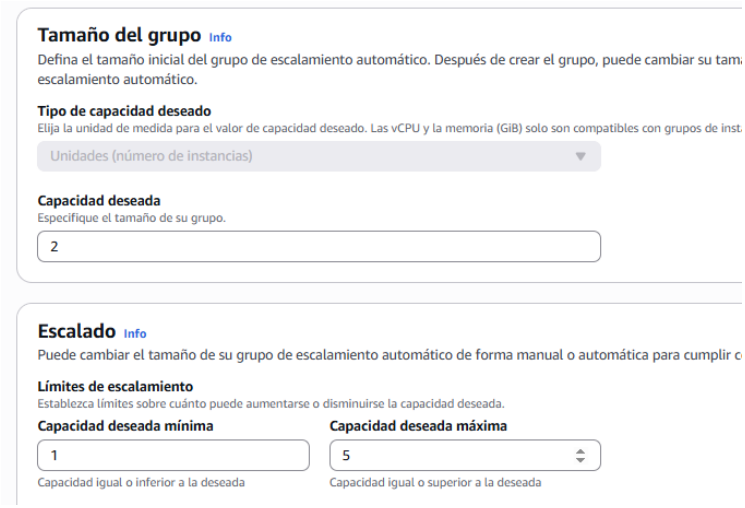
Activamos las comprobaciones de estado de las instancias para que realice monitoreo de las mismas y continuamos.

Figura 130 activación de comprobacion de estado.



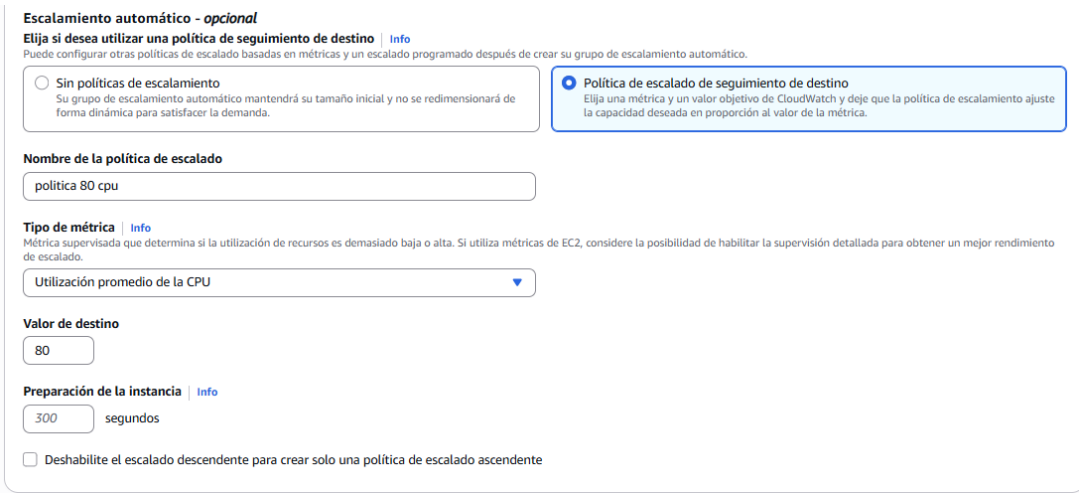
Ingresamos la capacidad de instancias deseadas, en este caso 2 y agregamos la capacidad mínima y máxima deseada.

Figura 131 capacidad instancias creadas.

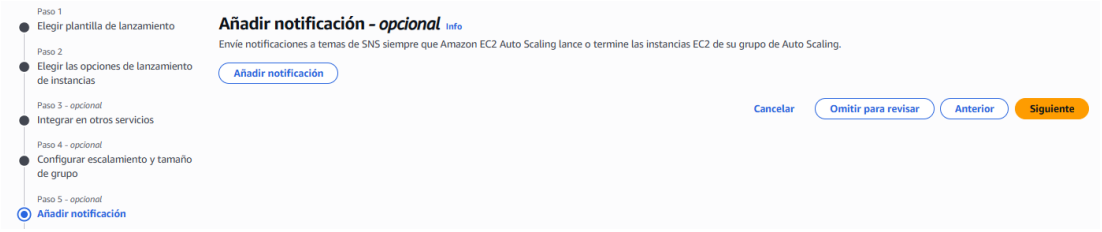


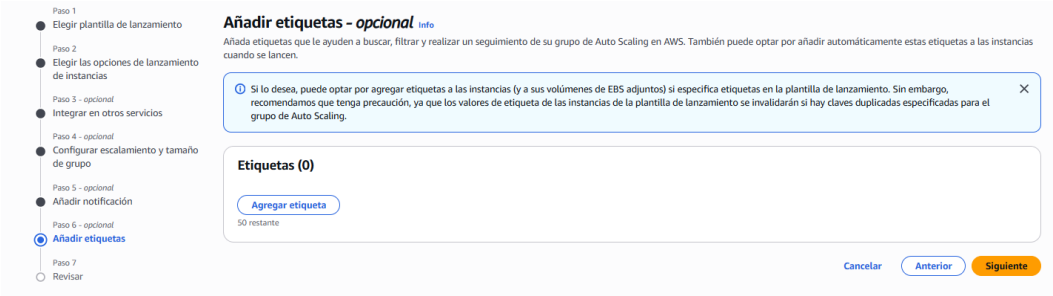
Creamos nuestra política de escalado y continuamos.

Figura 132 creación politica de escalado.



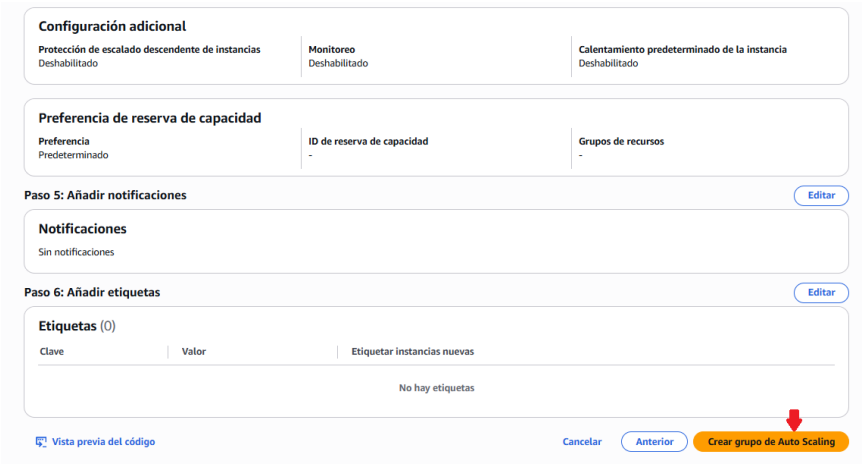
No configuramos notificaciones ni etiquetas por el momento, para continuar con la creación del auto scaling.





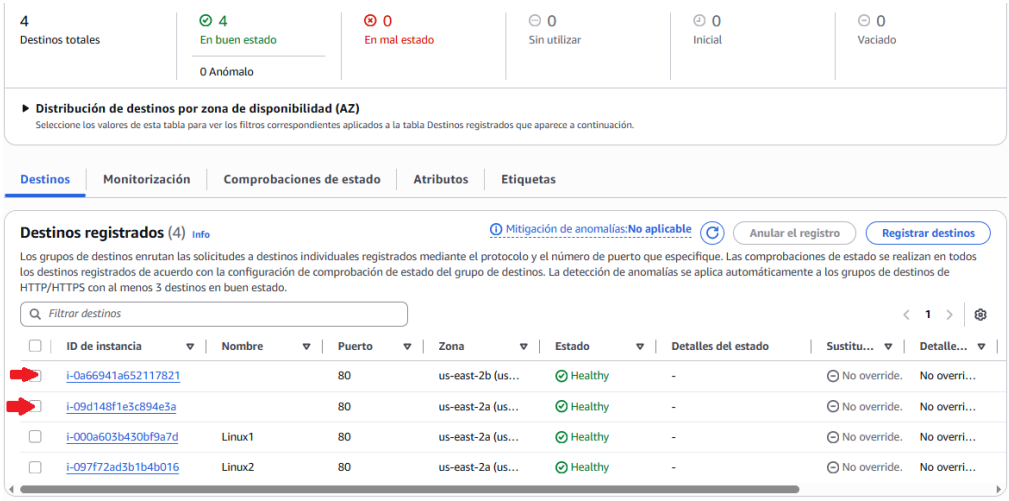
Posteriormente nos muestra un resumen de la configuración y damos clic en crear grupo de auto scaling.

Figura 133 resumen de la configuración.



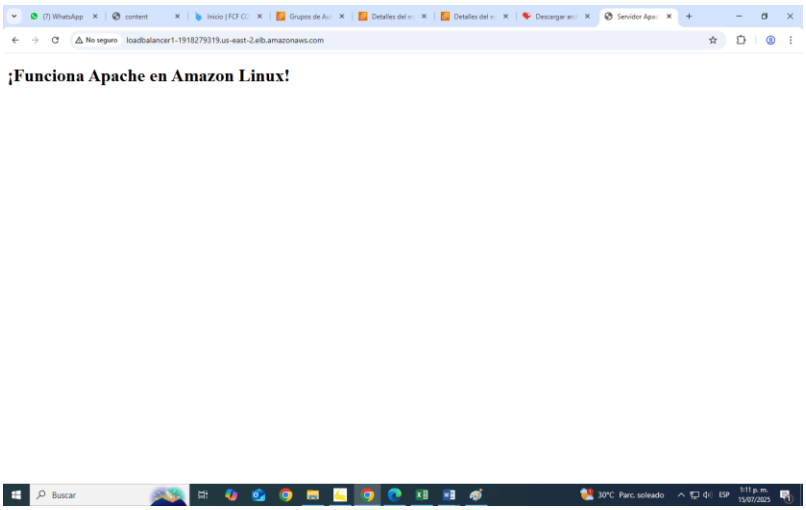
Efectivamente nuestro grupo de auto scaling se creó de forma correcta, creando en nuestro balanceador 2 nuevas instancias de destino.

Figura 134 prueba de auto scaling funcionando.



Para comprobar que esté funcionando nos dirigimos a nuestro balanceador de carga, copiamos nuestro DNS para hacer la prueba en nuestro navegador.

Figura 135 prueba en navegar web.

