

**Implementación de Arquitectura en AWS
para la startup “FastFoodGo”**

Corporación Universitaria Remington.
Facultad de Ingenierías
Ingeniería de Sistemas

Rony Reyes Agudelo
Nombre del Tutor: Juan Pablo Berrio López
Opción de Trabajo de grado Seminario Amazon Web Services (AWS)
2025

Agradecimientos

En primer lugar, agradezco a Dios por brindarme la fuerza, la sabiduría y el entendimiento necesarios para superar esta experiencia, la cual me permitió crecer no solo como profesional, sino también en el ámbito personal.

Agradezco también a cada persona que conocí en este camino y que, de alguna manera, formó parte de este proceso que llegó a mi vida para dejarme una enseñanza. A mi familia, a mi esposa y a mi hija, quienes durante este tiempo sacrificaron valiosos momentos en familia para brindarme el espacio y el apoyo necesario para alcanzar esta tan esperada meta. A mi suegra, gracias por creer en mí y acompañarme con su apoyo incondicional a lo largo de este trayecto.

Y, por último, pero no menos importante, a cada profesor que, con su experiencia y conocimiento, me permitió llegar hasta este trabajo final. Extiendo también mi agradecimiento a cada miembro de la universidad, ya que sin su valioso esfuerzo y dedicación no sería posible el adecuado funcionamiento de toda la comunidad educativa.

Tabla de Contenido

Resumen.....	6
Palabras clave.....	6
Marco conceptual y contextual	7
Desarrollo e Implementación del Aprendizaje	8
Implementación de una red en AWS con servidores Windows y Linux accesibles desde Internet	8
Parte 1: Documentación Técnica	8
Parte 2: Actividades Prácticas (Implementación y pruebas).....	12
Implementación de Arquitectura en AWS con Balanceador de Carga y Contenedores	25
El startup "FastFoodGo"	25
Conclusiones	38
Referencias.....	39

Tabla de Ilustraciones

Figura 1: Diagrama de Arquitectura AWS de la implementación de dos servidores	8
Figura 2: Información los VPC disponibles.....	12
Figura 3: Información detallada de seminario-vcpc	12
Figura 4: Información de subredes en el seminario-vcpc.....	13
Figura 5: Información de las dos instancias creadas para Windows y para Linux	13
Figura 6: Resumen de la Instancia Windows Server 2016	14
Figura 7: Detalle de la Instancia Windows Server 2016.....	14
Figura 8: Resumen de la Instancia Amazon Linux 2023	15
Figura 9: Detalle de la Instancia Amazon Linux 2023	15
Figura 10: Configuración del grupo de seguridad para RDP y HTTP.....	16
Figura 11: Configuración del grupo de seguridad para SSH y HTTP	16
Figura 12: Conexión por RDP a la Instancia Windows Server 2016	17
Figura 13: Conexión a RPD a Instancia Windows Server 2016.....	17
Figura 14: Conexión por SSH a la Instancia Amazon Linux 2023.....	18
Figura 15: Acceso por SSH a Instancia de Amazon Linux 2023	18
Figura 16: Instalación del rol Web Server IIS	19
Figura 17: Sitio por defecto del rol Web Server IIS	19
Figura 18: Instalación de Apache por medio del comando “dnf install httpd”.....	20
Figura 19: Activación del servicio Apache por el comando “systemctl start httpd”	20
Figura 20: Protocolo ICMP configurado en el Grupo de Seguridad de Windows Server	21
Figura 21: Protocolo ICMP configurado en el Grupo de Seguridad de Amazon Linux.....	21
Figura 22: Ping desde el servidor Windows a la IP privada del servidor Linux	22
Figura 23: Ping desde el servidor Linux a la IP privada del servidor Windows	23
Figura 24: Acceso por medio de la IP publica 18.220.39.92 de la Instancia de Windows.....	24
Figura 25: Acceso por medio de la IP publica 18.226.186.167 de la Instancia de Linux	24
Figura 26: Creación de Balanceador de carga para Docker.....	25
Figura 27: Detalles de balanceador de carga para Docker.....	26
Figura 28: Detalles del target Group para Docker	26
Figura 29: Destinos registrados en el target group para Docker.....	27
Figura 30: Aplicación que se visualiza por medio de DNS del balanceador de carga	27
Figura 31: Instancias EC2 multizona para Linux	28
Figura 32: Imagen AMI creada para Linux con Docker.....	28
Figura 33: Verificación de puertos utilizados para implementar Nginx	29
Figura 34: Instalación por SSH de nginx	29
Figura 35: Configuración de nginx	30
Figura 36: Modificación de nginx.conf para crear el proxy reverso.....	30
Figura 37: Instalación por SSH de Docker	31
Figura 38: Configuración de Docker	31
Figura 39: Instalación de una aplicación de prueba en Docker	32
Figura 40: Instalación de otra aplicación de prueba en Docker.....	32
Figura 41: Configuración de los puertos para las aplicaciones de prueba en Docker	33
Figura 42: Ingreso a aplicaciones de prueba por medio de internet con el puerto.....	33

	5
Figura 43: Platilla de lanzamiento para el Auto Scaling.....	34
Figura 44: Detalles de plantilla de lanzamiento para Auto Scaling	34
Figura 45: Creación del grupo de Auto Scaling.....	35
Figura 46: Detalles del grupo de Auto Scaling	35
Figura 47: Verificación de la creación de dos instancias EC2 por Auto Scaling	36
Figura 48: Historial de la actividad del Auto Scaling.....	36
Figura 49: Verificación de la creación automática de una nueva instancia EC2	37
Figura 50: Verificación de la eliminación automática de la instancia EC2 detenida	37

Resumen

Este trabajo está basado en implementar una aplicación startup como “FastFoodGo” sobre una arquitectura basada en servicios de Amazon Web Services (AWS).

Amazon Web Services (AWS) no solo permite desplegar infraestructura de forma ágil y bajo demanda, sino que también garantiza escalabilidad y eficiencia operativa, aspectos fundamentales para startups que deben adaptarse rápidamente a un entorno altamente competitivo.

Usar instancias EC2 permite virtualizar servidores de manera flexible, ajustando recursos a la medida del tráfico o la carga de trabajo, mientras que el uso de contenedores Docker gestionados manualmente aporta portabilidad, aislamiento y rapidez en el despliegue. La incorporación de un proxy reverso como Nginx mejora la administración del tráfico y la seguridad de las peticiones, actuando como punto de entrada para las aplicaciones y servicios. Complementariamente, el Balanceador de carga (ALB) distribuye el tráfico entre múltiples instancias, mejorando la disponibilidad del servicio. Finalmente, los grupos de Auto Scaling permiten ajustar automáticamente la capacidad de la infraestructura en función de la demanda, reduciendo costos operativos y asegurando un rendimiento óptimo sin intervención manual. En resumen, estos servicios de AWS brindan a FastFoodGo la robustez de una arquitectura moderna, la eficiencia del modelo "pago por uso", y la agilidad necesaria para innovar.

Palabras clave

Instancias (EC2): Sus siglas significan Elastic Compute Cloud, las instancias son servidores virtuales en la nube que se utilizan para ejecutar aplicaciones.

Balanceador de carga (ALB): Sus siglas significan Application Load Balancer, los balanceadores de carga son un servicio de AWS que permiten distribuir el tráfico web entrante entre múltiples servidores, de esta forma mejora el rendimiento y la disponibilidad de las aplicaciones.

Docker: Los Docker también conocidos como contenedores, son una plataforma que permite desarrollar y ejecutar aplicaciones de forma sencilla, a diferencia de las máquinas virtuales son más ligeros y consumen menos recursos.

Proxy Reverso Nginx: Un proxy reverso como Nginx es un intermediario muy similar a un balanceador de carga, recibe solicitudes de los clientes y las reenvía a los servidores adecuados sin importar que el servidor este en un puerto diferente al puerto 80.

Grupos de Auto Scaling: son colecciones cuyas políticas se usan para aumentar o reducir instancias según la carga de tráfico de la aplicación, lo que permite una administración de forma automática.

Marco conceptual y contextual

Hoy en día, emprender con tecnología no solo es una ventaja, es una necesidad. FastFoodGo es un startup pensado para agilizar las entregas de comida rápida conectando restaurantes y usuarios en línea. Para lograr que esta aplicación funcione sin interrupciones y mantenga seguros los datos de los usuarios, se requiere una infraestructura sólida como la que nos brinda Amazon Web Services (AWS). Con servicios como servidores remotos llamados instancias EC2 que permite crear servidores virtuales en la nube sin necesidad de comprar equipos físicos, FastFoodGo puede iniciar operaciones de forma ágil y económica, creciendo a medida que crecen sus usuarios, sin inversiones iniciales grandes.

Uno de los conceptos fundamentales detrás de esta implementación es la computación en la nube, la cual permite a empresas como FastFoodGo alquilar servidores, almacenamiento y servicios de red que antes requerían grandes inversiones e infraestructura física. Según lo explica Rodríguez Ramírez (2022), Cloud Computing es un modelo que optimiza el uso de recursos tecnológicos según demanda, garantizando flexibilidad, ahorro y eficiencia operativa. Por ejemplo, gracias a los grupos de Auto Scaling de AWS, FastFoodGo puede ajustar automáticamente la cantidad de servidores en función del tráfico: si muchas personas hacen pedidos a la vez, el sistema escala para evitar caídas; si hay pocos pedidos, reduce su tamaño, ahorrando dinero.

Finalmente, para garantizar que los pedidos lleguen rápido y sin errores, FastFoodGo utiliza herramientas modernas como Docker, que permite empaquetar y ejecutar la aplicación en cualquier servidor sin preocuparse por el sistema operativo, y Nginx como proxy reverso, el cual dirige cada petición del usuario al servidor correcto sin importar en que puerto se encuentre la aplicación. El uso del balanceador de carga de AWS permite distribuir las solicitudes entre varios servidores o instancias EC2, asegurando que el sistema no se sobrecargue. Amazon web Services (AWS) hace posible que una persona común pueda usar una app estable, segura y rápida, sin saber que detrás hay una arquitectura compleja funcionando en la nube.

Desarrollo e Implementación del Aprendizaje

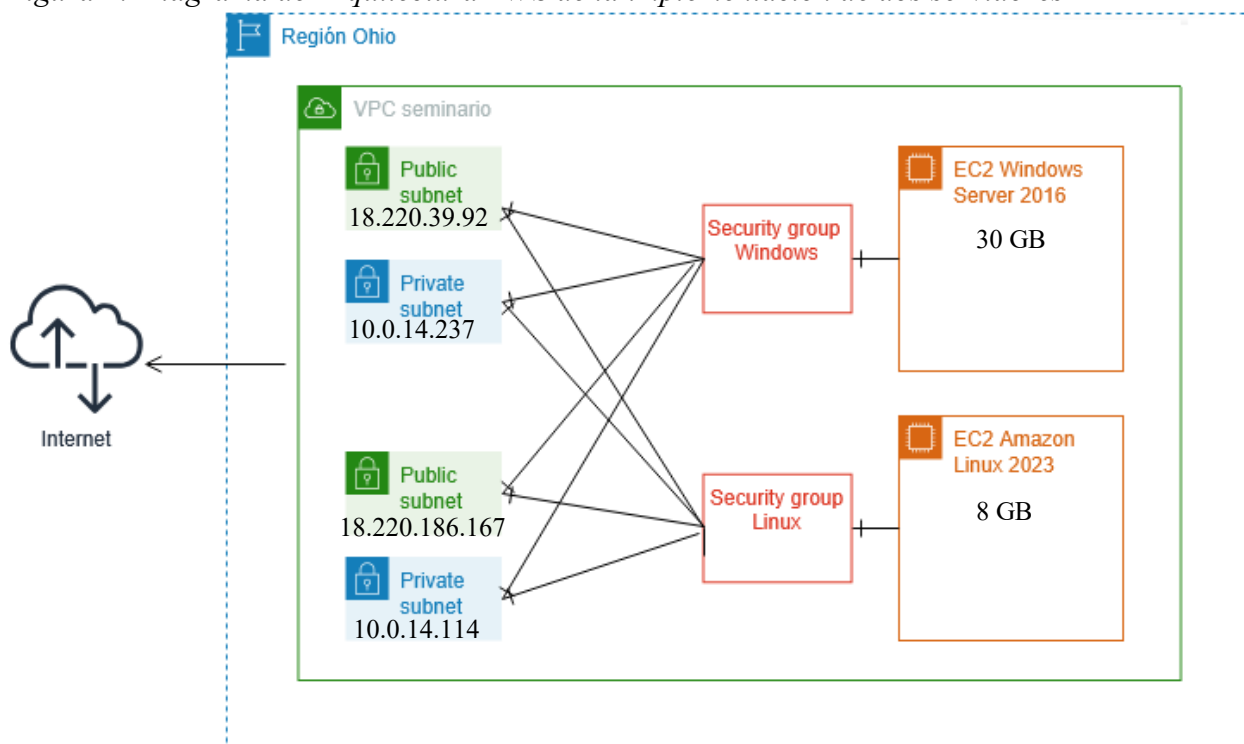
Implementación de una red en AWS con servidores Windows y Linux accesibles desde Internet

Parte 1: Documentación Técnica

1. Diagrama de arquitectura

Realizar una representación gráfica de la red (EC2s, subredes, IPs públicas/ privadas, grupos de seguridad, VPC, etc.)

Figura 1: Diagrama de Arquitectura AWS de la implementación de dos servidores



2. Descripción de la arquitectura

Se diseñó una red en AWS mediante la creación de una VPC llamada "seminario-vpc". Dentro de esta VPC se crearon dos subredes públicas, permitiendo que ambas instancias EC2, una con sistema operativo Windows Server 2016 y otra con Amazon Linux 2023, tengan conectividad directa a Internet a través de un Internet Gateway.

El uso de una VPC permite mayor control sobre los componentes de red, y la configuración de subredes y grupos de seguridad garantizando el acceso controlado y seguro a

las instancias. Se asignaron a las instancias direcciones IP públicas para facilitar el acceso externo.

Tipo de instancias usadas:

Linux: Amazon Linux 2023, tipo t2.micro.

Windows: Windows Server 2016 Base, tipo t2.micro.

