



TRABAJO DE GRADO
Opción Seminario-Diplomado.

QUALITY RUN

Corporación Universitaria Remington.

Ingeniería de Sistemas

Seminario Amazon Web Services AWS

David Alejandro Herrera Acevedo

Carlos David Betancur Salazar

Dairon Yesid Lizcano Machado

Juan Pablo Barrio López

Seminario-Diplomado.

2025

Tabla de Contenidos

RESUMEN	3
Palabras clave.....	4
MARCO CONCEPTUAL y CONTEXTUAL.....	4
DESARROLLO e IMPLEMENTACIÓN DEL APRENDIZAJE	7
ENTREGA 1	7
IMPLEMENTACIÓN de UNA RED en AWS CON SERVIDORES WINDOWS y LINUX ACCESIBLES DESDE INTERNET.	7
Objetivo General.....	7
Parte 1: Escrita (Documentación Técnica)	8
Parte 2: Práctica (Implementación y pruebas)	32
ENTREGA 2	48
IMPLEMENTACIÓN de ARQUITECTURA en AWS CON BALANCEADOR de CARGA y CONTENEDORES.....	49
Introducción: La startup.....	49
CONCLUSIONES	79
Referencias.....	80

RESUMEN

El presente trabajo de grado se enmarca dentro del Seminario-Diplomado sobre Amazon Web Services (AWS), con el objetivo de aplicar de manera práctica los conocimientos adquiridos en el diseño, implementación y gestión de infraestructuras en la nube. A través de dos entregas diferenciadas, se llevó a cabo una arquitectura que integra redes privadas virtuales, instancias de servidores (Windows y Linux), servidores web, balanceo de carga, contenedores Docker y escalamiento automático.

En la primera entrega, se diseñó y desplegó una red virtual en AWS que integraba dos instancias EC2: una con Windows Server 2016 y otra con Amazon Linux 2023. Ambas instancias fueron configuradas para ser accesibles desde Internet mediante protocolos seguros (RDP y SSH) y se instalaron servidores web (IIS en Windows y Apache en Linux). Este entorno sirvió como base para validar conceptos como conectividad interna mediante ICMP, configuración de VPC, subredes públicas, Internet Gateway y reglas de seguridad.

En la segunda fase, se abordó la necesidad de escalabilidad y alta disponibilidad. Se incorporaron contenedores Docker en ambas instancias, dentro de los cuales se ejecutaba una aplicación web simple. Posteriormente, se implementó un servidor NGINX como proxy reverso y se configuró un Application Load Balancer (ALB) para distribuir el tráfico entre las instancias. Finalmente, se añadió un grupo de Auto Scaling con políticas basadas en el uso de CPU, permitiendo que el sistema responda dinámicamente a la carga de trabajo.

Este proyecto no solo permitió afianzar conocimientos técnicos sobre servicios de AWS, sino también comprender la importancia de la automatización, la redundancia y la disponibilidad en sistemas modernos. Además, representó un reto práctico de integración de múltiples servicios, confirmando que la nube ofrece soluciones eficientes y escalables para empresas de cualquier tamaño.

La experiencia obtenida aporta competencias fundamentales en infraestructura como código, seguridad de redes, administración de sistemas y arquitectura de aplicaciones en la nube, preparando al estudiante para desafíos reales del entorno laboral contemporáneo.

Palabras clave

- Amazon Web Services (AWS)
- Infraestructura en la nube
- EC2
- Auto Scaling
- Balanceador de carga (ALB)
- Instancias
- Volúmenes
- RDP y SSH
- Subredes y grupos
- Grupos y plantillas

MARCO CONCEPTUAL y CONTEXTUAL

Este apartado está pensado para cualquier persona que quiera leer el trabajo, incluso si no tiene conocimientos previos sobre computación en la nube. El objetivo es introducir de manera clara los conceptos fundamentales que se utilizan a lo largo del documento, para que sea comprensible y accesible desde el inicio.

La computación en la nube es un modelo que permite acceder a servicios informáticos como almacenamiento, procesamiento o redes a través de Internet. En lugar de depender de servidores físicos propios, las empresas pueden utilizar infraestructura proporcionada por plataformas como Amazon Web Services (AWS), pagando solo por lo que usan. Esta modalidad se adapta fácilmente a diferentes necesidades de escala y presupuesto.

Uno de los servicios más importantes dentro de AWS es Amazon EC2 (Elastic Compute Cloud). EC2 permite lanzar servidores virtuales, conocidos como instancias, que funcionan como si fueran computadoras físicas. Estas instancias pueden tener diferentes sistemas operativos, como Windows Server o Linux, y se pueden configurar para cumplir funciones específicas, por ejemplo, alojar un sitio web.

Otro componente clave es la VPC (Virtual Private Cloud), que permite crear una red privada dentro de AWS. En esta red se definen subredes, reglas de acceso y conexiones a Internet mediante elementos como el Internet Gateway. La seguridad se controla a través de los grupos de seguridad, que son como cortafuegos que permiten o bloquean ciertos tipos de tráfico.

Además, este trabajo hace uso de un Balanceador de Carga (Application Load Balancer), que distribuye el tráfico entrante entre varias instancias EC2. Esto evita que una sola instancia se sobrecargue, mejorando así la disponibilidad del sistema. También se aplicaron conceptos de Autoescalado, que permiten aumentar o disminuir automáticamente la cantidad de servidores según la carga de trabajo.

Por último, se implementaron contenedores Docker, que permiten ejecutar aplicaciones de manera rápida y eficiente dentro de las instancias. Los contenedores se gestionaron usando servidores web como Apache y Nginx, siendo este último también utilizado como proxy reverso.

Este conjunto de tecnologías y conceptos fue aplicado en el contexto simulado de un startup tecnológico llamada “Quality Run”, la cual requería una infraestructura flexible, escalable y segura para atender su crecimiento. El trabajo busca demostrar cómo AWS permite responder a estas necesidades mediante herramientas prácticas y accesibles.

DESARROLLO e IMPLEMENTACIÓN DEL APRENDIZAJE

La transformación digital ha generado la necesidad de adoptar arquitecturas flexibles y resilientes en la gestión de servicios tecnológicos. Con el auge del cómputo en la nube, plataformas como Amazon Web Services se han convertido en una opción esencial para empresas que requieren escalar, automatizar y garantizar la disponibilidad de sus aplicaciones.

En este contexto, el presente proyecto de grado surge como una respuesta académica y técnica a las demandas de infraestructura moderna, aplicando los conocimientos adquiridos en el Seminario-Diplomado de AWS. A lo largo del proceso, se ejecutaron dos fases: primero, la configuración de una red básica con servidores () accesibles y funcionales; y segundo, la evolución hacia una arquitectura resiliente con balanceo de carga, contenedores Docker y políticas de autoescalado.

Este documento detalla la implementación paso a paso de dicha arquitectura, evidenciando la integración de conceptos de redes, seguridad, servicios web, virtualización y monitoreo de recursos. Además, se presentan capturas de pantalla y pruebas que validan el correcto funcionamiento de cada componente desplegado.

ENTREGA 1

IMPLEMENTACIÓN de UNA RED en AWS CON SERVIDORES WINDOWS y LINUX ACCESIBLES DESDE INTERNET.

Objetivo General

Diseñar, desplegar y documentar una red en AWS que incluya dos instancias EC2 (una Windows y una Linux), asegurando su accesibilidad pública, conectividad entre ellas y la instalación de un servidor web funcional en cada instancia.

Parte 1: Escrita (Documentación Técnica)

Diagrama de arquitectura

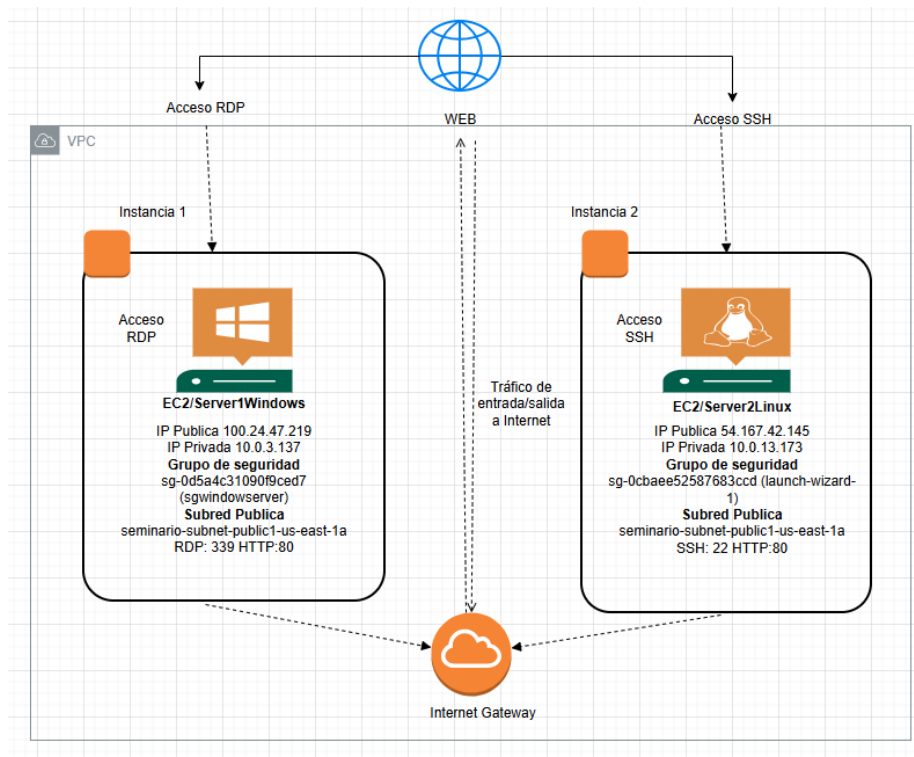


Ilustración 0.1

Descripción de la arquitectura

La arquitectura implementada en AWS consiste en una red virtual privada (VPC) que alberga dos instancias EC2: una con sistema operativo Windows y otra con sistema operativo Linux. Ambas instancias se encuentran en subredes públicas dentro de la zona de disponibilidad us-east-1a, lo cual permite su accesibilidad desde Internet a través de un Internet Gateway.

Cada instancia cuenta con una IP pública para acceso externo y una IP privada para comunicación interna. La instancia Windows permite acceso mediante el protocolo RDP (puerto 3389) y ofrece servicio web por HTTP (puerto 80), mientras que la instancia Linux permite conexión SSH (puerto 22) y también sirve contenido por HTTP.

La comunicación entre los usuarios e Internet con cada instancia está protegida mediante grupos de seguridad específicos, que controlan estrictamente los accesos permitidos.

Tipo de instancias usadas (Linux: Amazon Linux, Ubuntu, etc. | Windows: versión de Windows Server).

Instancia 1 – Server1Windows

Sistema operativo: *Windows Server 2016*

IP pública: 100.24.47.219

IP privada: 10.0.3.137

Subred: *seminario-subnet-public1-us-east-1a*

Grupo de seguridad: *sg-0d5a4c310909fced7* (sgwindowserver)

Puertos habilitados: RDP (3389), HTTP (80)

Instancia 2 – Server2Linux

Sistema operativo: *Amazon Linux 2023*

IP pública: 54.167.42.145

IP privada: 10.0.13.173

Subred: *seminario-subnet-public1-us-east-1a*

Grupo de seguridad: *sg-0cbaee52587683ccd* (launch-wizard-1)

Puertos habilitados: SSH (22), HTTP (80)

Justificación de las configuraciones de red (por ejemplo, uso de VPC, subredes públicas, Internet Gateway).

VPC (Virtual Private Cloud): Se utilizó una VPC personalizada para tener control completo sobre la red, incluyendo direccionamiento IP, subredes y conexiones externas, facilitando la administración y la seguridad.

Subredes públicas: Ambas instancias fueron ubicadas en subredes públicas para que puedan ser accedidas desde Internet. Esto es necesario para realizar pruebas remotas y verificar el funcionamiento de los servicios web.

Internet Gateway: Se configuró un Internet Gateway asociado a la VPC, permitiendo el acceso entrante y saliente a través de Internet. Es un componente esencial para garantizar la comunicación con recursos fuera de AWS.

Grupos de seguridad: Se configuraron reglas específicas para permitir únicamente los protocolos necesarios:

La instancia Windows permite acceso por RDP y HTTP.

La instancia Linux permite acceso por SSH y HTTP.

Esto refuerza la seguridad, limitando los vectores de ataque a lo estrictamente necesario.

Configuraciones realizadas

Pasos para crear las instancias EC2.

Para este proyecto se crearon dos instancias EC2 desde la consola de administración de AWS: una con Windows Server y otra con Amazon Linux.

Ingresar al servicio **EC2 > Instancias > Lanzar instancia.**

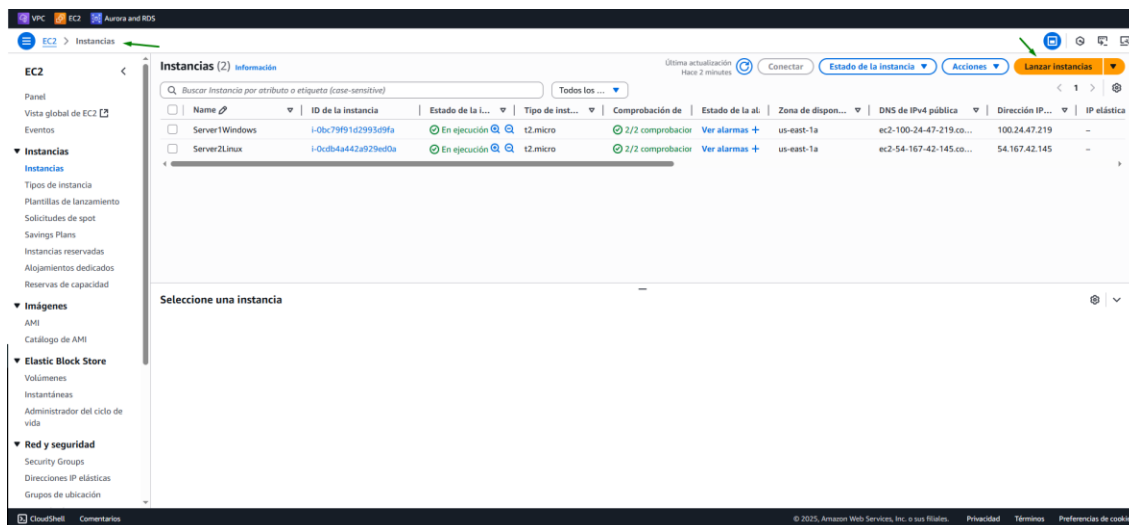


Ilustración 0.2

Asignar un nombre descriptivo a cada instancia:

En nuestro caso: EC2-Server1Windows / EC2-Server2Linux

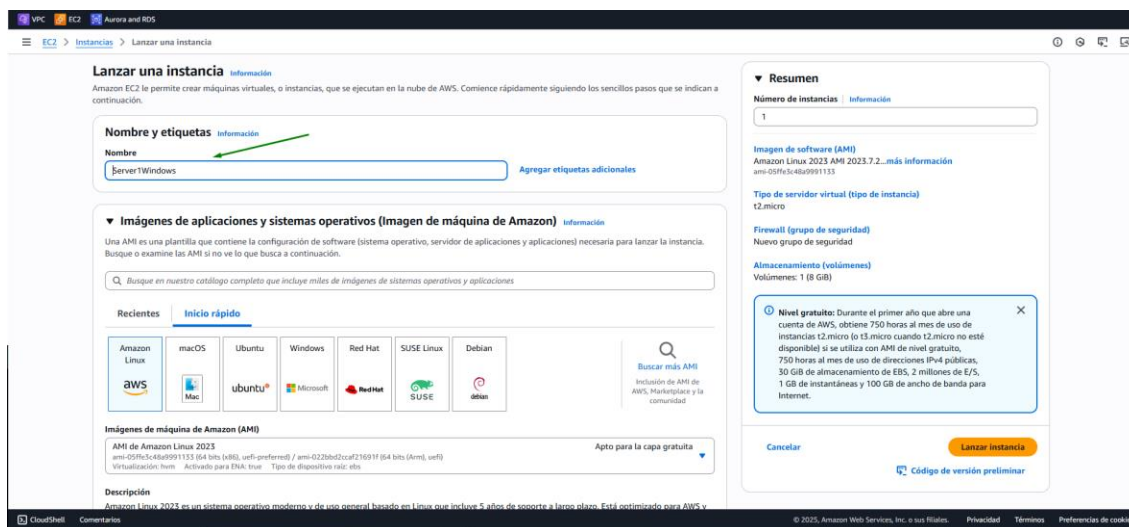


Ilustración 0.3

Seleccionar la AMI correspondiente:

Microsoft Windows 2016 Datacenter edition.

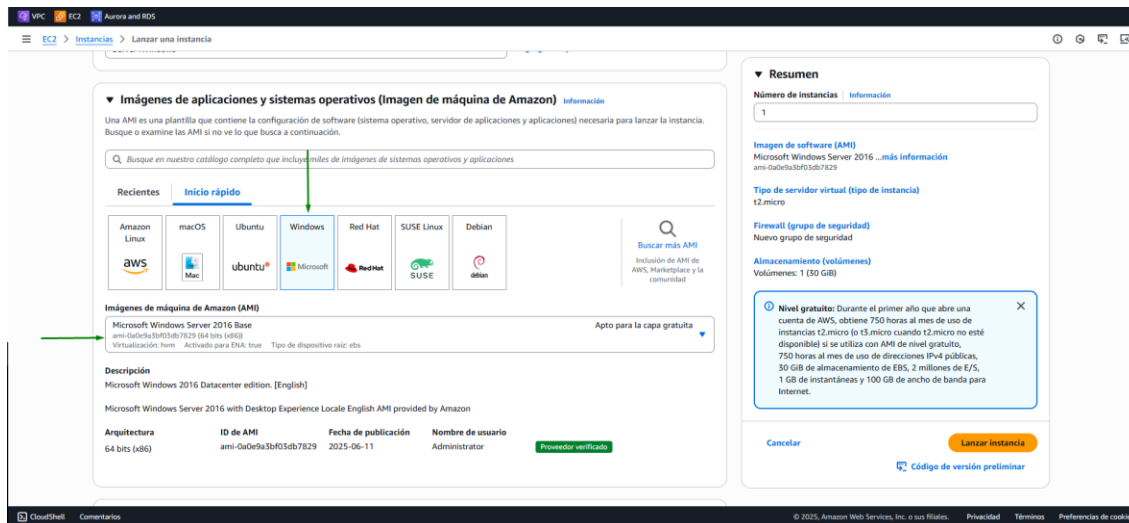


Ilustración 0.4

Amazon Linux 2023 AMI 2023.7.20250623.1 x86_64 HVM kernel-6.1

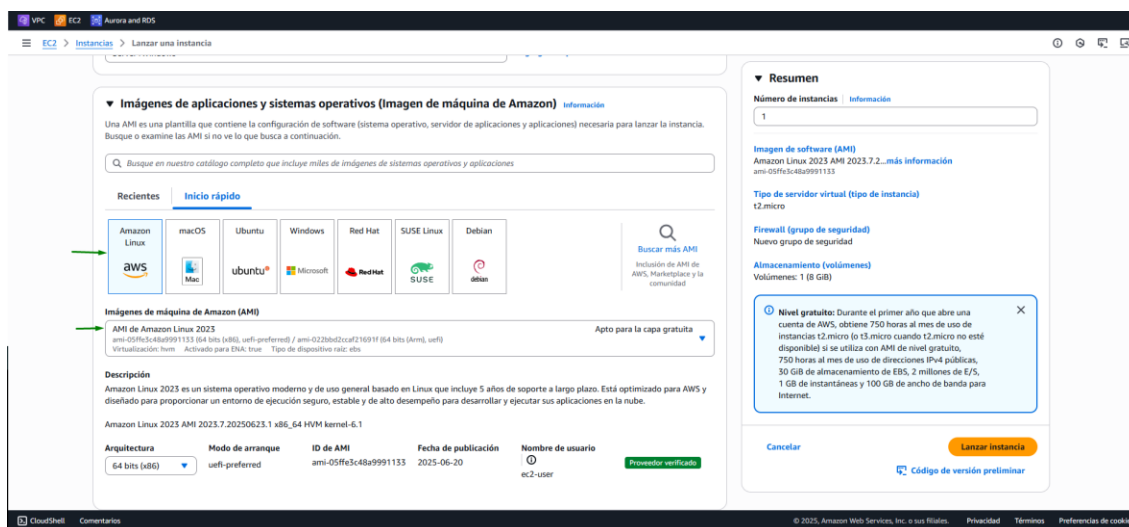


Ilustración 0.5

Elegir el tipo de instancia: t2.micro

The screenshot shows the 'Launch Instance' wizard in the AWS Management Console. The 'Instance Type' section is highlighted with a green arrow, showing 't2.micro' selected. The 'Key Pair' section is also visible, with a 'Select' button highlighted. The 'Summary' section on the right shows the instance configuration, including the software image (Microsoft Windows Server 2016), instance type (t2.micro), and storage (1 EBS volume).

Ilustración 0.6

Seleccionar o crear par llaves (PEM) para conexión remota.

The screenshot shows the 'Launch Instance' wizard in the AWS Management Console. The 'Key Pair' section is highlighted with a green arrow, showing the 'Select' button and the 'Create a new key pair' link. The 'Summary' section on the right shows the instance configuration, including the software image (Microsoft Windows Server 2016), instance type (t2.micro), and storage (1 EBS volume).

Ilustración 0.7

Escoger la subred pública dentro de la VPC seminario-vpc (zona us-east-1b).

Activar la asignación automática de IP pública.

Par de claves (inicio de sesión) Información

Puede utilizar un par de claves para conectarse de forma segura a la instancia. Asegúrese de que tiene acceso al par de claves seleccionado antes de lanzar la instancia.

Nombre del par de claves - obligatorio

WindowsServer [Crear un nuevo par de claves](#)

Para las instancias de Windows, utilice un par de claves para descifrar la contraseña del administrador y, a continuación, utilice la contraseña descifrada para conectarse a la instancia.

Configuraciones de red Información

VPC: obligatorio Información

vpc-0c97794c9f7334de (seminario-vpc) 10.0.0.0/16 [Crear nueva VPC](#)

Subred Información

subnet-05bfeb3f676a958da [Crear nueva subred](#) **seminario-subnet-public-1-us-east-1a**

VPC: vpc-0c97794c9f7334de Proprietario: 865362166802 Zona de disponibilidad: us-east-1a Tipo de zona: Zona de disponibilidad. Direcciones IP disponibles: 4089 CIDR: 10.0.0.0/20

Asignar automáticamente la IP pública Información

Habilitar [¿Qué es un tráfico específico que llegue a la instancia?](#)

Desactivar

Crear grupo de seguridad Seleccionar un grupo de seguridad existente

Grupos de seguridad comunes Información

Seleccionar grupos de seguridad [Compare reglas de grupo de seguridad](#)

Los grupos de seguridad que agrega o elimina aquí se agregarán a todas las interfaces de red o se eliminarán de ellas.

Configuración de red avanzada

Resumen

Número de instancias Información

1

Imagen de software (AMI) [Microsoft Windows Server 2016 ... más información](#) ami-0a6e930f50b7829

Tipo de servidor virtual (tipo de instancia) t2.micro

Firewall (grupo de seguridad) -

Almacenamiento (volúmenes) Volúmenes: 1 (30 GiB)

Nivel gratuito: Durante el primer año que abre una cuenta de AWS, obtiene 750 horas al mes de uso de instancias t2.micro o t3.micro cuando t2.micro no esté disponible si se utiliza con AMI de nivel gratuito, 750 horas al mes de uso de direcciones IPv4 públicas, 30 GiB de almacenamiento de EBS, 2 millones de E/S, 1 GB de instantáneas y 100 GB de ancho de banda para Internet.

[Cancelar](#) [Lanzar instancia](#) [Código de versión preliminar](#)

Ilustración 0.8

Asignar un grupo de seguridad nuevo o existente según la instancia.

vpc-0c97794c9f7334de (seminario-vpc) 10.0.0.0/16 [Crear nueva VPC](#)

Subred Información

subnet-05bfeb3f676a958da [Crear nueva subred](#) **seminario-subnet-public-1-us-east-1a**

VPC: vpc-0c97794c9f7334de Proprietario: 865362166802 Zona de disponibilidad: us-east-1a Tipo de zona: Zona de disponibilidad. Direcciones IP disponibles: 4089 CIDR: 10.0.0.0/20

Asignar automáticamente la IP pública Información

Habilitar [Se aplican cargos adicionales cuando no se cumplen los límites del nivel gratuito.](#)

Desactivar

Firewall (grupos de seguridad) Información

Un grupo de seguridad es un conjunto de reglas de Firewall que controlan el tráfico de la instancia. Agregue reglas para permitir que un tráfico específico llegue a la instancia.

Crear grupo de seguridad Seleccionar un grupo de seguridad existente

Grupos de seguridad comunes Información

Seleccionar grupos de seguridad [Compare reglas de grupo de seguridad](#)

default VPC: vpc-0c97794c9f7334de sg-05b24c7291ebf1c536

launch-wizard-1 VPC: vpc-0c97794c9f7334de sg-0c9aee52587683ccd

sgwindowner VPC: vpc-0c97794c9f7334de sg-0d5a4c31090f9ced7 **Avanzado**

[Los clientes que cumplan los requisitos de la capa gratuita pueden obtener hasta 30 GiB de almacenamiento magnético o de uso general \(SSD\) de EBS.](#) [Agregar un nuevo volumen](#)

The selected AMI contains instance store volumes, however the instance does not allow any instance store volumes. None of the instance store volumes from the

Resumen

Número de instancias Información

1

Imagen de software (AMI) [Microsoft Windows Server 2016 ... más información](#) ami-0a6e930f50b7829

Tipo de servidor virtual (tipo de instancia) t2.micro

Firewall (grupo de seguridad) -

Almacenamiento (volúmenes) Volúmenes: 1 (30 GiB)

Nivel gratuito: Durante el primer año que abre una cuenta de AWS, obtiene 750 horas al mes de uso de instancias t2.micro o t3.micro cuando t2.micro no esté disponible si se utiliza con AMI de nivel gratuito, 750 horas al mes de uso de direcciones IPv4 públicas, 30 GiB de almacenamiento de EBS, 2 millones de E/S, 1 GB de instantáneas y 100 GB de ancho de banda para Internet.

[Cancelar](#) [Lanzar instancia](#) [Código de versión preliminar](#)

Ilustración 0.9

Lanzar la instancia y esperar a que el estado esté como “En ejecución”.