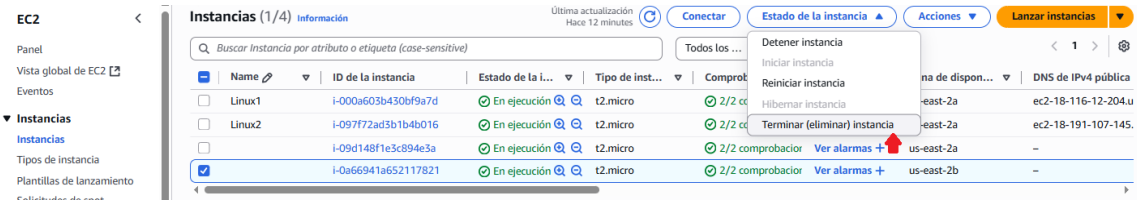


Efectivamente al recargar la imagen los dos servidores me muestran la misma imagen y no se nota el cambio, ya que se sacó la imagen de mi servidor 1.

Probamos que nuestro auto scaling funcione correctamente, eliminando una instancia, este servicio al notar ese proceso, lo que hace es generar automáticamente una nueva instancia con las mismas características.

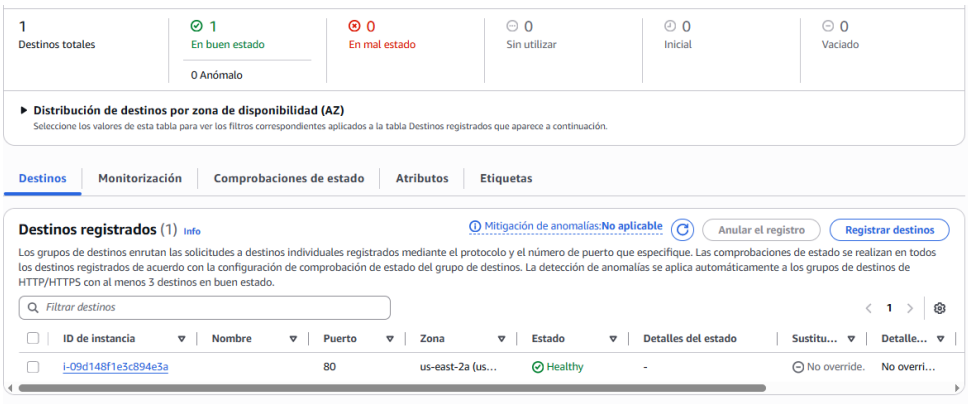
Damos terminar a una de nuestras instancias.

Figura 136 prueba auto scaling .



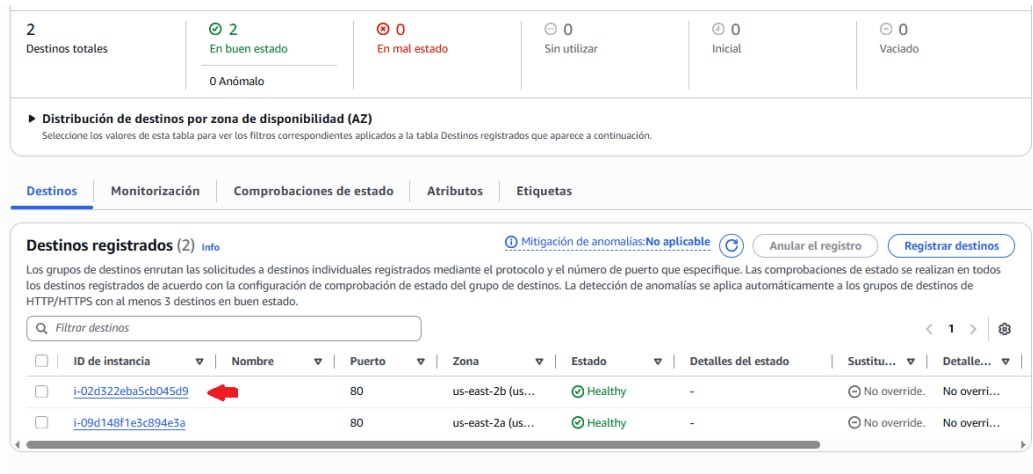
Revisamos en nuestro grupo de destino evidenciando que queda una instancia y esperamos que el auto scaling cree una nueva.

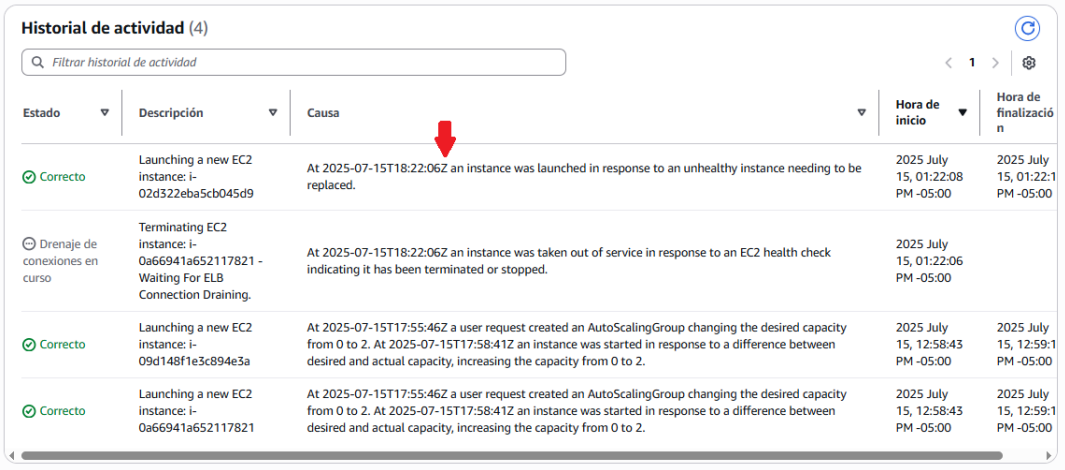
Figura 137 prueba auto scaling.

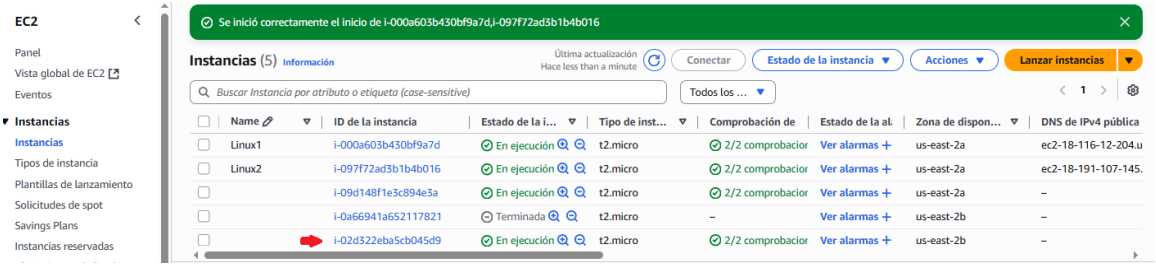


Efectivamente creó una nueva.

Figura 138 prueba creación auto scaling.



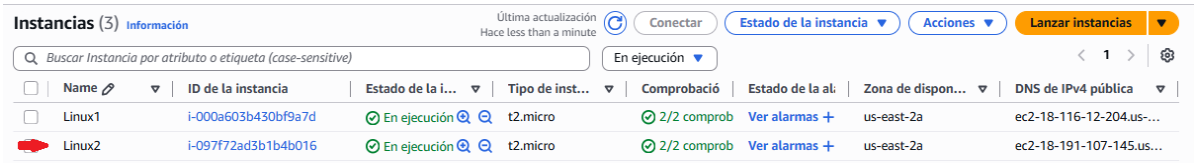




Implementación del servicio de Docker

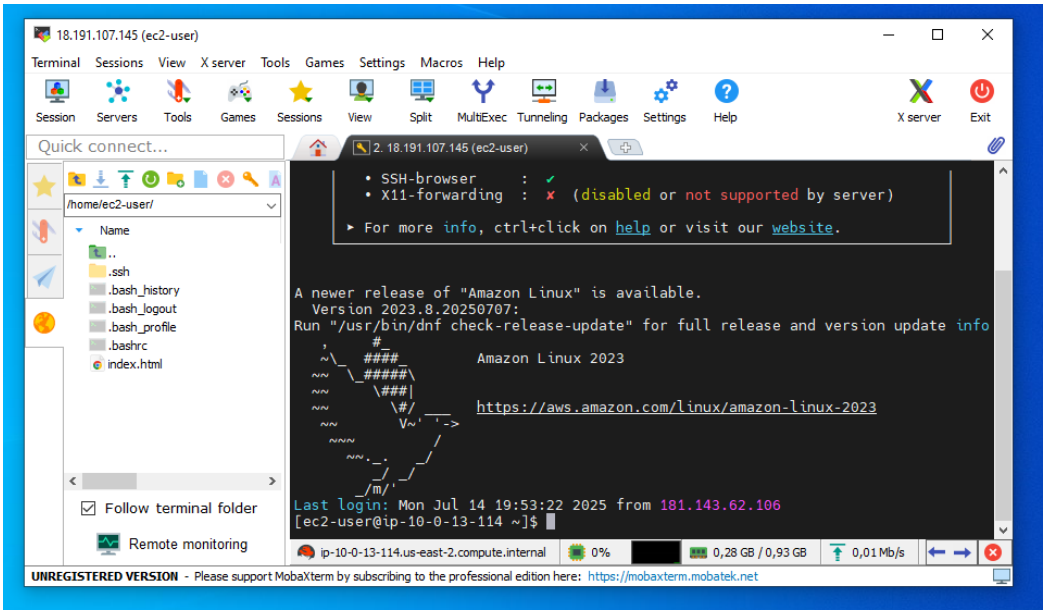
Para la implementación de Docker vamos a utilizar una de las instancias que tenemos creada en correcto funcionamiento.

Figura 139 seleccion de instancia.