Justificación de las configuraciones de red:

VPC: control completo de direccionamiento, ruteo y seguridad.

Subred pública: permite acceso directo desde Internet.

Internet Gateway: proporciona acceso de salida a Internet.

Grupos de seguridad: permiten accesos específicos (RDP, SSH, HTTP) con restricción por IP en caso de que aplique para algún requerimiento.

3. Configuraciones realizadas

Pasos para crear las instancias EC2:

La instancia para Windows Server 2016 se creó ingresando al panel de EC2 y seleccionamos la opción “Lanzar instancias”, se seleccionó la AMI llamada “Windows Server 2016 Base” para la capa gratuita, en el tipo de instancia se seleccionó “tipo t2.micro”, creamos un par de claves privadas para el inicio de sesión (PEM), habilitamos la creación de IP pública, configuramos el grupo de seguridad, seleccionamos un disco duro de 30GB.

Para crear la instancia de Linux, ingresamos al panel de EC2 y seleccionamos la opción “Lanzar instancias”, se seleccionó la AMI llamada “Amazon Linux 2023” que es la última versión para la capa gratuita, el tipo de instancia se seleccionó “tipo t2.micro”, se utilizó el mismo par de claves privadas para el inicio de sesión (PEM) de la instancia de Windows, habilitamos la creación de IP pública, configuramos el grupo de seguridad, seleccionamos un disco duro de 8GB.

Detalles de los Grupos de Seguridad:

Para dejar un puerto abierto a todo Internet se configura así (0.0.0.0/0).

RDP: se configura en el puerto 3389.

SSH: se configura en el puerto 22.

HTTP: se configura en el puerto 80.

ICMP: se habilita para que permita hacer ping.

Asignación de IPs públicas y privadas:

Las IPs privadas se crean automáticamente, para las IPs públicas se debe habilitar la opción “asignar automáticamente la IP pública” al momento de crear la instancia.

4. Procedimiento de acceso

Cómo acceder al servidor Windows Server 2016 utilizando cliente de RDP:

Para acceder al servidor de Windows utilizando cliente de RDP, primero seleccionamos la instancia de Windows Server 2016 en el panel EC2 y le damos en la opción “conectar”, seleccionamos la pestaña “cliente de RDP”, cargamos la clave privada en el archivo PEM en la opción “obtener contraseña”. Abrimos la aplicación de “Conexión Escritorio Remoto” que viene en el sistema operativo, colocamos la IP pública generada por la instancia, luego nos solicita usuario que es “Administrator” y la contraseña generada del archivo PEM, le damos conectar y nos abre una ventana para trabajar en nuestro servidor remoto Windows Server 2016

Cómo acceder al servidor Amazon Linux 2023 utilizando cliente SSH:

Para conectarnos utilizando cliente SSH debemos instalar una aplicación que nos permita esa conexión por consola como “MobaXterm”. Seleccionamos la instancia de Amazon Linux 2023 en el panel EC2 y le damos en la opción “conectar”, seleccionamos la pestaña “cliente SSH” para copiar el “remote host” y el “username” con el que nos conectaremos remotamente por medio de la aplicación “MobaXterm”, cargamos la clave privada del archivo PEM en “Advanced SSH settings” y le damos “ok”. La aplicación no da un mensaje de bienvenida y ya podemos ingresar por medio de comandos SSH en nuestro servidor Amazon Linux 2023.

Consideraciones de seguridad:

uso de llaves PEM: es una clave única que permite la conexión remota a un servidor.
contraseñas seguras: son esenciales por razones de seguridad, para la protección de recursos críticos y la prevención de accesos no autorizados.

5. Configuración del servidor web

Pasos seguidos para instalar IIS en Windows Server 2016:

Por medio de la aplicación “Conexión Escritorio Remoto” ingresamos a la “tecla de Windows” del servidor y seleccionamos “Server Manager”, luego seleccionamos la opción número 2 para agregar roles, seguido seleccionamos la pestaña “Server Roles” y habilitamos el rol “Web Server IIS” y le damos siguiente para continuar con la instalación. Ya instalado lo podemos configurar ingresando a la “tecla Windows” del servidor, seleccionamos la carpeta “Windows Administrative Tools” y seguido abrimos “Internet Information Services (IIS)”.

Pasos para instalar Apache en Amazon Linux 2023:

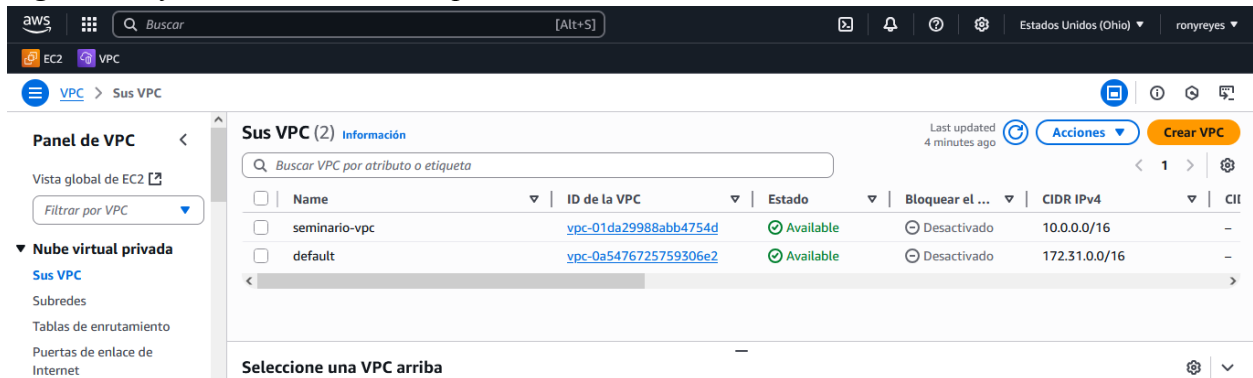
Nos conectamos por medio la aplicación “MobaXterm” al SSH del servidor Amazon Linux 2023, escribimos el comando “sudo su” para acceder a los permisos de administrador (#), luego escribimos el comando “dnf install httpd” lo que hace es buscar la versión de Apache para instalarla nos pide confirmación y le decimos que si (y). Al finalizar la instalación de Apache nos sale el mensaje: “Complete!”, verificamos el estado del Apache con el comando “systemctl status httpd”, en caso de que el Apache este inactivo, utilizamos el comando “systemctl start httpd” para activarlo, y se puede verificar nuevamente si ya está activado con el comando “systemctl status httpd”.

Parte 2: Actividades Prácticas (Implementación y pruebas)

1. Crear la infraestructura

VPC y subredes: se creó un VPC llamado “seminario-vcpc” en la región de Ohio, este VPC se creó con dos subredes públicas y dos subredes privadas.

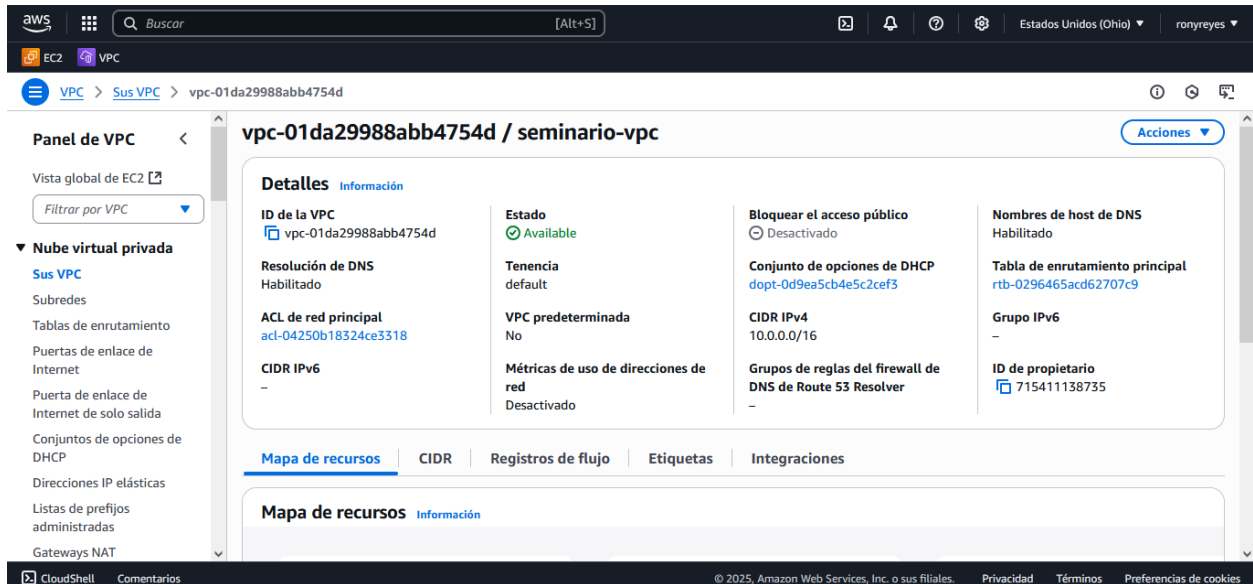
Figura 2: Información los VPC disponibles



The screenshot shows the AWS Management Console interface for VPCs. On the left, there is a navigation menu with options like 'Panel de VPC', 'Nube virtual privada', 'Subredes', 'Tablas de enrutamiento', 'Puertas de enlace de Internet', 'Puerta de enlace de Internet de solo salida', 'Conjuntos de opciones de DHCP', 'Direcciones IP elásticas', 'Listas de prefijos administradas', and 'Gateways NAT'. The main content area displays a table of VPCs under the heading 'Sus VPC (2) Información'. The table has columns for Name, ID de la VPC, Estado, Bloquear el..., CIDR IPv4, and CII. Two VPCs are listed: 'seminario-vcpc' with ID 'vpc-01da29988abb4754d' and 'default' with ID 'vpc-0a5476725759306e2'. Both are in an 'Available' state. Below the table, there is a prompt 'Seleccione una VPC arriba'.

Nombre	ID de la VPC	Estado	Bloquear el...	CIDR IPv4	CII
seminario-vcpc	vpc-01da29988abb4754d	Available	Desactivado	10.0.0.0/16	-
default	vpc-0a5476725759306e2	Available	Desactivado	172.31.0.0/16	-

Figura 3: Información detallada de seminario-vcpc



The screenshot shows the detailed view of the VPC 'vpc-01da29988abb4754d / seminario-vcpc'. The interface includes a navigation menu on the left and a main content area with a 'Detalles' section. The details are organized into a grid:

- ID de la VPC:** vpc-01da29988abb4754d
- Estado:** Available
- Bloquear el acceso público:** Desactivado
- Nombres de host de DNS:** Habilitado
- Resolución de DNS:** Habilitado
- Tenencia:** default
- Conjunto de opciones de DHCP:** dopt-0d9ea5cb4e5c2cef3
- Tabla de enrutamiento principal:** rtb-0296465acd62707c9
- ACL de red principal:** acl-04250b18324ce3318
- VPC predeterminada:** No
- CIDR IPv4:** 10.0.0.0/16
- Grupo IPv6:** -
- CIDR IPv6:** -
- Métricas de uso de direcciones de red:** Desactivado
- Grupos de reglas del firewall de DNS de Route 53 Resolver:** -
- ID de propietario:** 715411138735

Below the details, there are tabs for 'Mapa de recursos', 'CIDR', 'Registros de flujo', 'Etiquetas', and 'Integraciones'. The 'Mapa de recursos' tab is selected, showing a section for 'Mapa de recursos Información'.

Figura 4: Información de subredes en el seminario-vpc

The screenshot shows the AWS Management Console interface for a VPC named 'seminario-vpc'. The main area displays a 'Mapa de recursos' (Resource Map) with three panels:

- VPC (seminario-vpc):** Su red virtual de AWS.
- Subredes (4):** Subredes dentro de esta VPC.
 - us-east-2a:**
 - seminario-subnet-public1-us-east-2a
 - seminario-subnet-private1-us-east-2a
 - us-east-2b:**
 - seminario-subnet-public2-us-east-2b
 - seminario-subnet-private2-us-east-2b
- Tablas de enrutamiento (4):** Dirigir el tráfico de red a los recursos.
 - seminario-rtb-public
 - seminario-rtb-private2-us-east-2b
 - rtb-0296465acd62707c9
 - seminario-rtb-private1-us-east-2a

Lanzar dos instancias EC2: se creó una instancia en Windows (Windows Server 2016 Base, tipo t2.micro) y se creó otra instancia en Linux (Amazon Linux 2, tipo t2.micro).

Figura 5: Información de las dos instancias creadas para Windows y para Linux

The screenshot shows the AWS Management Console 'Instancias (2)' page. The table below lists the two instances:

Name	ID de la instancia	Estado de la instancia	Tipo de instancia	Comprobación de estado	Estado de la alarma	Zona de disponibilidad	DNS de IPv4 pública
AmazonLinux2023	i-008c90248a2d14b7b	En ejecución	t2.micro	2/2 comprobaciones	Ver alarmas +	us-east-2a	ec2-18-226-186-167.us-...
WindowsServer2016	i-0b0d36ef6f31c1af7	En ejecución	t2.micro	2/2 comprobaciones	Ver alarmas +	us-east-2a	ec2-18-220-39-92.us-...