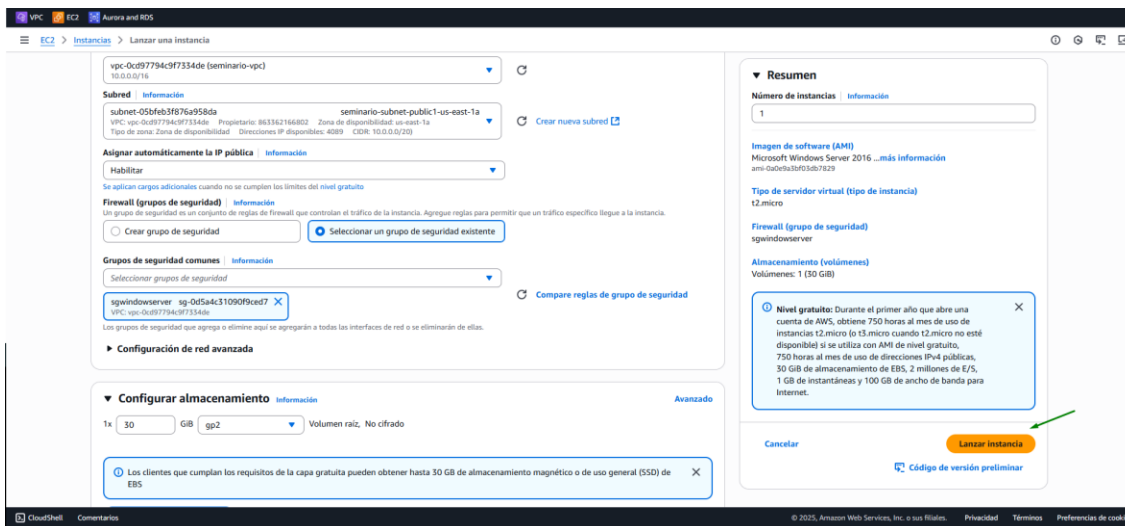


Ilustración 0.10

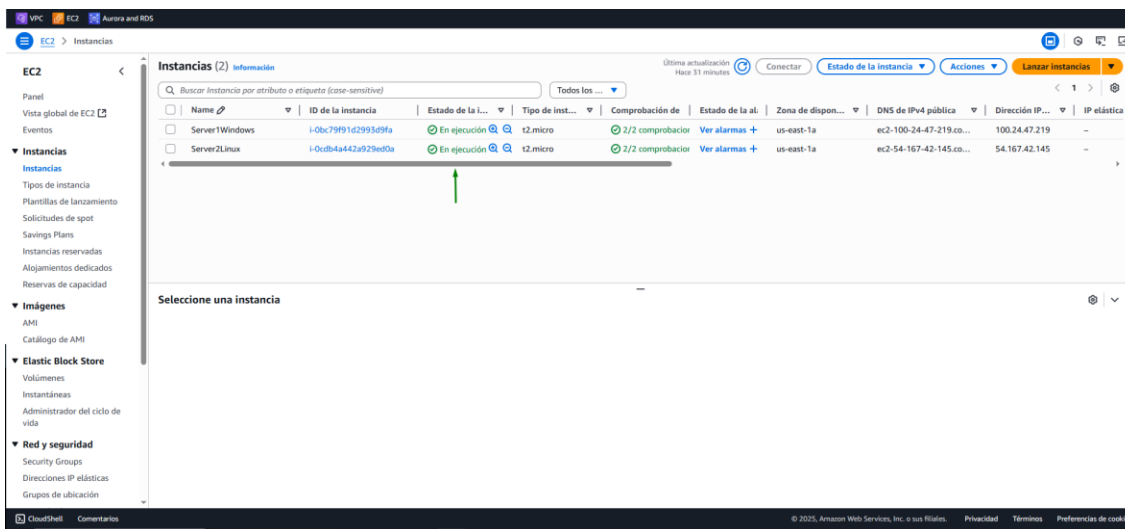


Ilustración 0.11

Detalles de los Grupos de Seguridad (puertos abiertos: RDP, SSH, HTTP).

Grupo de seguridad para Server1Windows – sg-windowserver:

RDP (puerto 3389) habilitado desde cualquier IP (0.0.0.0/0).

HTTP (puerto 80) habilitado desde cualquier IP (0.0.0.0/0).

EC2 > Instancias

Instancias (1/2) Información

Nombre	ID de la instancia	Estado de la I...	Tipo de inst...	Comprobación de	Estado de la at...	Zona de dispon...	DNS de IPv4 pública	Dirección IP...	IP elástica
Server1Windows	i-0bc79f91d2993d9fa	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-1a	ec2-100-24-47-219.co...	100.24.47.219	-
Server2Linux	i-0cdb4442a929ed0a	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-1a	ec2-54-167-42-145.co...	54.167.42.145	-

i-0bc79f91d2993d9fa (Server1Windows)

Reglas de entrada

Nombre	ID de la regla del grupo d...	Intervalo de pu...	Protocolo	Origen	Grupos de seguridad	Descripción
-	sgr-0ca6a84b479c5b0bf	80	TCP	0.0.0.0/0	sg-windowserver	Web
-	sgr-09b68ac671e9eebf	3389	TCP	0.0.0.0/0	sg-windowserver	-

Reglas de salida

Nombre	ID de la regla del grupo d...	Intervalo de pu...	Protocolo	Destino	Grupos de seguridad	Descripción
-	sgr-0fd4051fb2f445f5c	Todo	Todo	0.0.0.0/0	sg-windowserver	-

Ilustración 0.12

Grupo de seguridad para Server2Linux – launch-wizard-1:

SSH (puerto 22) habilitado desde cualquier IP (0.0.0.0/0).

HTTP (puerto 80) habilitado desde cualquier IP (0.0.0.0/0).

EC2 > Instancias

Instancias (1/2) Información

Nombre	ID de la instancia	Estado de la I...	Tipo de inst...	Comprobación de	Estado de la at...	Zona de dispon...	DNS de IPv4 pública	Dirección IP...	IP elástica
Server1Windows	i-0bc79f91d2993d9fa	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-1a	ec2-100-24-47-219.co...	100.24.47.219	-
Server2Linux	i-0cdb4442a929ed0a	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-1a	ec2-54-167-42-145.co...	54.167.42.145	-

i-0cdb4442a929ed0a (Server2Linux)

Reglas de entrada

Nombre	ID de la regla del grupo d...	Intervalo de pu...	Protocolo	Origen	Grupos de seguridad	Descripción
-	sgr-02396b98e77169378	22	TCP	0.0.0.0/0	launch-wizard-1	-
-	sgr-03ec762a46bad73a	80	TCP	0.0.0.0/0	launch-wizard-1	-

Reglas de salida

Nombre	ID de la regla del grupo d...	Intervalo de pu...	Protocolo	Destino	Grupos de seguridad	Descripción
-	sgr-0c626071e9fca36bc	Todo	Todo	0.0.0.0/0	launch-wizard-1	-

Ilustración 0.13

Asignación de IPs públicas y privadas.

Ambas instancias fueron configuradas para recibir automáticamente una dirección IP pública al momento de su creación. Las direcciones asignadas fueron:

Server1Windows

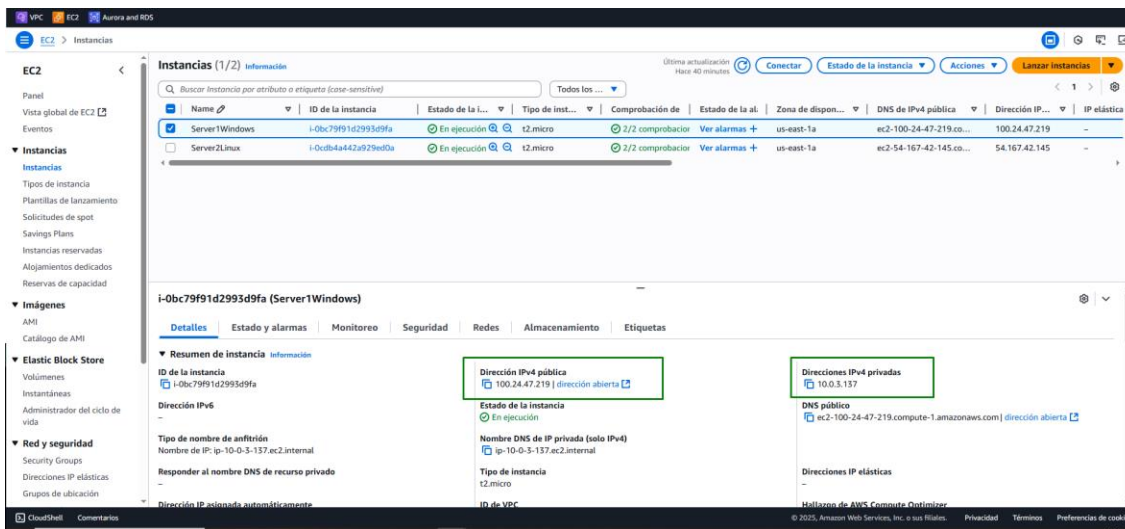


Ilustración 0.14

Server2Linux

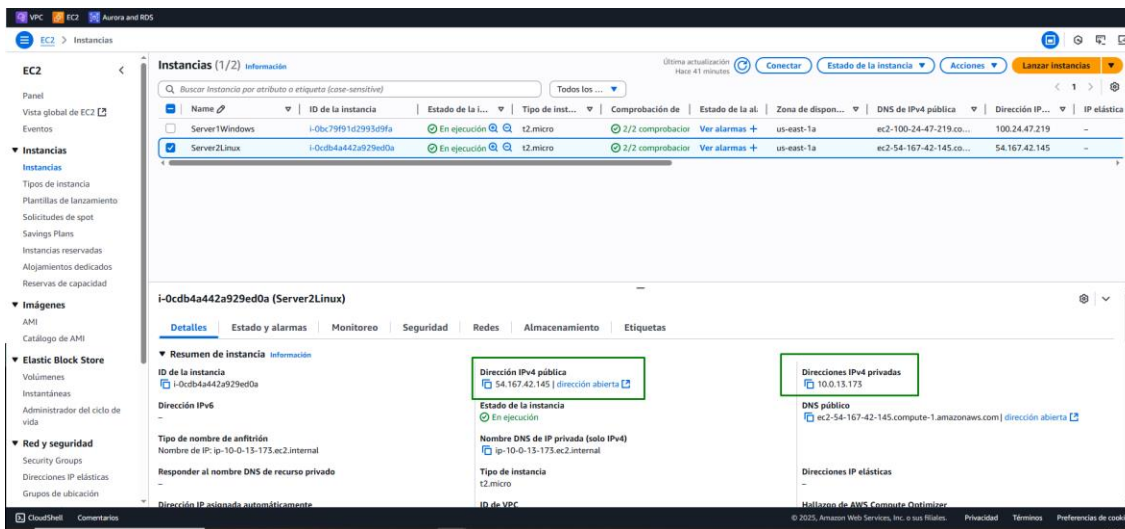


Ilustración 0.15

Estas IPs permiten la conexión remota desde Internet por RDP (Windows), SSH (Linux) y HTTP para ambos casos.

Procedimiento de acceso

Cómo acceder a cada servidor (cliente RDP para Windows, SSH para Linux).

Server1Windows (acceso vía RDP)

Para acceder a la instancia Windows, se utilizó el protocolo Remote Desktop Protocol (RDP) a través de un cliente de Escritorio Remoto:

En el panel superior, se hizo clic en el botón "Conectar".

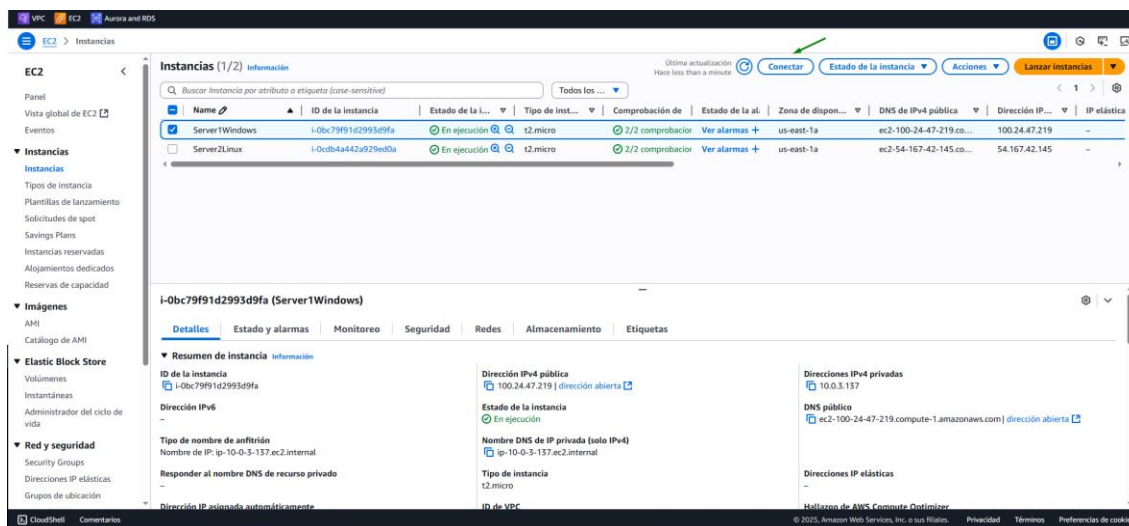


Ilustración 0.16

Se eligió la opción "Cliente de Escritorio Remoto", y se descargó el archivo .rdp (En este acaso ya se había descargado)

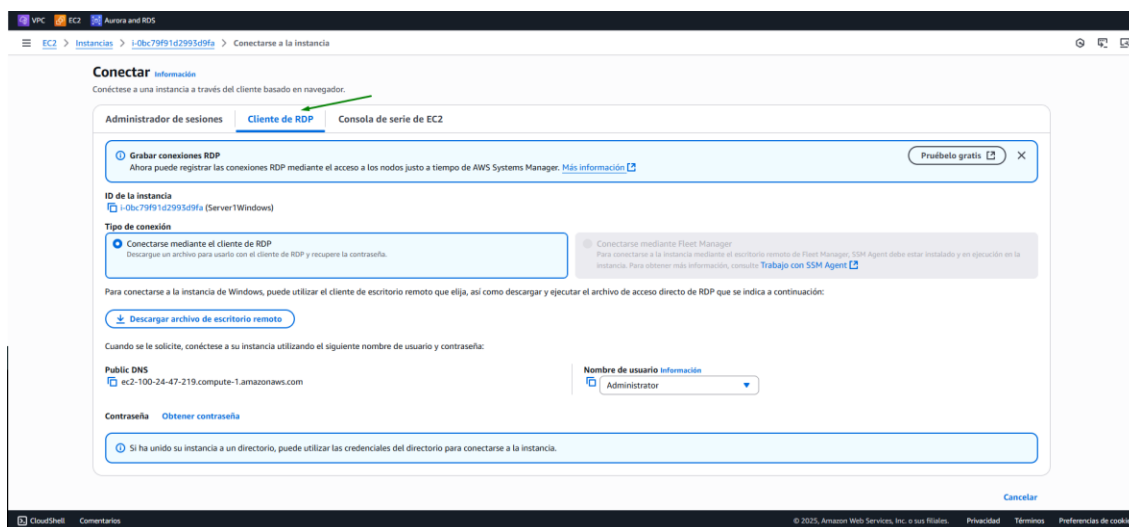


Ilustración 0.17

Se hizo clic en "Obtener contraseña" e ingresó el archivo .pem asociado al par de claves creado en el lanzamiento.

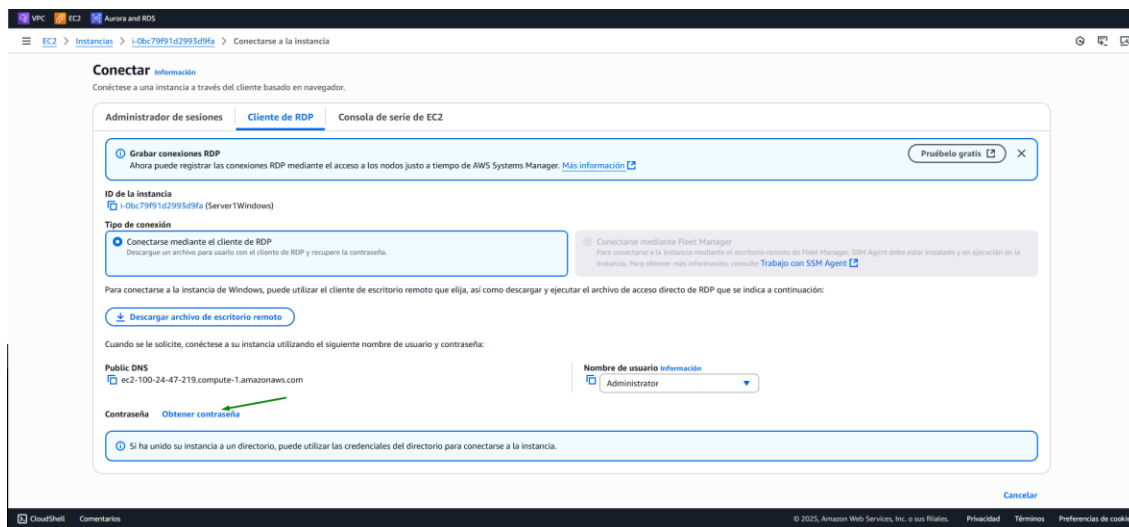


Ilustración 0.18

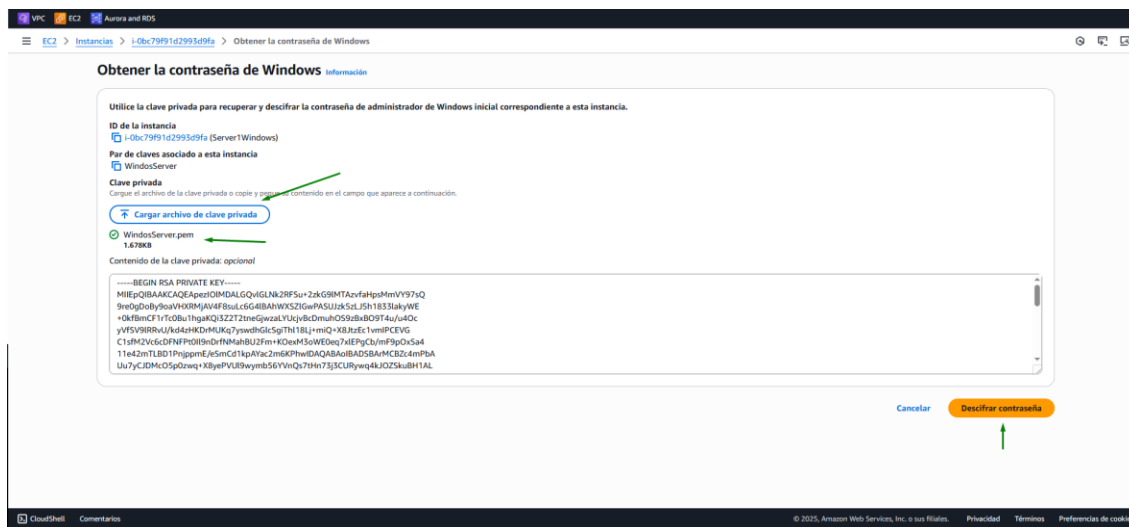


Ilustración 0.19

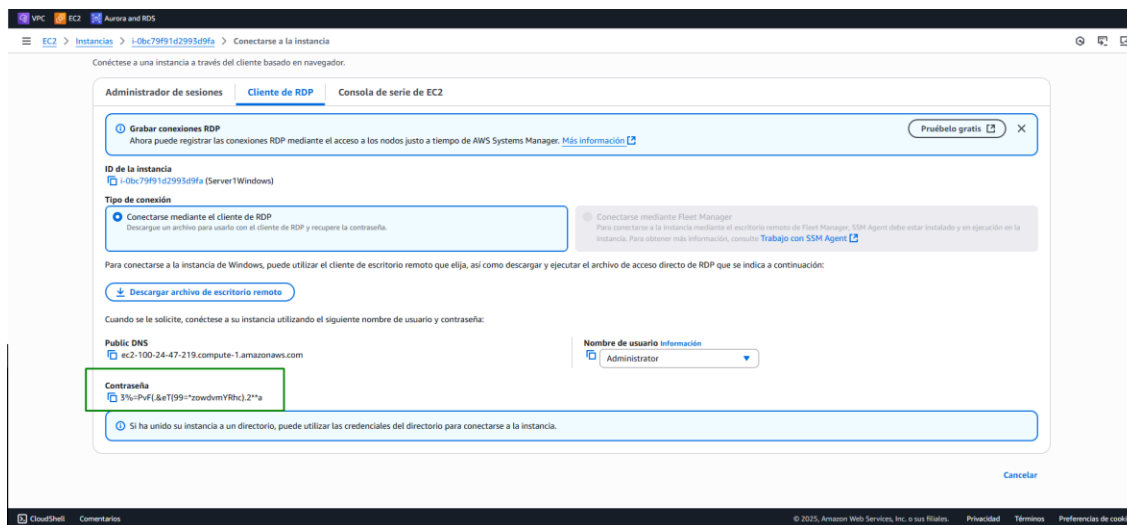


Ilustración 0.20

Este procedimiento permite ingresar al entorno gráfico del servidor Windows desde cualquier equipo compatible con RDP.

Server2Linux (acceso vía SSH)

Para la conexión a la instancia Linux, se utilizó MobaXterm, una herramienta todo-en-uno que facilita el acceso remoto mediante SSH desde entornos Windows.

Se abrió MobaXterm y se creó una nueva sesión tipo SSH.

En el campo "Remote host" se ingresó la IP pública de la instancia Linux: 54.167.42.145.

Se estableció el usuario como ec2-user, correspondiente a Amazon Linux.

En la pestaña "Advanced SSH settings", se marcó la opción "Use private key" y se cargó la llave privada .pem.

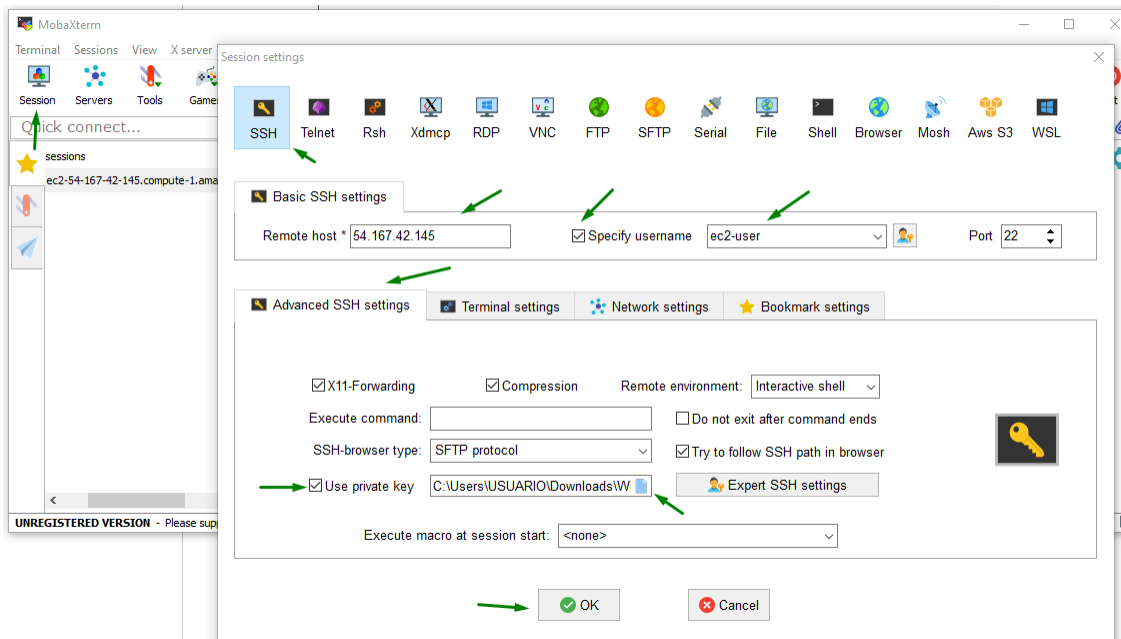


Ilustración 0.21

Una vez iniciada la conexión, MobaXterm abrió una terminal SSH funcional para administrar el servidor.

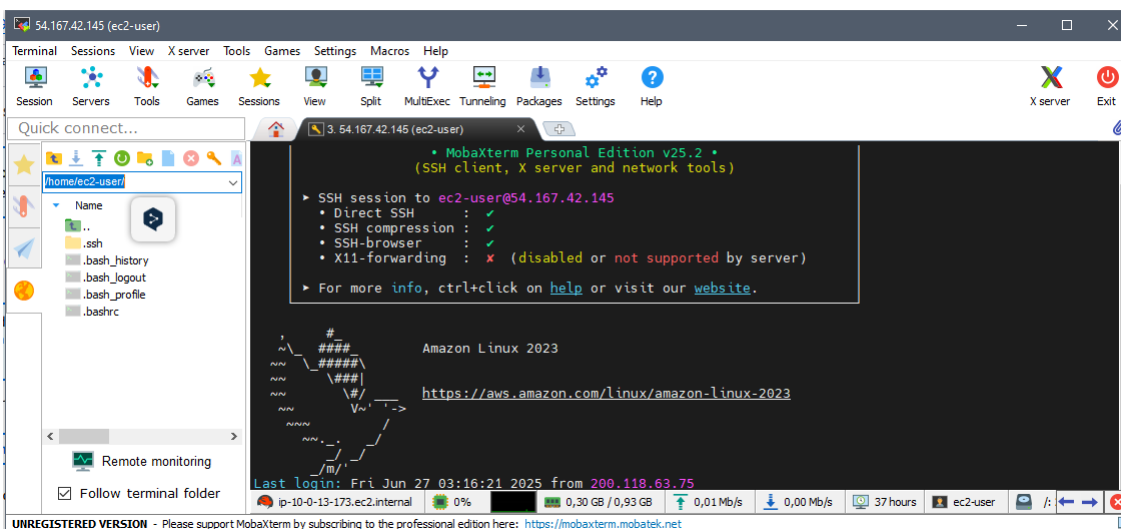


Ilustración 0.22

Consideraciones de seguridad (por ejemplo, uso de llaves PEM, contraseñas seguras).

Llaves PEM: Para ambos servidores se utilizó una clave privada tipo .pem, generada durante la creación de las instancias. Esta clave es esencial para descriptar la contraseña del servidor Windows y para autenticarse vía SSH en Linux.