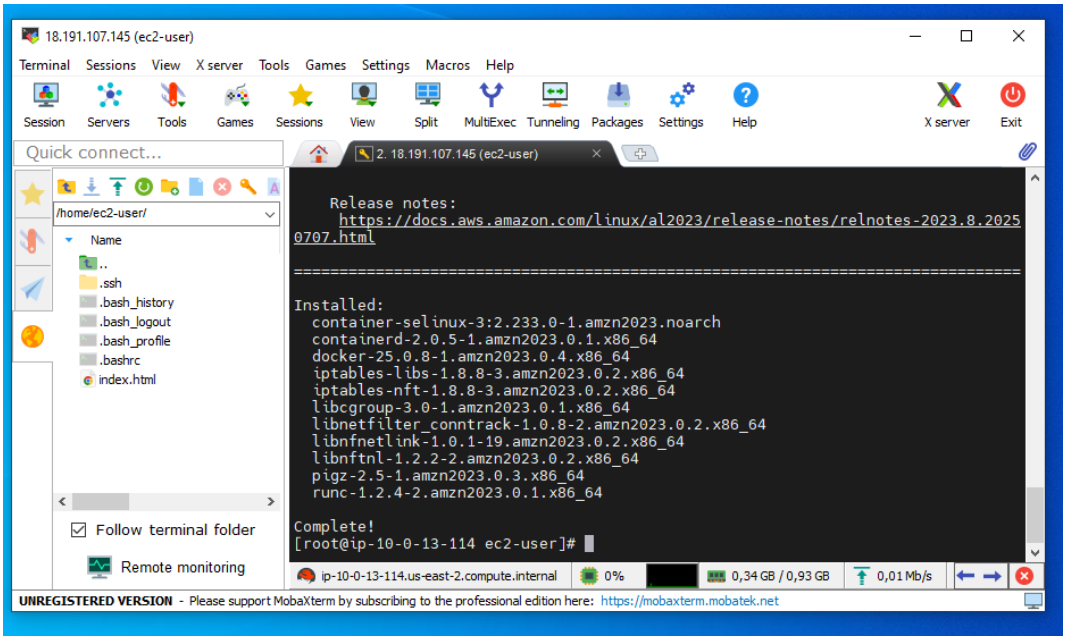
Para iniciar con la creación del Docker accedes a nuestro servidor a través de MobaXterm e ingresamos los datos necesarios para la conexión a través del protocolo SSH.

Figura 140 acceso al servidor.



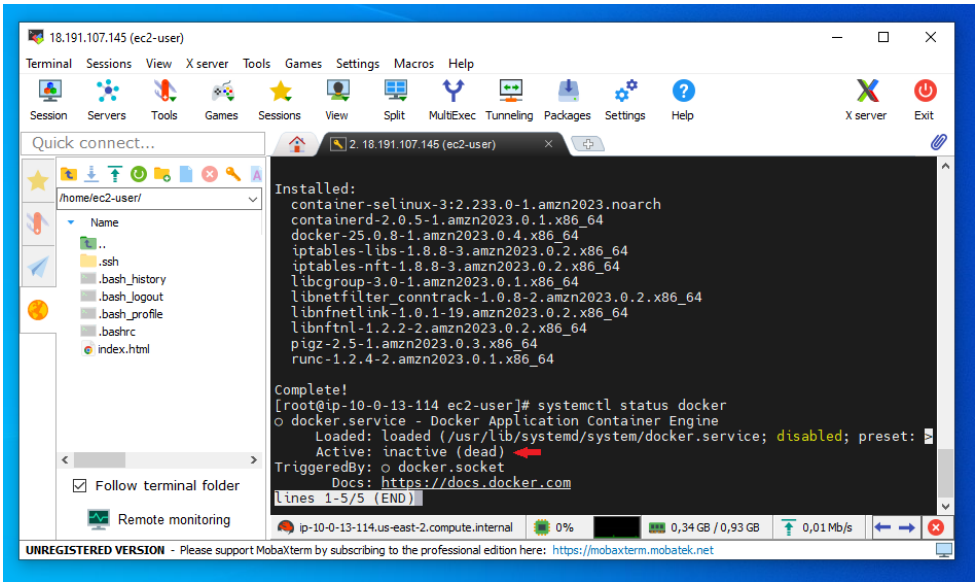
Ingresamos el comando (sudo su) para obtener el rol de administrador y procedemos a usar el comando (yum install Docker) para la instalación del Docker.

Figura 141 rol administrador.



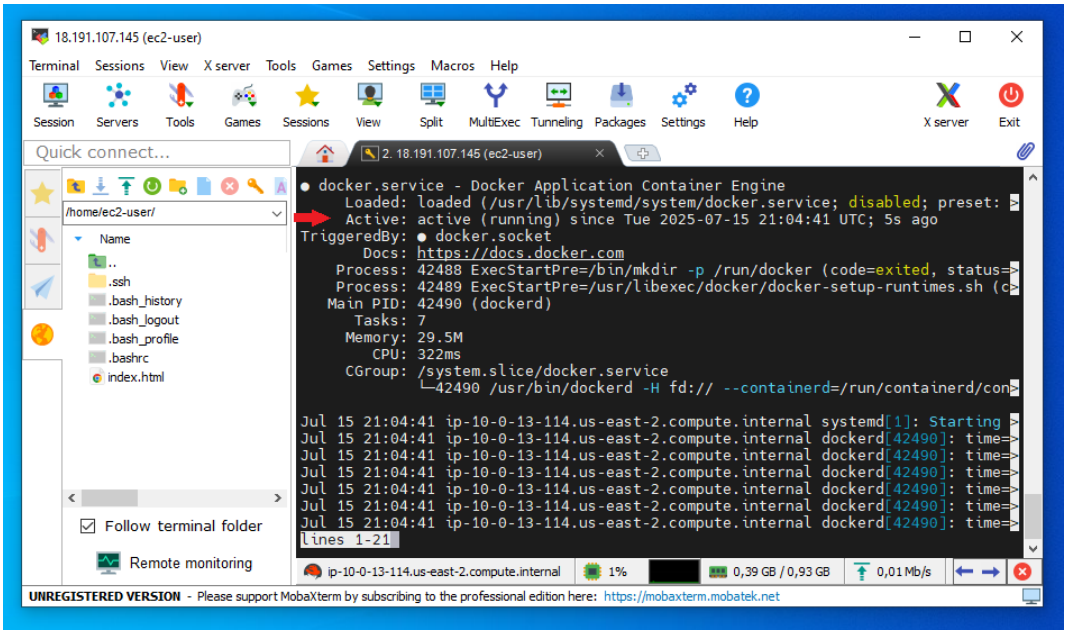
Visualizamos en qué estado está el servicio con el comando (systemctl status Docker).

Figura 142 estado del servidor.



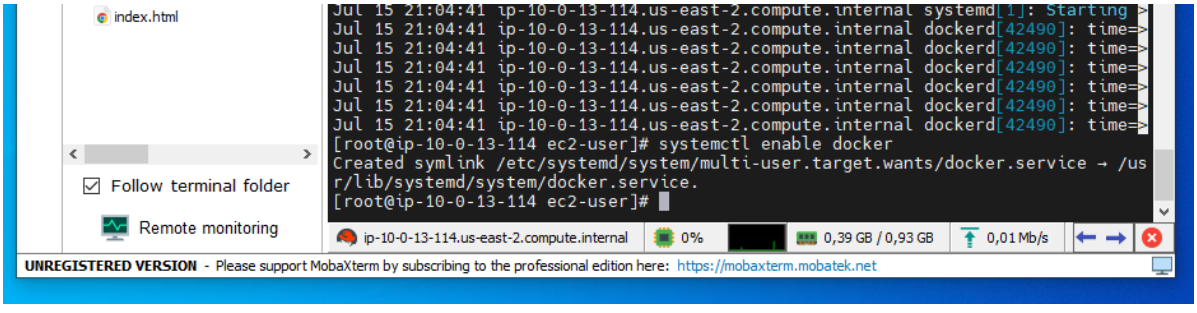
Para activar el servicio escribimos (systemctl start Docker) y volvemos a verificar el estado con el comando anterior.

Figura 143 servidor activado.



Procedemos a ingresar el comando (systemctl enable docker) para que este servicio arranque de forma automática.

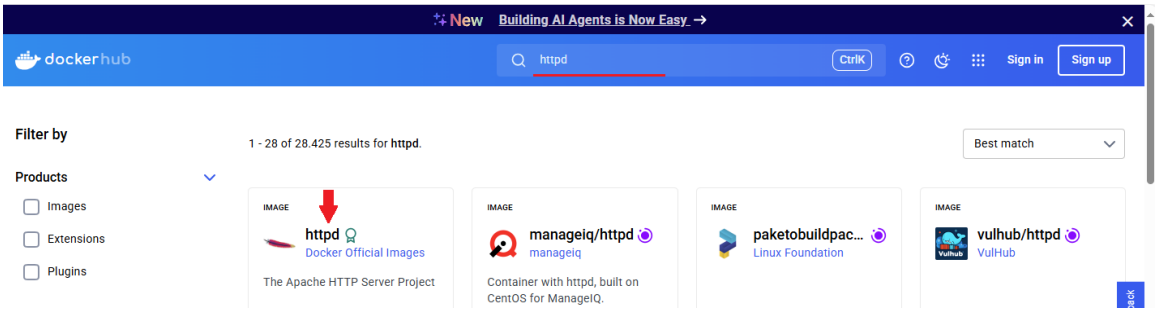
Figura 144 comando de arranque automatico.



Procedemos a crear nuestro contenedor, pero para eso necesitamos una imagen que va a funcionar dentro del mismo. Para esto vamos a descargar una imagen desde el repositorio de docker, el cual se llama docker hub.

Dentro del repositorio de docker, buscamos httpd y seleccionamos el más actualizado.

Figura 145 Busqueda desde el navegador docker hub.

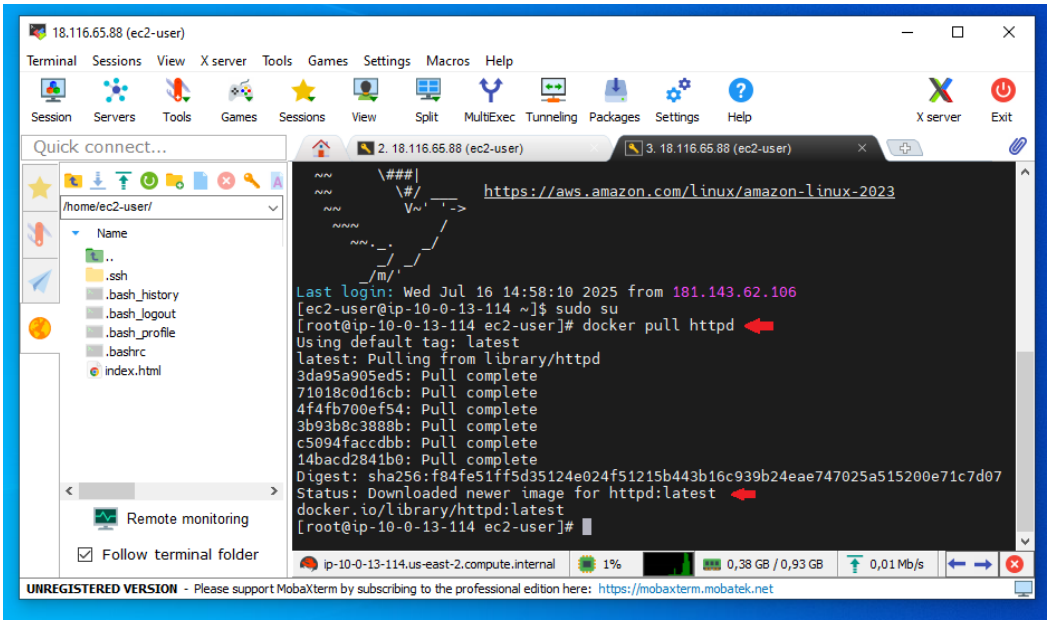


Para descargar la imagen usamos el comando que nos proporciona.