Instancia Windows Server 2016:

Figura 6: Resumen de la Instancia Windows Server 2016

Resumen de instancia de i-0b0d36ef6f31c1af7 (WindowsServer2016) Información

Se ha actualizado hace less than a minute

ID de la instancia
i-0b0d36ef6f31c1af7

Dirección IPv6
-

Tipo de nombre de anfitrión
Nombre de IP: ip-10-0-14-237.us-east-2.compute.internal

Responder al nombre DNS de recurso privado
-

Dirección IP asignada automáticamente
3.133.143.184 [IP pública]

Rol de IAM
-

Dirección IPv4 pública
3.133.143.184 | dirección abierta

Estado de la instancia
En ejecución

Nombre DNS de IP privada (solo IPv4)
ip-10-0-14-237.us-east-2.compute.internal

Tipo de instancia
t2.micro

ID de VPC
vpc-01da2998abb4754d (seminario-vpc)

ID de subred
subnet-05825adf560a0c138 (seminario-subnet-public1-us-east-2a)

Direcciones IPv4 privadas
10.0.14.237

DNS público
ec2-3-133-143-184.us-east-2.compute.amazonaws.com | dirección abierta

Direcciones IP elásticas
-

Hallazgo de AWS Compute Optimizer
Suscribirse a AWS Compute Optimizer para recibir recomendaciones. | Más información

Nombre del grupo de Auto Scaling
-

Figura 7: Detalle de la Instancia Windows Server 2016

▼ Detalles de la instancia Información

ID de AMI
ami-0101d36a1f0ada268

Nombre de AMI
Windows_Server-2016-English-Full-Base-2025.06.11

Detener la protección
desactivado

Migración por reinicio de instancias
Predeterminada (activado)

Comportamiento de detención de hibernación
desactivado

Motivo de transición de estado
-

Mensaje de transición de estado
-

Propietario
715411138735

Monitoreo
desactivado

Imagen permitida
-

Hora de lanzamiento
Thu Jul 03 2025 23:11:59 GMT-0500 (hora estándar de Colombia) (about 1 hour)

Recuperación automática de instancias
Predeterminada

Índice de lanzamiento de AMI
0

Especificación de crédito
standard

Operación de uso
RunInstances:0002

Compatibilidad con enclaves
-

Detalles de la plataforma
Windows

Protección de terminación
desactivado

Ubicación de AMI
amazon/Windows_Server-2016-English-Full-Base-2025.06.11

Ciclo de vida
normal

Par de claves asignado en el lanzamiento
WindowsServer1

ID de kernel
-

ID de disco RAM
-

Modo de arranque
-

Instancia Linux (Amazon Linux 2023):

Figura 8: Resumen de la Instancia Amazon Linux 2023

The screenshot displays the 'Resumen de instancia de i-008c90248a2d14b7b (AmazonLinux2023)' page in the AWS console. The instance is in the 'En ejecución' state. Key details include:

- ID de la instancia:** i-008c90248a2d14b7b
- Dirección IPv4 pública:** 18.226.186.167
- Direcciones IPv4 privadas:** 10.0.14.114
- Tipo de instancia:** t2.micro
- Nombre DNS de IP privada (solo IPv4):** ip-10-0-14-114.us-east-2.compute.internal
- Dirección IP asignada automáticamente:** 18.226.186.167 [IP pública]
- ID de VPC:** vpc-01da29988abb4754d (seminario-vpc)
- ID de subred:** subnet-05825adf560a0c138 (seminario-subnet-public1-us-east-2a)

Figura 9: Detalle de la Instancia Amazon Linux 2023

The screenshot displays the 'Detalles de la instancia' page in the AWS console, providing a comprehensive view of the instance's configuration:

- ID de AMI:** ami-0c803b171269e2d72
- Nombre de AMI:** al2023-ami-2023.7.20250623.1-kernel-6.1-x86_64
- Monitoreo:** desactivado
- Imagen permitida:** -
- Detalles de la plataforma:** Linux/UNIX
- Protección de terminación:** desactivado
- Ubicación de AMI:** amazon/al2023-ami-2023.7.20250623.1-kernel-6.1-x86_64
- Detener la protección:** desactivado
- Hora de lanzamiento:** Fri Jul 04 2025 06:23:37 GMT-0500 (hora estándar de Colombia) (6 minutos)
- Migración por reinicio de instancias:** Predeterminada (activado)
- Recuperación automática de instancias:** Predeterminada
- Ciclo de vida:** normal
- Comportamiento de detención de hibernación:** desactivado
- Índice de lanzamiento de AMI:** 0
- Par de claves asignado en el lanzamiento:** WindowsServer1
- Motivo de transición de estado:** -
- Especificación de crédito:** standard
- ID de kernel:** -
- Mensaje de transición de estado:** -
- Operación de uso:** RunInstances
- ID de disco RAM:** -
- Propietario:** 715411138735
- Compatibilidad con enclaves:** -
- Modo de arranque:** uefi-preferred

2. Configurar Grupos de Seguridad

HTTP (puerto 80) abierto a todo Internet (0.0.0.0/0) para ambos.

RDP (puerto 3389) para Windows desde la IP pública del alumno.

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

Figura 10: Configuración del grupo de seguridad para RDP y HTTP

The screenshot shows the AWS Management Console interface for a security group named 'sg-03b8d8b28fe81bffe' in the 'sgWindowsServer1' group. The details section shows the group name, ID, description, and VPC ID. The 'Reglas de entrada' (Inbound Rules) section is active, displaying two rules:

Nombre	ID de la regla del gr...	Versión de IP	Tipo	Protocolo	Intervalo de pu...	Origen
-	sgr-064fe16939e59c79d	IPv4	HTTP	TCP	80	0.0.0.0/0
-	sgr-0e92851ab72e59b9f	IPv4	RDP	TCP	3389	190.108.79.138/32

Figura 11: Configuración del grupo de seguridad para SSH y HTTP

The screenshot shows the AWS Management Console interface for a security group named 'sg-08093de33613c9188' in the 'sgAmazonLinux' group. The details section shows the group name, ID, description, and VPC ID. The 'Reglas de entrada' (Inbound Rules) section is active, displaying two rules:

Nombre	ID de la regla del gr...	Versión ...	Tipo	Protocolo	Intervalo de puertos	Origen
-	sgr-0fe411801b5241db6	IPv4	HTTP	TCP	80	0.0.0.0/0
-	sgr-0a918a154cb76acd5	IPv4	SSH	TCP	22	190.108.79.138/32

3. Acceder a las instancias

Acceder vía RDP a la instancia Windows

Figura 12: Conexión por RDP a la Instancia Windows Server 2016

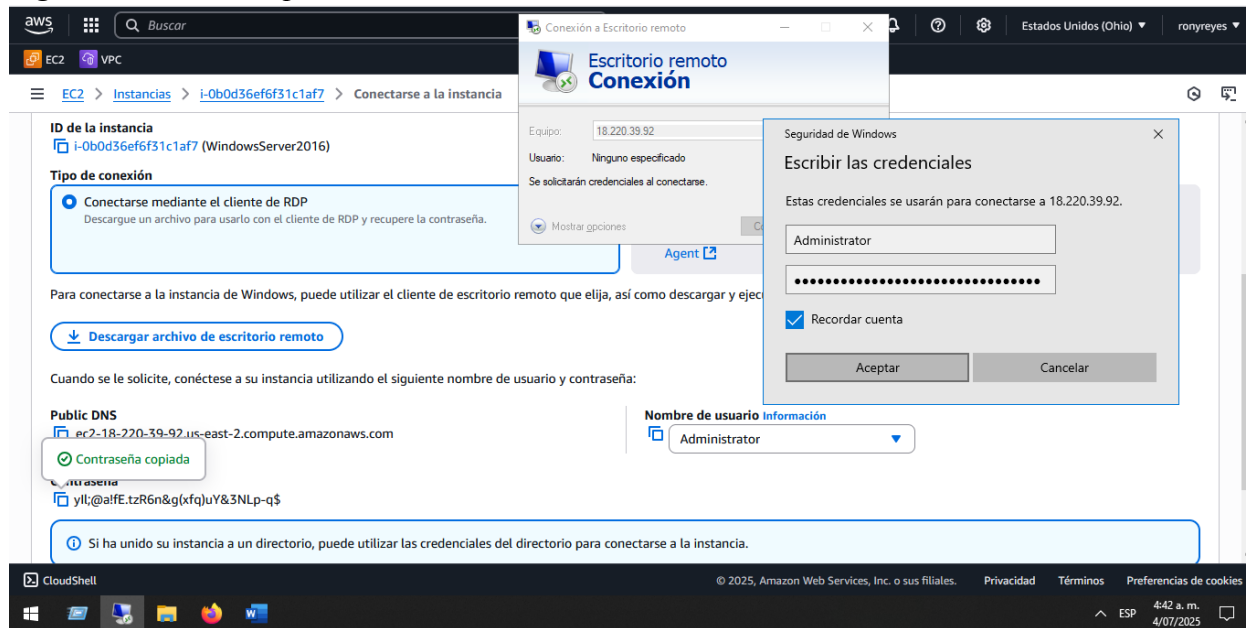
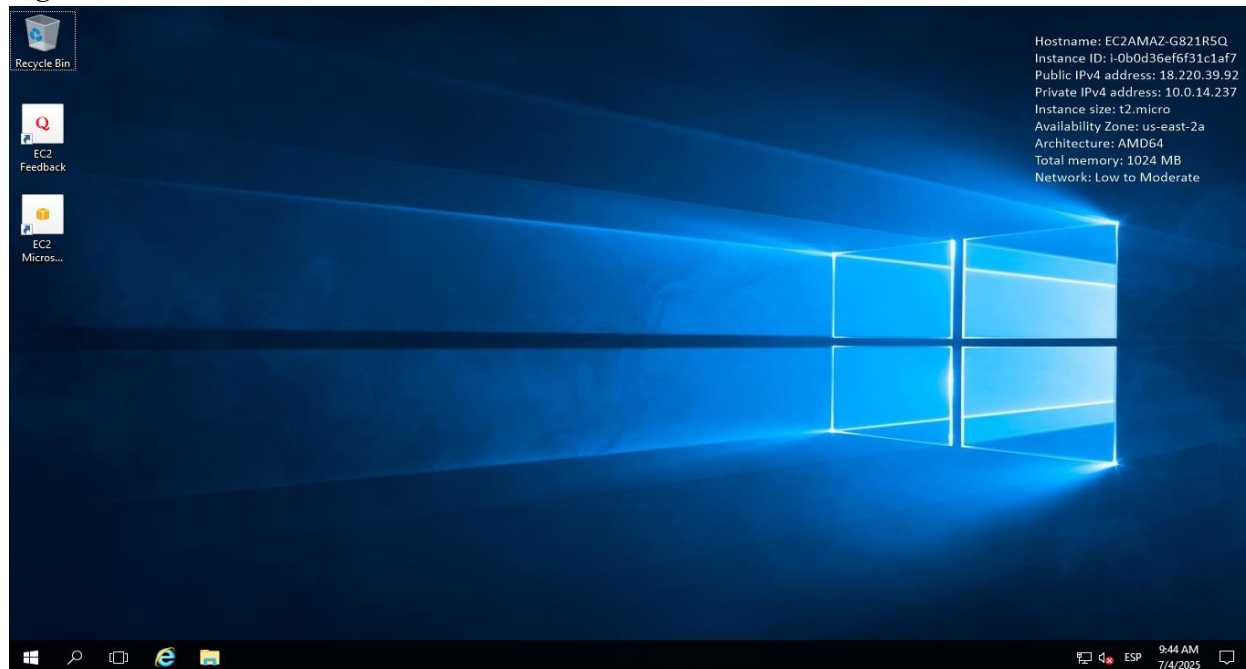


Figura 13: Conexión a RPD a Instancia Windows Server 2016



Acceder vía SSH a la instancia Linux:

Figura 14: Conexión por SSH a la Instancia Amazon Linux 2023

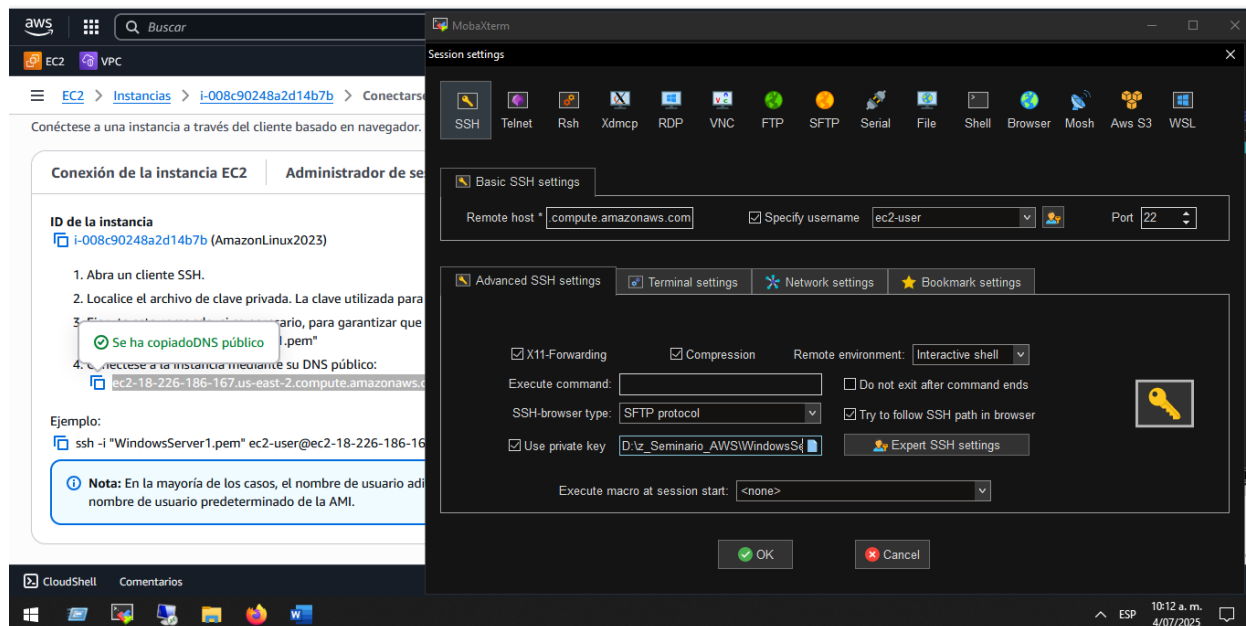
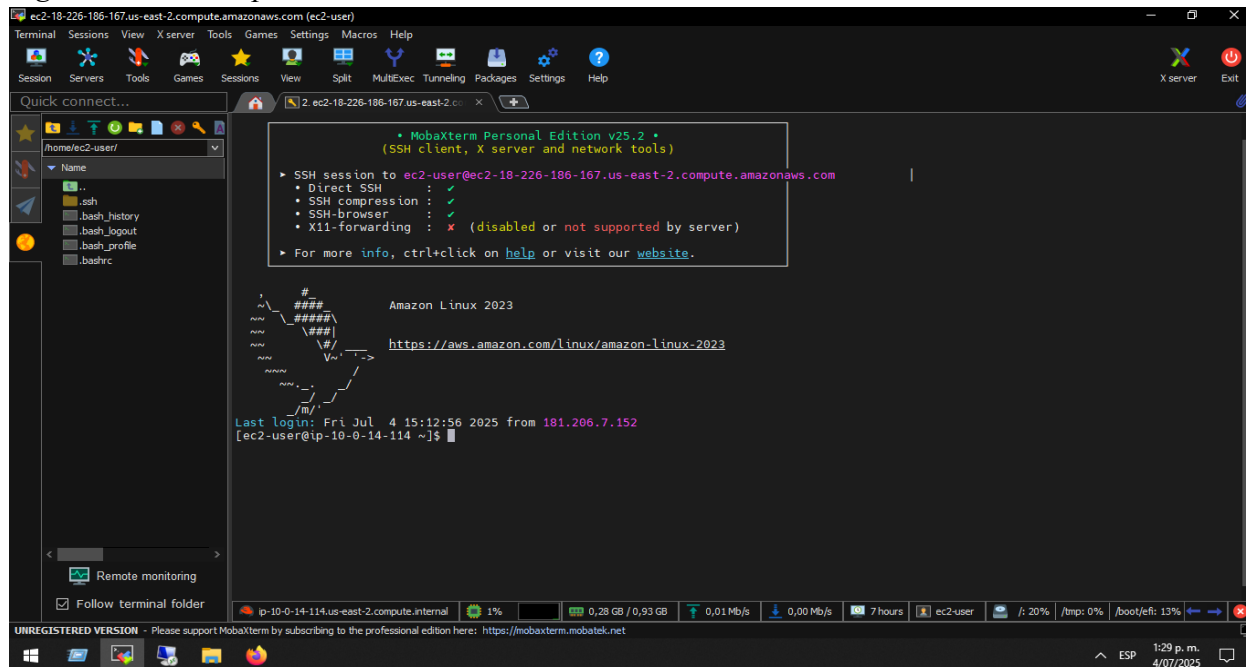