Contraseñas seguras: La contraseña del servidor Windows se generó de forma aleatoria y se descriptó únicamente con la clave privada, garantizando que solo el propietario del archivo pem tenga acceso.

Reglas de seguridad: Los grupos de seguridad fueron configurados para permitir únicamente los puertos necesarios, reduciendo la exposición de la red.

Control de acceso mediante grupos de seguridad: Se configuraron reglas específicas en los grupos de seguridad para permitir únicamente los puertos esenciales:

RDP (3389) en Windows.

SSH (22) en Linux.

HTTP (80) en ambos, para acceso web.

Configuración del servidor web

Pasos seguidos para instalar IIS en Windows Server.

Instalación de IIS en Windows Server

Para convertir la instancia EC2-Server1 Windows en un servidor web funcional, se instaló IIS (Internet Information Services), que es el servidor web nativo de Windows Server.

Se descriptó la contraseña y se copió para el acceso mediante RDP.

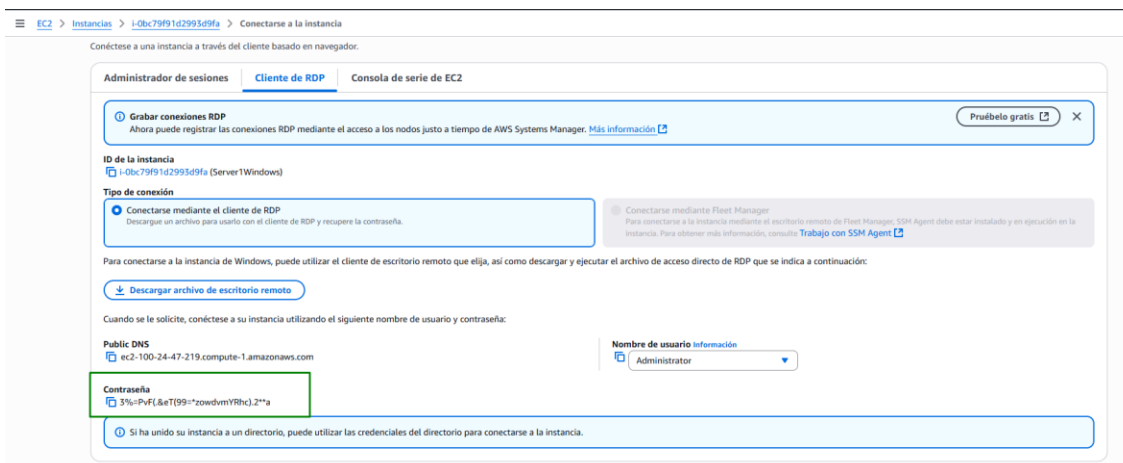


Ilustración 0.23

Conexión RDP mediante IP Pública

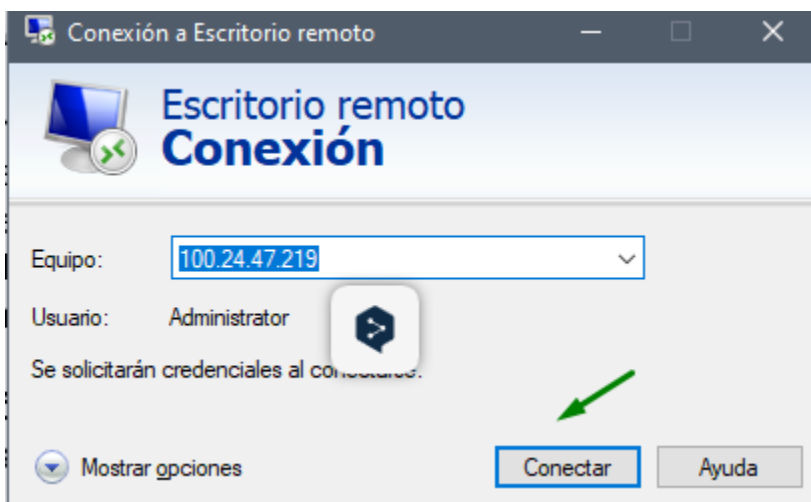


Ilustración 0.24

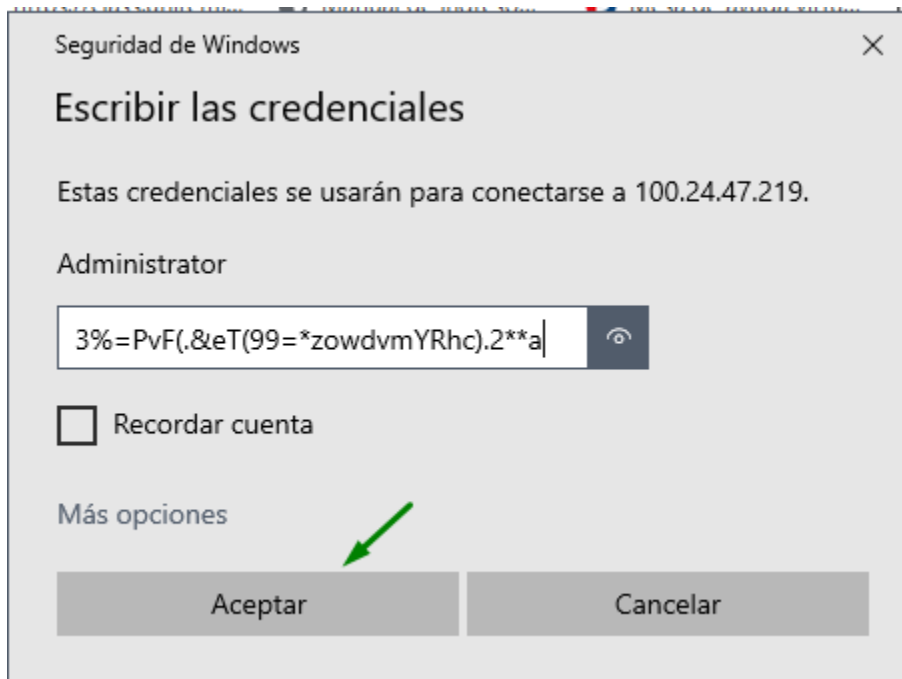


Ilustración 0.25

Se abrió el Administrador del servidor (Server Manager).

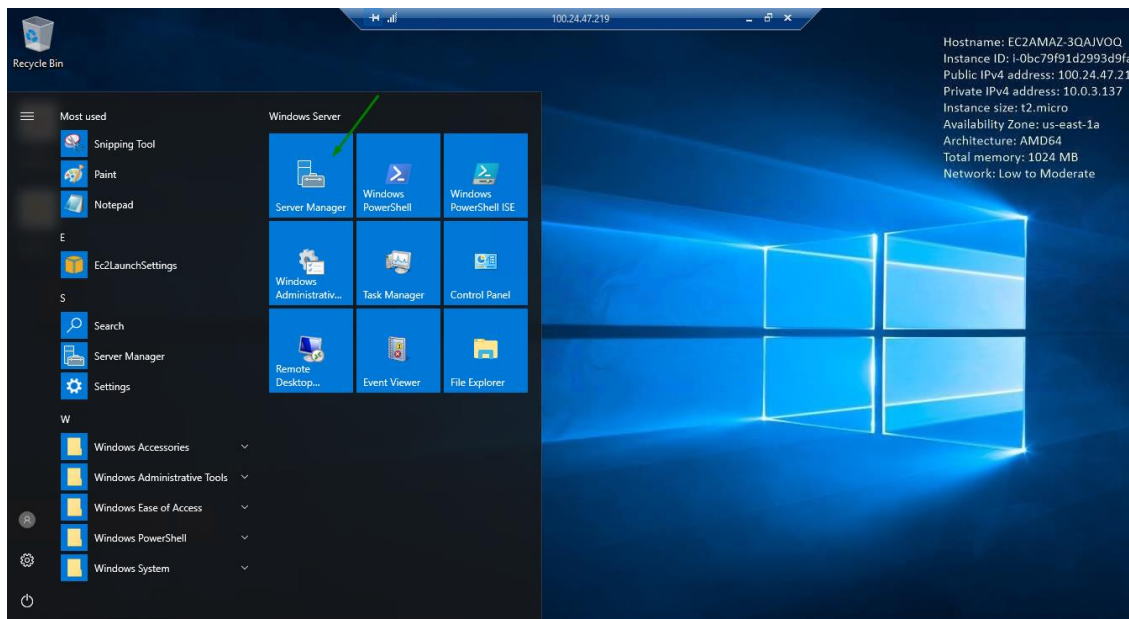


Ilustración 0.26

En el panel lateral, se hizo clic en "Agregar roles y características".

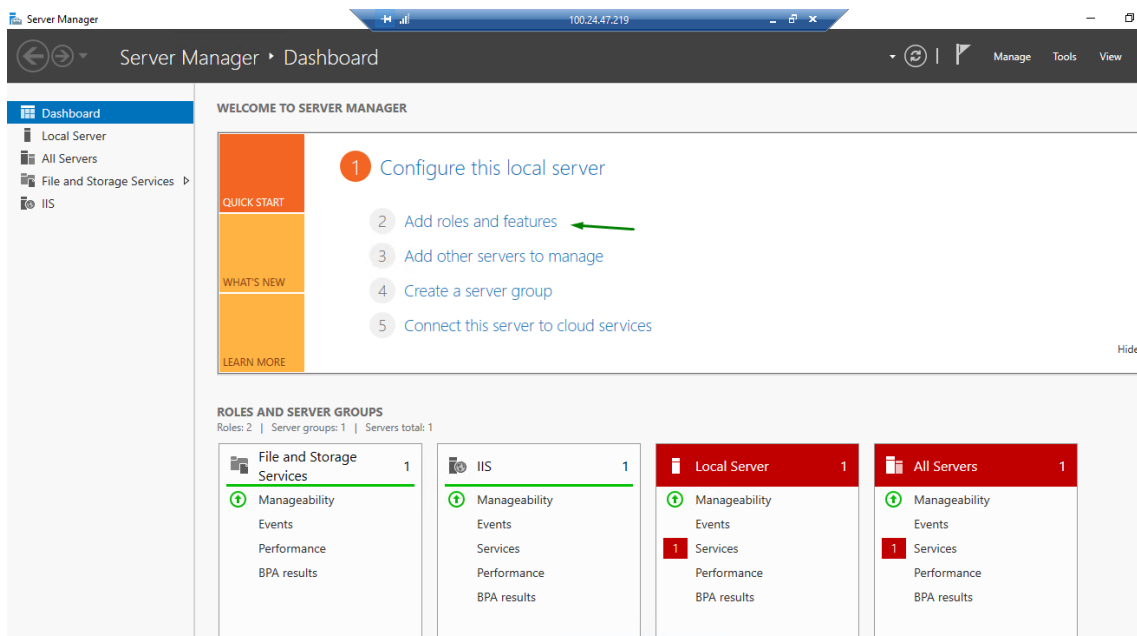


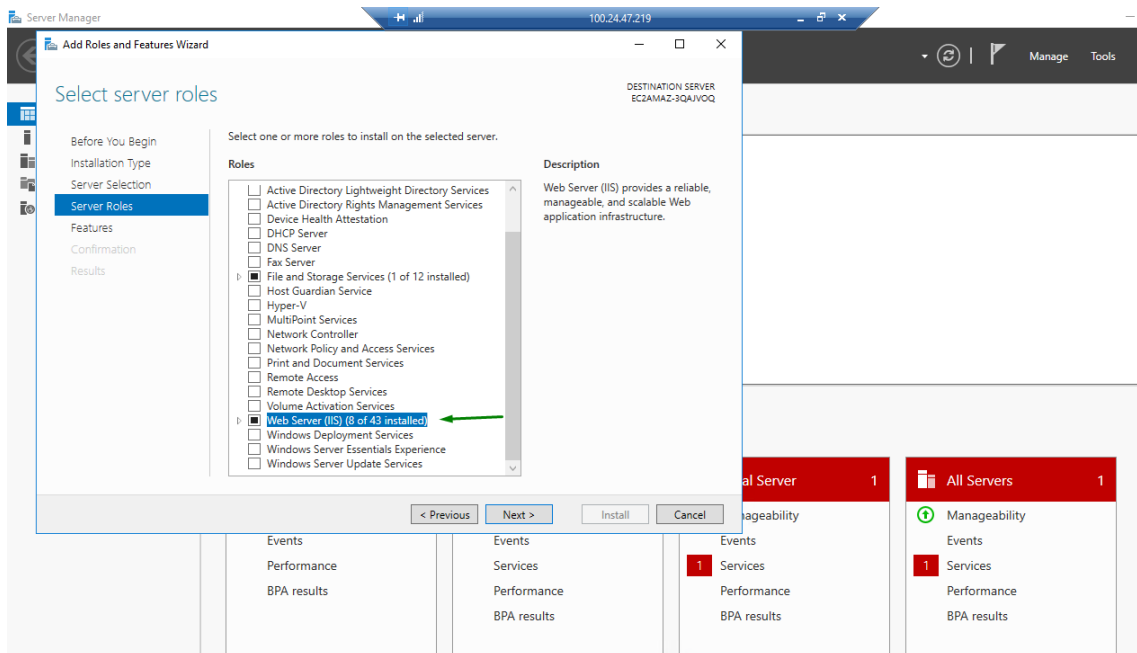
Ilustración 0.27

En el asistente, se seleccionó:

Instalación basada en características o roles.

La instancia local como servidor de destino.

El rol "Servidor web (IIS)".



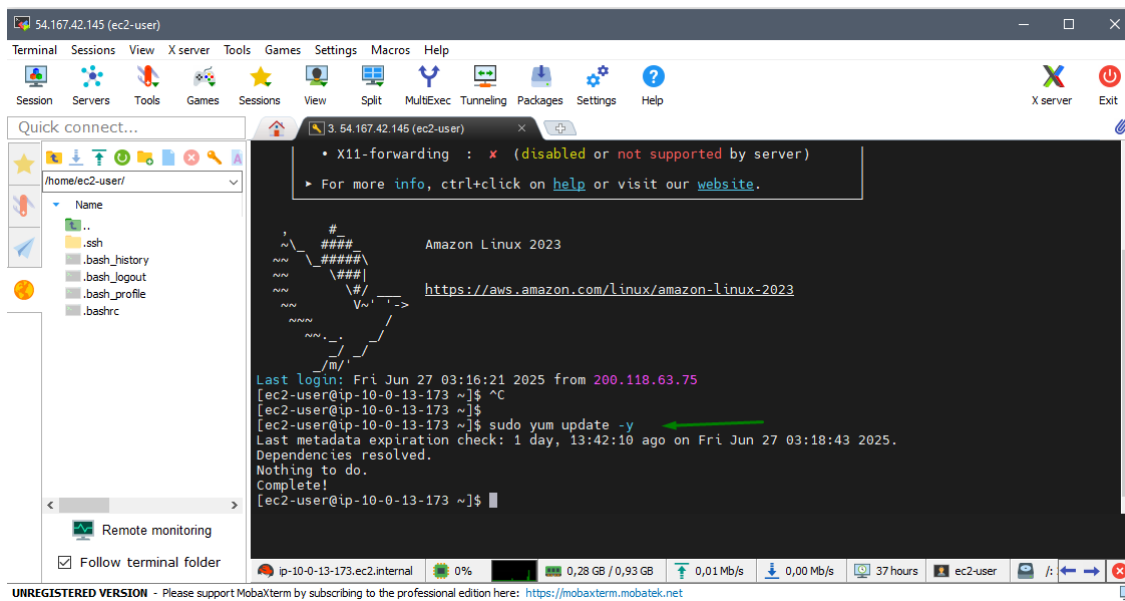


Ilustración 0.30

Instalación paquete de Apache.

```
sudo yum install httpd -y
```

sudo: root

yum: Gestor de paquetes.

install: Ordena instalar un paquete.

httpd: Es el nombre del paquete del servidor Apache en Amazon Linux.

-y: Acepta la instalación sin pedir confirmación manual.

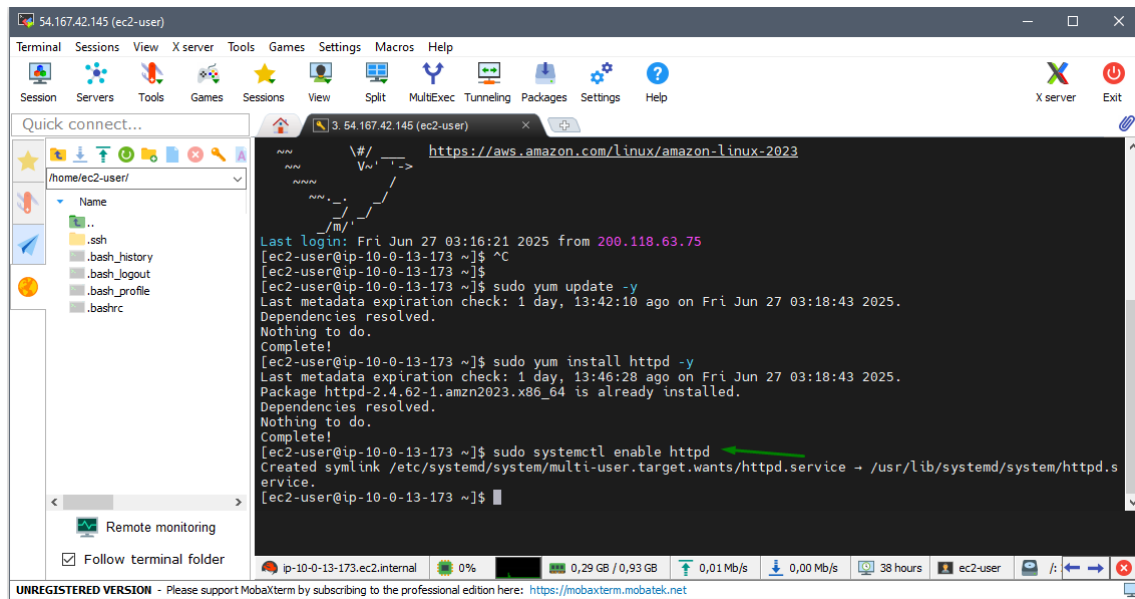


Ilustración 0.32

Iniciar el servicio de Apache

`sudo systemctl start httpd`

`systemctl`: Administra servicios.

`start`: Inicia manualmente el servicio (no espera al próximo reinicio).

`httpd`: El nombre del servicio.

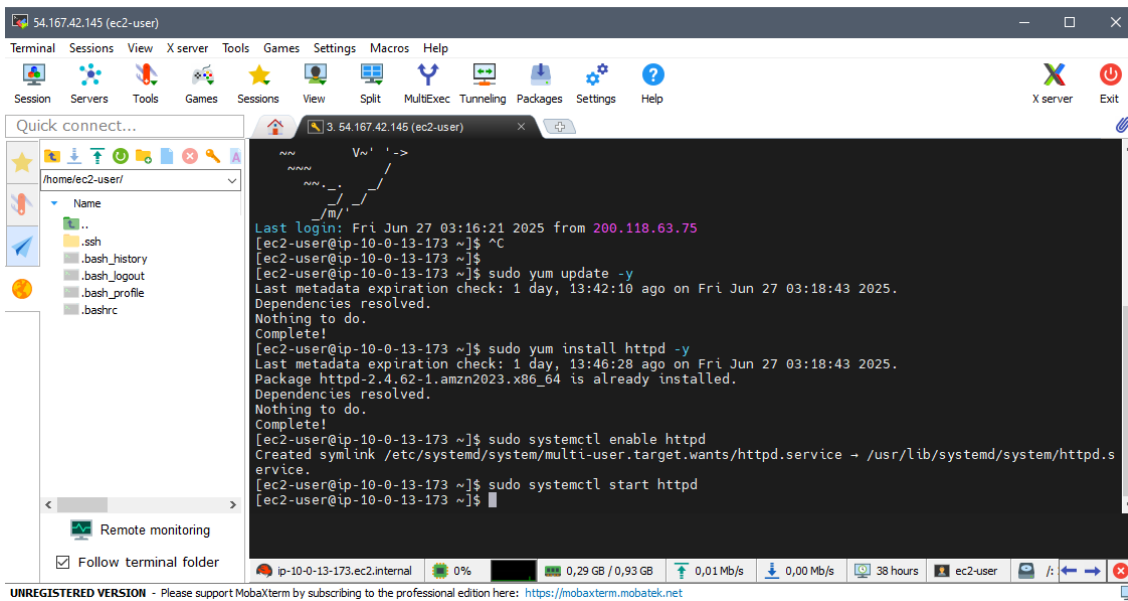


Ilustración 0.33

Verificación del estado del servicio

Sudo systemctl status httpd

Status: Muestra el estado actual del servicio.

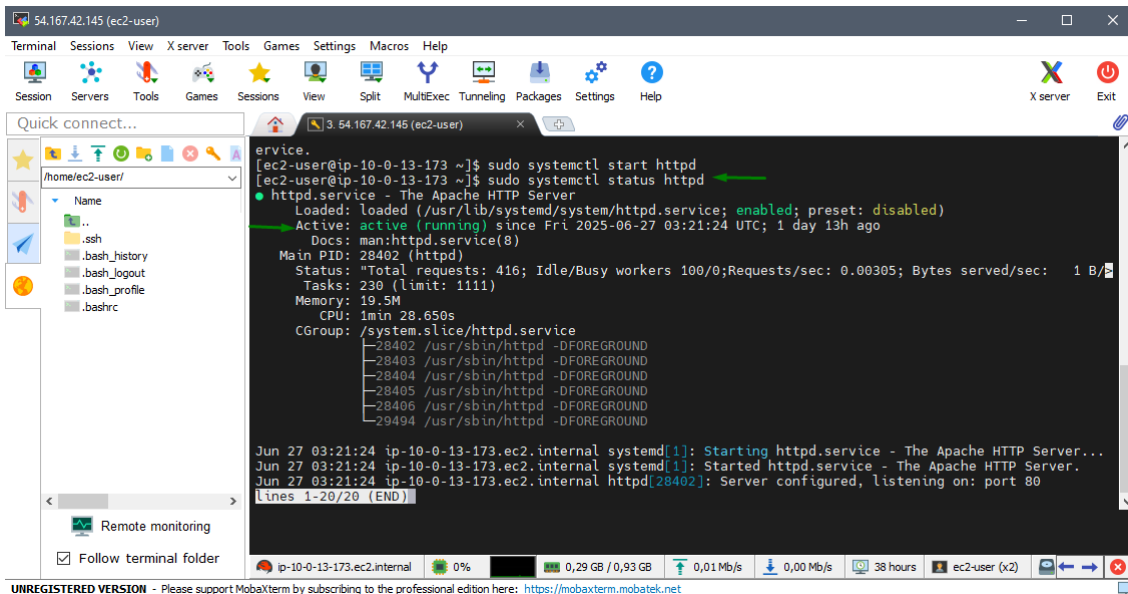


Ilustración 0.34

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

Página de IIS al acceder a <http://100.24.47.219>

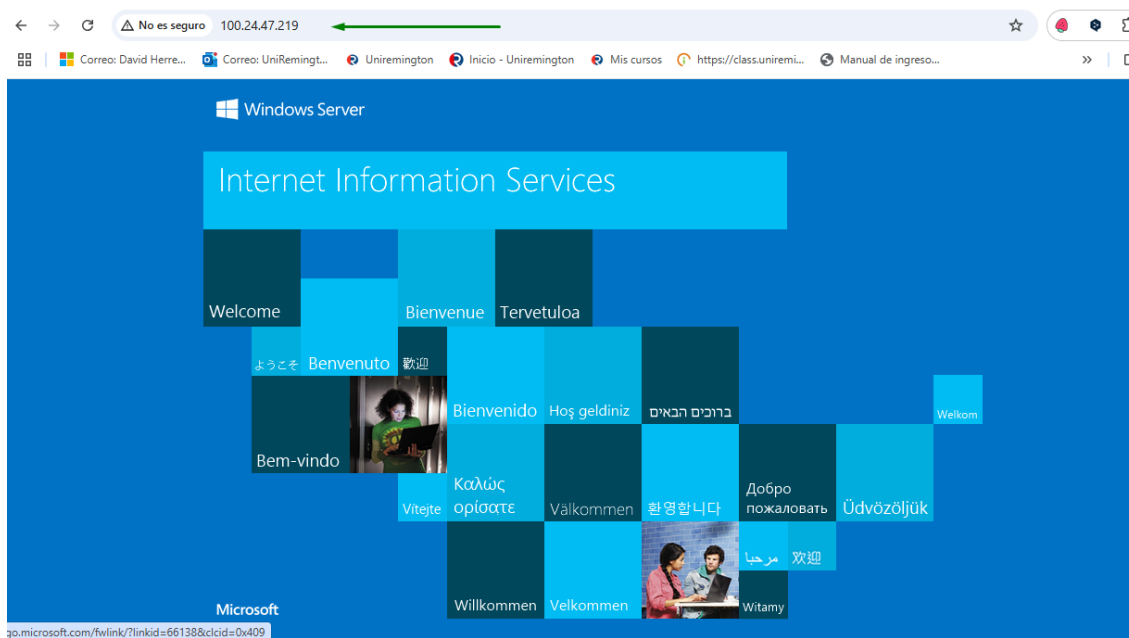
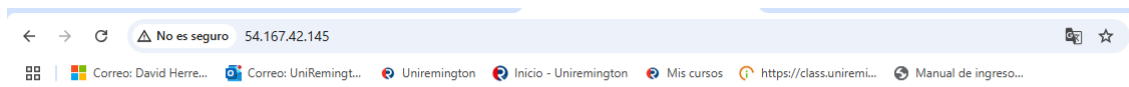


Ilustración 0.35

Página de Apache al acceder a <http://54.167.42.145>



It works!