Figura 146 descarga del sitio.

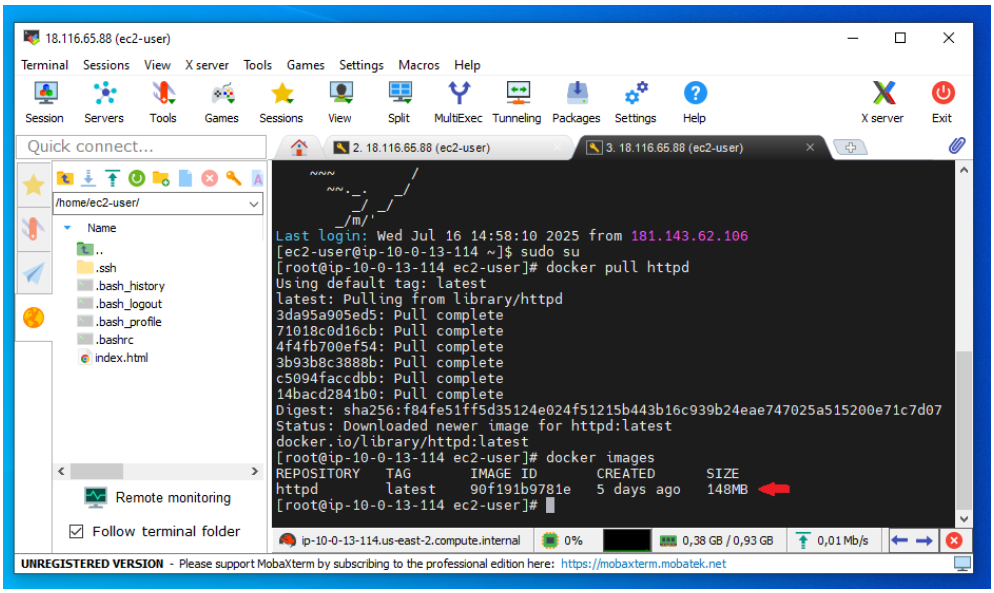
Ejecutamos ese comando en el servidor y esperamos la instalación de la imagen.

Figura 147 instalación de la imagen.



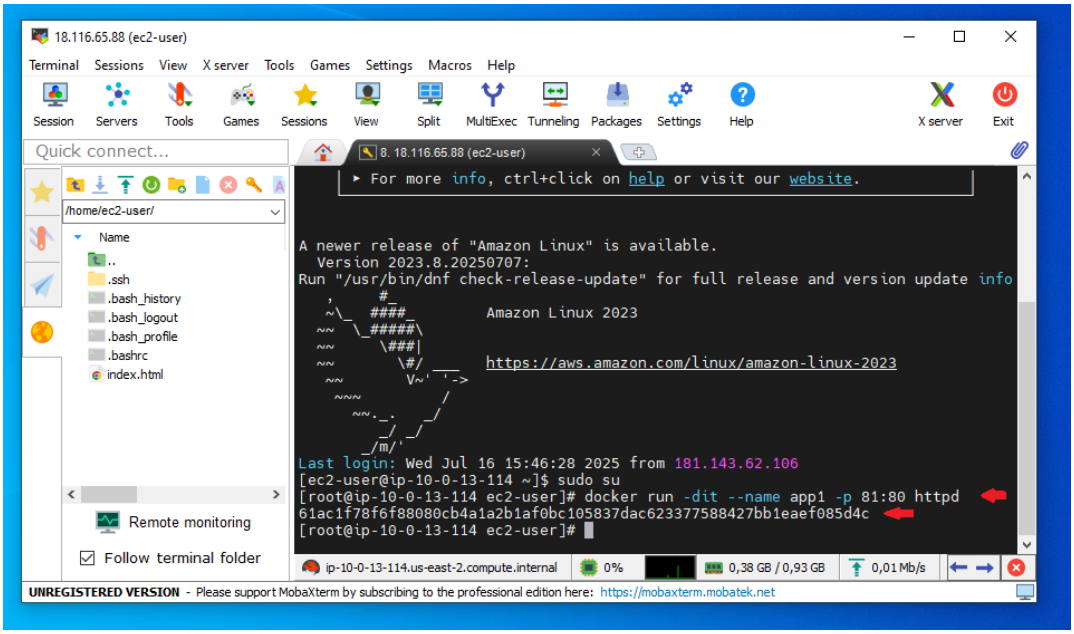
Revisamos la imagen que tenemos con el comando (docker images).

Figura 148 comprobacion de imagen instalada.



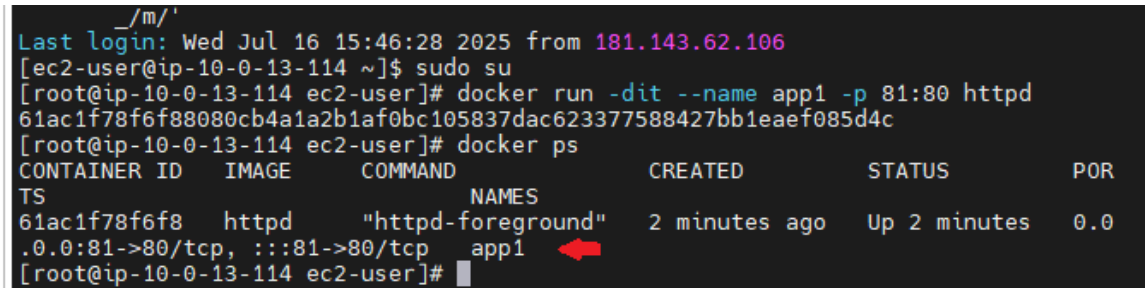
Una vez instalada la imagen, iniciamos a crear nuestro primer contenedor con el comando (docker run -dit --name app1 -p 81:80 httpd).

Figura 149 creación de contenedor.



Observamos los contenedores tenemos con el comando (docker ps).

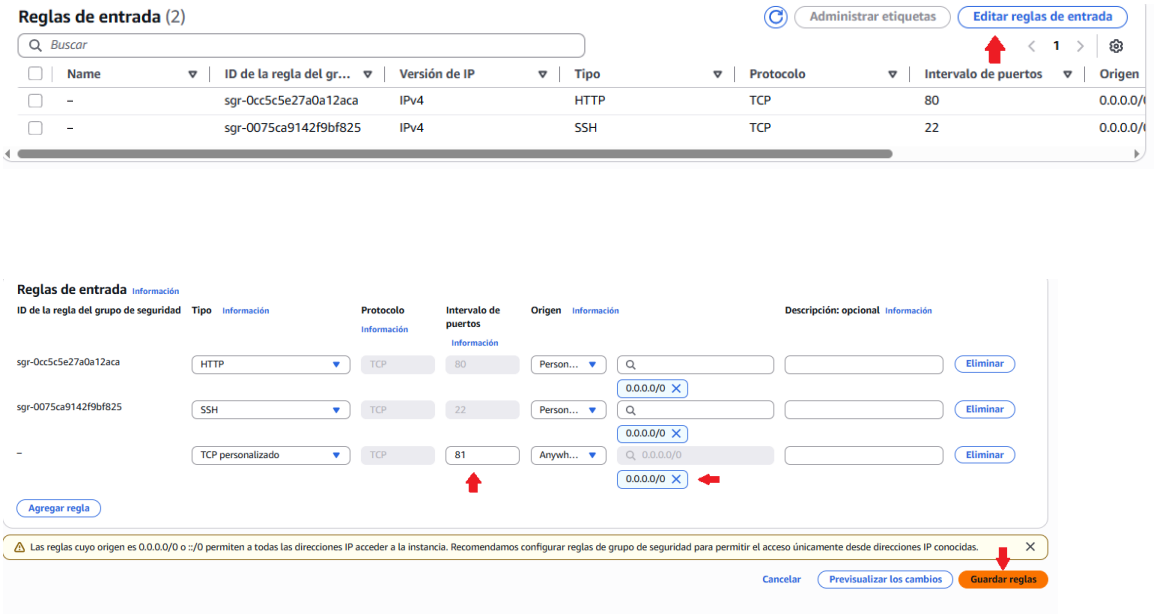
Figura 150 contenedores disponibles



Para verificar que nuestro contenedor esté funcionando desde el puerto 81, debemos agregarlo en el grupo de seguridad de nuestra instancia y luego probarlo.

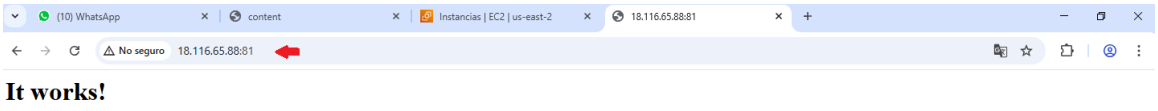
En las reglas de entrada damos editar para agregar el puerto 81.

Figura 151 puerto 81 agregado a las reglas de entrada.



Probamos nuestro contenedor ingresando la IP publica seguida del puerto habilitado para el contenedor y efectivamente está funcionando.

Figura 152 prueba en navegar web del contenedor.

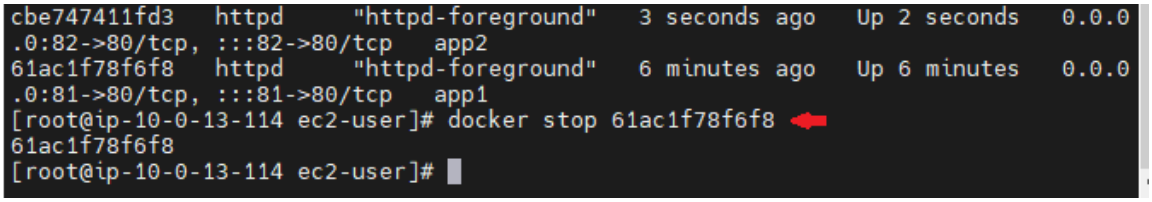


Continuamos ingresando una aplicación propia en la ubicación en la ubicación de la carpeta del servidor.

Para la administración de nuestro docker debemos tener en cuenta el identificar que se genera al momento de crear los contenedores.