Figura 15: Acceso por SSH a Instancia de Amazon Linux 2023



4. *Instalar y configurar los servidores web*

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

Figura 16: Instalación del rol Web Server IIS

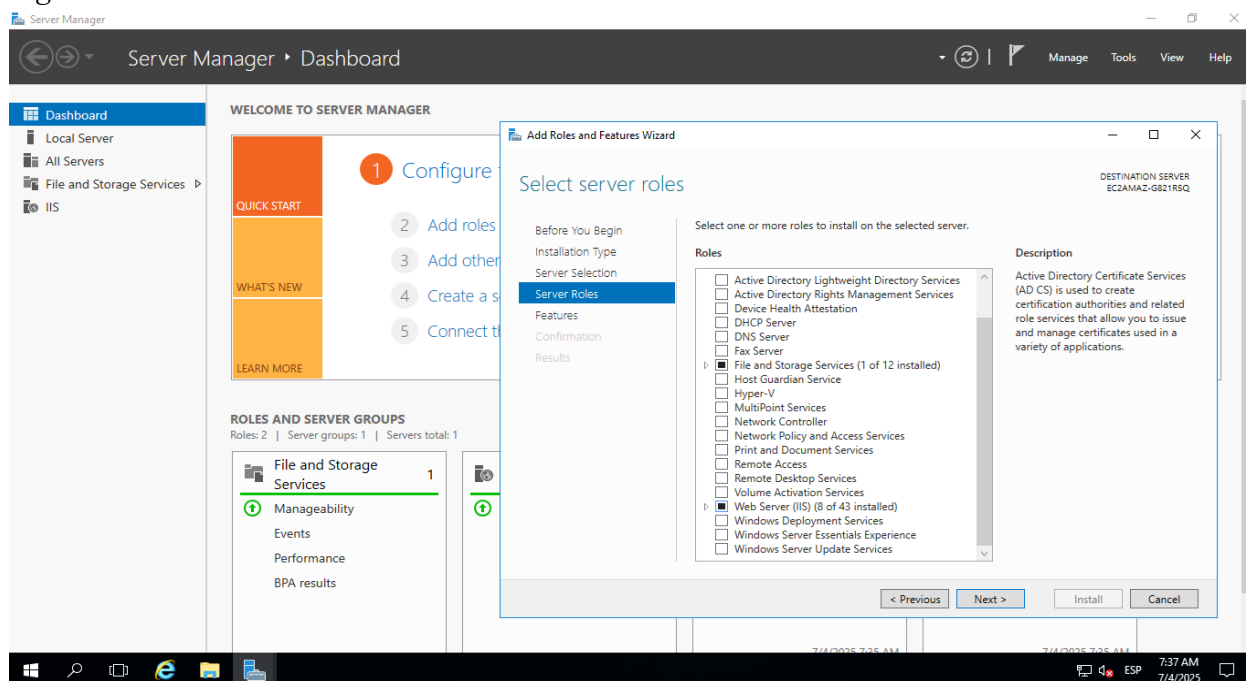
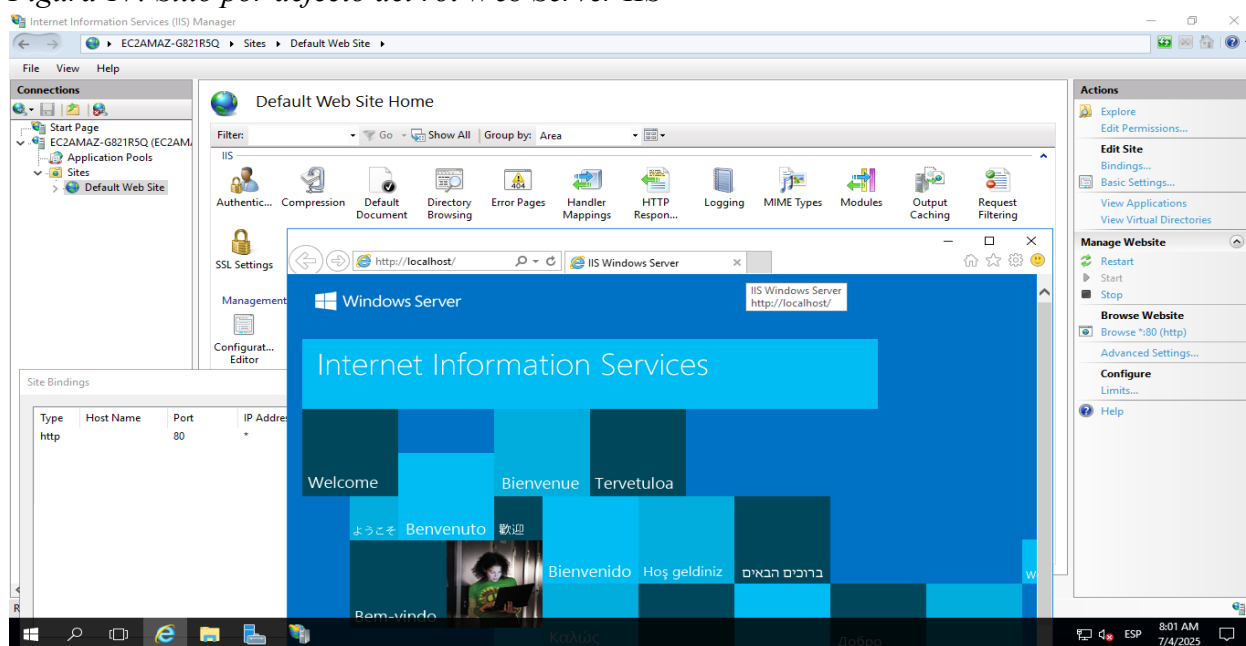


Figura 17: Sitio por defecto del rol Web Server IIS



Linux: Instalar Apache y levantar el sitio por defecto.

Figura 18: Instalación de Apache por medio del comando “dnf install httpd”

```

[ec2-user@ip-10-0-14-114 ~]$ sudo su
[root@ip-10-0-14-114 ec2-user]# dnf install httpd
Amazon Linux 2023 Kernel Livepatch repository                               161 kB/s | 17 kB   00:00
Dependencies resolved.
=====
Package                        Architecture      Version            Repository          Size
-----
Installing:
httpd                          x86_64            2.4.62-1.amzn2023  amazonlinux         48 k
Installing dependencies:
apr                             x86_64            1.7.5-1.amzn2023.0.4  amazonlinux         129 k
apr-util                       x86_64            1.6.3-1.amzn2023.0.1  amazonlinux         98 k
generic-logos-httpd           noarch            18.0-0-12.amzn2023.0.3  amazonlinux         1.4 M
httpd-core                    x86_64            2.4.62-1.amzn2023    amazonlinux          14 k
httpd-filesystem              noarch            2.4.62-1.amzn2023    amazonlinux          14 k
httpd-tools                   x86_64            2.4.62-1.amzn2023    amazonlinux          81 k
libbrotli                     x86_64            1.0.9-4.amzn2023.0.2  amazonlinux         315 k
mailcap                       noarch            2.1.49-3.amzn2023.0.3  amazonlinux         33 k
Installing weak dependencies:
apr-util-openssl              x86_64            1.6.3-1.amzn2023.0.1  amazonlinux         17 k
mod_http2                     x86_64            2.0.27-1.amzn2023.0.3  amazonlinux         166 k
mod_lua                        x86_64            2.4.62-1.amzn2023    amazonlinux          61 k
Transaction Summary
-----
Install 12 Packages
Total download size: 2.3 M
Installed size: 6.9 M
Is this ok [y/N]: y
  
```

Figura 19: Activación del servicio Apache por el comando “systemctl start httpd”

```

Verifying : mod_lua-2.4.62-1.amzn2023.x86_64                               12/12
Installed:
apr-1.7.5-1.amzn2023.0.4.x86_64      apr-util-1.6.3-1.amzn2023.0.1.x86_64      apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch  httpd-2.4.62-1.amzn2023.x86_64      httpd-core-2.4.62-1.amzn2023.x86_64
httpd-filesystem-2.4.62-1.amzn2023.noarch          httpd-tools-2.4.62-1.amzn2023.x86_64      libbrotli-1.0.9-4.amzn2023.0.2.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch              mod_http2-2.0.27-1.amzn2023.0.3.x86_64  mod_lua-2.4.62-1.amzn2023.x86_64

Complete!
[root@ip-10-0-14-114 ec2-user]# systemctl status httpd
o httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: inactive (dead)
   Docs: man:httpd.service(8)
[root@ip-10-0-14-114 ec2-user]# systemctl start httpd
[root@ip-10-0-14-114 ec2-user]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: active (running) since Fri 2025-07-04 15:46:53 UTC; 28s ago
     Docs: man:httpd.service(8)
    Main PID: 45810 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/sec"
     Tasks: 177 (limit: 1111)
    Memory: 13.0M
       CPU: 80ms
    CGroup: /system.slice/httpd.service
           └─45810 /usr/sbin/httpd -DFOREGROUND
             └─45811 /usr/sbin/httpd -DFOREGROUND
               └─45812 /usr/sbin/httpd -DFOREGROUND
                 └─45813 /usr/sbin/httpd -DFOREGROUND
                   └─45814 /usr/sbin/httpd -DFOREGROUND

Jul 04 15:46:53 ip-10-0-14-114.us-east-2.compute.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Jul 04 15:46:53 ip-10-0-14-114.us-east-2.compute.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Jul 04 15:46:53 ip-10-0-14-114.us-east-2.compute.internal httpd[45810]: Server configured, listening on: port 80
[root@ip-10-0-14-114 ec2-user]#
  
```

5. Pruebas de conectividad

Para realizar las pruebas de conectividad realizando un ping a la IP privada del otro servidor y viceversa debemos agregar una regla a los Grupos de Seguridad de los dos servidores (Windows y Linux) y configurar el protocolo ICMP para que permita tráfico de un servidor a otro.

Figura 20: Protocolo ICMP configurado en el Grupo de Seguridad de Windows Server

The screenshot shows the AWS Management Console interface for the 'sg-03b8d8b28fe81bffe - sgWindowsServer1' security group. The 'Reglas de entrada' (Inbound rules) tab is selected, showing a table with three rules:

Nombre	ID de la regla del grupo de seguridad	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Origen
-	sgr-02bdae719f9b5dd19	IPv4	Todos los ICMP IPv4	ICMP	Todo	0.0.0.0/0
-	sgr-064fe16939e59c79d	IPv4	HTTP	TCP	80	0.0.0.0/0
-	sgr-0e92851ab72e59b9f	IPv4	RDP	TCP	3389	190.108.79.138/32

Figura 21: Protocolo ICMP configurado en el Grupo de Seguridad de Amazon Linux

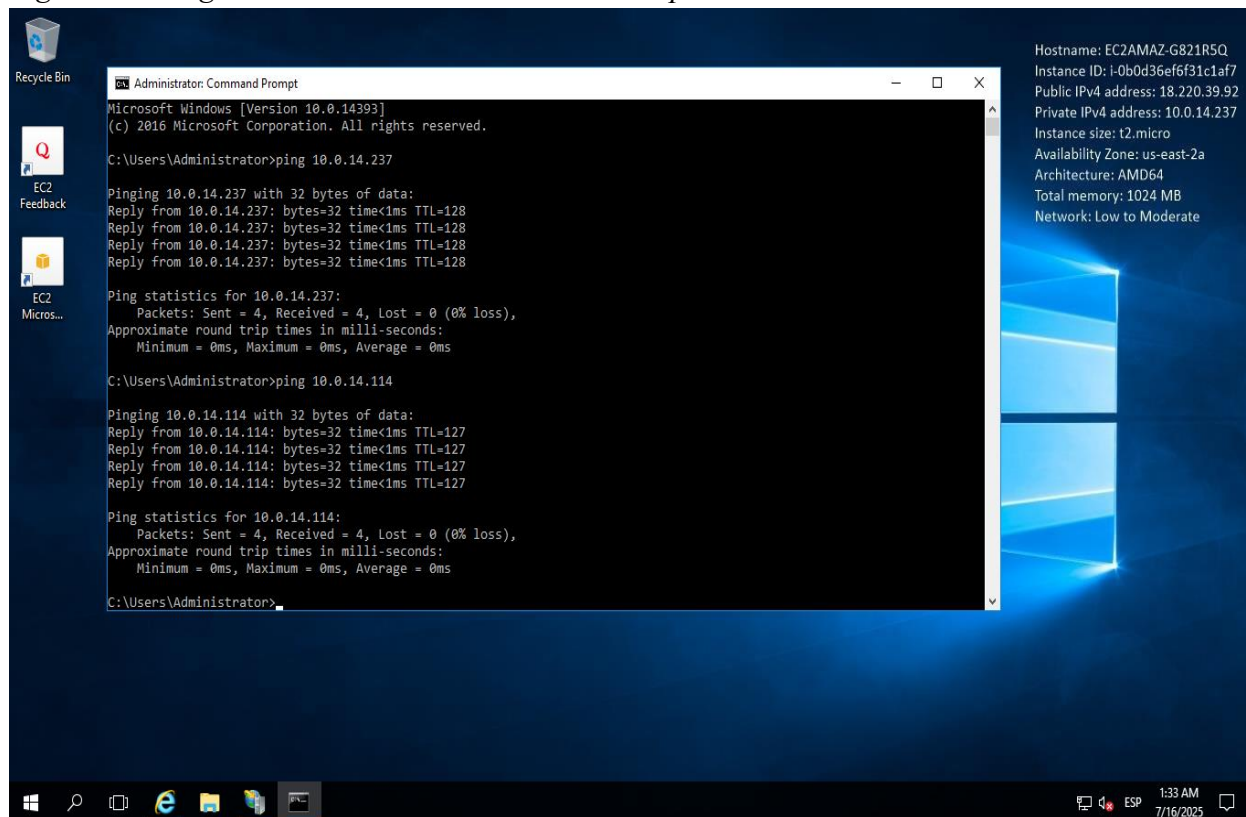
The screenshot shows the AWS Management Console interface for the 'sg-08093de33613c9188 - sgAmazonLinux' security group. The 'Reglas de entrada' (Inbound rules) tab is selected, showing a table with three rules:

Nombre	ID de la regla del grupo de seguridad	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Origen
-	sgr-00f0f4244194f057a	IPv4	Todos los ICMP IPv4	ICMP	Todo	0.0.0.0/0
-	sgr-0fe411801b5241db6	IPv4	HTTP	TCP	80	0.0.0.0/0
-	sgr-0a918a154cb76acd5	IPv4	SSH	TCP	22	190.108.79.138/32

Desde la instancia Windows hacer *ping* a la IP privada de la instancia Linux:

Nos conectamos al servidor Windows Server 2016 por medio de cliente RDP, abrimos la aplicación de “Command” y le hacemos un ping a la IP privada del servidor de Amazon Linux 2023 (IP privada 10.0.14.114).

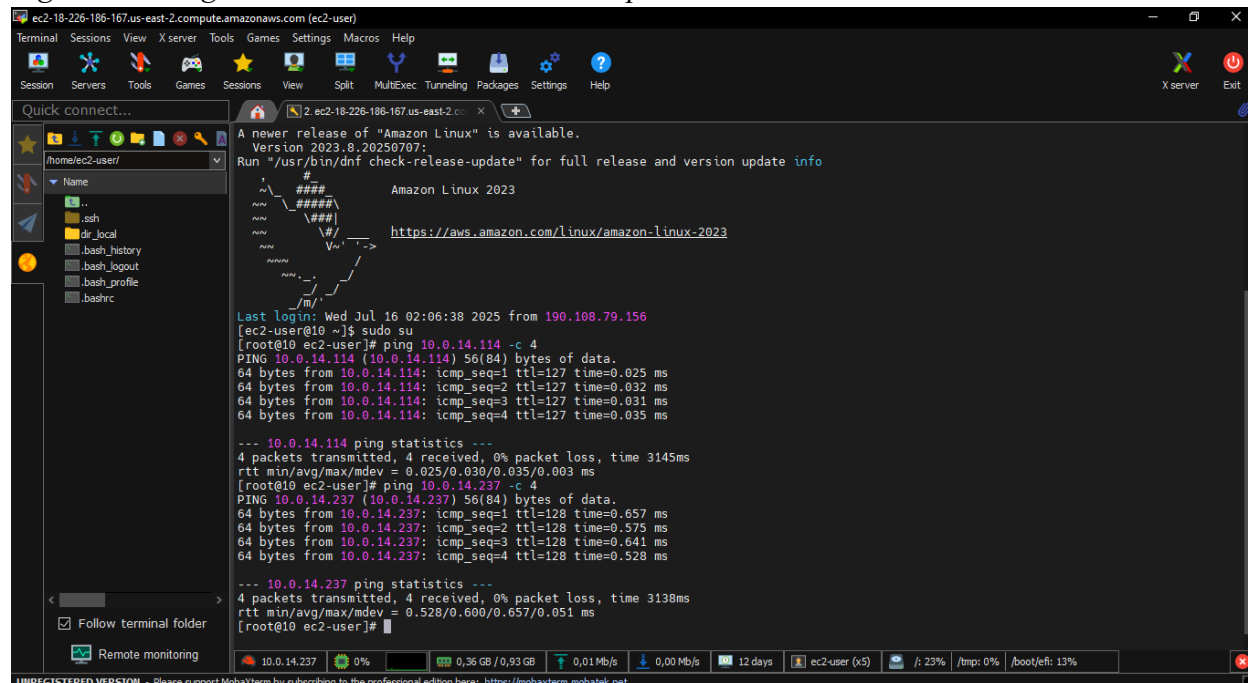
Figura 22: Ping desde el servidor Windows a la IP privada del servidor Linux



Desde la instancia Linux hacer *ping* a la IP privada de la instancia Windows:

Para hacer ping al servidor de Windows debemos ingresar al Firewall de Windows server y añadir una nueva regla de entrada que me permita recibir los Ping de otros servidores por medio del protocolo ICMP.

Figura 23: Ping desde el servidor Linux a la IP privada del servidor Windows



```
ec2-18-226-186-167.us-east-2.compute.amazonaws.com (ec2-user)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
/home/ec2-user/
Name
..
.ssh
dir_local
.bash_history
.bash_logout
.bash_profile
.bashrc
A newer release of "Amazon Linux" is available.
Version 2023.8.20250707:
Run "/usr/bin/dnf check-release-update" for full release and version update info
#
#####
Amazon Linux 2023
#####
|
\#|
V#|
#####
|/m/|
Last login: Wed Jul 16 02:06:38 2025 from 190.168.79.156
[ec2-user@10 ~]$ sudo su
[root@10 ec2-user]# ping 10.0.14.114 -c 4
PING 10.0.14.114 (10.0.14.114) 56(84) bytes of data:
64 bytes from 10.0.14.114: icmp_seq=1 ttl=127 time=0.025 ms
64 bytes from 10.0.14.114: icmp_seq=2 ttl=127 time=0.032 ms
64 bytes from 10.0.14.114: icmp_seq=3 ttl=127 time=0.031 ms
64 bytes from 10.0.14.114: icmp_seq=4 ttl=127 time=0.035 ms

--- 10.0.14.114 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3145ms
rtt min/avg/max/mdev = 0.025/0.030/0.035/0.003 ms
[root@10 ec2-user]# ping 10.0.14.237 -c 4
PING 10.0.14.237 (10.0.14.237) 56(84) bytes of data:
64 bytes from 10.0.14.237: icmp_seq=1 ttl=128 time=0.657 ms
64 bytes from 10.0.14.237: icmp_seq=2 ttl=128 time=0.575 ms
64 bytes from 10.0.14.237: icmp_seq=3 ttl=128 time=0.641 ms
64 bytes from 10.0.14.237: icmp_seq=4 ttl=128 time=0.528 ms

--- 10.0.14.237 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3138ms
rtt min/avg/max/mdev = 0.528/0.600/0.657/0.051 ms
[root@10 ec2-user]#
```

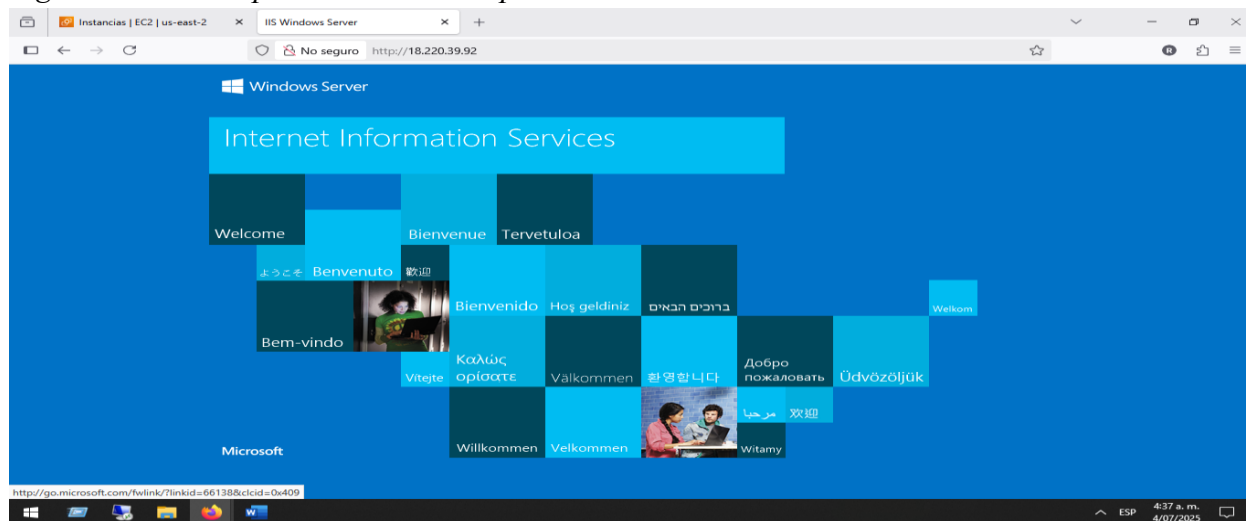
Hay necesidad de habilitar ICMP en los Grupos de Seguridad para permitir ping:

Para hacer ping desde un servidor de AWS a otro servidor utilizando solo la IP privada, se deben de tener en cuenta algunos requisitos como que los dos servidores deben estar en el misma VPC, deben de tener rutas y reglas de seguridad correctamente configuradas. El Grupo de Seguridad del servidor de destino debe permitir tráfico ICMP entrante de otro servidor.

6. Validación de acceso web

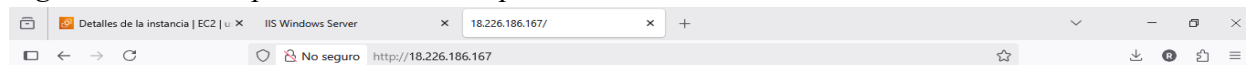
Acceder desde el navegador local al sitio web de la instancia Windows (<http://18.220.39.92/>):

Figura 24: Acceso por medio de la IP pública 18.220.39.92 de la Instancia de Windows



Acceder desde el navegador local al sitio web de la instancia Linux (<http://18.226.186.167/>).

Figura 25: Acceso por medio de la IP pública 18.226.186.167 de la Instancia de Linux



It works!



Implementación de Arquitectura en AWS con Balanceador de Carga y Contenedores

El startup "FastFoodGo"

Bienvenidos al proyecto de implementación. En esta tarea, ustedes asumirán el rol de arquitectos en la nube para un startup llamado **FastFoodGo**, una plataforma innovadora que conecta a restaurantes con clientes mediante entregas rápidas. La empresa ha experimentado un rápido crecimiento en los últimos meses y ahora necesita escalar su infraestructura tecnológica para manejar una mayor demanda, garantizar la disponibilidad y mejorar los tiempos de respuesta.

El CTO de **FastFoodGo** ha diseñado una arquitectura preliminar y necesita de su ayuda para implementarla en **Amazon Web Services (AWS)**. La solución debe ser altamente disponible, escalable y estar diseñada para manejar una gran cantidad de tráfico de manera eficiente.

1. Balanceador de Carga:

Se configuro una Application Load Balancer (ALB) para distribuir el tráfico entrante a múltiples instancias EC2.

Figura 26: Creación de Balanceador de carga para Docker

The screenshot displays the AWS Management Console interface for configuring an Elastic Load Balancing (ALB). The main view shows a list of load balancers under the heading "Balanceadores de carga (1/2)". Two load balancers are listed:

Nombre	Nombre de DNS	Estado	ID de VPC	Zonas de disponibili...	Tipo	Fecha cr...
LoadBalancerLinux1	LoadBalancerLinux1-159217898...	Activo	vpc-01da29988abb4754d	2 Zonas de disponibilidad	application	17 de julio ...
LoadBalancerLinuxDocker	LoadBalancerLinuxDocker-32135...	Activo	vpc-01da29988abb4754d	2 Zonas de disponibilidad	application	20 de julio ...

Below the list, the configuration details for the selected "Equilibrador de carga: LoadBalancerLinuxDocker" are shown. The "Reenviar al grupo de destino" (Forward to target group) section is expanded, showing the following configuration:

- Protocol: HTTP:80
- Acción predeterminada: Reenviar al grupo de destino
- Reglas: 1 regla
- ARN: ARN
- Política de seguridad: No aplicable
- Certificado SSL/TLS predet.: No aplicable

The target group configuration includes:

- targetGroupDocker: 1 (100%)
- Permanencia del grupo de destino: Desactivada

Figura 27: Detalles de balanceador de carga para Docker

The screenshot displays the AWS Management Console interface for the 'LoadBalancerLinuxDocker' resource. The page is titled 'LoadBalancerLinuxDocker' and includes a navigation menu with options like 'Agentes de escucha y reglas', 'Mapeo de red', 'Mapa de recursos', 'Seguridad', 'Monitorización', 'Integraciones', 'Atributos', and 'Capacidad'. The main content area is divided into several sections:

- Detalles:**
 - Tipo de equilibrador de carga:** Aplicación
 - Esquema:** Internet-facing
 - Estado:** Activo (with a green checkmark icon)
 - Zona hospedada:** Z3AADJGX6KTTL2
 - VPC:** vpc-01da29988abb4754d
 - Zonas de disponibilidad:**
 - subnet-08e4ef4629211b671 (use2-az2) us-east-2b
 - subnet-05825adf560a0c138 (use2-az1) us-east-2a
 - Tipo de dirección IP del equilibrador de carga:** IPv4
 - Fecha creada:** 20 de julio de 2025, 13:09 (UTC-05:00)
- ARN del equilibrador de carga:** arn:aws:elasticloadbalancing:us-east-2:715411138735:loadbalancer/app/LoadBalancerLinuxDocker/472ac2a0699454e3
- Nombre de DNS info:** LoadBalancerLinuxDocker-321353602.us-east-2.elb.amazonaws.com (Registro A)

Figura 28: Detalles del target Group para Docker

The screenshot displays the AWS Management Console interface for the 'targetGroupDocker' resource. The page is titled 'targetGroupDocker' and includes a navigation menu with options like 'Destinos', 'Monitorización', 'Comprobaciones de estado', 'Atributos', and 'Etiquetas'. The main content area is divided into several sections:

- Detalles:**
 - Tipo de destino:** Instancia
 - Tipo de dirección IP:** IPv4
 - Protocolo : Puerto:** HTTP: 80
 - Versión del protocolo:** HTTP1
 - Balaceador de carga:** LoadBalancerLinuxDocker
 - VPC:** vpc-01da29988abb4754d
- Destinos totales:** 2 (with a green checkmark icon)
 - En buen estado: 2
 - En mal estado: 0 (with a red X icon)
 - Sin utilizar: 0 (with a grey circle icon)
 - Inicial: 0 (with a grey circle icon)
 - Vaciado: 0 (with a grey circle icon)
 - 0 Anómalo
- Distribución de destinos por zona de disponibilidad (AZ):**

Seleccione los valores de esta tabla para ver los filtros correspondientes aplicados a la tabla Destinos registrados que aparece a continuación.