Ilustración 0.36

Parte 2: Práctica (Implementación y pruebas)

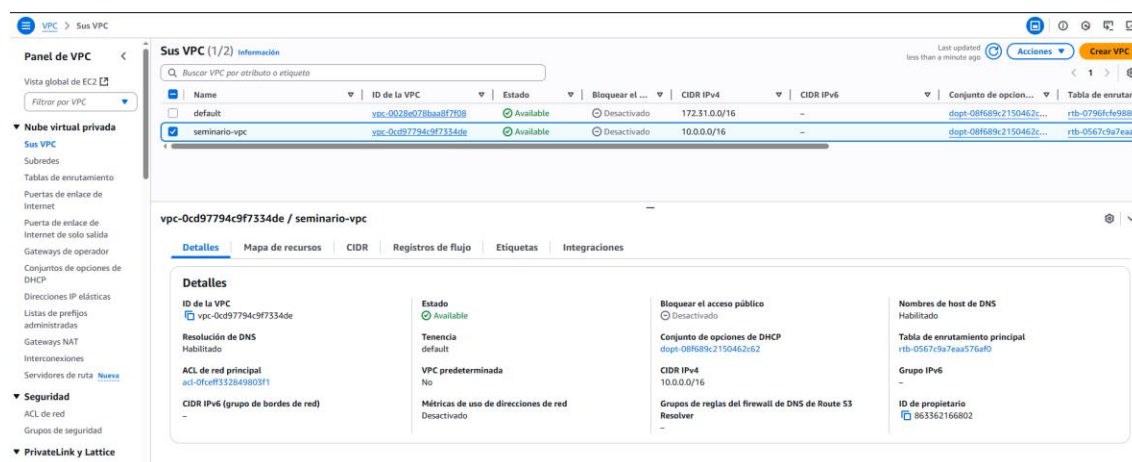
Actividades prácticas:

Crear la infraestructura

A continuación, se presentan las evidencias gráficas que respaldan la implementación de la infraestructura en AWS. Estas incluyen la creación de la VPC, subredes públicas, instancias EC2, y el Internet Gateway utilizado para permitir la conexión a Internet.

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

Se creó una VPC llamada `seminario-vpc`, que permite controlar de forma centralizada el direccionamiento IP, las subredes, las rutas y los elementos de red asociados a las instancias EC2.



Name	ID de la VPC	Estado	Bloquear el ...	CIDR IPv4	CIDR IPv6	Conjunto de opción...	Tabla de enrutami
default	vpc-0028e078baa8f7808	Available	Desactivado	172.31.0.0/16	-	dopt-08f689c2150462c...	rtb-0796fc498805
seminario-vpc	vpc-0cd97794c9f7334de	Available	Desactivado	10.0.0.0/16	-	dopt-08f689c2150463...	rtb-0567c9a7eaa5

Detalles	
ID de la VPC vpc-0cd97794c9f7334de	Estado Available
Resolución de DNS Habilitado	Tenencia default
ACL de red principal acl-0fceff332849803f1	VPC predeterminada No
CIDR IPv6 (grupo de bordes de red) -	Métricas de uso de direcciones de red Desactivado
Bloquear el acceso público Desactivado	Conjunto de opciones de DHCP dopt-08f689c2150462c62
CIDR IPv4 10.0.0.0/16	Grupos de reglas del firewall de DNS de Route 53 Resolver
Nombres de host de DNS Habilitado	Tabla de enrutamiento principal rtb-0567c9a7eaa576af0
Grupo IPv6 -	ID de propietario 863362166802

Ilustración 0.1

Subred pública

La subred pública configurada fue `seminario-subnet-public1-us-east-1a`, asignada a la zona de disponibilidad `us-east-1b`. Esta subred permite que las instancias reciban tráfico desde y hacia Internet.

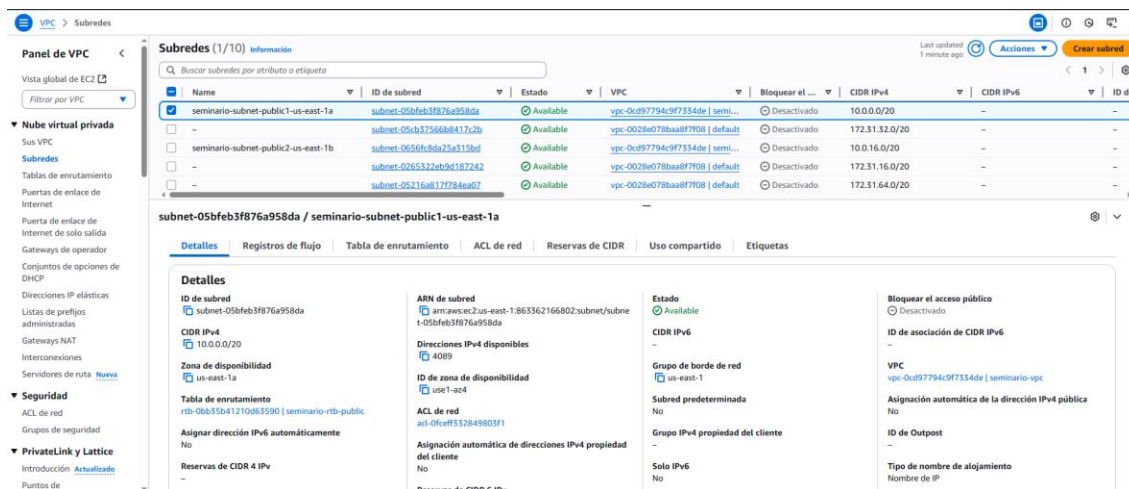


Ilustración 0.2

Lanzar dos instancias EC2:

Una instancia Windows Server.

EC2/Server1Windows

Esta instancia fue creada con Windows Server 2016, asignada a la subred pública y con IP pública habilitada para conexión RDP.

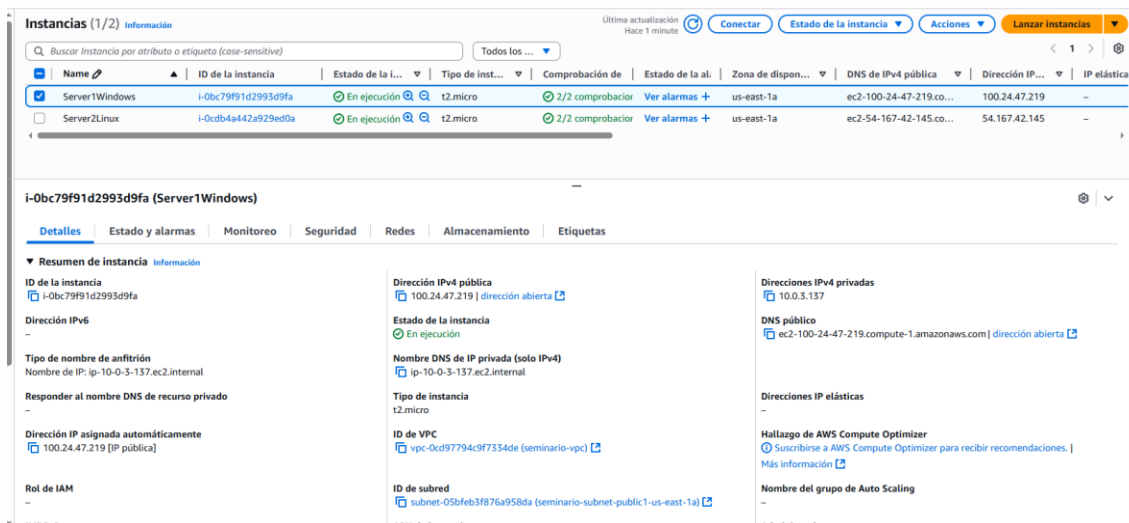


Ilustración 0.3

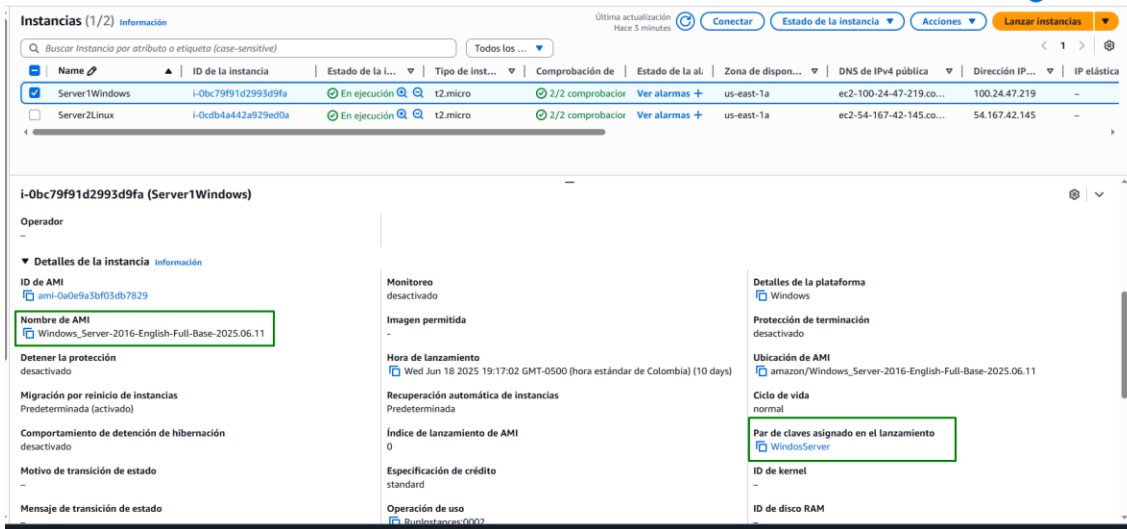


Ilustración 0.4

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

EC2/Server2Linux

Esta instancia corre Amazon Linux 2023, en la misma subred pública, con acceso SSH habilitado.

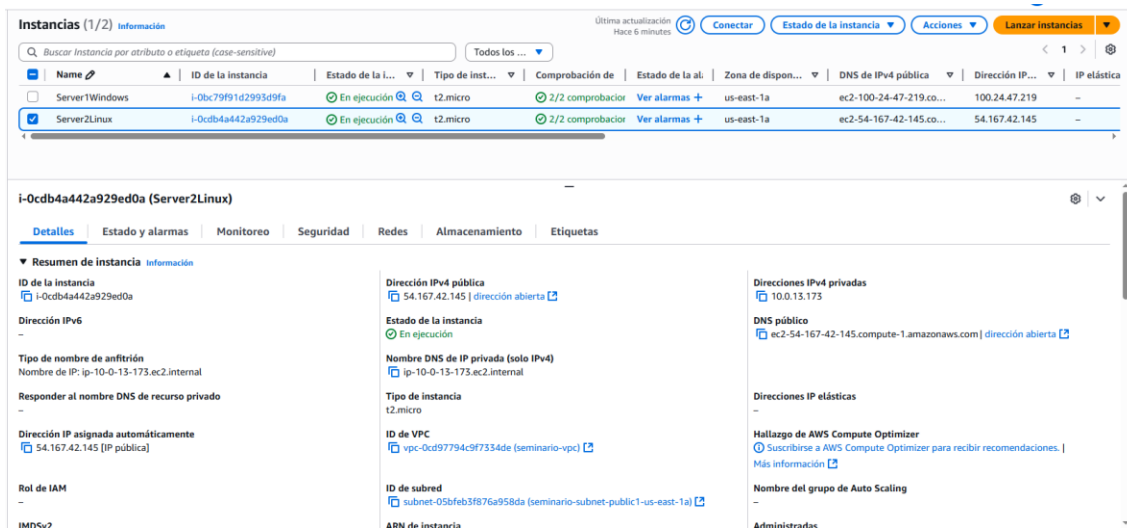


Ilustración 0.5

The screenshot displays the AWS Management Console interface for EC2 instances. At the top, there's a header for 'Instancias (1/2) Información' with a search bar and various action buttons like 'Conectar', 'Estado de la instancia', and 'Lanzar instancias'. Below this is a table listing instances. The second instance, 'Server2Linux' with ID 'i-0cdb4a442a929ed0a', is selected and highlighted in blue. Below the table, the detailed view for this instance is shown, categorized into 'Detalles de la instancia Información'. This view is split into three columns: 'ID de AMI', 'Monitoreo', and 'Detalles de la plataforma'. In the 'ID de AMI' column, the 'Nombre de AMI' is highlighted with a green box: 'al2023-ami-2023.7.20250623.1-kernel-6.1-x86_64'. In the 'Detalles de la plataforma' column, the 'Par de claves asignado en el lanzamiento' is highlighted with a green box: 'WinDowsServer'. Other details include 'Monitoreo desactivado', 'Imagen permitida', 'Hora de lanzamiento' (Thu Jun 26 2025 22:04:01 GMT-0500), 'Recuperación automática de instancias' (Predeterminada), 'Índice de lanzamiento de AMI' (0), 'Especificación de crédito' (standard), 'Operación de uso' (RunInstances), 'Compatibilidad con enclaves', '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), 'Ciclo de vida' (normal), 'ID de kernel', 'ID de disco RAM', and 'Modo de arranque' (uefi-preferred).

Ilustración 0.6

Internet Gateway

Como parte de la infraestructura de red, se configuró un Internet Gateway (igw-03244fb9bd4e89ee1) adjunto a la VPC seminario-vpc. Este componente permite que las instancias EC2 dentro de la red puedan comunicarse con Internet, ya sea para conexión remota (SSH/RDP) o para servir contenido web (HTTP).

La subred utilizada (seminario-subnet-public1-us-east-1a) está asociada a la tabla de enrutamiento seminario-rtb-public, la cual incluye la siguiente regla clave:

0.0.0.0/0 → igw-03244fb9bd4e89ee1

Esto garantiza que todo el tráfico de salida de la subred pueda llegar a Internet a través del Internet Gateway, siempre que las instancias cuenten con una dirección IP pública y grupos de seguridad configurados correctamente.

The screenshot displays the AWS Management Console interface for subnets. At the top, there's a search bar and a table of subnets. Below the table, the 'Tabla de enrutamiento' (Routing Table) section is expanded, showing a list of routes. A green arrow points to the route for the destination 0.0.0.0/0, which is associated with the gateway 'igw-03244fb9bd4e89ee1'.

Name	ID de subred	Estado	VPC	Bloquear el ...	CIDR IPv4	CIDR IPv6	ID de
seminario-subnet-public1-us-east-1a	subnet-05bfeb3f876a958da	Available	vpc-0cd97794c9f7334de semi...	Desactivado	10.0.0.0/20	-	-
-	subnet-05cb37566b8417c2b	Available	vpc-0028e078baa8f708 default	Desactivado	172.31.32.0/20	-	-
seminario-subnet-public2-us-east-1b	subnet-0c56f8da25a315bd	Available	vpc-0cd97794c9f7334de semi...	Desactivado	10.0.16.0/20	-	-
-	subnet-0265322eb9d187242	Available	vpc-0028e078baa8f708 default	Desactivado	172.31.16.0/20	-	-
-	subnet-05216a817784ea07	Available	vpc-0028e078baa8f708 default	Desactivado	172.31.64.0/20	-	-

Destino	Destino
10.0.0.0/16	local
0.0.0.0/0	igw-03244fb9bd4e89ee1

Ilustración 0.7

Configurar Grupos de Seguridad

Como parte de la configuración de red en AWS, se crearon grupos de seguridad (Security Groups) específicos para cada instancia EC2, con el objetivo de controlar el tráfico entrante hacia los servidores. Las reglas fueron definidas para permitir el acceso remoto desde la IP pública del alumno y el acceso web desde cualquier lugar del mundo, cumpliendo los requerimientos del proyecto.

Permitir:

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

Server1Windows – Grupo de seguridad sgwindowserver

Este grupo fue asignado a la instancia EC2 que ejecuta Windows Server 2016. Se establecieron dos reglas de entrada y una de salida:

sg-0d5a4c31090f9ced7 - sgwindowserver

sg-0d5a4c31090f9ced7 - sgwindowserver Acciones

Detalles

Nombre del grupo de seguridad sgwindowserver	ID del grupo de seguridad sg-0d5a4c31090f9ced7	Descripción launch-wizard-1 created 2025-06-18T23:54:25.006Z	ID de la VPC vpc-5cd97794c9f7334de
Propietario 863362166802	Número de reglas de entrada 2 Entradas de permisos	Número de reglas de salida 1 Entrada de permiso	

Reglas de entrada | Reglas de salida | Compartiendo : *novedad* | Asociaciones de VPC : *novedad* | Etiquetas

Reglas de entrada (1/2) Administrar etiquetas Editar reglas de entrada

<input type="checkbox"/>	Name	ID de la regla del gr...	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción
<input type="checkbox"/>	-	sgr-0ca6a84b479c5b0bf	IPv4	HTTP	TCP	80	0.0.0.0/0	Web
<input checked="" type="checkbox"/>	-	sgr-09b68ac671e9eebf	IPv4	RDP	TCP	3389	0.0.0.0/0	-

Ilustración 0.8

sg-0d5a4c31090f9ced7 - sgwindowserver

sg-0d5a4c31090f9ced7 - sgwindowserver Acciones

Detalles

Nombre del grupo de seguridad sgwindowserver	ID del grupo de seguridad sg-0d5a4c31090f9ced7	Descripción launch-wizard-1 created 2025-06-18T23:54:25.006Z	ID de la VPC vpc-5cd97794c9f7334de
Propietario 863362166802	Número de reglas de entrada 2 Entradas de permisos	Número de reglas de salida 1 Entrada de permiso	

Reglas de entrada | **Reglas de salida** | Compartiendo : *novedad* | Asociaciones de VPC : *novedad* | Etiquetas

Reglas de salida (1) Administrar etiquetas Editar reglas de salida

<input type="checkbox"/>	Name	ID de la regla del gr...	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Destino	Descripción
<input type="checkbox"/>	-	sgr-0fd4051fbbf445f5c	IPv4	Todo el tráfico	Todo	Todo	0.0.0.0/0	-

Ilustración 0.9

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

Server2Linux – Grupo de seguridad launch-wizard-1

Este grupo se asignó a la instancia Amazon Linux 2023. Las reglas de entrada y salida, fueron las siguientes:

sg-0cbaee52587683ccd - launch-wizard-1

sg-0cbaee52587683ccd - launch-wizard-1 Acciones

Detalles

Nombre del grupo de seguridad launch-wizard-1	ID del grupo de seguridad sg-0cbaee52587683ccd	Descripción launch-wizard-1 created 2025-06-27T02:58:48.975Z	ID de la VPC vpc-5cd97794c9f7334de
Propietario 863362166802	Número de reglas de entrada 2 Entradas de permisos	Número de reglas de salida 1 Entrada de permiso	

Reglas de entrada | Reglas de salida | Compartiendo : *novedad* | Asociaciones de VPC : *novedad* | Etiquetas

Reglas de entrada (1/2) Administrar etiquetas Editar reglas de entrada

<input checked="" type="checkbox"/>	Name	ID de la regla del gr...	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción
<input checked="" type="checkbox"/>	-	sgr-02396b98e77169378	IPv4	SSH	TCP	22	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-03ec762a46badb73a	IPv4	HTTP	TCP	80	0.0.0.0/0	-

Ilustración 0.10

sg-0cbaee52587683ccd - launch-wizard-1 Acciones

Detalles

Nombre del grupo de seguridad launch-wizard-1	ID del grupo de seguridad sg-0cbaee52587683ccd	Descripción launch-wizard-1 created 2025-06-27T02:58:48.975Z	ID de la VPC vpc-0cd97794c9f7334de
Propietario 863362166802	Número de reglas de entrada 2 Entradas de permisos	Número de reglas de salida 1 Entrada de permiso	

Reglas de entrada | **Reglas de salida** | Compartiendo : *novedad* | Asociaciones de VPC : *novedad* | Etiquetas

Reglas de salida (1) Administrar etiquetas Editar reglas de salida

Nombre	ID de la regla del gr...	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Destino	Descripción
-	sgr-0c626071e9fca36bc	IPv4	Todo el tráfico	Todo	Todo	0.0.0.0/0	-

Ilustración 0.11

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

Server1Windows

sg-0d5a4c31090f9ced7 - sgwindowserver Acciones

Detalles

Nombre del grupo de seguridad sgwindowserver	ID del grupo de seguridad sg-0d5a4c31090f9ced7	Descripción launch-wizard-1 created 2025-06-18T23:54:25.006Z	ID de la VPC vpc-0cd97794c9f7334de
Propietario 863362166802	Número de reglas de entrada 2 Entradas de permisos	Número de reglas de salida 1 Entrada de permiso	

Reglas de entrada | Reglas de salida | Compartiendo : *novedad* | Asociaciones de VPC : *novedad* | Etiquetas

Reglas de entrada (1/2) Administrar etiquetas Editar reglas de entrada

Nombre	ID de la regla del gr...	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción
-	sgr-0ca6a84b479c5b0bf	IPv4	HTTP	TCP	80	0.0.0.0/0	Web
-	sgr-09b68ac671e9eefbf	IPv4	RDP	TCP	3389	0.0.0.0/0	-

Ilustración 0.12

Server2Linux

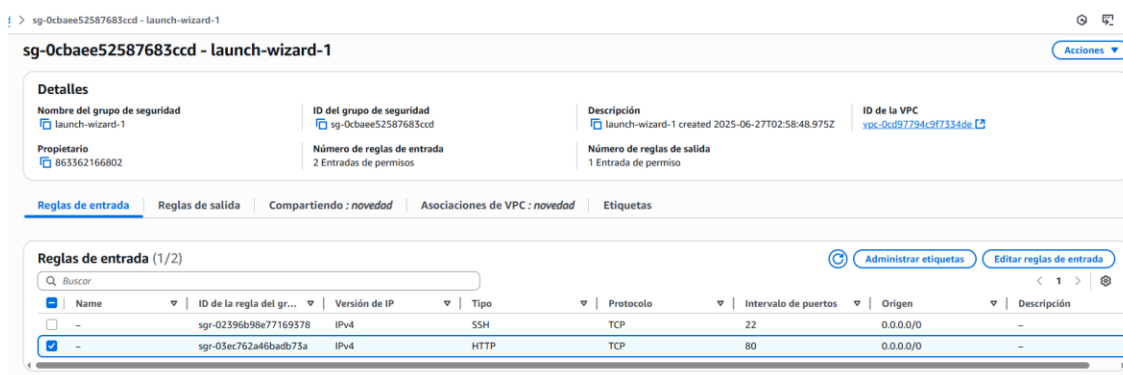


Ilustración 0.13

Restricción por IP en los puertos RDP y SSH reduce la exposición a ataques externos y garantiza que solo el administrador tenga acceso remoto a las instancias.

Puerto 80 abierto al mundo (0.0.0.0/0) necesario para probar los servidores web desde navegadores.

Acceder a las instancias

Una vez creadas las instancias y configurados los grupos de seguridad, se procedió a acceder a cada una de ellas utilizando los protocolos correspondientes: RDP para Windows y SSH para Linux. Estas conexiones remotas permiten al administrador realizar configuraciones internas, instalar servicios y validar el funcionamiento de la infraestructura desde el entorno del sistema operativo.

Acceder vía RDP a la instancia Windows.

Para acceder al servidor Windows (Server1 Windows), se utilizó el protocolo RDP (Remote Desktop Protocol) mediante el cliente de Escritorio Remoto de Windows.

En la consola de AWS, se seleccionó la instancia EC2 con Windows.

Se hizo clic en el botón “Conectar”.

En la pestaña “Cliente de escritorio remoto”, se descargó el archivo .rdp.

Se generó la contraseña del usuario administrador, cargando la clave privada (.pem) usada al crear la instancia.

Con el archivo .rdp abierto, se ingresó la contraseña generada y se estableció conexión con el servidor.

Una vez dentro del escritorio remoto, se pudo instalar y configurar IIS como servidor web.

Nota: Esto quedo documentado con el paso a paso en la primera parte de esta entrega.

Acceder vía SSH a la instancia Linux.

Para el servidor Linux (Server2Linux), se utilizó el protocolo SSH (Secure Shell), mediante la herramienta MobaXterm, que permite establecer conexiones SSH desde sistemas Windows.

Se abrió MobaXterm y se creó una nueva sesión de tipo SSH.

En el campo “Remote host” se ingresó la IP pública de la instancia (54.167.42.145).

En “Username” se indicó ec2-user, que es el usuario por defecto para Amazon Linux.

En la sección de configuración avanzada, se cargó la clave privada (.pem) correspondiente.

Se hizo clic en “OK” para iniciar la sesión.

Al conectarse, se obtuvo acceso completo a la terminal del servidor Linux, desde donde se pudieron ejecutar comandos de administración, actualización del sistema e instalación del servidor Apache.

Nota: Esto quedo documentado con el paso a paso en la primera parte de esta entrega.

Instalar y configurar los servidores web

Para validar el funcionamiento de las instancias y su accesibilidad desde Internet, se instalaron y configuraron servidores web en ambos entornos: IIS en Windows Server y Apache en Amazon Linux. El objetivo fue levantar el sitio por defecto de cada servidor y verificar su disponibilidad desde un navegador web externo.

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

Windows: Instalar IIS (Internet Information Services)

En la instancia EC2 con Windows Server 2016, se utilizó el Administrador del servidor para instalar el rol de servidor web (IIS), el cual es una solución integrada de Microsoft para publicar contenido HTTP.

Se accedió a la instancia mediante Escritorio Remoto (RDP).

En el escritorio, se abrió el Administrador del servidor.

Se hizo clic en “Agregar roles y características”.

En el asistente, se seleccionó:

Tipo de instalación: “Basada en características o roles”.

Servidor local como destino.

Rol: “Servidor Web (IIS)”.

Se dejaron las configuraciones por defecto y se continuó con la instalación.

Al finalizar, se abrió el navegador dentro del servidor y se accedió a <http://localhost> para comprobar que el sitio por defecto estaba activo.

Se verificó el acceso desde un navegador externo ingresando la IP pública de la instancia (<http://100.24.47.219>), mostrando correctamente la página por defecto de IIS.

Nota: Esto quedo documentado con el paso a paso en la primera parte de esta entrega.

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

Linux: Instalar Apache

En la instancia EC2 con Amazon Linux 2023, se instaló el servidor web Apache HTTP Server, una de las soluciones más utilizadas en entornos Linux.

Se accedió a la instancia mediante MobaXterm vía SSH.

Se ejecutó una actualización general del sistema: `sudo yum update -y`

Se instaló Apache con el comando: `sudo yum install httpd -y`

Se habilitó el servicio para que inicie automáticamente: `sudo systemctl enable httpd`.

Se inició el servicio: `sudo systemctl start httpd`

Se validó el estado del servicio: `sudo systemctl status httpd`

Para verificar la instalación, se ingresó la IP pública de la instancia (<http://54.167.42.145>) desde un navegador externo, y se mostró correctamente la página por defecto de Apache.

Nota: Esto quedo documentado con el paso a paso en la primera parte de esta entrega.

Resultado:

Ambas instancias presentan sus sitios web activos en el puerto 80.

Las páginas por defecto fueron accesibles desde cualquier navegador conectado a Internet.

Las pruebas confirman que las reglas del grupo de seguridad, el Internet Gateway y la subred pública fueron configuradas correctamente.