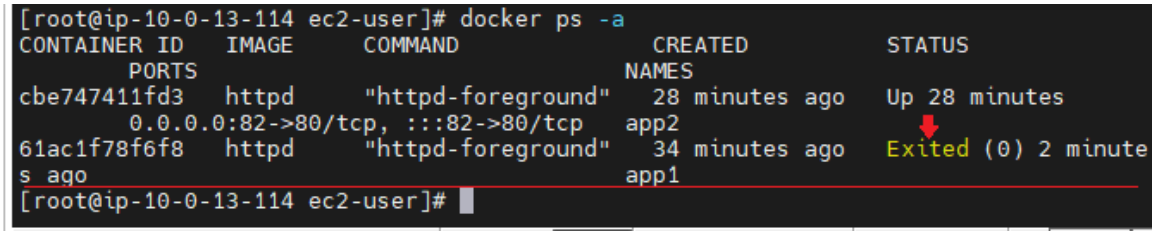
Para detener el contenedor ingresamos el comando (docker stop #idcontenedor).

Figura 153 detener contenedor.



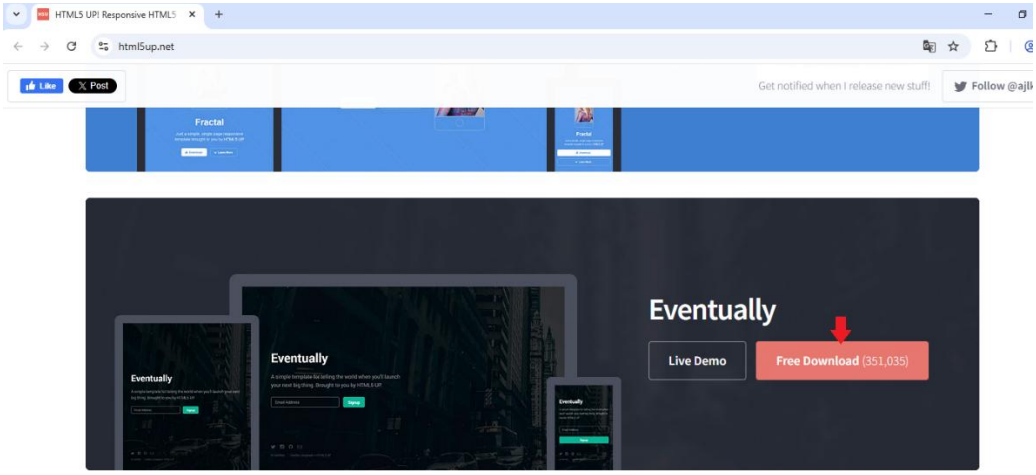
Para revisar que nuestro contenedor se detuvo ingresamos el comando (docker ps –a).

Figura 154 observar contenedor detenido.



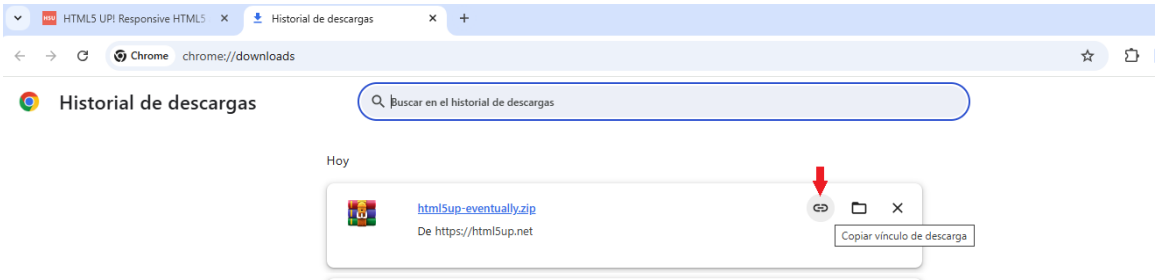
Vamos a probar montando una página en nuestro contenedor, buscamos en nuestro navegador web html5 template y elegimos una descargando el archivo.

Figura 155 prueba montando imagen.



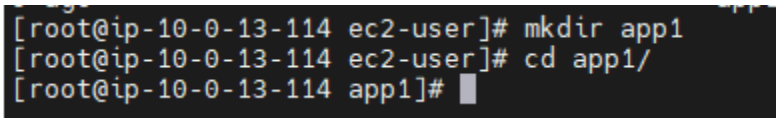
Una vez se descarga necesitamos copiar el enlace de descargar para utilizarlo en nuestra instancia de Linux.

Figura 156 copiar el enlace de la imagen descargada.



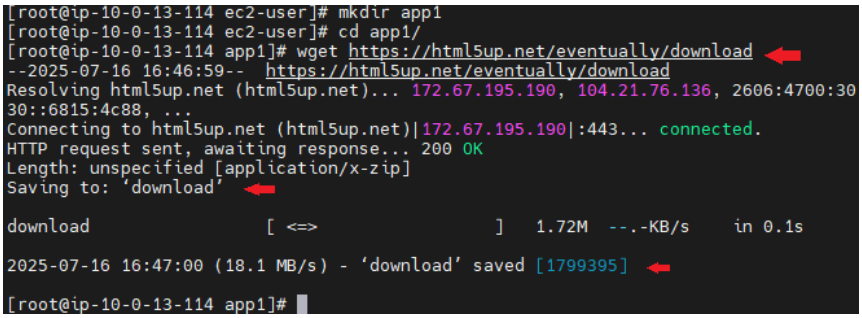
Creamos la carpeta para ingresar el archivo con el comando (mkdir app1) y accedemos a la carpeta con el comando (cd app1/), para descargar el sitio.

Figura 157 creación de carpeta.



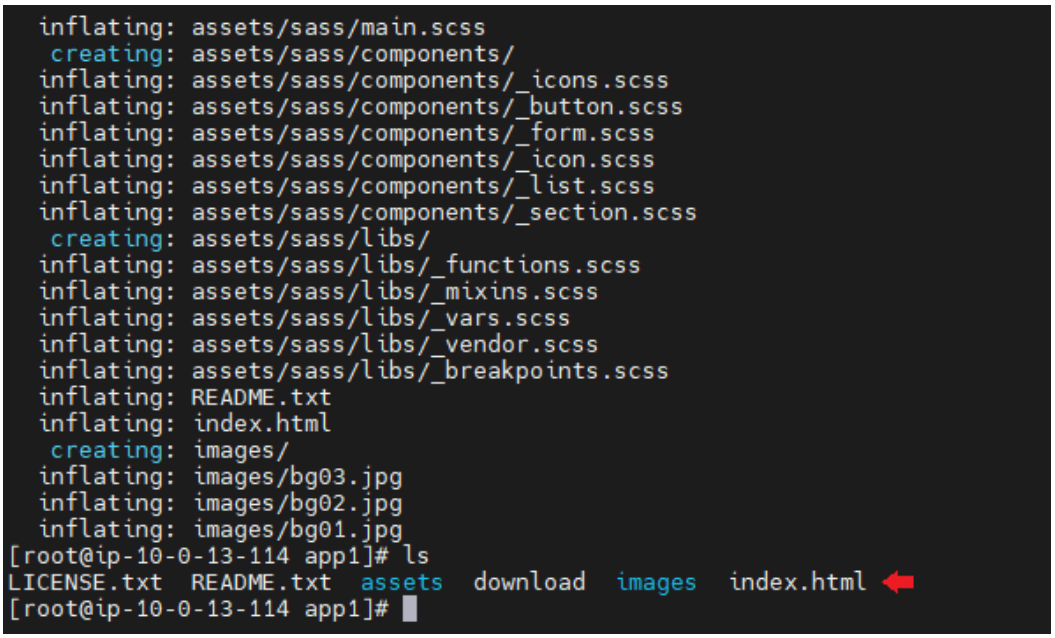
Descargamos el sitio ingresando el comando (wget) y pegando el enlace de descarga del sitio.

Figura 158 descarga de sitio.



Comprobamos que es un archivo Zip con el comando (file download) y lo descomprimimos con el comando (unzip download).

Figura 159 comprobacion de tipo de archivo.



Luego agregamos el archivo a la carpeta creada anteriormente para que los contenedores puedan acceder a él.

Para eso vamos a crear un nuevo contenedor y a este le vamos a agregar el archivo que descargamos para que este dentro de la carpeta del contenedor y pueda mostrar el sitio que descargamos.

Figura 160 creación de contenedor con el archivo descargado.

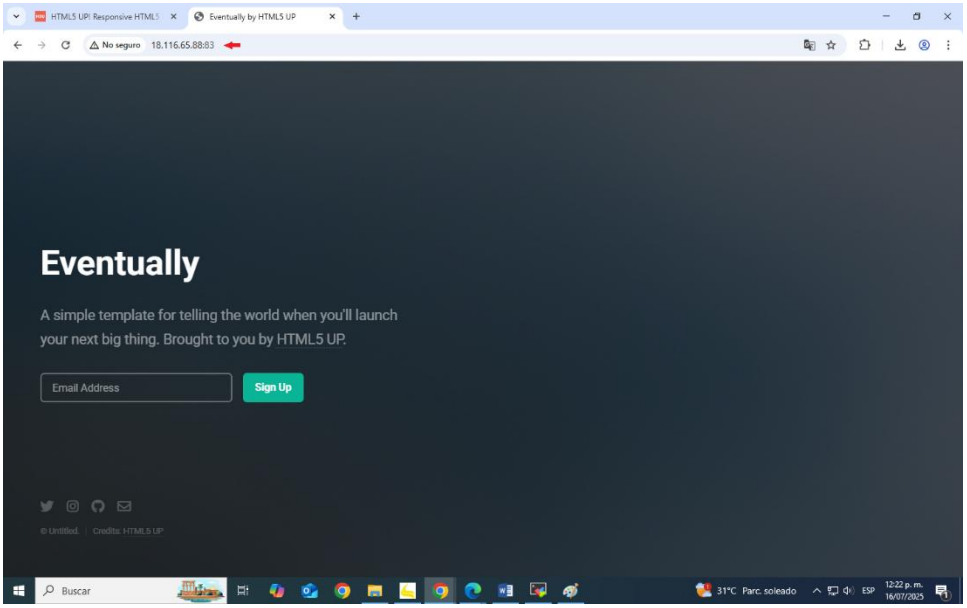
```
Create and run a new container from an image
[root@ip-10-0-13-114 app1]# docker run -dit --name app3 -p 83:80 -v /home/ec2-user
/app1/:/usr/local/apache2/htdocs/ httpd
2abc4bdf0c4831e36003f5f0b0761c766094face9d4bc8a37b6cfd9de8cb2cd2
[root@ip-10-0-13-114 app1]#
```

Verificamos en nuestro navegador, pero antes le damos el acceso a nuestro puerto 83 desde la instancia en grupos de seguridad, reglas de entrada.