Figura 29: Destinos registrados en el target group para Docker

Distribución de destinos por zona de disponibilidad (AZ)
 Seleccione los valores de esta tabla para ver los filtros correspondientes aplicados a la tabla Destinos registrados que aparece a continuación.

Destinos | Monitorización | Comprobaciones de estado | Atributos | Etiquetas

Destinos registrados (2) [Info](#) [Mitigación de anomalías: No aplicable](#) [Anular el registro](#) [Registrar destinos](#)

Los grupos de destinos enrutan las solicitudes a destinos individuales registrados mediante el protocolo y el número de puerto que especifique. Las comprobaciones de estado se realizan en todos los destinos registrados de acuerdo con la configuración de comprobación de estado del grupo de destinos. La detección de anomalías se aplica automáticamente a los grupos de destinos de HTTP/HTTPS con al menos 3 destinos en buen estado.

Buscar destinos

<input type="checkbox"/>	ID de instancia	Nombre	Puerto	Zona	Estado	Detalles d...	Sustitu...	Detalle...	Hora d...
<input type="checkbox"/>	i-059750397b2d5021a	LinuxDockerNginx	80	us-east-2b (use2-az2)	Healthy	-	No override	No overri...	20 de julio...
<input type="checkbox"/>	i-008c90248a2d14b7b	AmazonLinux2023	80	us-east-2a (use2-az1)	Healthy	-	No override	No overri...	4 de julio ...

CloudShell | Comentarios | © 2025, Amazon Web Services, Inc. o sus filiales. | Privacidad | Términos | Preferencias de cookies

Figura 30: Aplicación que se visualiza por medio de DNS del balanceador de carga

Junky
SINCE 1998

CHEDDAR JUNKY BURGOS DE PELADAS

El chef Wesley Genovart fabrica esta hamburguesa de inspiración abierta Shake Shackins con dos hamburguetas delgadas, tocino de corte grueso, kimchi y una salsa casera picante.

[INFO Y NUTRICIÓN](#)

http://loadbalancerlinuxdocker-321353602.us-east-2.elb.amazonaws.com

2. Instancias EC2:

Se realizó la implementación de dos instancias EC2 en una configuración multizona para garantizar una alta disponibilidad.

Figura 31: Instancias EC2 multizona para Linux

The screenshot displays the AWS Management Console interface for EC2 instances. The top navigation bar shows the AWS logo, search bar, and user profile 'ronryeyes'. The main content area is titled 'Instancias (6) Información' and includes a search bar and a table of instances. The table columns are: Name, ID de la instancia, Estado de la instancia, Tipo de instancia, Comprobación de estado, Estado de la alarma, Zona de disponibilidad, and DNS de IPv4 pública. All instances are in the 'En ejecución' state and are distributed across multiple availability zones (us-east-2a and us-east-2b). Below the table, there is a section for 'Seleccione una instancia' and a footer with 'CloudShell' and 'Comentarios'.

Name	ID de la instancia	Estado de la instancia	Tipo de instancia	Comprobación de estado	Estado de la alarma	Zona de disponibilidad	DNS de IPv4 pública
WindowsServer2016	i-0b0d36ef6f31c1af7	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-2a	ec2-18-220-39-92.us-east-2.compute.amazonaws.com
LinuxDockerNginx	i-059750397b2d3021a	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-2b	ec2-18-117-96-197.us-east-2.compute.amazonaws.com
ECS Instance - Cluster_FastFoodGo	i-0b45cc86765c6b936	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-2b	ec2-18-222-202-215.us-east-2.compute.amazonaws.com
AmazonLinux2023	i-008c90248a2d14b7b	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-2a	ec2-18-226-186-167.us-east-2.compute.amazonaws.com
	i-070e2863383f08458	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-2b	-
	i-055a5fcb7563e8ebe	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-2a	-

Figura 32: Imagen AMI creada para Linux con Docker

The screenshot displays the AWS Management Console interface for Amazon Machine Images (AMI). The top navigation bar shows the AWS logo, search bar, and user profile 'ronryeyes'. The main content area is titled 'Imágenes de Amazon Machine Image (AMI) (1/3) Información' and includes a search bar and a table of AMIs. The table columns are: Name, Nombre de AMI, ID de AMI, Origen, Propietario, Visibilidad, Estado, and Fecha de creación. The AMI 'LinuxDocker_Nginx_V1' is selected and highlighted. Below the table, there is a section for 'ID de la AMI: ami-0a3e15dfbeec13460' and a footer with 'CloudShell' and 'Comentarios'.

Name	Nombre de AMI	ID de AMI	Origen	Propietario	Visibilidad	Estado	Fecha de creación
	LinuxApacheV2	ami-0a73387f404c17f86	715411138735/LinuxApacheV2	715411138735	Privado	Disponible	2025/07/17
	LinuxDockerV1	ami-02b953bb02ce4a4ba	715411138735/LinuxDockerV1	715411138735	Privado	Disponible	2025/07/20
<input checked="" type="checkbox"/>	LinuxDocker_Nginx_V1	ami-0a3e15dfbeec13460	715411138735/LinuxDocker_Nginx_V1	715411138735	Privado	Disponible	2025/07/20

ID de la AMI: ami-0a3e15dfbeec13460

ID de AMI ami-0a3e15dfbeec13460	Tipo de imagen machine	Detalles de la plataforma Linux/UNIX	Tipo de dispositivo raíz EBS
Nombre de AMI LinuxDocker_Nginx_V1	ID de cuenta del propietario 715411138735	Arquitectura x86_64	Operación de uso RunInstances

Figura 35: Configuración de nginx

```

Complete!
[root@10 ec2-user]# systemctl status nginx
Unit nginx.service could not be found.
[root@10 ec2-user]# systemctl status nginx
○ nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; preset: disabled)
   Active: inactive (dead)
[root@10 ec2-user]# systemctl start nginx
[root@10 ec2-user]# systemctl enable nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
[root@10 ec2-user]# systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
   Active: active (running) since Sun 2025-07-20 16:27:45 UTC; 17s ago
     Main PID: 2043053 (nginx)
       Tasks: 2 (limit: 1111)
         Memory: 2.5M
           CPU: 61ms
      CGroup: /system.slice/nginx.service
              └─2043053 "nginx: master process /usr/sbin/nginx"
                 └─2043056 "nginx: worker process"

Jul 20 16:27:44 10.0.14.237 systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Jul 20 16:27:45 10.0.14.237 nginx[2043053]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Jul 20 16:27:45 10.0.14.237 nginx[2043053]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Jul 20 16:27:45 10.0.14.237 systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[root@10 ec2-user]# cd /etc/nginx/
[root@10 nginx]# ls
conf.d      fastcgi.conf.default  koi-utf      mime.types.default  scgi_params      uwsgi_params.default
default.d  fastcgi_params        koi-win      mime.types           scgi_params.default  uwsgi_params
fastcgi.conf  fastcgi_params.default  mime.types  nginx.conf.default  uwsgi_params
[root@10 nginx]# cp nginx.conf nginx.conf.BK
[root@10 nginx]# nano nginx.conf
[root@10 nginx]# systemctl restart nginx

```

Figura 36: Modificación de nginx.conf para crear el proxy reverso

```

GNU nano 8.3          nginx.conf          Modified
events {}

http {
    upstream fastfoodgo {
        server localhost:85;
        server localhost:90;
    }

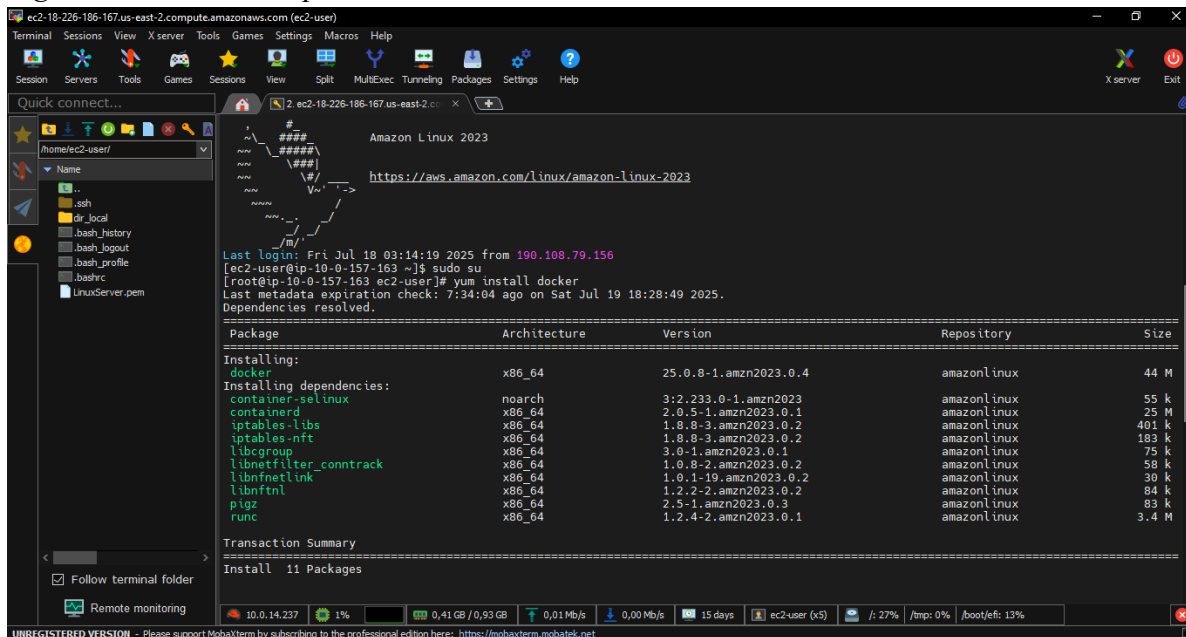
    server {
        listen 80;
        server_name nginx;
        location / {
            proxy_pass http://fastfoodgo;
        }
    }
}

```

4. Servicio de Docker de forma manual

Se realizo la implementación del servicio de Docker de forma manual con una aplicación de prueba.

Figura 37: Instalación por SSH de Docker



```

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

Last login: Fri Jul 19 03:14:19 2025 from 190.108.79.156
[ec2-user@ip-10-0-157-163 ~]$ sudo su
[root@ip-10-0-157-163 ec2-user]# yum install docker
Last metadata expiration check: 7:34:04 ago on Sat Jul 19 18:28:49 2025.
Dependencies resolved.

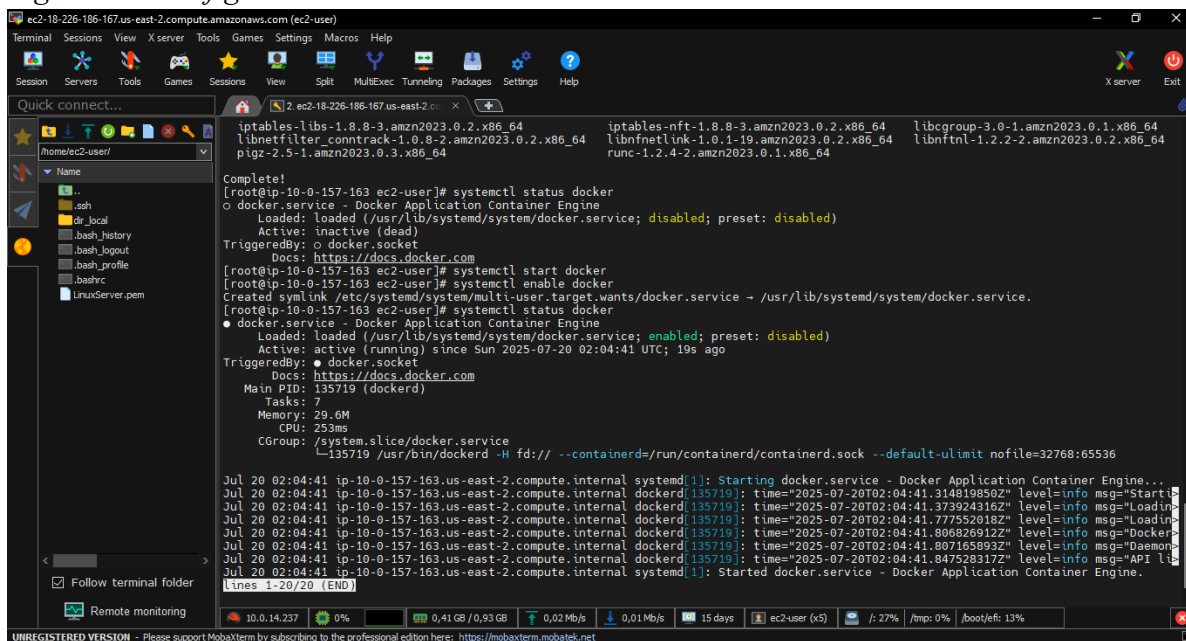
Package Architecture Version Repository Size
-----
Installing:
docker x86_64 25.0.8-1.amzn2023.0.4 amazonlinux 44 M
Installing dependencies:
container-selinux x86_64 3:2.233.0-1.amzn2023 amazonlinux 55 k
containerd x86_64 2.0.5-1.amzn2023.0.1 amazonlinux 25 M
iptables-libs x86_64 1.8.8-3.amzn2023.0.2 amazonlinux 401 k
iptables-nft x86_64 1.8.8-3.amzn2023.0.2 amazonlinux 183 k
libcgroup x86_64 3.0-1.amzn2023.0.1 amazonlinux 75 k
libnetfilter_conntrack x86_64 1.0.8-2.amzn2023.0.2 amazonlinux 58 k
libnftnl x86_64 1.0.1-19.amzn2023.0.2 amazonlinux 30 k
libnftnl x86_64 1.2.2-2.amzn2023.0.2 amazonlinux 84 k
pigz x86_64 2.5-1.amzn2023.0.3 amazonlinux 83 k
runc x86_64 1.2.4-2.amzn2023.0.1 amazonlinux 3.4 M

Transaction Summary
Install 11 Packages

Follow terminal folder
Remote monitoring
10.0.14.237 1% 0.41 GB / 0.93 GB 0.01 Mb/s 0.00 Mb/s 15 days ec2-user (x5) /: 27% /tmp: 0% /boot/efi: 13%
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