Pruebas de conectividad

Como parte de la validación de la red privada dentro de la VPC, se realizaron pruebas de conectividad entre las dos instancias EC2 creadas (una con Windows y otra con Linux), utilizando el comando `ping`. Esta prueba permite comprobar si las instancias pueden comunicarse entre sí mediante sus IP privadas, sin necesidad de salir a Internet.

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

Inicialmente, el comando no recibió respuesta. Luego de revisar la configuración, se determinó que el firewall de Windows estaba configurado con perfil de red pública, lo cual bloqueaba las solicitudes ICMP entrantes, a pesar de que la regla correspondiente estaba habilitada en el grupo de seguridad.

Cambiar el perfil del firewall de Windows

Se ingresó al Firewall de Windows con seguridad avanzada.

Se verificó que la regla “Echo Request - ICMPv4-In” estaba habilitada. Se habilita regla.

Se cambió el perfil de red de la instancia de público a privado, permitiendo que las reglas de ICMP se aplicaran correctamente.

Este ajuste fue necesario porque el firewall de Windows aplica reglas distintas según el tipo de red (pública o privada).

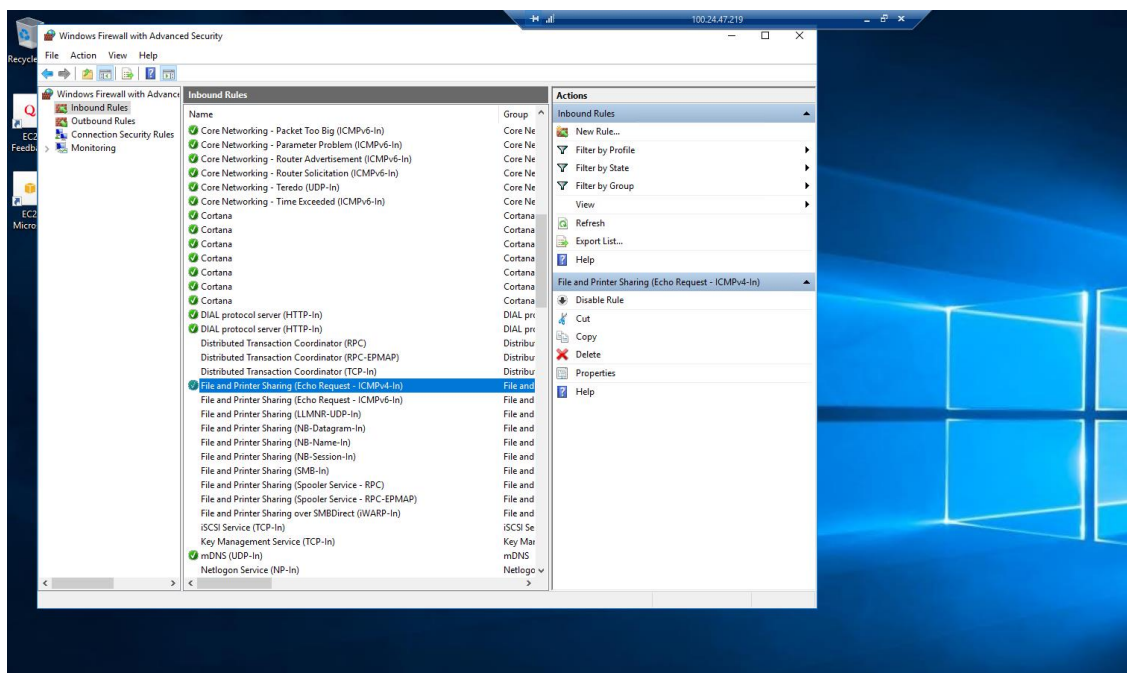


Ilustración 0.15

Ping de Linux a Windows

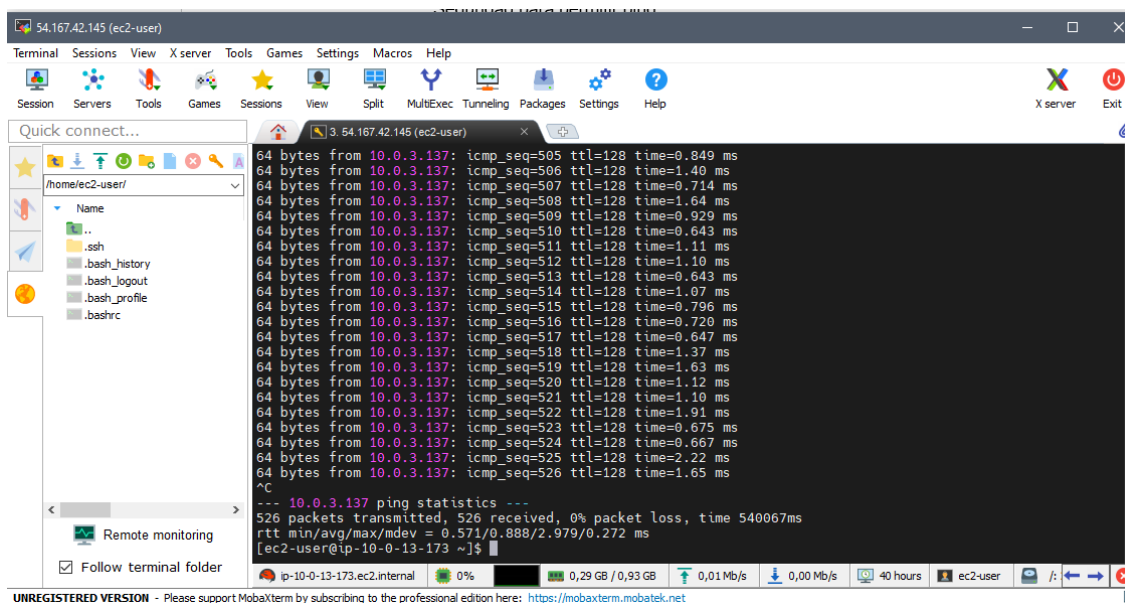


Ilustración 0.16

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

Configuración de ICMP en los Grupos de Seguridad

Para permitir los pings entre instancias, se agregó una regla adicional en ambos grupos de seguridad (sgwindowserver y launch-wizard-1):

Regla aplicada:

Versión de IP: IPv4

Tipo: Todos los ICMP IPv4

Protocolo: ICMP

Intervalo de puertos: Todo

Origen: 10.0.0.0/16

Windows: sgwindowserver

sg-0d5a4c31090f9ced7 - sgwindowserver Acciones ▾

Detalles

Nombre del grupo de seguridad sgwindowserver	ID del grupo de seguridad sg-0d5a4c31090f9ced7	Descripción launch-wizard-1 created 2025-06-18T23:54:25.006Z	ID de la VPC vpc-0cd97794c9f7334de
Propietario 863362166802	Número de reglas de entrada 3 Entradas de permisos	Número de reglas de salida 1 Entrada de permiso	

Reglas de entrada | Reglas de salida | Compartiendo : *novedad* | Asociaciones de VPC : *novedad* | Etiquetas

Reglas de entrada (1/3) Administrar etiquetas Editar reglas de entrada

<input type="checkbox"/>	Name	ID de la regla del gr...	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción
<input checked="" type="checkbox"/>	-	sgr-0fa83a7cf1cb1494d	IPv4	Todos los ICMP IPv4	ICMP	Todo	10.0.0.0/16	Ping entre instancias d...
<input type="checkbox"/>	-	sgr-0ca6a84b479c5b0bf	IPv4	HTTP	TCP	80	0.0.0.0/0	Web
<input type="checkbox"/>	-	sgr-09b68ac671e9eebf	IPv4	RDP	TCP	3389	0.0.0.0/0	-

Ilustración 0.17

Linux: launch-wizard-1

sg-0cbaee52587683ccd - launch-wizard-1 Acciones ▾

Detalles

Nombre del grupo de seguridad launch-wizard-1	ID del grupo de seguridad sg-0cbaee52587683ccd	Descripción launch-wizard-1 created 2025-06-27T02:58:48.975Z	ID de la VPC vpc-0cd97794c9f7334de
Propietario 863362166802	Número de reglas de entrada 3 Entradas de permisos	Número de reglas de salida 1 Entrada de permiso	

Reglas de entrada | Reglas de salida | Compartiendo : *novedad* | Asociaciones de VPC : *novedad* | Etiquetas

Reglas de entrada (1/3) Administrar etiquetas Editar reglas de entrada

<input type="checkbox"/>	Name	ID de la regla del gr...	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción
<input type="checkbox"/>	-	sgr-02396b98e77169378	IPv4	SSH	TCP	22	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-03ec762a46badb73a	IPv4	HTTP	TCP	80	0.0.0.0/0	-
<input checked="" type="checkbox"/>	-	sgr-0dd2e22ee85dd4f91	IPv4	Todos los ICMP IPv4	ICMP	Todo	10.0.0.0/16	Ping entre instancias d...

Ilustración 0.18

10.0.0.0/16 abarca todo el rango de IP privadas de la VPC, lo que permite ICMP entre todas las instancias internas.

Validación de acceso web

Una vez instalados y configurados los servidores web en ambas instancias EC2, se procedió a validar su accesibilidad desde un navegador web externo. Para ello, se utilizó el navegador local del equipo del administrador, ingresando directamente las direcciones IP públicas de las instancias en la barra de direcciones, utilizando el protocolo HTTP por el puerto 80.

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

Acceso al sitio web en la instancia Windows

La instancia Server1 Windows fue configurada con el rol de IIS (Internet Information Services).

En el navegador, se ingresó la siguiente URL: <http://100.24.47.219>

Al cargar la dirección, el navegador mostró la página de bienvenida por defecto de IIS.

Resultado: acceso exitoso al sitio web en Windows desde el navegador local.

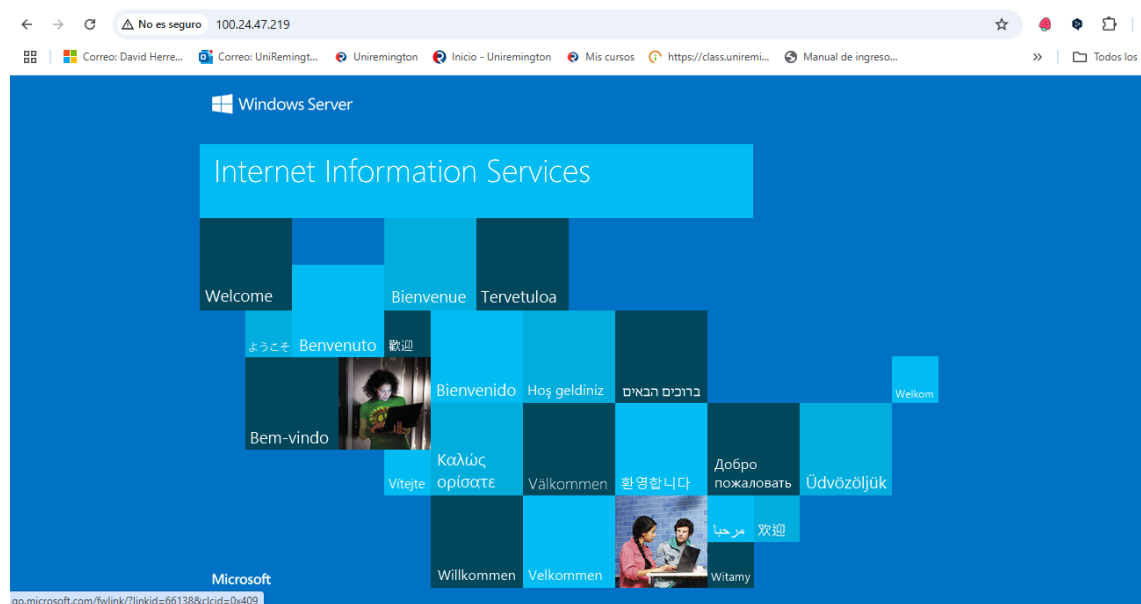


Ilustración 0.19

Acceder desde el navegador local al sitio web de la instancia Linux

(http://<IP_Pública_Linux>).

Acceso al sitio web en la instancia Linux

La instancia Server2Linux fue configurada con el servidor web Apache HTTP Server.

En el navegador local, se accedió a la siguiente URL: <http://54.167.42.145>

El navegador respondió con la página por defecto de Apache

Resultado: El navegador respondió con la página por defecto de Apache.

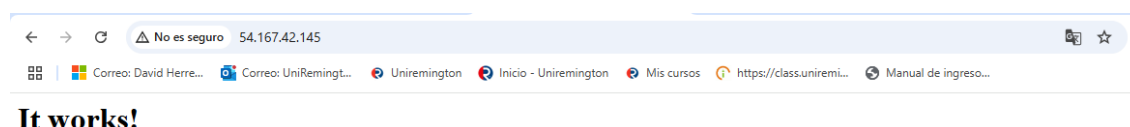


Ilustración 0.20

Conclusión:

El proyecto permitió desplegar una red funcional en AWS de arquitectura en la nube. Se configuraron correctamente las instancias EC2, el acceso remoto seguro, y los servicios web en Windows y Linux. La conectividad interna fue validada mediante pruebas ICMP, y el acceso externo por HTTP demostró que las configuraciones de red (VPC, subred pública, IGW y grupos de seguridad) fueron correctamente implementadas. Esta experiencia fortaleció el entendimiento práctico sobre servicios de red en la nube y seguridad de infraestructura.

ENTREGA 2

IMPLEMENTACIÓN de ARQUITECTURA en AWS CON BALANCEADOR de CARGA y CONTENEDORES

Introducción: La startup

Bienvenidos al proyecto de implementación. En esta tarea, ustedes asumirán el rol de arquitectos en la nube para un startup llamado **Quality Run**, 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 **Quality Run** a 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.

Antes de iniciar con el objetivo general de la segunda entrega, Se lanzo/creo una segunda instancia con un IAM de Linux 2023, que es la misma que tiene la instancia de Linux que se creó en la primera entrega, esta nueva instancia tiene los siguientes detalles:

Instancia 2, Server 2 Linux

Sistema operativo: *Linux*

IP pública: 13.217.25.177

IP privada: 10.0.26.111

Subred: subnet-0656fc8da25a315bd (seminario-subnet-public2-us-east-1b) se usó la sub red us-east-1b ya que la us-east-1a está configurada para el servidor uno de Linux

Grupo de seguridad: sg-074f1d11dca7285f3 (launch-wizard-2)

Puertos habilitados: SSH (22), HTTP (80)

Según lo que puede averiguar y no es estrictamente necesario, pero si es recomendado es, desinstalar Apache para evitar conflictos en el puerto 80 y permitir el correcto funcionamiento de Nginx como proxy reverso. Apache está instalado en el Server2Linux, IP pública: 54.167.42.145.

Proceso para desinstalar Apache en el Server2Linux, IP pública: 54.167.42.145, mediante comandos, se ejecutaron los siguientes comandos en el siguiente orden:

```
sudo systemctl stop httpd
```

```
sudo systemctl disable httpd
```

```
sudo yum remove httpd -y
```

Esto permite detener Apache, deshabilitarlo y removerlo.

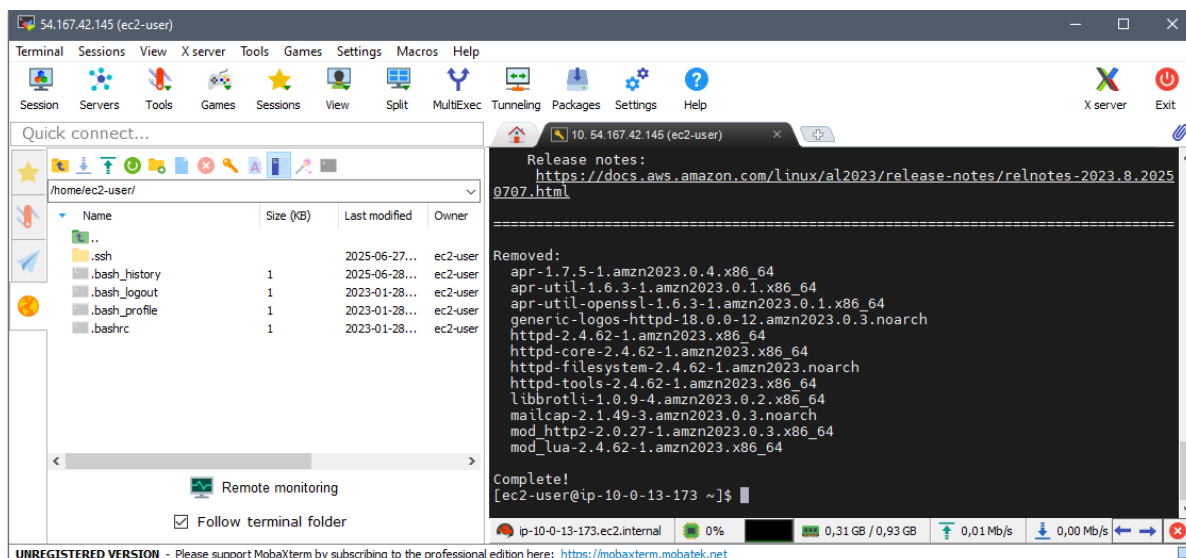


Ilustración 0.1

Verifico si se desinstalo con el siguiente con el siguiente comando:

```
sudo systemctl status httpd
```

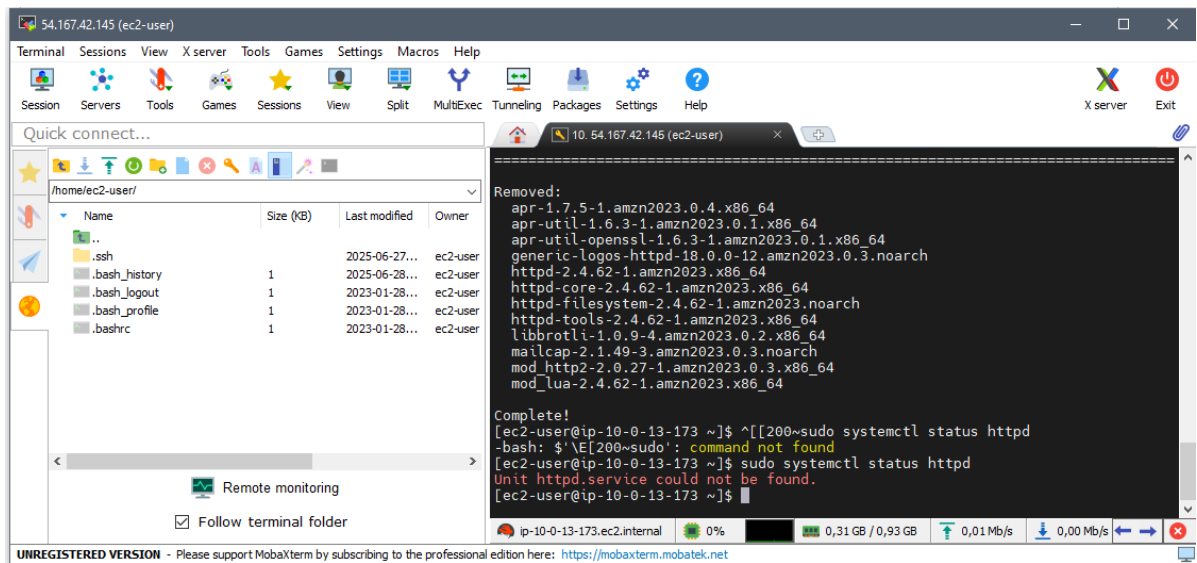


Ilustración 0.2

Foro de stackover flow

<https://stackoverflow.com/questions/14972792/nginx-nginx-emerg-bind-to-80-failed-98-address-already-in-use>

Instalación de Docker en ambas instancias, Server2Linux y Server3Linux.

Se ejecutan los siguientes comandos en orden en ambas instancias:

sudo yum update -y

sudo yum install docker -y

sudo systemctl enable docker

sudo systemctl start Docker

Valido en ambas instancias que Docker quede instalado, con el siguiente comando:

sudo docker info

Server2Linux

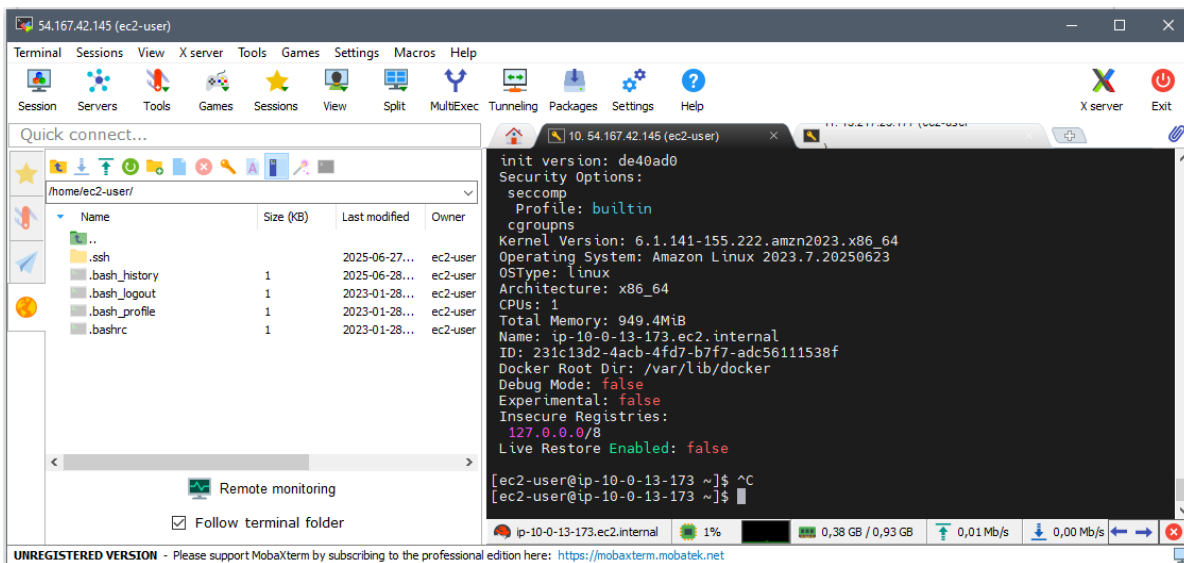


Ilustración 0.3

Server3Linux

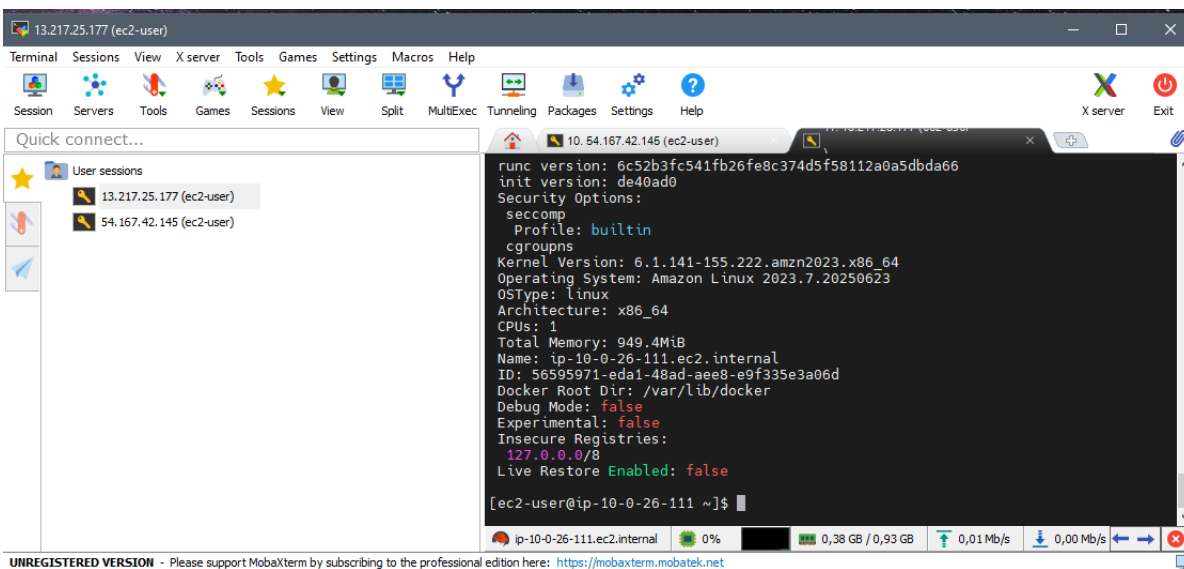


Ilustración 0.4

Creo 1 carpeta por cada instancia y dentro de cada instancia.

Con el siguiente comando:

`mkdir ~/miappseminarioS2` y `mkdir ~/miappseminarioS3` dependiendo cada instancia.

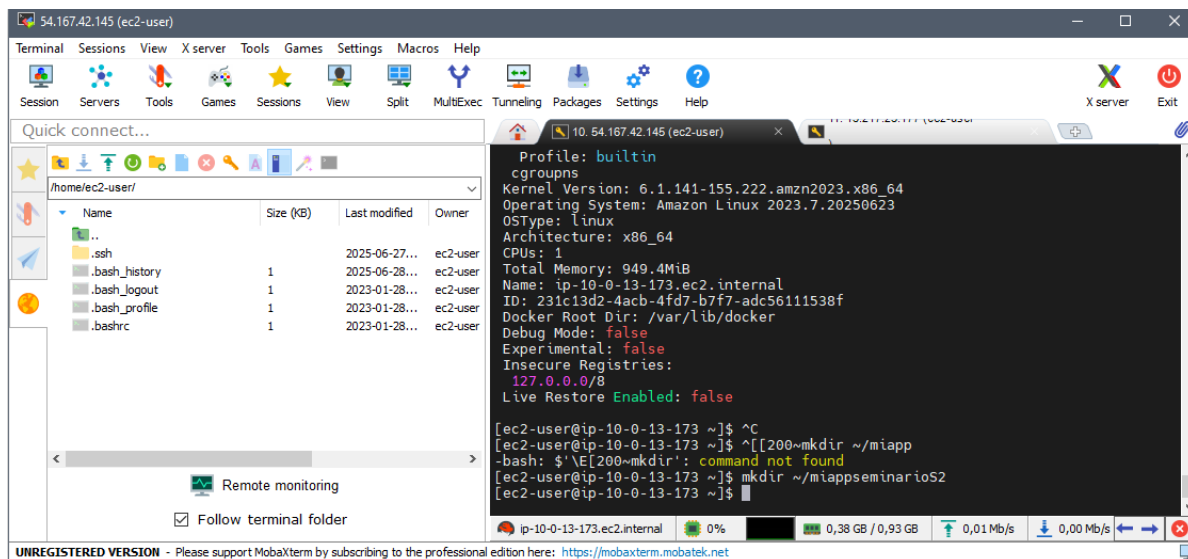


Ilustración 0.5

Posterior a ello, creo un index.html dentro de la carpeta.