Figura 161 acceso a puerto para prueba.

Efectivamente nos carga el contenedor con la aplicación.

Figura 162 prueba del contenedor en el navegador web.



Creamos otro contenedor, pero en este caso ingresamos un nuevo comando para garantizar que, aunque se apague el servidor, el contenedor vuelva a arrancar de manera automática.

Figura 163 creación de contenedor y comando de inicio automáticamente.

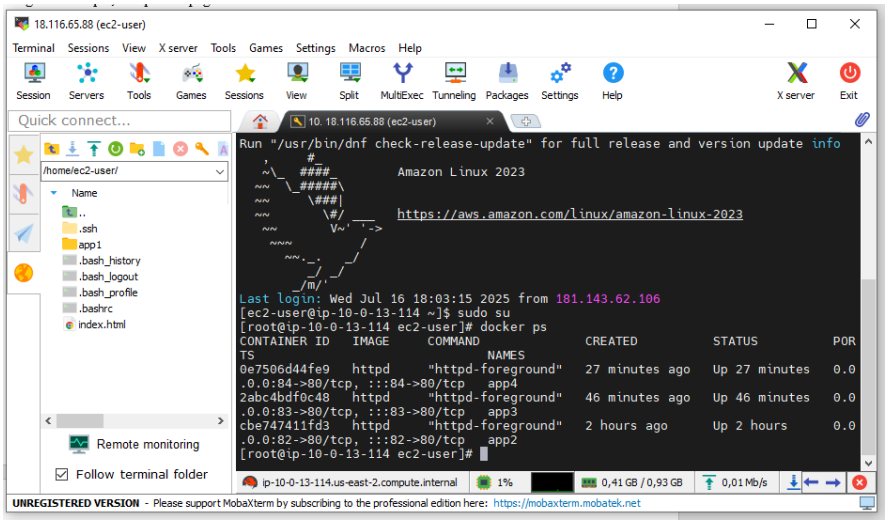
```
[root@ip-10-0-13-114 app1]# docker run -dit --name app4 -d --restart always -p 84:
80 -v /home/ec2-user/app1/:/usr/local/apache2/htdocs/ httpd
0e7506d44fe9f18b83508bd7338aedbe6de3713eed7b555e90fc33045cd4c6a7
[root@ip-10-0-13-114 app1]#
```

Proxy Reverso con instancias

Implementación del proxy inverso para que gestione el tráfico entrante entre los servidores.

Verificamos los contenedores que tenemos creados, con el comando (Docker ps).

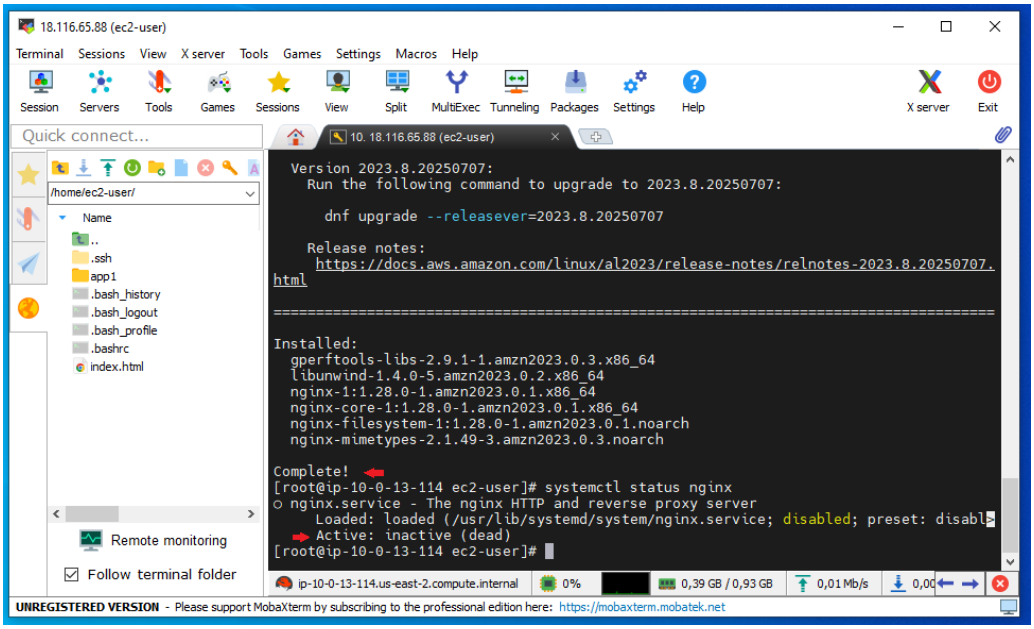
Figura 164 comprobacion de contenedores.



Ponemos a funcionar varios contenedores dentro de la misma instancia con la misma aplicación, para tener acceso desde una URL sin puertos y necesitamos instalar el servicio de Nginx.

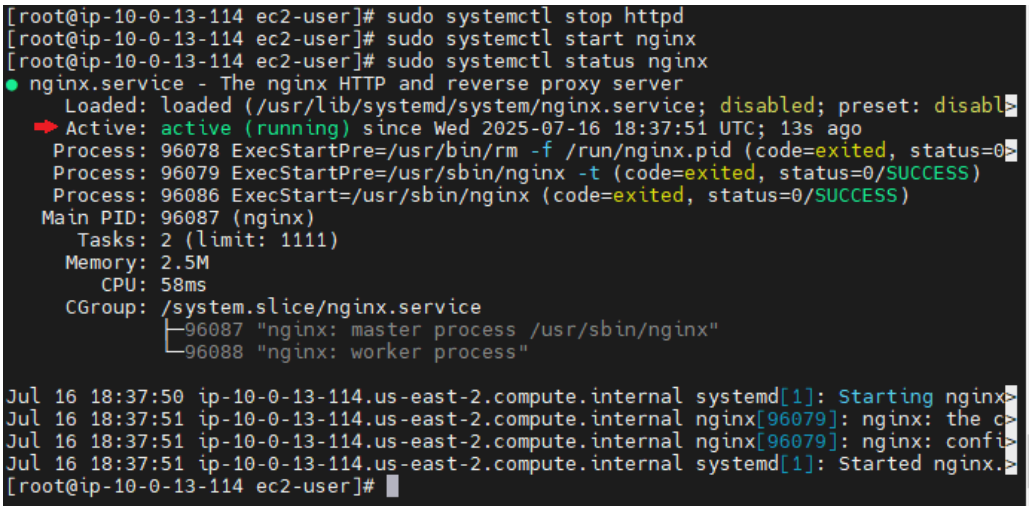
Instalamos el servicio de Nginx ejecutando el comando (dnf install nginx) y verificamos el estado.

Figura 165 instalación del servicio Nginx.



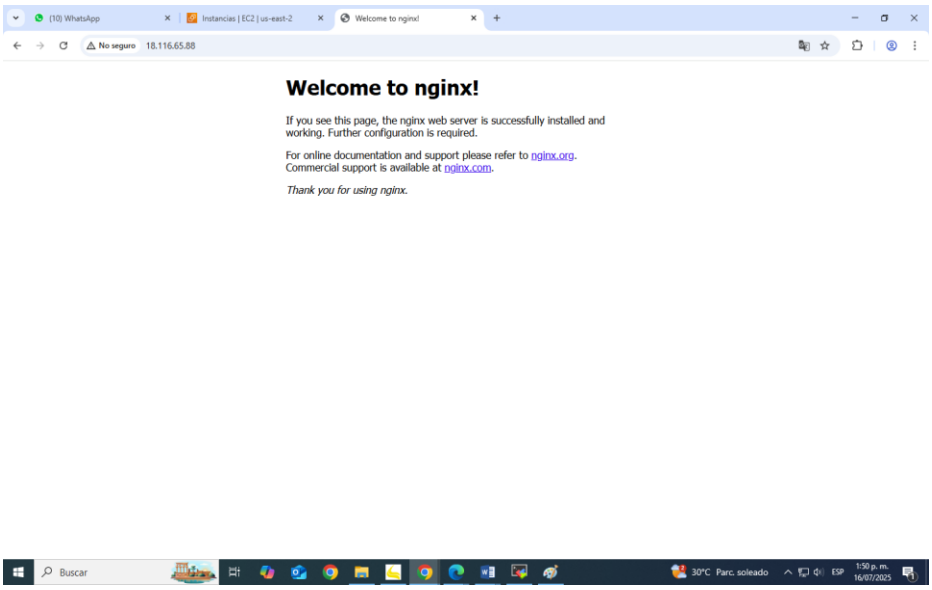
Activamos el servicio con (systemctl start nginx).

Figura 166 activación del servicio.



Verificamos que este corriendo, accediendo desde internet por medio del puerto 80, usando la IP publica de nuestra instancia.

Figura 167 prueba de acceso.



Seguimos con la configuración para poner a funcionar el servicio como un proxy inverso.

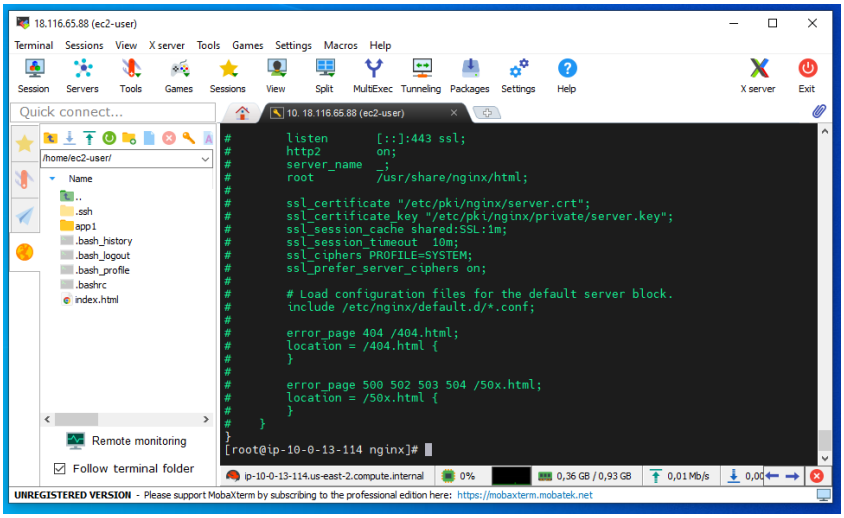
Ingresamos a la carpeta de Nginx para acceder a la configuración con la ruta (cd /etc/nginx/) y escribimos ls para ver que lo que está dentro.