```

Figura 38: Configuración de Docker



```

iptables-libs-1.8.8-3.amzn2023.0.2.x86_64 iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 libcgroup-3.0-1.amzn2023.0.1.x86_64
libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 libnftnl-1.0.1-19.amzn2023.0.2.x86_64 libnftnl-1.2.2-2.amzn2023.0.2.x86_64
pigz-2.5-1.amzn2023.0.3.x86_64 runc-1.2.4-2.amzn2023.0.1.x86_64

Complete!
[root@ip-10-0-157-163 ec2-user]# systemctl status docker
○ docker.service - Docker Application Container Engine
Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; preset: disabled)
Active: inactive (dead)
TriggeredBy: ○ docker.socket
Docs: https://docs.docker.com

[root@ip-10-0-157-163 ec2-user]# systemctl start docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[root@ip-10-0-157-163 ec2-user]# systemctl enable docker
● docker.service - Docker Application Container Engine
Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: disabled)
Active: active (running) since Sun 2025-07-20 02:04:41 UTC; 19s ago
TriggeredBy: ● docker.socket
Docs: https://docs.docker.com
Main PID: 135719 (dockerd)
Tasks: 7
Memory: 29.6M
CPU: 253ms
CGroup: /system.slice/docker.service
└─135719 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

Jul 20 02:04:41 ip-10-0-157-163.us-east-2.compute.internal systemd[1]: Starting docker.service - Docker Application Container Engine...
Jul 20 02:04:41 ip-10-0-157-163.us-east-2.compute.internal dockerd[135719]: time="2025-07-20T02:04:41.314819850Z" level=info msg="Startin
Jul 20 02:04:41 ip-10-0-157-163.us-east-2.compute.internal dockerd[135719]: time="2025-07-20T02:04:41.777552618Z" level=info msg="Loadin
Jul 20 02:04:41 ip-10-0-157-163.us-east-2.compute.internal dockerd[135719]: time="2025-07-20T02:04:41.886826911Z" level=info msg="Docker
Jul 20 02:04:41 ip-10-0-157-163.us-east-2.compute.internal dockerd[135719]: time="2025-07-20T02:04:41.807165893Z" level=info msg="Daemon
Jul 20 02:04:41 ip-10-0-157-163.us-east-2.compute.internal dockerd[135719]: time="2025-07-20T02:04:41.847528317Z" level=info msg="API li
Jul 20 02:04:41 ip-10-0-157-163.us-east-2.compute.internal systemd[1]: Started docker.service - Docker Application Container Engine.
lines 1-20/20 (END)
10.0.14.237 0% 0.41 GB / 0.93 GB 0.02 Mb/s 0.01 Mb/s 15 days ec2-user (x5) /: 27% /tmp: 0% /boot/efi: 13%
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

```

Figura 39: Instalación de una aplicación de prueba en Docker

```

[ec2-user]# cd app/
[ec2-user]# wget https://plantillashtmlgratis.com/wp-content/themes/helium-child/descargas/page230/burger.zip
--2025-07-20 05:03:37-- https://plantillashtmlgratis.com/wp-content/themes/helium-child/descargas/page230/burger.zip
Resolving plantillashtmlgratis.com (plantillashtmlgratis.com)... 51.77.201.228
Connecting to plantillashtmlgratis.com (plantillashtmlgratis.com)|51.77.201.228|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 674928 (659K) [application/zip]
Saving to: 'burger.zip'

burger.zip 100%[=====] 659.11K 421KB/s in 1.6s

2025-07-20 05:03:39 (421 KB/s) - 'burger.zip' saved [674928/674928]

[ec2-user]# file burger.zip
burger.zip: Zip archive data, at least v1.0 to extract
[ec2-user]# unzip burger.zip
Archive: burger.zip
  creating: burger-free-html5-landing-page-template/
  creating: burger-free-html5-landing-page-template/css/
  inflating: burger-free-html5-landing-page-template/css/animate.css
  creating: _____MACOSX/
  creating: _____MACOSX/burger-free-html5-landing-page-template/
  creating: _____MACOSX/burger-free-html5-landing-page-template/css/
  inflating: _____MACOSX/burger-free-html5-landing-page-template/css/_animate.css
  inflating: burger-free-html5-landing-page-template/css/bootsnav.css
  inflating: _____MACOSX/burger-free-html5-landing-page-template/css/_bootsnav.css
  inflating: burger-free-html5-landing-page-template/css/color.css
  inflating: _____MACOSX/burger-free-html5-landing-page-template/css/_color.css
  inflating: burger-free-html5-landing-page-template/css/custom.css
  inflating: _____MACOSX/burger-free-html5-landing-page-template/css/_custom.css
  inflating: burger-free-html5-landing-page-template/css/flaticon.css
  inflating: _____MACOSX/burger-free-html5-landing-page-template/css/_flaticon.css
  inflating: burger-free-html5-landing-page-template/css/_overwrite.css
  inflating: _____MACOSX/burger-free-html5-landing-page-template/_css
  creating: burger-free-html5-landing-page-template/fonts/
  inflating: burger-free-html5-landing-page-template/fonts/Flaticon.eot
  
```

Figura 40: Instalación de otra aplicación de prueba en Docker

```

CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                               NAMES
13e2c8a48     httpd     "httpd-foreground"      45 minutes ago Up 45 minutes 0.0.0.0:90->80/tcp, :::90->80/tcp  app3
1263457436d     httpd     "httpd-foreground"      3 hours ago   Exited (0)   33 seconds ago

[ec2-user]# mkdir app2
[ec2-user]# cd app2/
[ec2-user]# wget https://plantillashtmlgratis.com/wp-content/themes/helium-child/descargas/page162/steak-house.zip
--2025-07-20 06:20:55-- https://plantillashtmlgratis.com/wp-content/themes/helium-child/descargas/page162/steak-house.zip
Resolving plantillashtmlgratis.com (plantillashtmlgratis.com)... 51.77.201.228
Connecting to plantillashtmlgratis.com (plantillashtmlgratis.com)|51.77.201.228|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 420815 (411K) [application/zip]
Saving to: 'steak-house.zip'

steak-house.zip 100%[=====] 410.95K 412KB/s in 1.0s

2025-07-20 06:20:57 (412 KB/s) - 'steak-house.zip' saved [420815/420815]

[ec2-user]# file steak-house.zip
steak-house.zip: Zip archive data, at least v2.0 to extract
[ec2-user]# unzip steak-house.zip
Archive: steak-house.zip
  creating: steak-house/
  inflating: steak-house/about.html
  inflating: steak-house/beer.html
  inflating: steak-house/blog.html
  inflating: steak-house/contact.html
  creating: steak-house/css/
  inflating: steak-house/css/ie6.css
  inflating: steak-house/css/ie7.css
  inflating: steak-house/css/style.css
  inflating: steak-house/food.html
  creating: steak-house/images/
  inflating: steak-house/images/arrow.gif
  inflating: steak-house/images/barbeque.jpg
  inflating: steak-house/images/beer.jpg
  inflating: steak-house/images/bg-body-header.gif
  
```

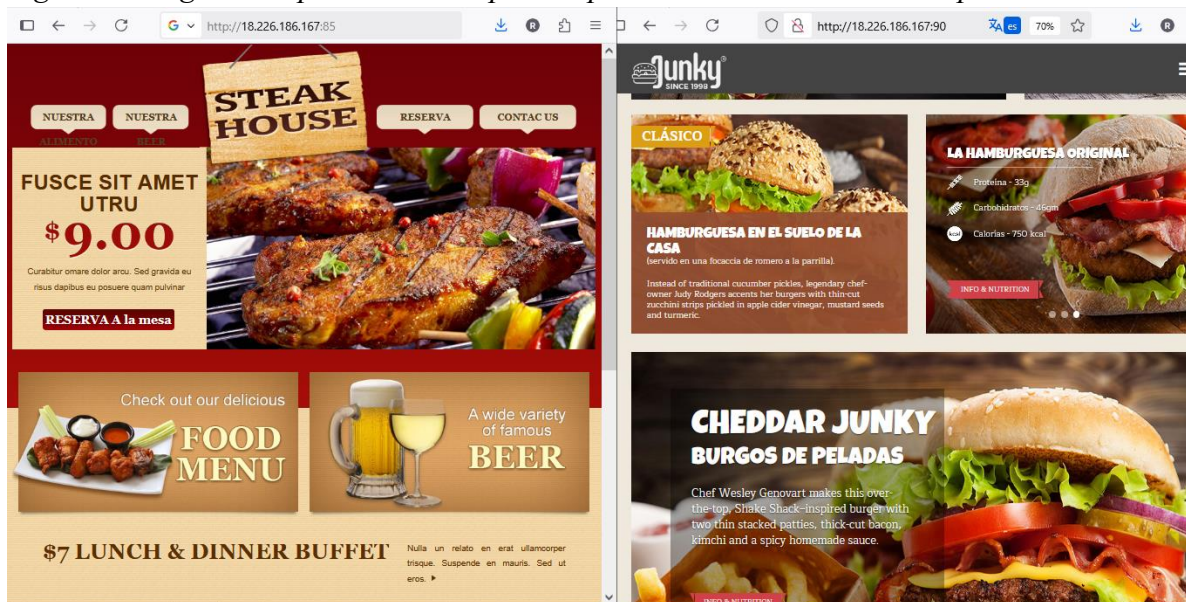
Figura 41: Configuración de los puertos para las aplicaciones de prueba en Docker

```

inflatng: steak-house/images/bg-menu2.jpg
inflatng: steak-house/images/bg-menu3.jpg
inflatng: steak-house/images/bg-menu4.jpg
inflatng: steak-house/images/bg-page.jpg
inflatng: steak-house/images/border-dashed.gif
inflatng: steak-house/images/border-dashed2.gif
inflatng: steak-house/images/bullet.gif
inflatng: steak-house/images/button.gif
inflatng: steak-house/images/button1.gif
inflatng: steak-house/images/food-menu.jpg
inflatng: steak-house/images/food.jpg
inflatng: steak-house/images/frame.jpg
inflatng: steak-house/images/header-bg2.gif
inflatng: steak-house/images/icon-facebook.gif
inflatng: steak-house/images/icon-twitter.gif
inflatng: steak-house/images/icon.gif
inflatng: steak-house/images/logo.jpg
inflatng: steak-house/images/logo2.jpg
inflatng: steak-house/images/menu.gif
inflatng: steak-house/images/nine-dollars.gif
inflatng: steak-house/images/waiter.jpg
inflatng: steak-house/index.html
inflatng: steak-house/menu.html
inflatng: steak-house/reservation.html
[root@10 app2]# cd steak-house/
[root@10 steak-house]# pwd
/home/ec2-user/app2/steak-house
[root@10 steak-house]# docker run -dit --name appfood -d --restart always -p 85:80 -v /home/ec2-user/app2/steak-house:/usr/local/apache2
/htdocs/ httpd
35a6d4fb13a2e112e86304851d718555800132c8e8c51e6005afb421097082bf
[root@10 steak-house]# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED        STATUS        PORTS                               NAMES
35a6d4fb13a2      httpd              "httpd-foreground"     20 seconds ago Up 19 seconds 0.0.0.0:85->80/tcp, :::85->80/tcp  appfood
37f13e2c8a48      httpd              "httpd-foreground"     57 minutes ago Up 57 minutes  0.0.0.0:90->80/tcp, :::90->80/tcp  appburger
12634c57436d      httpd              "httpd-foreground"     4 hours ago   Exited (0) 12 minutes ago
[root@10 steak-house]#

```

Figura 42: Ingreso a aplicaciones de prueba por medio de internet con el puerto



5. Auto escalado

Se realizo la configuración de políticas de Auto Scaling para aumentar o reducir las instancias EC2 según la carga.