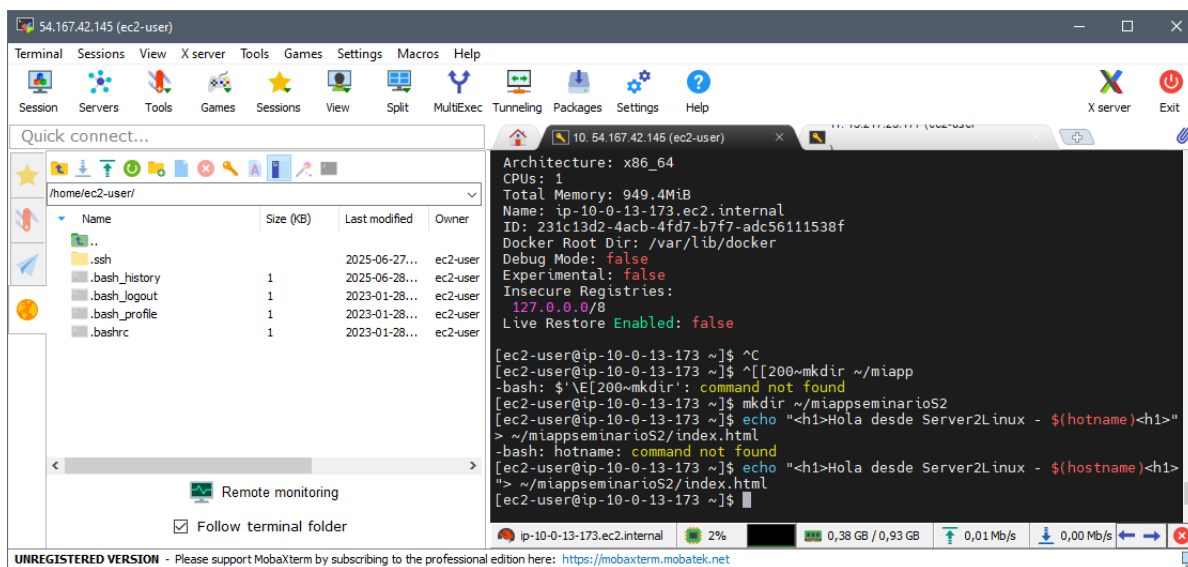


Ilustración 0.6

Paso siguiente es crear un archivo Docker dentro de la carpeta.

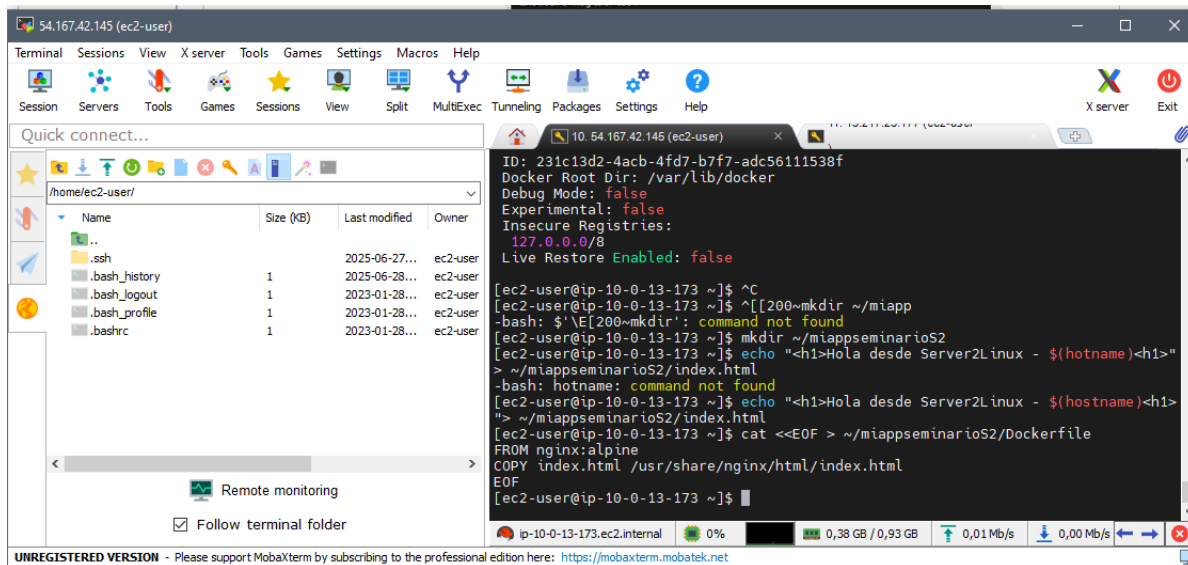


Ilustración 0.7

Continuamos con la construcción de la imagen docker y la ejecución:

Acceso a la carpeta:

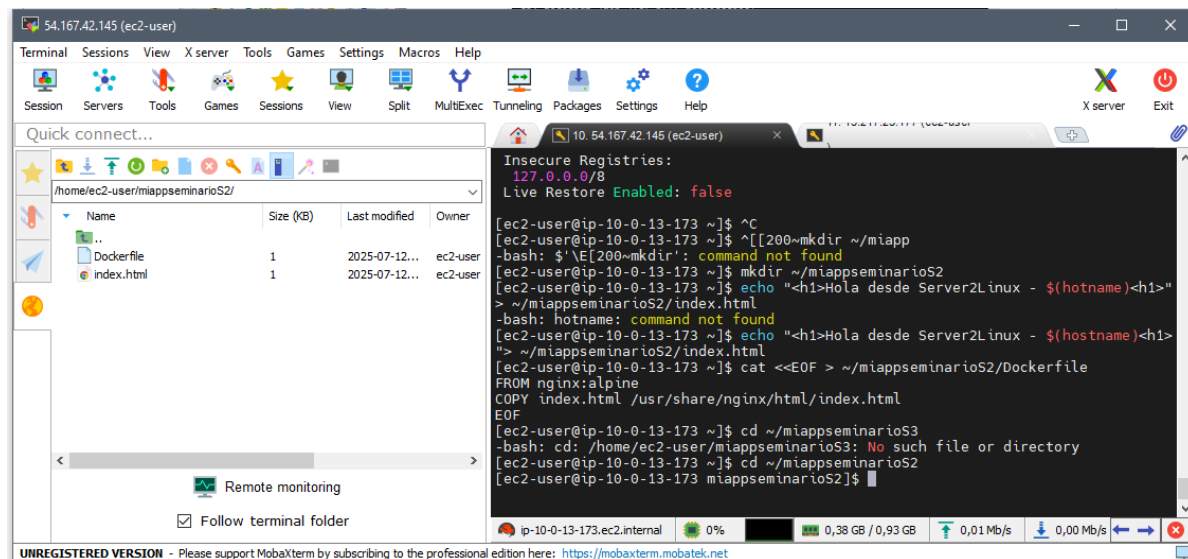


Ilustración 0.8

Construcción del archivo Docker, con el siguiente comando: `sudo docker build -t miapp .`

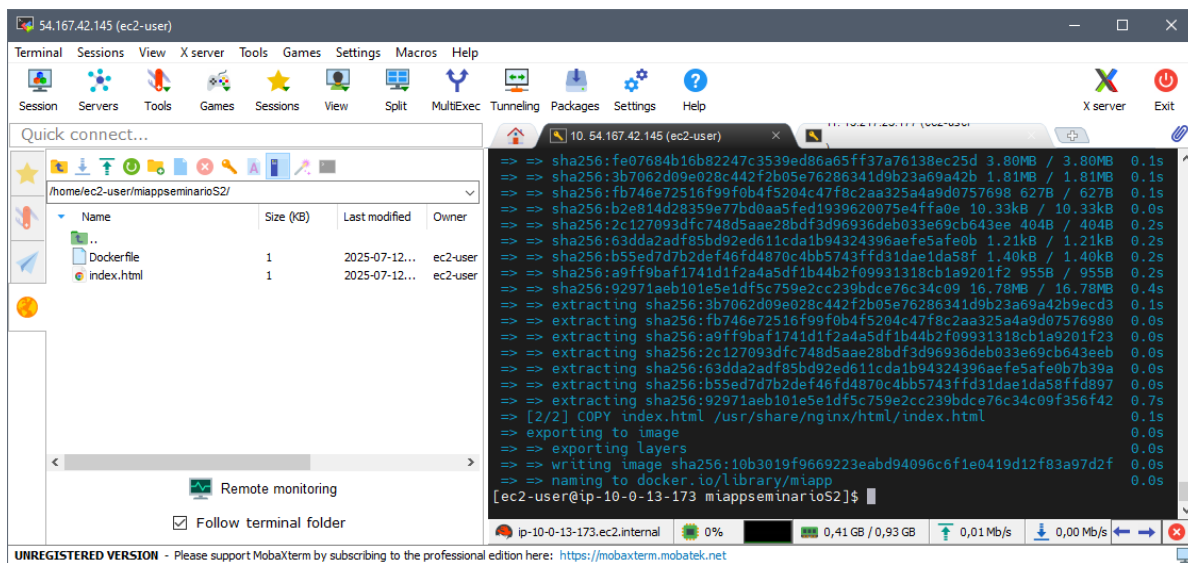


Ilustración 0.9

Ejecución:

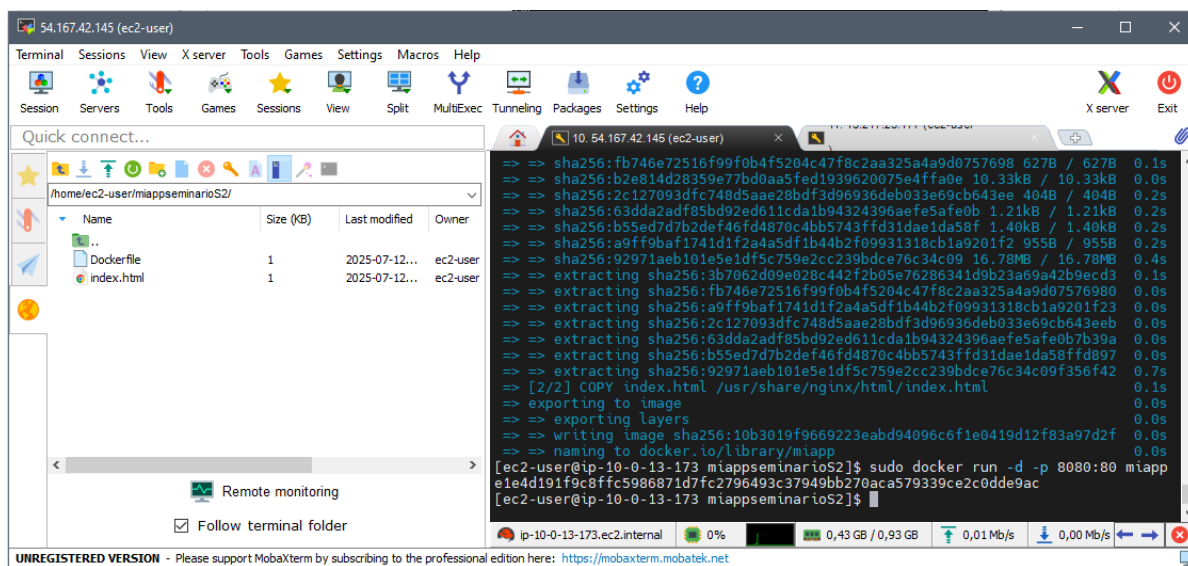


Ilustración 0.10

Verificar que los contenedores este ejecutándose:

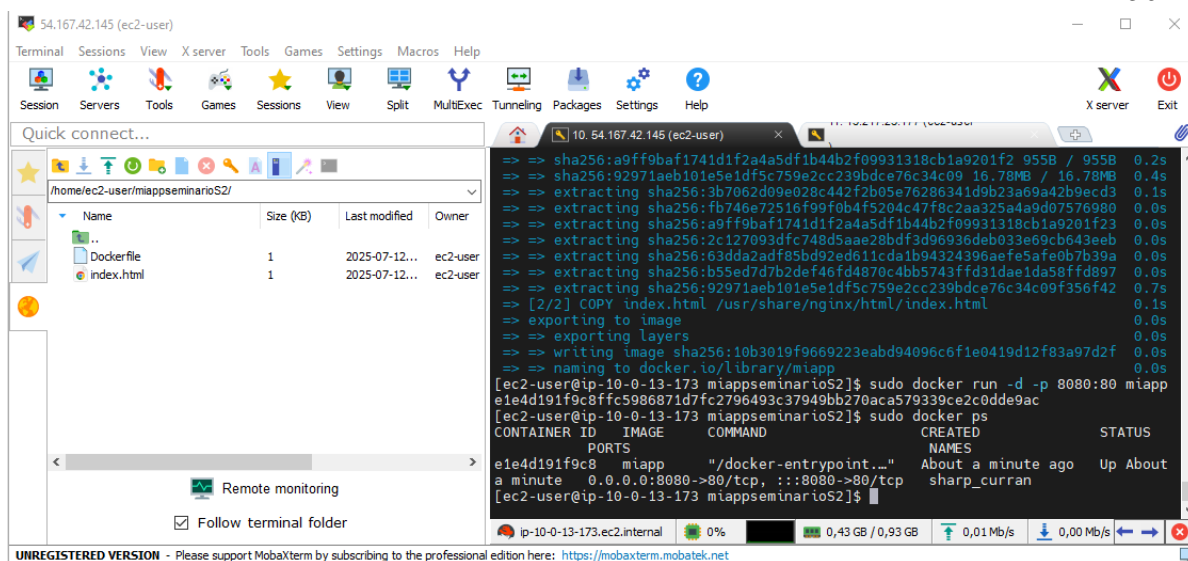


Ilustración 0.11

Instalación y configuración NGINX

Para la instalación ejecuto los siguientes comandos en orden:

```
sudo yum install nginx -y
```

```
sudo systemctl enable nginx
```

```
sudo systemctl start nginx
```

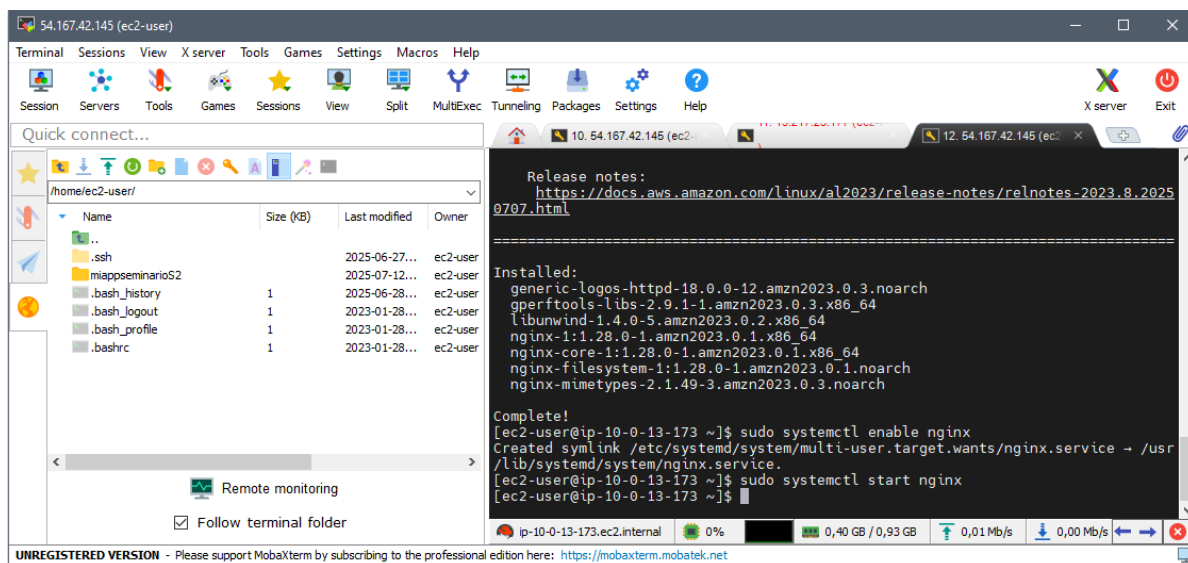


Ilustración 0.12

Editar configuración para el proxy reverso:

Ejecuto el siguiente comando: `sudo nano /etc/nginx/nginx.conf`, para acceder a la ruta de la configuración de nginx

Se agrega la locación del proxy reverso con esta línea:

```
location / {
    proxy_pass http://localhost:8080;
}
```

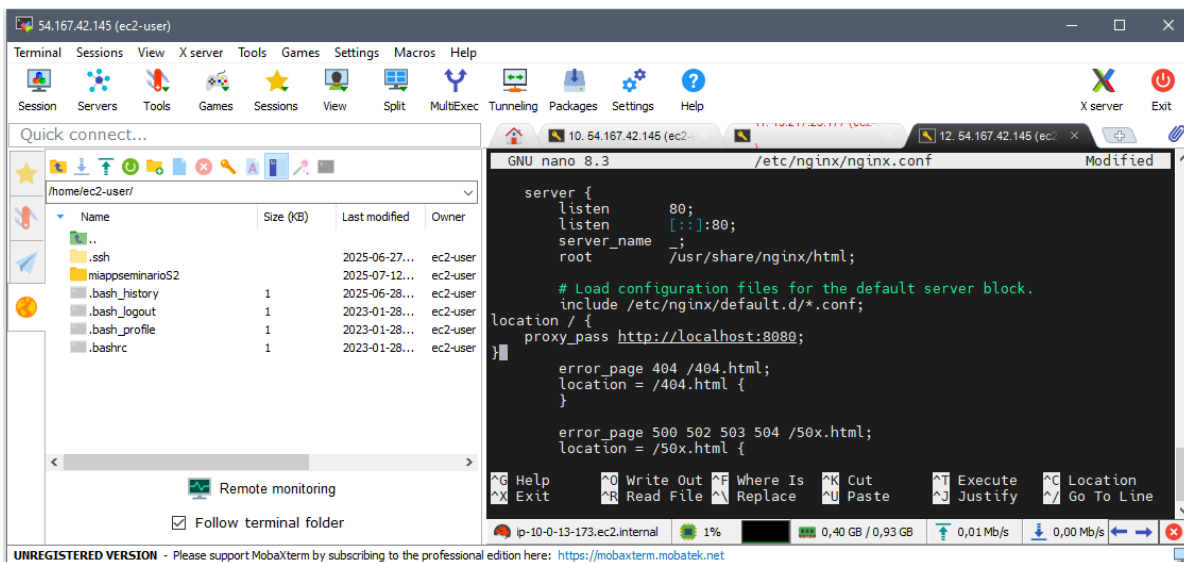


Ilustración 0.13

Guardo la configuración y salgo del .config.

Reinicio NGINX

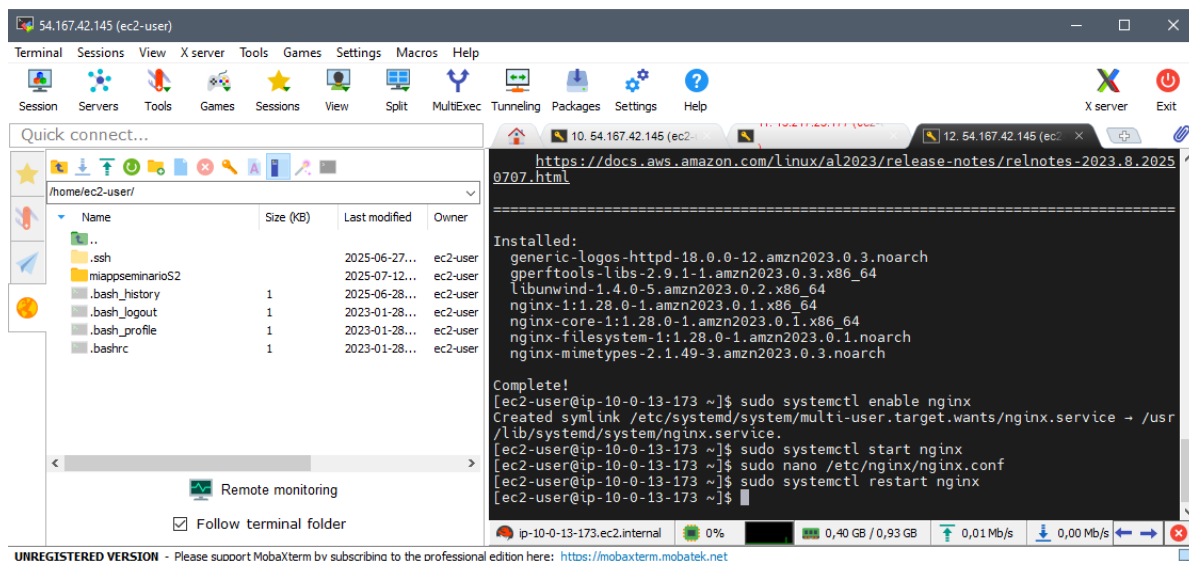


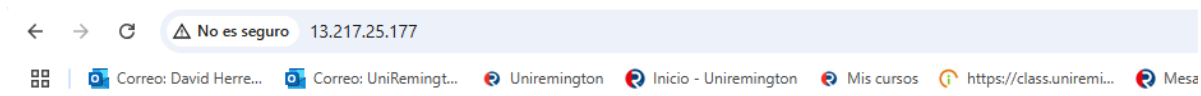
Ilustración 0.14

Nota: Recordar que esta configuración se realiza en ambas instancias, solo que en las imágenes de los resultados esta solo el Server2Linux, para una mejor comprensión.

Verifico la IP pública del Server2Linux/ Server3Linux en mi explorador y ya puedo ver el index.html



Ilustración 0.15



Hola desde Server3Linux - ip-10-0-26-111.ec2.internal

Ilustración 0.16

Crear Application Load Balancer (ALB) para distribuir el tráfico entre ambas instancias y acceder desde un DNS público.

Creación grupo destino

Ingresar a EC2, ir a grupos de destino y dará clic en el botón crear grupo destino.

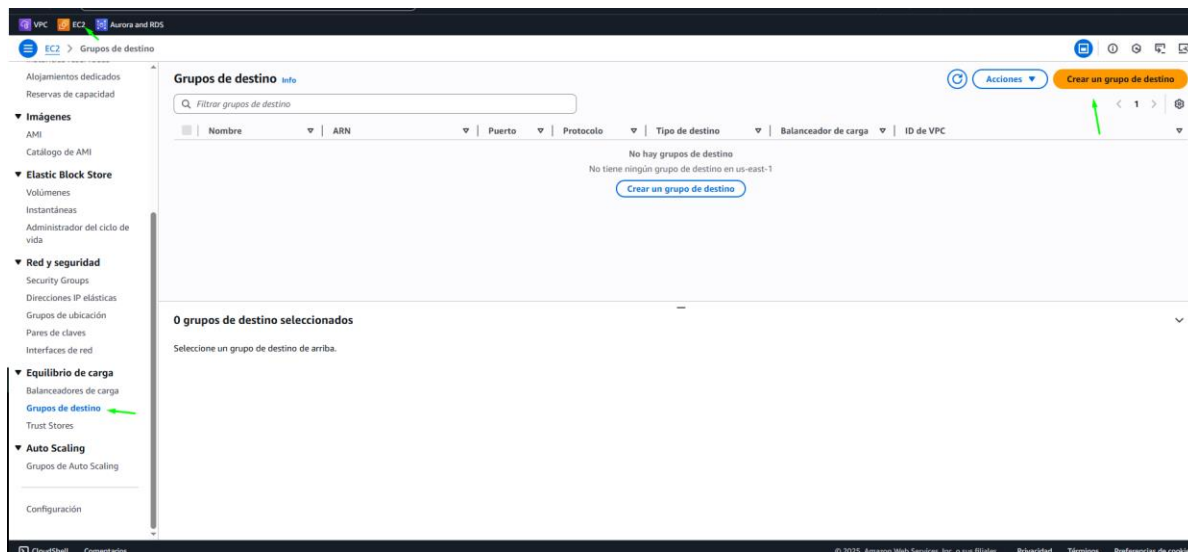


Ilustración 0.17

En el paso 1 se debe realizar la siguiente configuración:

Seleccionar instancias.

Nombre del grupo de destino: QualityRun.

Protocolo debe ser HTTP.

Puerto 80

VPC: seminarioVPC.

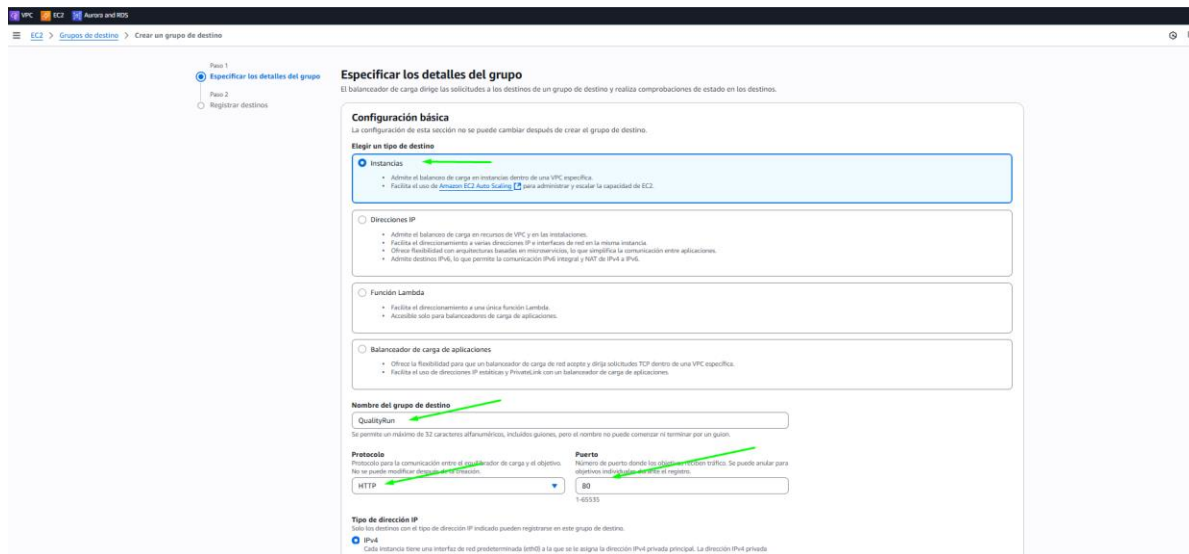


Ilustración 0.18

En Comprobaciones de estado:

Protocolo HTTP.

Ruta de comprobación de estado: / (Ambas vienen por defecto).