Figura 168 configuración de la ruta.

```
[root@ip-10-0-13-114 ec2-user]# cd /etc/nginx/
[root@ip-10-0-13-114 nginx]# ls
conf.d                  koi-utf             scgi_params
default.d               koi-win             scgi_params.default
fastcgi.conf            mime.types          uwsgi_params
fastcgi.conf.default    mime.types.default  uwsgi_params.default
fastcgi_params          nginx.conf          win-utf
fastcgi_params.default  nginx.conf.default
[root@ip-10-0-13-114 nginx]#
```

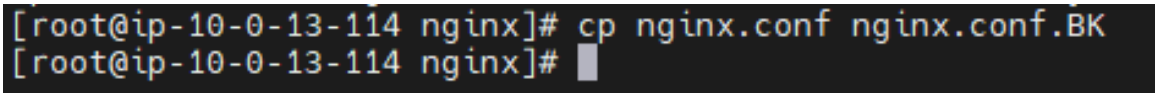
Revisamos el funcionamiento del servidor ingresando (cat nginx.conf)

Figura 169 revisar funcionamiento.



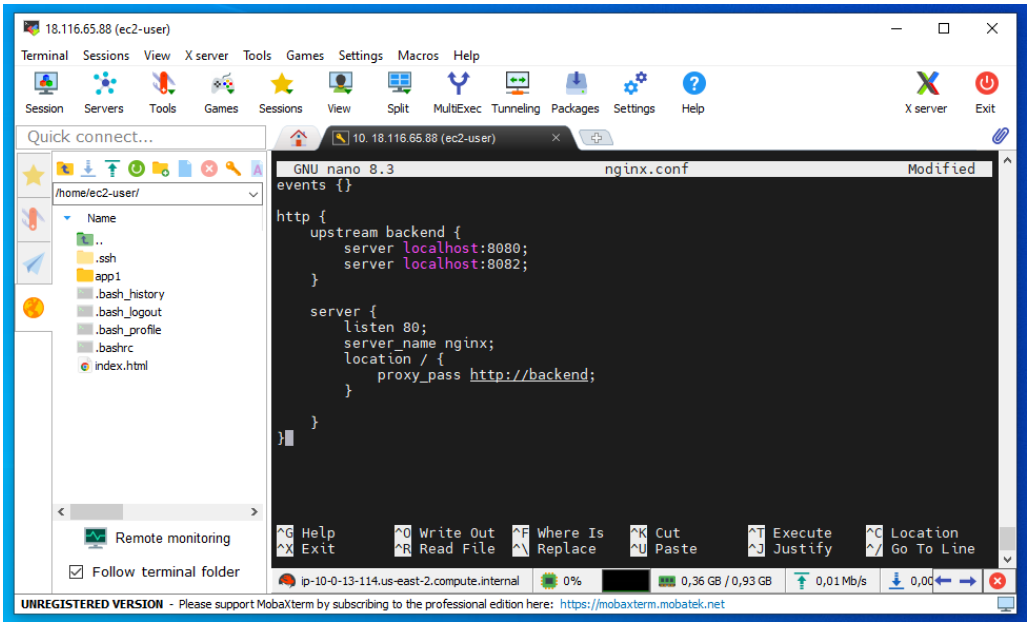
Realizamos una copia del archivo por si deseamos regresar al estado como estaba el archivo.

Figura 170 Copia de seguridad.



Entramos al archivo para modificarlo con (nano nginx). Borramos la información y pegamos la documentación del sitio.

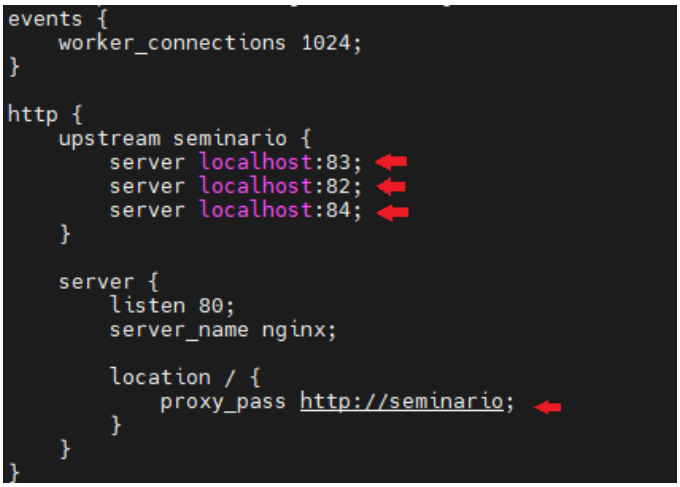
Figura 171 modificación de archivo para sitio.



Esa es la configuración que se debe usar para que funcione el servicio como proxy reverso.

Modificamos el nombre del servidor y las urls para que cuando llegue una petición al puerto 80 las redireccione al grupo de urls donde están mis contenedores.

Figura 172 modificacion nombre del servidor y url.



Prueba de proxy reverso desde cada una de las IP

Figura 173 prueba de contenedor con el puerto 85.

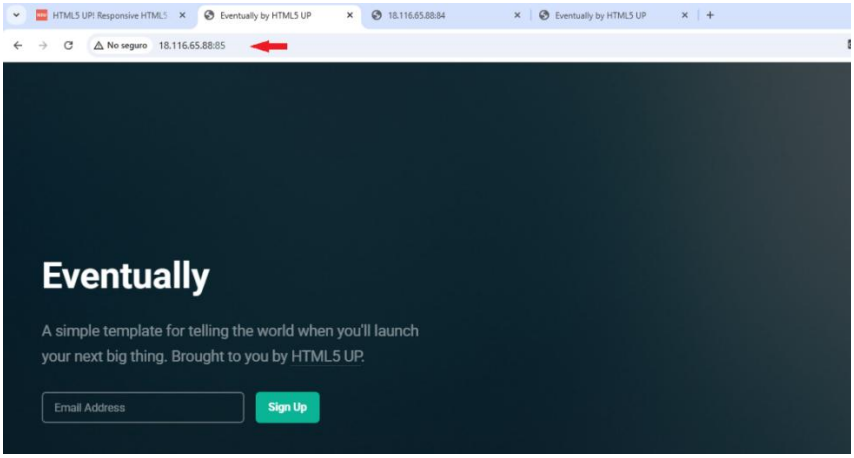


Figura 174 prueba de contenedor con el puerto 84.

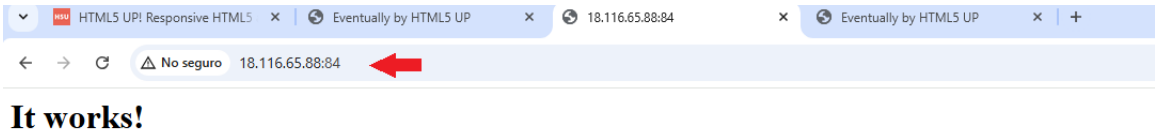
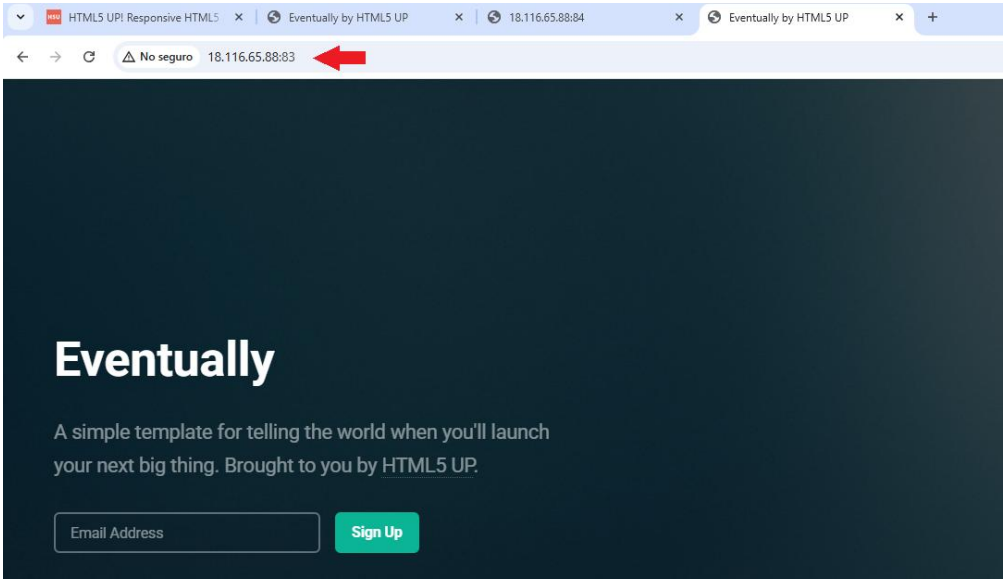
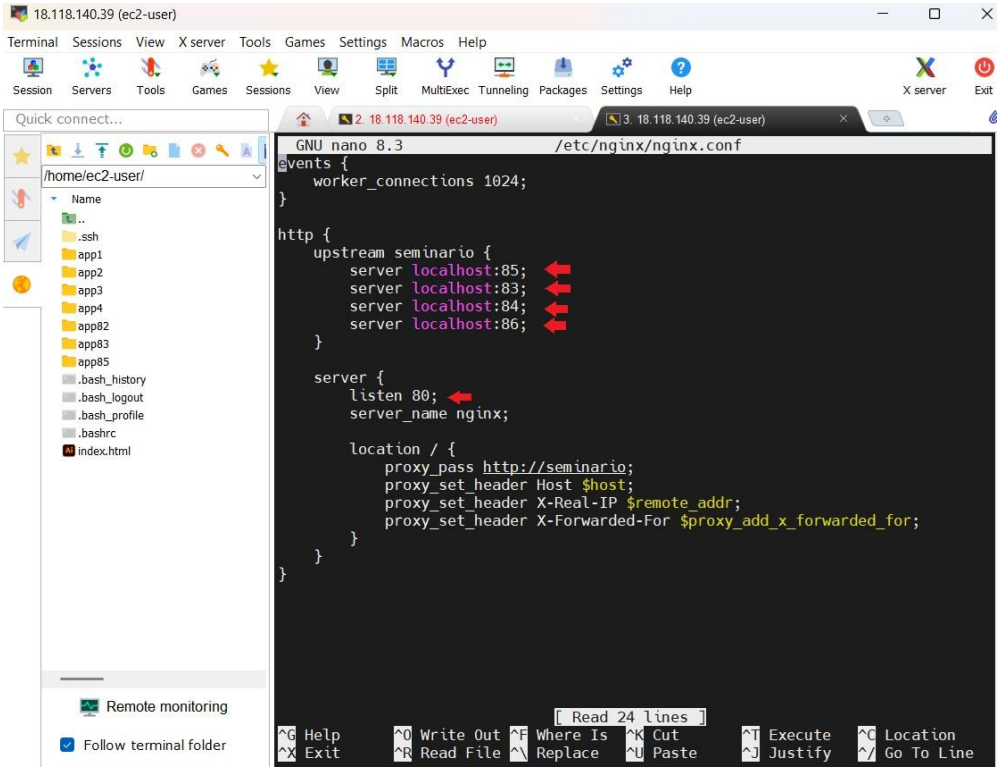


Figura 175 prueba de contenedor con el puerto 83.