Figura 43: Platilla de lanzamiento para el Auto Scaling

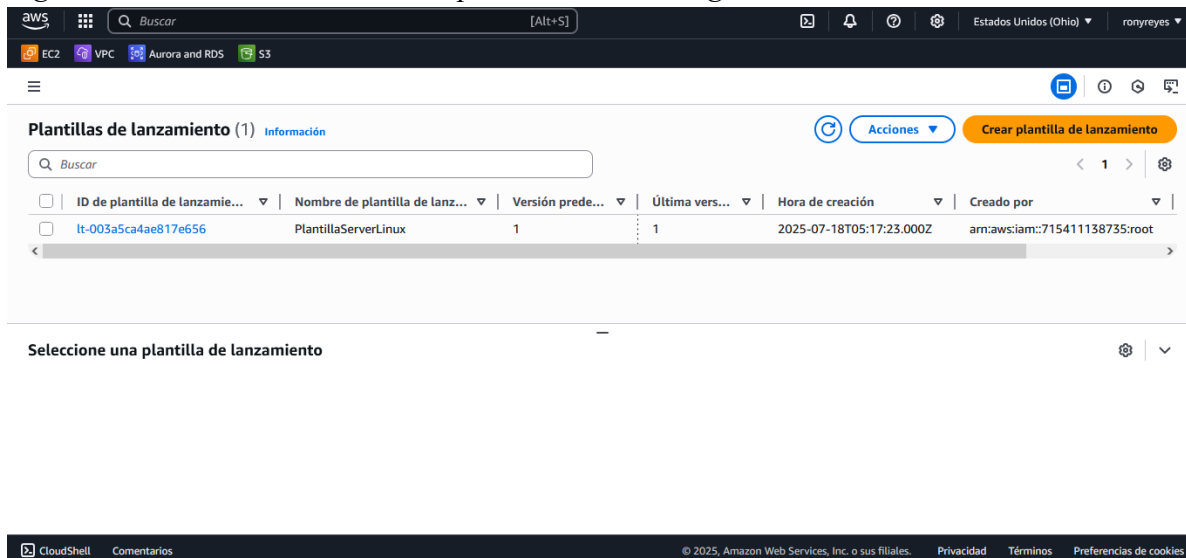


Figura 44: Detalles de plantilla de lanzamiento para Auto Scaling

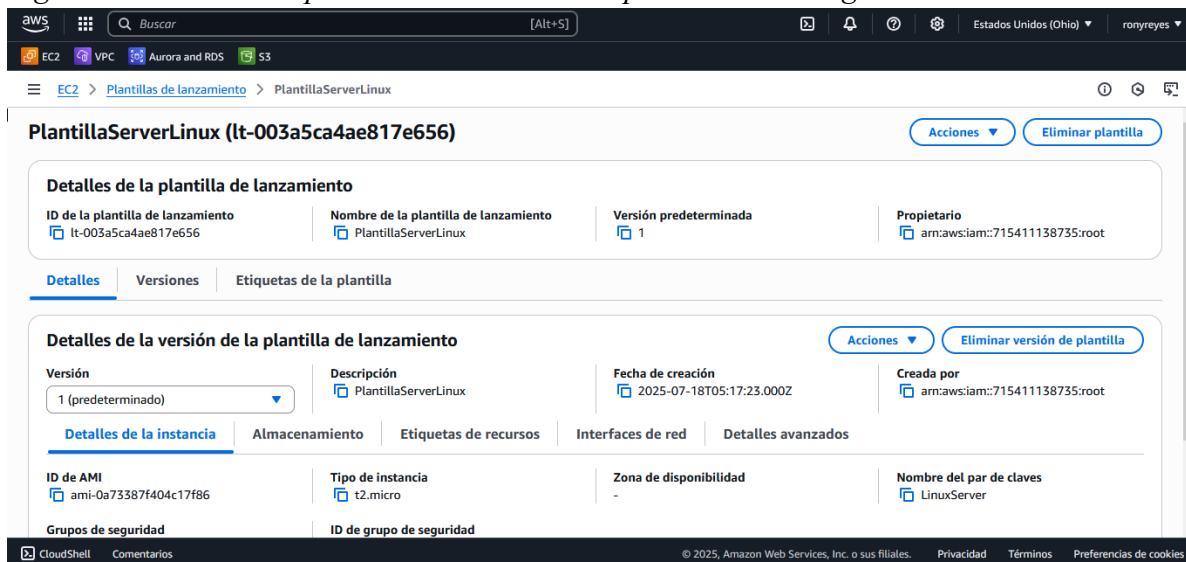


Figura 45: Creación del grupo de Auto Scaling

Grupos de Auto Scaling (1) [Info](#)

Última actualización hace menos de un minuto

Configuraciones de lanzamiento Plantillas de lanzamiento Acciones [Crear grupo de escalado automático](#)

Buscar sus grupos de Auto Scaling

<input type="checkbox"/>	Nombre	Plantilla de lanzamiento/config...	Instanc...	Estado	Capacidad ...	Mín	M...	Zonas de disponibilidad
<input type="checkbox"/>	AutoScalingSeminario	PlantillaServerLinux Versión Predetermin	2	-	2	1	5	2 Zonas de disponibilidad

0 Grupos de escalado automático seleccionados

CloudShell Comentarios © 2025, Amazon Web Services, Inc. o sus filiales. Privacidad Términos Preferencias de cookies

Figura 46: Detalles del grupo de Auto Scaling

AutoScalingSeminario

[Editar](#)

arn:aws:autoscaling:us-east-2:715411138735:autoScalingGroup:1d3ba88d-a7c6-4d58-a917-fd542c135bed:autoScalingGroupName/AutoScalingSeminario

Capacidad deseada 2	Límites de escalamiento (Mín. - Máx.) 1 - 5	Tipo de capacidad deseado Unidades (número de instancias)	Estado -
------------------------	--	--	-------------

Fecha de creación
Fri Jul 18 2025 00:35:49 GMT-0500 (hora estándar de Colombia)

[Detalles](#) | [Integraciones - nueva](#) | [Escalado automático](#) | [Administración de instancias](#) | [Actualización de instancias](#) | [Actividad](#) | [Monitoreo](#)

Plantilla de lanzamiento

[Editar](#)

Plantilla de lanzamiento lt-003a5ca4ae817e656 PlantillaServerLinux	ID de AMI ami-0a73387f404c17f86	Tipo de instancia t2.micro	Propietario arn:aws:iam::715411138735:root
--	--	-------------------------------	---

CloudShell Comentarios © 2025, Amazon Web Services, Inc. o sus filiales. Privacidad Términos Preferencias de cookies

Figura 47: Verificación de la creación de dos instancias EC2 por Auto Scaling

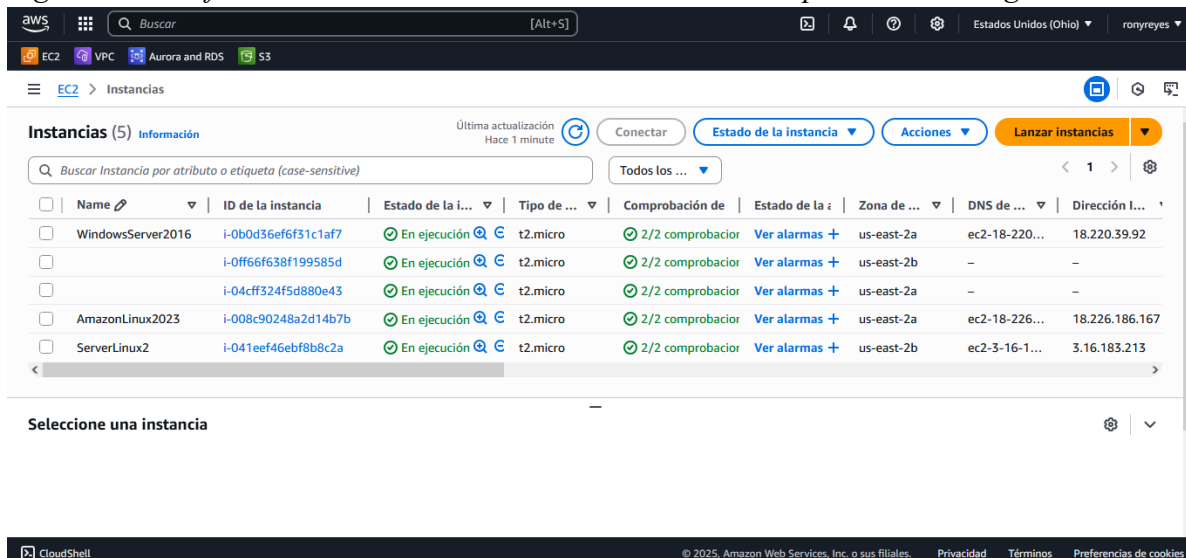


Figura 48: Historial de la actividad del Auto Scaling

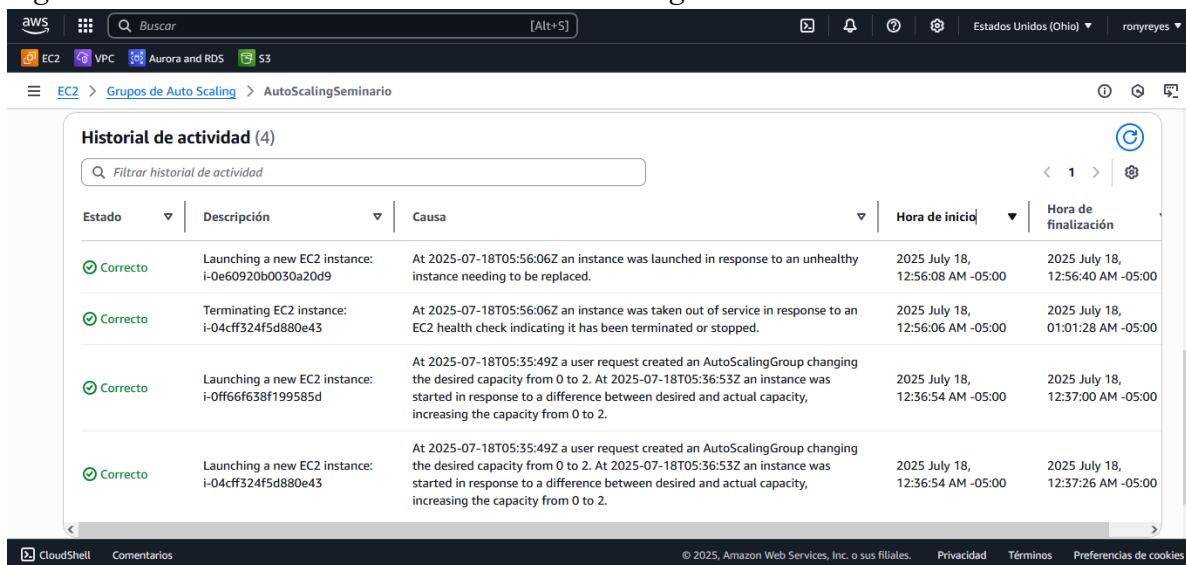


Figura 49: Verificación de la creación automática de una nueva instancia EC2

Se inició correctamente la detención de i-04cff324f5d880e43

Instancias (6) Información Última actualización Hace less than a minute Conectar Estado de la instancia Acciones Lanzar instancias

Buscar Instancia por atributo o etiqueta (case-sensitive) Todos los ...

Name	ID de la instancia	Estado de la instancia	Tipo de instancia	Comprobación de integridad	Estado de la instancia	Zona de disponibilidad	DNS de instancia	Dirección IP pública
	i-0ff66f638f199585d	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-2b	-	-
	i-04cff324f5d880e43	Detenida	t2.micro	-	Ver alarmas +	us-east-2a	-	-
	i-0e60920b0030a20d9	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-2a	-	-
AmazonLinux2023	i-008c90248a2d14b7b	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-2a	ec2-18-226...	18.226.186.167

Seleccione una instancia

Figura 50: Verificación de la eliminación automática de la instancia EC2 detenida

Se inició correctamente la detención de i-04cff324f5d880e43

Instancias (4) Información Última actualización Hace 2 minutos Conectar Estado de la instancia Acciones Lanzar instancias

Buscar Instancia por atributo o etiqueta (case-sensitive) Todos los ...

Name	ID de la instancia	Estado de la instancia	Tipo de instancia	Comprobación de integridad	Estado de la instancia	Zona de disponibilidad	DNS de instancia	Dirección IPv4 pública
WindowsServer2016	i-0b0d36ef6f31c1af7	En ejecución	t2.micro	2/2 comprob	Ver alarmas +	us-east-2a	ec2-18-220...	18.220.39.92
AmazonLinux2023	i-008c90248a2d14b7b	En ejecución	t2.micro	2/2 comprob	Ver alarmas +	us-east-2a	ec2-18-226...	18.226.186.167
	i-0ff66f638f199585d	En ejecución	t2.micro	2/2 comprob	Ver alarmas +	us-east-2b	-	-
	i-0e60920b0030a20d9	En ejecución	t2.micro	2/2 comprob	Ver alarmas +	us-east-2a	-	-

Seleccione una instancia

Conclusiones

La computación en la nube ha dejado de ser tendencia para convertirse en el estándar de arquitectura, AWS con servicios como instancias EC2 nos permiten desplegar servidores virtuales en minutos, con configuraciones adaptadas al tamaño de cada proyecto. Esta capacidad bajo demanda no solo evita costosas inversiones iniciales, sino que también ofrece una solución ágil para enfrentar los picos de tráfico inesperados, como ocurre en lanzamientos de apps o eventos masivos. Gracias a EC2, cualquier startup o negocio emergente puede acceder a tecnología de alto nivel sin comprometer su estabilidad financiera.

Se concluyo que no basta con lanzar instancias, para garantizar el rendimiento constante y la disponibilidad de aplicaciones es necesario usar el servicio de balanceadores de carga y proxies reversos configurados con Nginx, los cuales nos ofrecen una arquitectura más ágil y eficiente. Con esta combinación, podemos distribuir el tráfico de manera inteligente entre nuestras instancias EC2, garantizar la tolerancia a fallos y optimizar la experiencia del usuario final. Si a eso sumamos la implementación de contenedores con Docker gestionados manualmente, logramos un entorno más portátil, reproducible y fácil de mantener para nuestras aplicaciones, sin importar su complejidad.

Uno de los aspectos más valiosos que ofrece AWS es la automatización inteligente de la escalabilidad a través de los grupos de Auto Scaling. Desde mi experiencia, este servicio permite que nuestras aplicaciones crezcan o reduzcan su infraestructura de forma automática según la carga, sin intervención humana y sin afectar la operación. Esta característica no solo mejora la eficiencia operativa, sino que también cuida el presupuesto, evitando el sobredimensionamiento de recursos.

Referencias

Amazon Web Services. (s.f). *Referencia de comandos de la AWS CLI. EC2*. Sitio web: <https://docs.aws.amazon.com/cli/latest/reference/ec2/>

Amazon Web Services. (2025). *Guía del usuario: Amazon Elastic Compute Cloud*. https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/ec2-ug.pdf

Rodríguez, J. L. (2022, mayo). *Fundamentos de Cloud Computing. Cloud Computing*. https://www.academia.edu/91103206/Fundamentos_de_Cloud_Computing?auto=download