Dar clic en siguiente:

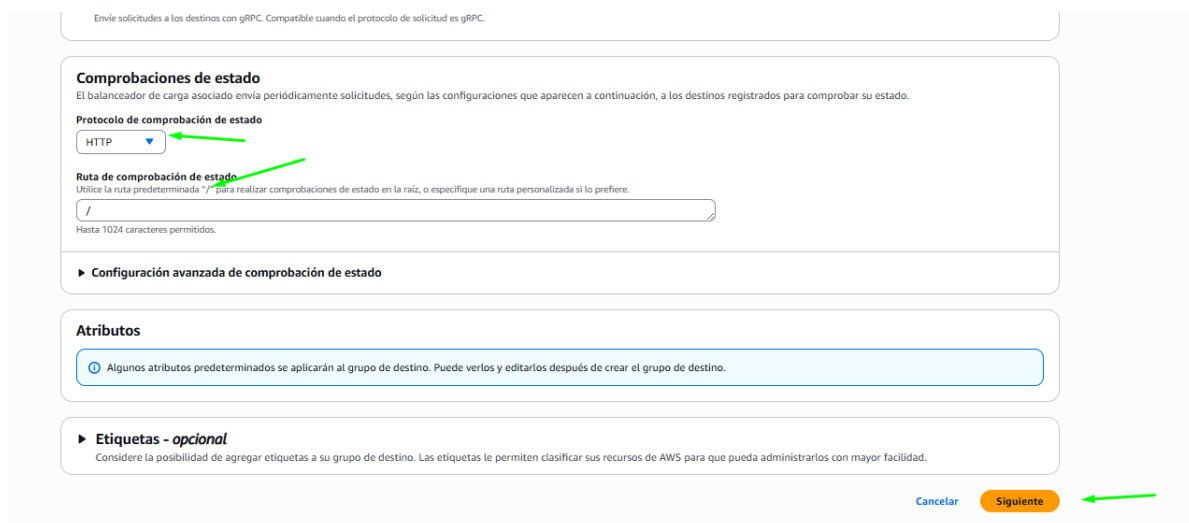


Ilustración 0.19

En registrar destinos seleccionamos las dos instancias previamente configuradas en Linux, damos clic en el botón incluir como pendientes a continuación y por último dar clic en crear grupo de destino.

Registrar destinos
Se trata de un paso opcional para crear un grupo de destino. Sin embargo, para asegurarse de que el balanceador de carga dirige el tráfico a este grupo de destino, debe registrar los destinos.

Instancias disponibles (3)

ID de instancia	Nombre	Estado	Grupos de seguridad	Zona	Dirección IPv4 privada	ID de subred	Hora de lanzamiento
<input type="checkbox"/> i-0b780e787e23016fe	Server3Linux	● Ejecutando	launch-wizard-2	us-east-1b	10.0.26.111	subnet-0656f68da25a315bd	10 de julio de 2025, 19:52 (UTC-05:00)
<input type="checkbox"/> i-0c0b4a442a929ed0a	Server2Linux	● Ejecutando	launch-wizard-1	us-east-1a	10.0.13.173	subnet-059feb3f876a958da	26 de junio de 2025, 22:04 (UTC-05:00)
<input type="checkbox"/> i-0bc79f91d2993d9fa	Server1Windows	● Ejecutando	sgwindowserver	us-east-1a	10.0.3.137	subnet-059feb3f876a958da	18 de junio de 2025, 19:17 (UTC-05:00)

0 seleccionados

Puertos para las instancias seleccionadas
Puertos para dirigir el tráfico a las instancias seleccionadas.
1-65535 (separe puertos múltiples con coma)

Tiene 2 selecciones pendientes a continuación. Incluya más a registrar los destinos cuando estén listas.

Revisar destinos

Destinos (2)

ID de instancia	Nombre	Puerto	Estado	Grupos de seguridad	Zona	Dirección IPv4 privada	ID de subred	Hora de lanzamiento
i-0b780e787e23016fe	Server3Linux	80	● Ejecutando	launch-wizard-2	us-east-1b	10.0.26.111	subnet-0656f68da25a315bd	10 de julio de 2025, 19:52 (UTC-05:00)
i-0c0b4a442a929ed0a	Server2Linux	80	● Ejecutando	launch-wizard-1	us-east-1a	10.0.13.173	subnet-059feb3f876a958da	26 de junio de 2025, 22:04 (UTC-05:00)

2 pendientes

Ilustración 0.20

QualityRun Acciones

Detalles
arn:aws:elasticloadbalancing:us-east-1:863362166802:targetgroup/QualityRun/G328ebaad6f2f7f9

Tipo de destino Instancia	Protocolo : Puerto HTTP: 80	Versión del protocolo HTTP1	VPC vpc-0cd97794c9f7334de
Tipo de dirección IP IPv4	Balanceador de carga Ninguno asociado		

2 Destinos totales	● 0 En buen estado	● 0 En mal estado	● 2 Sin utilizar	● 0 Inicial	● 0 Vaciado
-----------------------	--	---	---	--	--

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) Mitigación de anomalías: No aplicable

Los grupos de destinos enrutran 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.

ID de instancia	Nombre	Puerto	Zona	Estado	Detalles del estado	Sustitución administrativa
<input type="checkbox"/> i-0b780e787e23016fe	Server3Linux	80	us-east-1b (use1-a26)	● Unused	Target group is not co...	-
<input type="checkbox"/> i-0c0b4a442a929ed0a	Server2Linux	80	us-east-1a (use1-a24)	● Unused	Target group is not co...	-

Ilustración 0.21

El objetivo de esta configuración fue crear el grupo de destinos que nos sirve como puente entre el ALB y las instancias. Sin él, el balanceador no sabe a qué instancias redirige el tráfico.

Creacion Application Load Balancer (ALB)

Ir a EC2, ingresar a balanceadores de carga, crear balanceador de carga de aplicaciones.

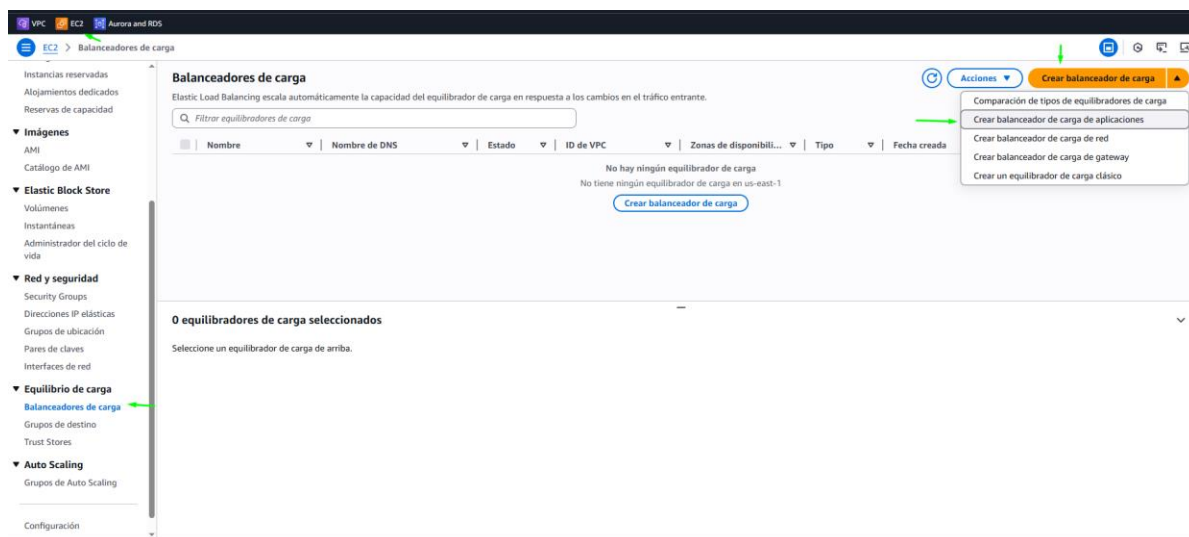


Ilustración 0.22

Allí debe se debe diligenciar, nombre del balanceador, el esquema es expuesto a internet, IPV4, el VPC es el que venimos manejando que es seminario-vpc

Configuración básica

Nombre del balanceador de carga
Debe ser nombre único dentro de su cuenta de AWS y no puede cambiarse después de crear el equilibrador de carga.
bal-QualityRun
Se permite un máximo de 32 caracteres alfanuméricos, incluidos guiones, pero el nombre no puede comenzar ni terminar por un guion.

Esquema | Info
El esquema no se puede cambiar después de crear el equilibrador de carga.

Expuesto a Internet

- Suministra el tráfico expuesto a Internet.
- Tiene direcciones IP públicas.
- El nombre DNS se resuelve en direcciones IP públicas.
- Requiere una subred pública.

Interno

- Suministra el tráfico interno.
- Tiene direcciones IP privadas.
- El nombre DNS se resuelve en direcciones IP privadas.
- Compatible con los tipos de direcciones IP IPv4 y Dualstack.

Tipo de dirección IP del equilibrador de carga | Info
Seleccione el tipo de dirección IP de frontend que desea asignar al equilibrador de carga. La VPC y las subredes asignadas a este equilibrador de carga deben incluir los tipos de direcciones IP seleccionados. Las direcciones IPv4 públicas tienen un costo adicional.

IPv4
Incluye solo direcciones IPv4.

Dualstack
Incluye direcciones IPv4 e IPv6.

Dualstack sin IPv4 pública
Incluye una dirección IPv6 pública y direcciones IPv4 e IPv6 privadas. Compatible solo con equilibradores de carga expuestos a Internet.

Mapeo de red | Info
El balanceador de carga dirige el tráfico a los destinos de las subredes seleccionadas y en función de la configuración de las direcciones IP.

VPC | Info
El equilibrador de carga existirá y escalará dentro de la VPC seleccionada. La VPC seleccionada también es el lugar donde se tienen que alojar los destinos del equilibrador de carga, a menos que se dirijan a destinos de Lambda o locales, o si se utiliza la interconexión de VPC. Para confirmar la VPC para sus objetivos, consulte [los grupos de destino](#). Para una VPC nueva, [cree una VPC](#).

seminario-vpc
vpc-0cd87794c9f7334de
CIDR de VPC IPv4: 10.0.0.0/16

Ilustración 0.23

En las zonas de disponibilidad seleccionar las dos zonas donde se encuentra cada instancia con la subred correspondiente.

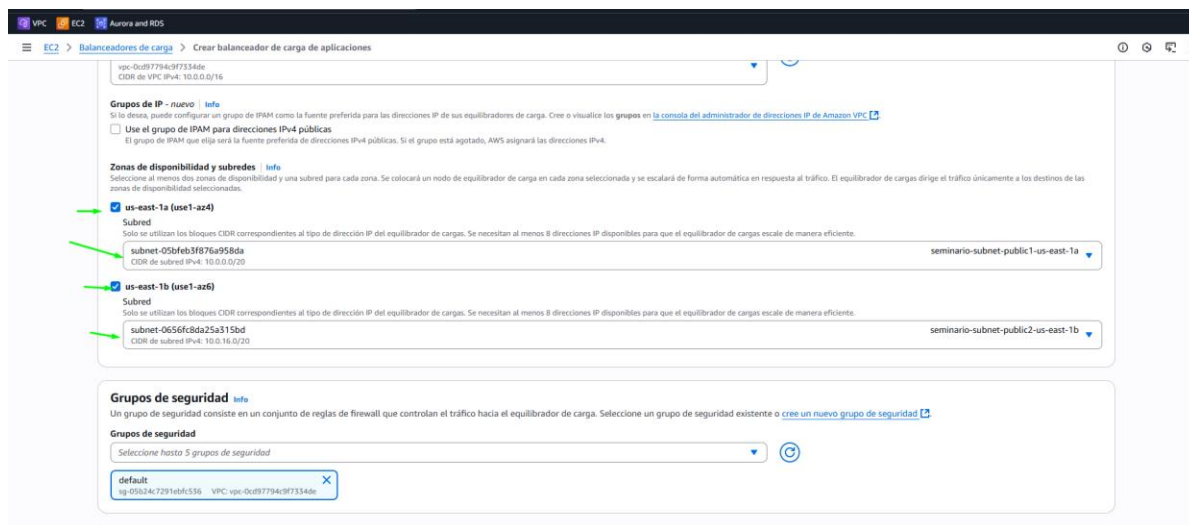


Ilustración 0.24

En los grupos de seguridad seleccione uno que ya tenía creado y tiene el puerto 80 abierto, en los Agentes de escucha y direccionamiento, ya venía por defecto HTTP:80

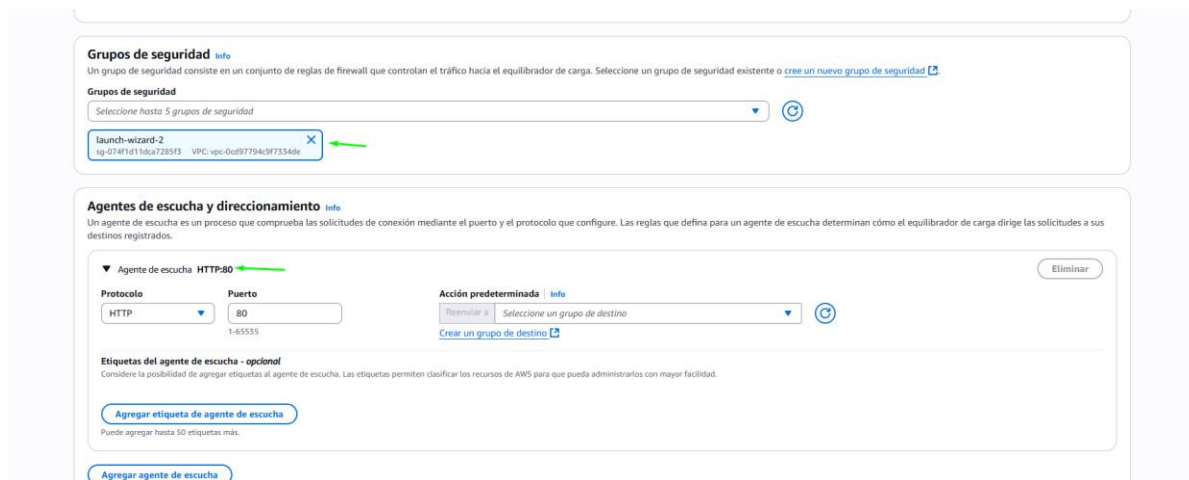


Ilustración 0.25

En la misma sección de Agentes de escucha y direccionamiento, seleccionar el grupo destino que se está construyendo en este caso **QualityRun**.

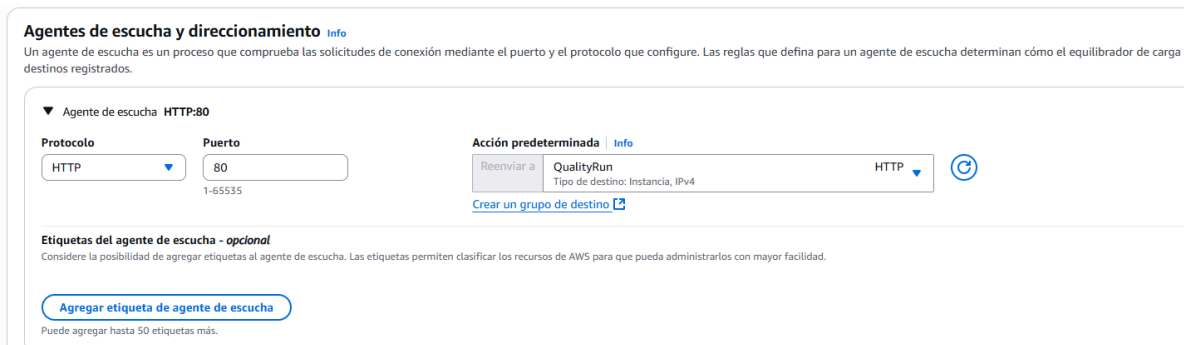


Ilustración 0.26

Ahora dar clic en el botón de crear balanceador de carga.

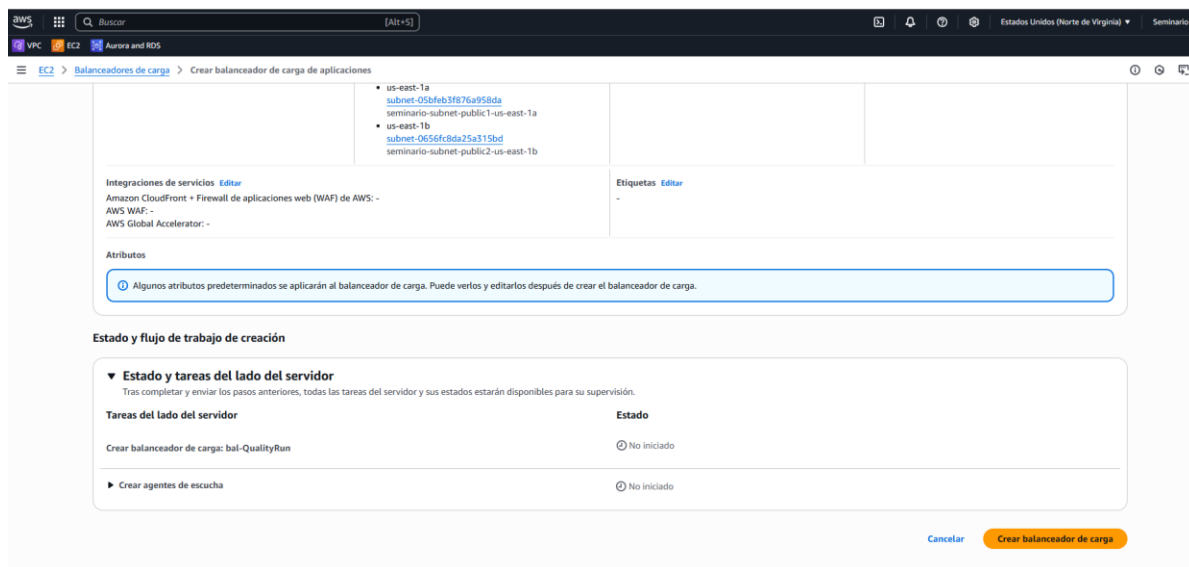


Ilustración 0.27

Luego de un rato el balanceador ya se encuentra en estado activo, y al ingresar mediante la DNS veo que se carga o se muestra los índices de la instancia 1 y 2.

Balanceadores de carga (1/1)

Elastic Load Balancing escala automáticamente la capacidad del equilibrador de carga en respuesta a los cambios en el tráfico entrante.

Q Filtrar equilibradores de carga

Nombre	Nombre de DNS	Estado	ID de VPC	Zonas de disponibili...	Tipo	Fecha creada
bal-QualityRun	bal-QualityRun-90513969....	Activo	vpc-0cd97794c9f7334de	2 Zonas de disponibilidad	application	12 de julio de 2025, 17:42 (UTC-05:00)

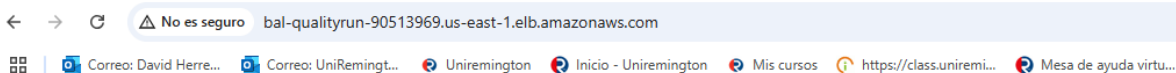
Equilibrador de carga: bal-QualityRun

Detalles

Tipo de equilibrador de carga Aplicación	Estado Activo	VPC vpc-0cd97794c9f7334de	Tipo de dirección IP del equilibrador de carga IPv4
Esquema Internet-facing	Zona hospedada Z355XD0TRQ7X7K	Zonas de disponibilidad subnet-05bfeb3f876a958da us-east-1a (use1-az4) subnet-0656fc8da25a515bd us-east-1b (use1-az6)	Fecha creada 12 de julio de 2025, 17:42 (UTC-05:00)
ARN del equilibrador de carga arn:aws:elasticloadbalancing:us-east-1:863362166802:loadbalancer/app/bal-QualityRun/f39f26b510b47c6	Nombre de DNS Info bal-QualityRun-90513969.us-east-1.elb.amazonaws.com (Registro A)		

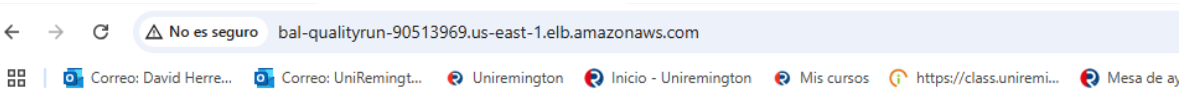
Ilustración 0.28

DNS: <http://bal-qualityrun-90513969.us-east-1.elb.amazonaws.com/>



Hola desde Server2Linux - ip-10-0-13-173.ec2.internal

Ilustración 0.29



Hola desde Server3Linux - ip-10-0-26-111.ec2.internal

Ilustración 0.30

Creación de auto escalable

Creación de plantilla de lanzamiento

Ingresar a EC2, ir a plantillas de lanzamiento, y dar clic en crear plantilla.



Ilustración 0.31

Diligenciar nombre y descripción de la plantilla

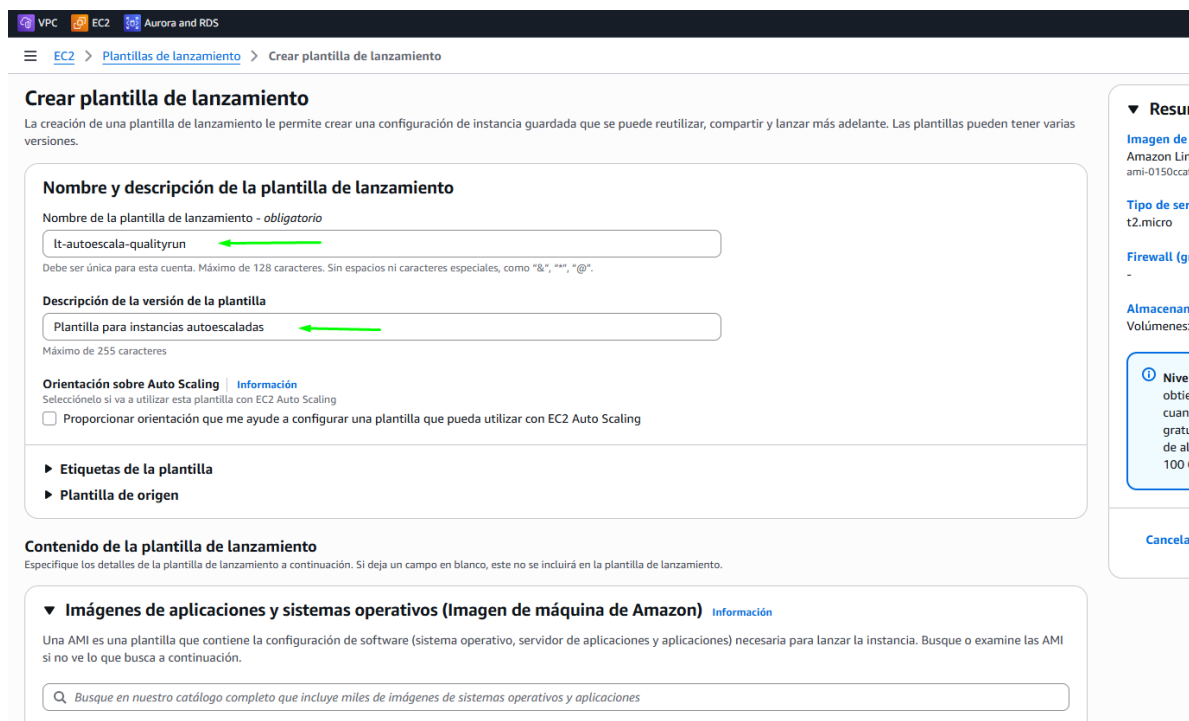


Ilustración 0.32

Seleccionar AMI, en este caso usamos la misma de Linux, tipo de instancia t2.micro, el par de claves que venimos usando dese la primera entrega y el grupo de seguridad que ya se había creado para Server3Linux.

Recientes | **Inicio rápido**

No incluir en la plantilla de lanzamiento | Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | Debian

Buscar más AMI
Inclusión de AMI de AWS, Marketplace y la comunidad

Imágenes de máquina de Amazon (AMI)

AMI de Amazon Linux 2023 kernel-6.1
ami-0150ccaf51ab55a51 (64 bits (x86), uefi-preferred) / ami-0cd4eb0ae8defb650 (64 bits (Arm), uefi)
Virtualización: hvm Activado para ENA: true Tipo de dispositivo raíz: ebs Apto para la capa gratuita

Descripción
Amazon Linux 2023 (kernel-6.1) es un sistema operativo moderno y de uso general basado en Linux que incluye 5 años de soporte a largo plazo. Está optimizado para AWS y diseñado para proporcionar un entorno de ejecución seguro, estable y de alto desempeño para desarrollar y ejecutar sus aplicaciones en la nube.

Amazon Linux 2023 AMI 2023.8.20250707.0 x86_64 HVM kernel-6.1

Arquitectura	Modo de arranque	ID de AMI	Fecha de publicación	Nombre de usuario
64 bits (x86)	uefi-preferred	ami-0150ccaf51ab55a51	2025-07-08	ec2-user

▼ Tipo de instancia Información | Obtener asesoramiento Avanzado

Tipo de instancia

t2.micro Apto para la capa gratuita

Familia: t2 1 vCPU 1 GiB Memoria Generación actual: true Bajo demanda Windows base precios: 0.0162 USD por hora
Bajo demanda Ubuntu Pro base precios: 0.0134 USD por hora Bajo demanda SUSE base precios: 0.0116 USD por hora
Bajo demanda RHEL base precios: 0.026 USD por hora Bajo demanda Linux base precios: 0.0116 USD por hora

[Se aplican costos adicionales a las AMI con software preinstalado](#)

▼ Par de claves (inicio de sesión) Información

Puede utilizar un par de claves para conectarse de forma segura a la instancia. Asegúrese de que tiene acceso al par de claves seleccionado antes de lanzar la instancia.

Nombre del par de claves

WindosServer Crear un nuevo par de claves

Ilustración 0.33

En la configuración de red lo dejo sin subred y zona de disponibilidad ya que el auto escalable se encarga de asignársela.

▼ **Configuraciones de red** [Información](#)

Subred | [Información](#)

No incluir en la plantilla de lanzamiento ↕ [Crear nueva subred](#)

Al especificar una subred, se agrega automáticamente una interfaz de red a la plantilla.