Para que nuestro proxy reverso funcione de manera adecuada y muestre lo que tienen los contenedores sin acceder a cada uno desde la IP individual es necesario abrir el archivo de configuración y editar para agregar los puertos de los contenedores que tenemos disponibles.

Figura 176 aplicación del proxy reverso en el archivo.



Verificación del funcionamiento del proxy reverso con la IP pública

Para verificar que está funcionando ingresamos la IP pública de nuestra instancia y desde ahí debemos observar el cambio de contenedores al recargar la página desde el mismo puerto 80.

Efectivamente carga la información que está en el index.html del contenedor al que corresponde el puerto.

Figura 177 prueba del proxy reverso.

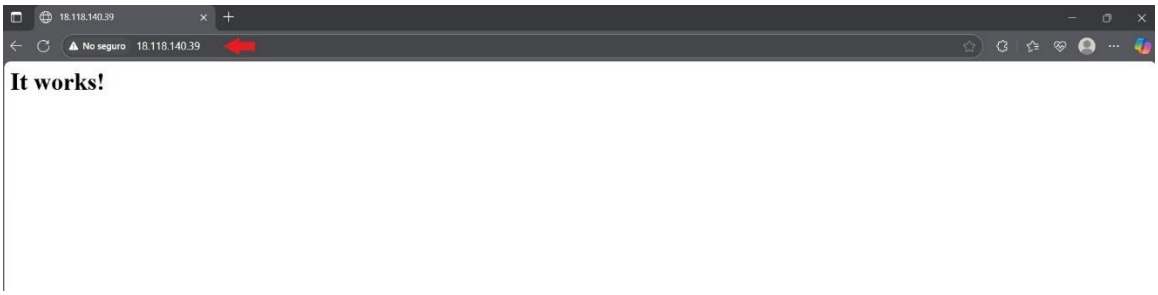


Figura 178 prueba del proxy reverso.

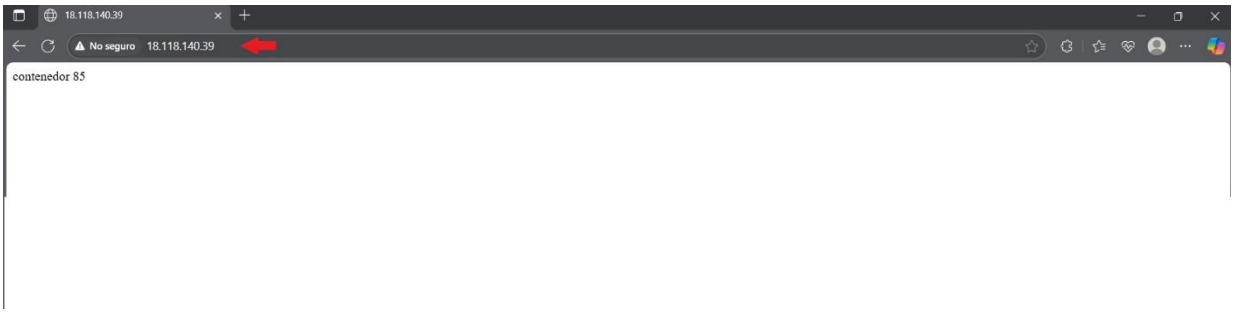


Figura 179 prueba del proxy reverso.

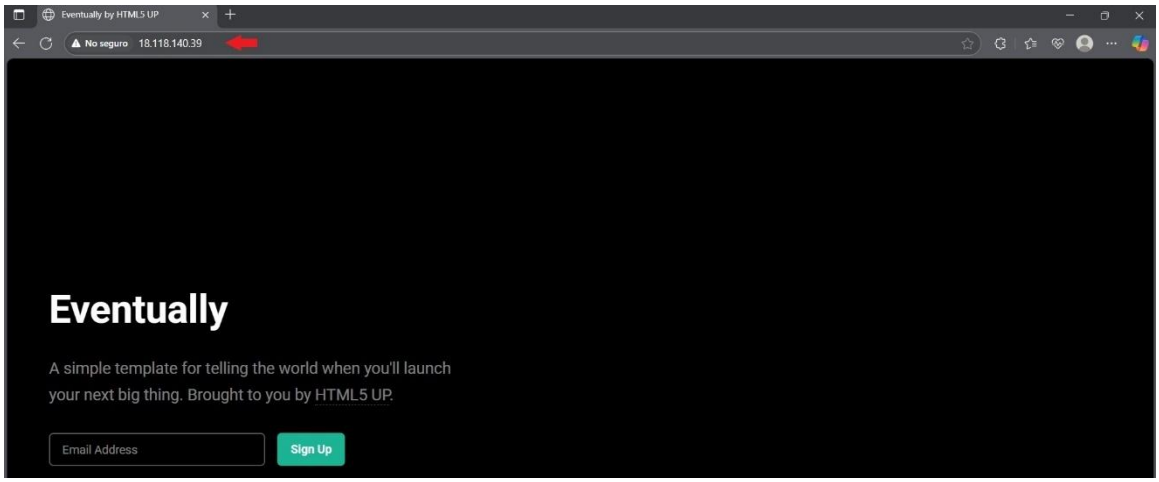
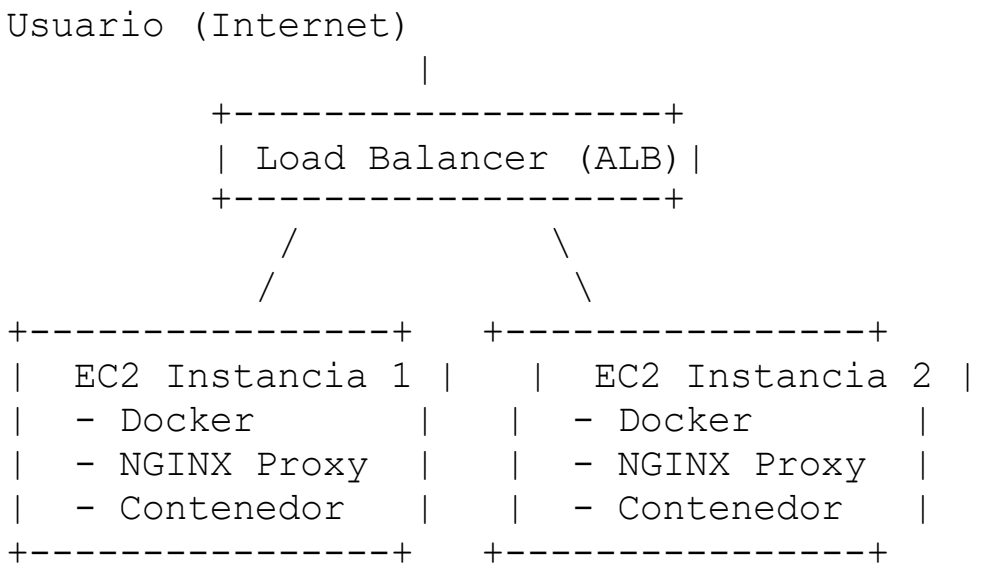


Diagrama de los servicios usados



Indice de figuras

Conclusiones

La arquitectura implementada durante el seminario AWS garantiza alta disponibilidad gracias al uso de los balanceadores de carga y la distribución en múltiples zonas de disponibilidad; ademas el uso de Docker permite un despliegue eficaz de las aplicaciones, optimizando los tiempos de configuración. Cabe destacar que el auto scaling asegura que la infraestructura puede adaptarse automáticamente a variaciones en la demanda, contribuyendo a una mayor eficiencia en el uso de recursos.

Finalmente, el uso del proxy reverso permite manejar múltiples contenedores en una sola instancia de manera ordenada y escalable. Los servicios de AWS demuestran ser herramientas solidas y en constante evolución, para desplegar infraestructuras modernas, robustas y preparadas para el crecimiento dinámico de las aplicaciones.

Referencias

AWS, A. (s.f.). *Amazon AWS.* Obtenido de https://docs.aws.amazon.com/es_es/elasticloadbalancing/latest/userguide/what-is-load-balancing.html

AWS, A. (s.f.). *Amazon AWS.* Obtenido de https://docs.aws.amazon.com/es_es/elasticloadbalancing/latest/userguide/what-is-load-balancing.html

AWS, A. (s.f.). *AWS Amazon.* Obtenido de https://docs.aws.amazon.com/es_es/ebs/latest/userguide/ebs-volumes.html

C, D. (30 de Julio de 2022). *Medium* . Obtenido de Medium.com: https://medium.com/@diego.coder/introducci%C3%B3n-a-aws-vpc-amazon-virtual-private-cloud-a8e8bd614e24

Mallón, X. (25 de Octubre de 2024). Obtenido de https://keepcoding.io/blog/que-es-una-imagen-de-maquina-de-amazon-ami/

Mallón, X. (17 de Mayo de 2024). *Keep Coding*. Obtenido de Keepcoding.io: https://keepcoding.io/blog/que-es-amazon-ec2/

Mendieta, A. d. (17 de Marzo de 2017). *OpenWebinars*. Obtenido de OpenWebinars.net: https://openwebinars.net/blog/que-es-aws/

Morales, E. V. (11 de Marzo de 2022). *Pragma*. Obtenido de pragma.co: https://www.pragma.co/es/blog/contenedores-de-aws-que-son-y-cuales-son-sus-usos-y-ventajas

NAKIVO, E. (6 de Mayo de 2024). *Naviko*. Obtenido de naviko.com: https://www.nakivo.com/es/blog/aws-ebs-snapshot-explained/

Susnjara, S., & Smalley, I. (10 de Febrero de 2025). *IBM*. Obtenido de IBM.COM: https://www.ibm.com/es-es/think/topics/cloud-computing

Takami, K. (25 de Julio de 2024). *Squads Ventures*. Obtenido de https://squads.ventures/la-importancia-de-la-escalabilidad-en-el-desarrollo-de-software/