Zona de disponibilidad [Información](#)

No incluir en la plantilla de lanzamiento ↕ [Enable additional zones](#)

Firewall (grupos de seguridad) | [Información](#)

Un grupo de seguridad es un conjunto de reglas de firewall que controlan el tráfico de la instancia. Agregue reglas para permitir que un tráfico específico llegue a la instancia.

Seleccionar un grupo de seguridad existente Crear grupo de seguridad

Grupos de seguridad | [Información](#)

Seleccionar grupos de seguridad ↕ [Compare reglas de grupo de seguridad](#)

launch-wizard-2 sg-074f1d11dca7285f3 ✕
VPC: vpc-0cd97794c9f7334de

► **Configuración de red avanzada**

Ilustración 0.34

En configuración de red habilitar Asignar automáticamente la IP pública.

▼ **Configuración de red avanzada**

Interfaz de red 1 [Eliminar](#)

Índice de dispositivos | [Información](#)

0

Interfaz de red | [Información](#)

Nueva interfaz ↕

Descripción | [Información](#)

Subred | [Información](#)

No incluir en la plantilla de lanzamiento

Grupos de seguridad | [Información](#)

Seleccionar grupos de seguridad ↕ [Muestre todos los seleccionados \(1\)](#)

Asignar automáticamente la IP pública | [Información](#)

Habilitar ↕

Ilustración 0.35

Ahora vamos a desplegar detalles avanzado y vamos a ir datos de usuario y dejamos este script allí, este script lo que hace es todo lo que hice de manera manual en los pasos anteriores, pero lo hace de manera automática.

Nota: Lo único diferente que hay en el script esta línea `#!/bin/bash` que le indica al sistema operativo en este caso Linux con que interprete debe de ejecutarlo.

Script

```
#!/bin/bash

yum update -y

yum install -y docker nginx

systemctl enable docker

systemctl start docker

systemctl enable nginx

systemctl start nginx

mkdir -p /home/ec2-user/miapp

echo "<h1>Hola desde instancia autoscaling - $(hostname)</h1>" > /home/ec2-
user/miapp/index.html

cat <<EOF > /home/ec2-user/miapp/Dockerfile

FROM nginx:alpine

COPY index.html /usr/share/nginx/html/index.html

EOF

cd /home/ec2-user/miapp

docker build -t miapp .

docker run -d -p 8080:80 miapp

cat <<EOF > /etc/nginx/conf.d/proxy.conf

location / {

    proxy_pass http://localhost:8080;

}

EOF

systemctl restart nginx
```

Plantilla creada

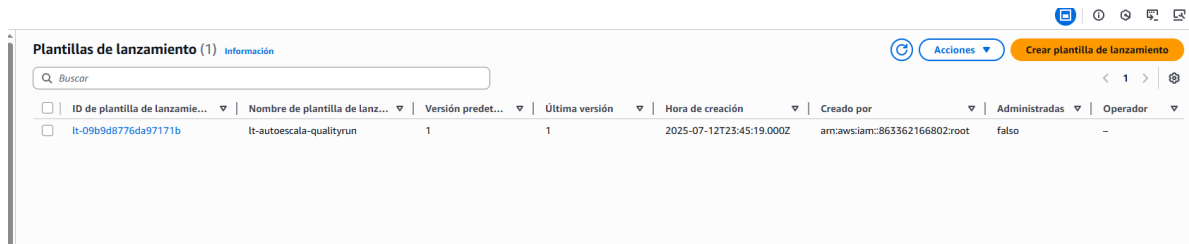


Ilustración 0.36

Crear auto escalable

Ir a EC2, dar clic en grupos de auto Scaling y dar clic en el botón crear grupos de auto scaling.

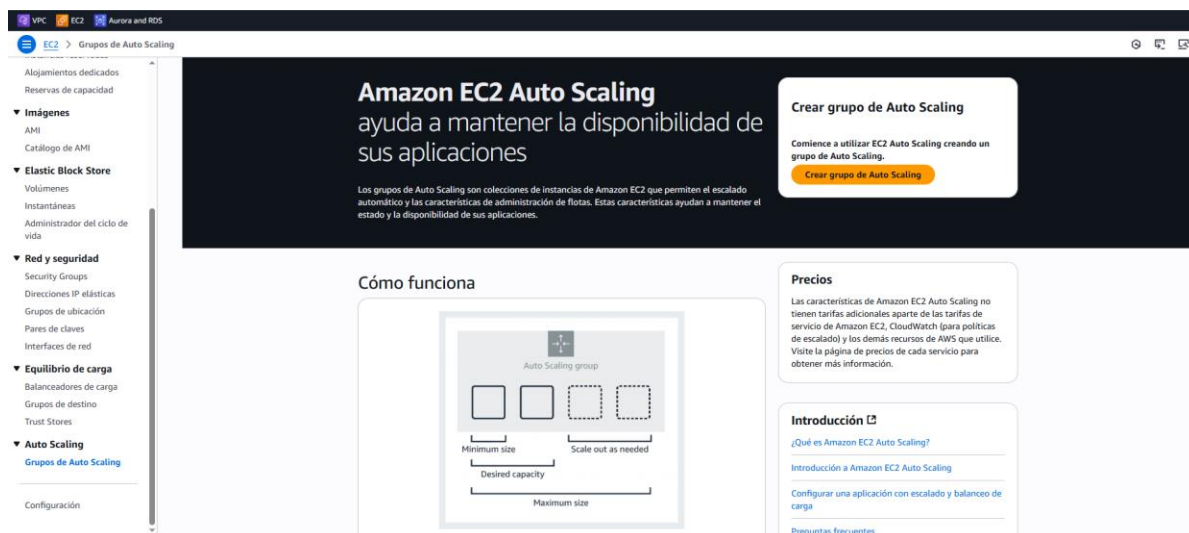


Ilustración 0.37

Allí se debe signarle un nombre y seleccionar la plantilla que creamos anteriormente. Y dar clic en siguiente:

Nombre

Nombre del grupo de Auto Scaling
Escriba un nombre para identificar el grupo.

Debe ser único para esta cuenta en la región actual y no puede superar los 255 caracteres.

Plantilla de lanzamiento Info

🕒 Para las cuentas creadas después del 31 de mayo de 2023, la consola de EC2 solo admite la creación de grupos de escalamiento automático con plantillas de lanzamiento. No se recomienda crear grupos de escalado automático con configuraciones de lanzamiento, pero aún se podrá hacer a través de la CLI y la API hasta el 31 de diciembre de 2023.

Plantilla de lanzamiento
Elija una configuración de lanzamiento que contenga la configuración de nivel de instancia, como la imagen de máquina de Amazon (AMI), el tipo de instancia, el par de claves y los grupos de seguridad.

[Crear una configuración de lanzamiento](#)

Versión
Default (1)

[Crear una versión de plantilla de lanzamiento](#)

<p>Descripción Plantilla para instancias autoescaladas</p> <p>AMI ID ami-0150ccaf51ab55a51</p> <p>Nombre del par de claves WindosServer</p>	<p>Plantilla de lanzamiento lt-autoescala-qualityrun lt-09b9d8776da97171b</p> <p>Grupos de seguridad -</p> <p>ID de grupos de seguridad sg-074f1d11dca7285f3</p>	<p>Tipo de instancia t2.micro</p> <p>Solicitar instancias de spot No</p>
--	---	--

Ilustración 0.38

En el paso 2, seleccionar VPC que ya tenemos y sub redes que igualmente ya tenemos.

Grupos de Auto Scaling > Crear grupo de Auto Scaling

Paso 3 - opcional

● Integrar en otros servicios

Paso 4 - opcional

● Configurar escalamiento y tamaño de grupo

Paso 5 - opcional

● Añadir notificación

Paso 6 - opcional

● Añadir etiquetas

Paso 7

● Revisar

Requisitos de tipo de instancia Info

Puede mantener los mismos atributos o tipos de instancias de la plantilla de lanzamiento, o bien puede optar por anular la plantilla de lanzamiento al especificar atributos de instancia diferentes o al agregar los tipos de instancias de forma manual.

Plantilla de lanzamiento	Versión	Descripción
lt-autoescala-qualityrun lt-09b9d8776da97171b	Default	Plantilla para instancias autoescaladas

Tipo de instancia
t2.micro

Red Info

Para la mayoría de las aplicaciones, puede utilizar varias zonas de disponibilidad y dejar que EC2 Auto Scaling equilibre sus instancias entre las zonas. La VPC predeterminada y las subredes predeterminadas son adecuadas para comenzar rápidamente.

VPC
Elija la VPC que define la red virtual para el grupo de Auto Scaling.

[Crear una VPC](#)

Zonas de disponibilidad y subredes
Defina qué zonas de disponibilidad y subredes puede utilizar el grupo de Auto Scaling en la VPC elegida.

Seleccionar zonas de disponibilidad y subredes

us-east-1a | subnet-05bfe3f876a958da (seminario-subnet-public1-us-east-1a)
10.0.0.0/20

us-east-1b | subnet-0656fc8da25a315bd (seminario-subnet-public2-us-east-1b)
10.0.16.0/20

[Crear una subred](#)

Distribución de zonas de disponibilidad - nuevo

Ilustración 0.39

En el paso 3 elegir el balanceador de carga que ya habíamos creado

Integrar en otros servicios - *opcional* [Info](#)

Use un equilibrador de carga para distribuir el tráfico de red entre varios servidores. Active las comunicaciones de servicio a servicio con VPC Lattice. Desvía los recursos de las zonas de disponibilidad con problemas mediante el cambio de zona. También puede personalizar los reemplazos y la supervisión de la comprobación de estado.

Balance de carga [Info](#)

Utilice las siguientes opciones para asociar su grupo de Auto Scaling a un balanceador de carga existente o a uno nuevo que defina.

No se encontró ningún balanceador de carga
El tráfico a su grupo de Auto Scaling no se llevará a cabo por un balanceador de carga.

Asociar a un balanceador de carga existente
Elija entre los balanceadores de carga existentes.

Asociar a un nuevo balanceador de carga
Cree rápidamente un balanceador de carga básico para asociarlo al grupo de Auto Scaling.

Asociar a un balanceador de carga existente

Seleccione los balanceadores de carga que desea asociar al grupo de Auto Scaling.

Elegir entre los grupos de destino del balanceador de carga
Esta opción le permite asociar balanceadores de carga de puerta de enlace, red o aplicaciones.

Elegir entre balanceadores de carga clásicos

Grupos de destino del balanceador de carga existentes
Solo están disponibles para su selección los grupos de destino de instancias que pertenecen a la misma VPC que el grupo de Auto Scaling.

Seleccionar grupos de destino

QualityRun | HTTP
Application Load Balancer: bal-QualityRun

Ilustración 0.40

En el paso 4 dejamos deseada en 1 (Es la cantidad inicial que se lanza al crear el ASG), mínima en 1 (Siempre que haya al menos 1 instancia funcionando) y máxima en dos (Límite superior: hasta cuántas instancias puede crecer automáticamente)

de Auto Scaling

ramiento

y tamaño

Tamaño del grupo [Info](#)

Defina el tamaño inicial del grupo de escalamiento automático. Después de crear el grupo, puede cambiar su tamaño para satisfacer la demanda, ya sea en forma manual o mediante el escalamiento automático.

Tipo de capacidad deseado
Elija la unidad de medida para el valor de capacidad deseado. Las vCPU y la memoria (GiB) solo son compatibles con grupos de instancias mixtos configurados con un conjunto de atributos de instancia.

Unidades (número de instancias)

Capacidad deseada
Especifique el tamaño de su grupo.

1

Escalado [Info](#)

Puede cambiar el tamaño de su grupo de escalamiento automático de forma manual o automática para cumplir con los cambios en la demanda.

Límites de escalamiento
Establezca límites sobre cuánto puede aumentarse o disminuirse la capacidad deseada.

Capacidad deseada mínima **Capacidad deseada máxima**

1 2

Capacidad igual o inferior a la deseada Capacidad igual o superior a la deseada

Escalamiento automático - *opcional*
Elija si desea utilizar una política de seguimiento de destino [Info](#)
Puede configurar otras políticas de escalado basadas en métricas y un escalado programado después de crear su grupo de escalamiento automático.

Sin políticas de escalamiento
Su grupo de escalamiento automático mantendrá su tamaño inicial y no se redimensionará de forma dinámica para satisfacer la demanda.

Política de escalado de seguimiento de destino
Elija una métrica y un valor objetivo de CloudWatch y deje que la política de escalamiento ajuste la capacidad deseada en proporción al valor de la métrica.

Política de mantenimiento de instancias

Ilustración 0.41

Escalamiento automático

Capacidad igual o inferior a la deseada Capacidad igual o superior a la deseada

Escalamiento automático - opcional
Elija si desea utilizar una política de seguimiento de destino | [Info](#)
 Puede configurar otras políticas de escalado basadas en métricas y un escalado programado después de crear su grupo de escalamiento automático.

Sin políticas de escalamiento
 Su grupo de escalamiento automático mantendrá su tamaño inicial y no se redimensionará de forma dinámica para satisfacer la demanda.

Política de escalado de seguimiento de destino
 Elija una métrica y un valor objetivo de CloudWatch y deje que la política de escalamiento ajuste la capacidad deseada en proporción al valor de la métrica.

Nombre de la política de escalado

cpu-auto

Tipo de métrica | [Info](#)
 Métrica supervisada que determina si la utilización de recursos es demasiado baja o alta. Si utiliza métricas de EC2, considere la posibilidad de habilitar la supervisión detallada para obtener un mejor rendimiento de escalado.

Utilización promedio de la CPU

Valor de destino

50

Preparación de la instancia | [Info](#)

300 segundos

Deshabilite el escalado descendente para crear solo una política de escalado ascendente

Ilustración 0.42

Aquí el punto más importante es el valor destino que es el que indica que, si la CPU supera 50%, lanzará otra instancia.

Paso 5 y 6 opcionales no los diligencie.

Paso 7, revisar y crear grupo de auto scaling.

Grupos de Auto Scaling (1/3) [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

Nombre	Plantilla de lanzamiento/config...	Instanc...	Estado	Capacidad des...	M...	M...	Zonas de disponibilidad
<input checked="" type="checkbox"/> aut-qualityrun3	lt-autoescala-qualityrun3 Versión Predet	1	-	1	1	2	2 Zonas de disponibilidad
<input type="checkbox"/> aut-qualityrun2	lt-autoescala-qualityrun2 Versión Predet	1	-	1	1	2	2 Zonas de disponibilidad
<input type="checkbox"/> aut-qualityrun	lt-autoescala-qualityrun Versión Predete	1	-	1	1	2	2 Zonas de disponibilidad

Grupo de Auto Scaling: aut-qualityrun3

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

aut-qualityrun3 Descripción general de la capacidad [Editar](#)

am:aws:autoScaling:us-east-1:863362166802:autoScalingGroup:27f0092b-dce4-4153-9f4e-d1d8aaeb021b:autoScalingGroupName/aut-qualityrun3

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

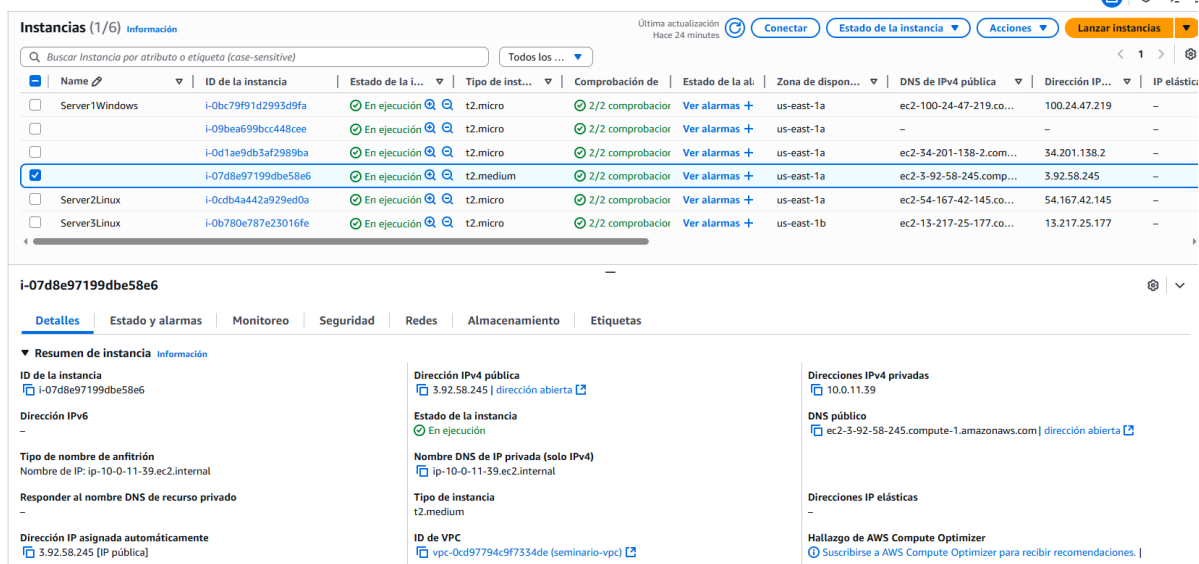
Fecha de creación
Sat Jul 12 2025 19:41:08 GMT-0500 (hora estándar de Colombia)

Plantilla de lanzamiento [Editar](#)

Plantilla de lanzamiento	ID de AMI	Tipo de instancia	Propietario
lt-015eb74c-5f377b774	ami-0150ccaf51ab55a51	t2.medium	arn:aws:iam::863362166802:root

Ilustración 0.43

Instancia creada del auto scaling



Instancias (1/6) Información

Última actualización Hace 24 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 i...	Tipo de inst...	Comprobación de	Estado de la al...	Zona de dispon...	DNS de IPv4 pública	Dirección IP...	IP elástic...
Server1Windows	i-0bc79f91d2993d9fa	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-1a	ec2-100-24-47-219.co...	100.24.47.219	-
	i-09bea699bcc448cee	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-1a	-	-	-
	i-0d1ae9db3af2989ba	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-1a	ec2-34-201-138-2.com...	34.201.138.2	-
	i-07d8e97199dbe58e6	En ejecución	t2.medium	2/2 comprobador	Ver alarmas +	us-east-1a	ec2-3-92-58-245.comp...	3.92.58.245	-
Server2Linux	i-0c0b4442a929ed0a	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-1a	ec2-54-167-42-145.co...	54.167.42.145	-
Server3Linux	i-0b780e787e23016fe	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-1b	ec2-13-217-25-177.co...	13.217.25.177	-

i-07d8e97199dbe58e6

Detalles Estado y alarmas Monitoreo Seguridad Redes Almacenamiento Etiquetas

Resumen de instancia Información

ID de la instancia
i-07d8e97199dbe58e6

Dirección IPv6
-

Tipo de nombre de anfitrión
Nombre de IP: ip-10-0-11-39.ec2.internal

Responder al nombre DNS de recurso privado
-

Dirección IP asignada automáticamente
3.92.58.245 [IP pública]

Dirección IPv4 pública
3.92.58.245 | dirección abierta

Estado de la instancia
En ejecución

Nombre DNS de IP privada (solo IPv4)
ip-10-0-11-39.ec2.internal

Tipo de instancia
t2.medium

ID de VPC
vpc-0cd97794c9f7334de (seminario-vpc)

Direcciones IPv4 privadas
10.0.11.39

DNS público
ec2-3-92-58-245.compute-1.amazonaws.com | dirección abierta

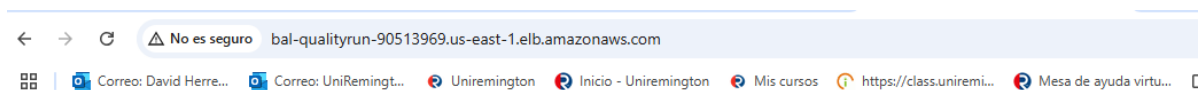
Direcciones IP elásticas
-

Hallazgo de AWS Compute Optimizer
Suscribirse a AWS Compute Optimizer para recibir recomendaciones.

Ilustración 0.44

Prueba ingresando desde la DNS: [http://bal-qualityrun-90513969.us-east-](http://bal-qualityrun-90513969.us-east-1.elb.amazonaws.com/)

1.elb.amazonaws.com/ Para ver que llega al auto scaling



← → ↻ No es seguro bal-qualityrun-90513969.us-east-1.elb.amazonaws.com

Correo: David Herre... Correo: UniRemingt... Uniremington Inicio - Uniremington Mis cursos https://class.unirem... Mesa de ayuda virtu...

Hola desde instancia autoscaling - ip-10-0-11-39.ec2.internal

Ilustración 0.45

Se implementaron dos instancias EC2 distribuidas en diferentes Zonas de Disponibilidad

Availability Zones (AZ) dentro de la región us-east-1

Server2Linux en us-east-1a

Instancias (1/6) Información 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 i...	Tipo de inst...	Comprobación de	Estado de la al:	Zona de dispon...	DNS de IPv4 pública	Dirección IP...	IP elástica
Server1Windows	i-0bc79f91d2993d9fa	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-1a	ec2-100-24-47-219.co...	100.24.47.219	-
	i-09bea699bcc448cee	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-1a	-	-	-
	i-0d1ae9db3af2989ba	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-1a	ec2-34-201-138-2.com...	34.201.138.2	-
	i-07d8e97199dbe58e6	En ejecución	t2.medium	2/2 comprobador	Ver alarmas +	us-east-1a	ec2-3-92-58-245.comp...	3.92.58.245	-
Server2Linux	i-0cdb4a442a929ed0a	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-1a	ec2-54-167-42-145.co...	54.167.42.145	-
Server3Linux	i-0b780e787e23016fe	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-1b	ec2-13-217-25-177.co...	13.217.25.177	-

i-0cdb4a442a929ed0a (Server2Linux)

Tipo de nombre de anfitrión
Nombre de IP: ip-10-0-13-173.ec2.internal

Responder al nombre DNS de recurso privado
-

Dirección IP asignada automáticamente
54.167.42.145 [IP pública]

Rol de IAM
-

IMDSv2
Required

Operador
-

Nombre DNS de IP privada (solo IPv4)
ip-10-0-13-173.ec2.internal

Tipo de instancia
t2.micro

ID de VPC
vpc-0cd97794c9f7334de (seminario-vpc)

ID de subred
subnet-05bf876a958da (seminario-subnet-public1-us-east-1a)

ARN de instancia
arn:aws:ec2:us-east-1:863362166802:instance/i-0cdb4a442a929ed0a

Direcciones IP elásticas
-

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

Nombre del grupo de Auto Scaling
-

Administradas
falso

Ilustración 0.46

Server3Linux en us-east-1b

Instancias (1/6) Información 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 i...	Tipo de inst...	Comprobación de	Estado de la al:	Zona de dispon...	DNS de IPv4 pública	Dirección IP...	IP elástica
Server1Windows	i-0bc79f91d2993d9fa	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-1a	ec2-100-24-47-219.co...	100.24.47.219	-
	i-09bea699bcc448cee	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-1a	-	-	-
	i-0d1ae9db3af2989ba	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-1a	ec2-34-201-138-2.com...	34.201.138.2	-
	i-07d8e97199dbe58e6	En ejecución	t2.medium	2/2 comprobador	Ver alarmas +	us-east-1a	ec2-3-92-58-245.comp...	3.92.58.245	-
Server2Linux	i-0cdb4a442a929ed0a	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-1a	ec2-54-167-42-145.co...	54.167.42.145	-
Server3Linux	i-0b780e787e23016fe	En ejecución	t2.micro	2/2 comprobador	Ver alarmas +	us-east-1b	ec2-13-217-25-177.co...	13.217.25.177	-

i-0b780e787e23016fe (Server3Linux)

Tipo de nombre de anfitrión
Nombre de IP: ip-10-0-26-111.ec2.internal

Responder al nombre DNS de recurso privado
-

Dirección IP asignada automáticamente
13.217.25.177 [IP pública]

Rol de IAM
-

IMDSv2
Required

Operador
-

Nombre DNS de IP privada (solo IPv4)
ip-10-0-26-111.ec2.internal

Tipo de instancia
t2.micro

ID de VPC
vpc-0cd97794c9f7334de (seminario-vpc)

ID de subred
subnet-0656fc8da25a315bd (seminario-subnet-public2-us-east-1b)

ARN de instancia
arn:aws:ec2:us-east-1:863362166802:instance/i-0b780e787e23016fe

Direcciones IP elásticas
-

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

Nombre del grupo de Auto Scaling
-

Administradas
falso

Ilustración 0.47

Confirmación de autoscaling multizonas

Scaling

Grupos de Auto Scaling (1/3) info Última actualización hace 6 minutos Configuraciones de lanzamiento Plantillas de lanzamiento Acciones Crear grupo de escalado automático

Buscar sus grupos de Auto Scaling

Nombre	Plantilla de lanzamiento/config...	Instanc...	Estado	Capacidad des...	M...	M...	Zonas de disponibilidad
<input checked="" type="checkbox"/> aut-qualityrun3	lt-autoescala-qualityrun3 Versión Predet	1	-	1	1	2	2 Zonas de disponibilidad
<input type="checkbox"/> aut-qualityrun2	lt-autoescala-qualityrun2 Versión Predet	1	-	1	1	2	2 Zonas de disponibilidad
<input type="checkbox"/> aut-qualityrun	lt-autoescala-qualityrun Versión Predete	1	-	1	1	2	2 Zonas de disponibilidad

Grupo de Auto Scaling: aut-qualityrun3

Red Editar

Zonas de disponibilidad use1-az4 (us-east-1a) use1-az6 (us-east-1b)	ID de subred subnet-05bfeb3f876a958da subnet-01b7c2d043f79509b	Distribución de zonas de disponibilidad Mejor esfuerzo equilibrado
---	--	---

Requisitos de tipo de instancias Editar

El grupo de Auto Scaling se adhiere a la plantilla de lanzamiento para la opción de compra y el tipo de instancia.

Ilustración 0.48

Diagrama de Arquitectura

Este diagrama ilustra cómo funciona una arquitectura en AWS que permite escalar automáticamente una aplicación web. Todo comienza con el usuario, quien accede desde Internet, su solicitud llega primero a un Application Load Balancer (ALB), que se encarga de distribuir el tráfico entre varias instancias EC2 dentro de una VPC. Cada una de estas instancias tiene Nginx instalado como proxy reverso en el sistema operativo, y además ejecuta un contenedor Docker que también tiene Nginx, pero este último sirve una página HTML sencilla o un texto simple que muestra en que instancia esta. Las instancias están distribuidas en diferentes zonas de disponibilidad para asegurar una alta disponibilidad, y el grupo de Auto Scaling se ocupa de lanzar instancias automáticamente según la demanda del sistema.

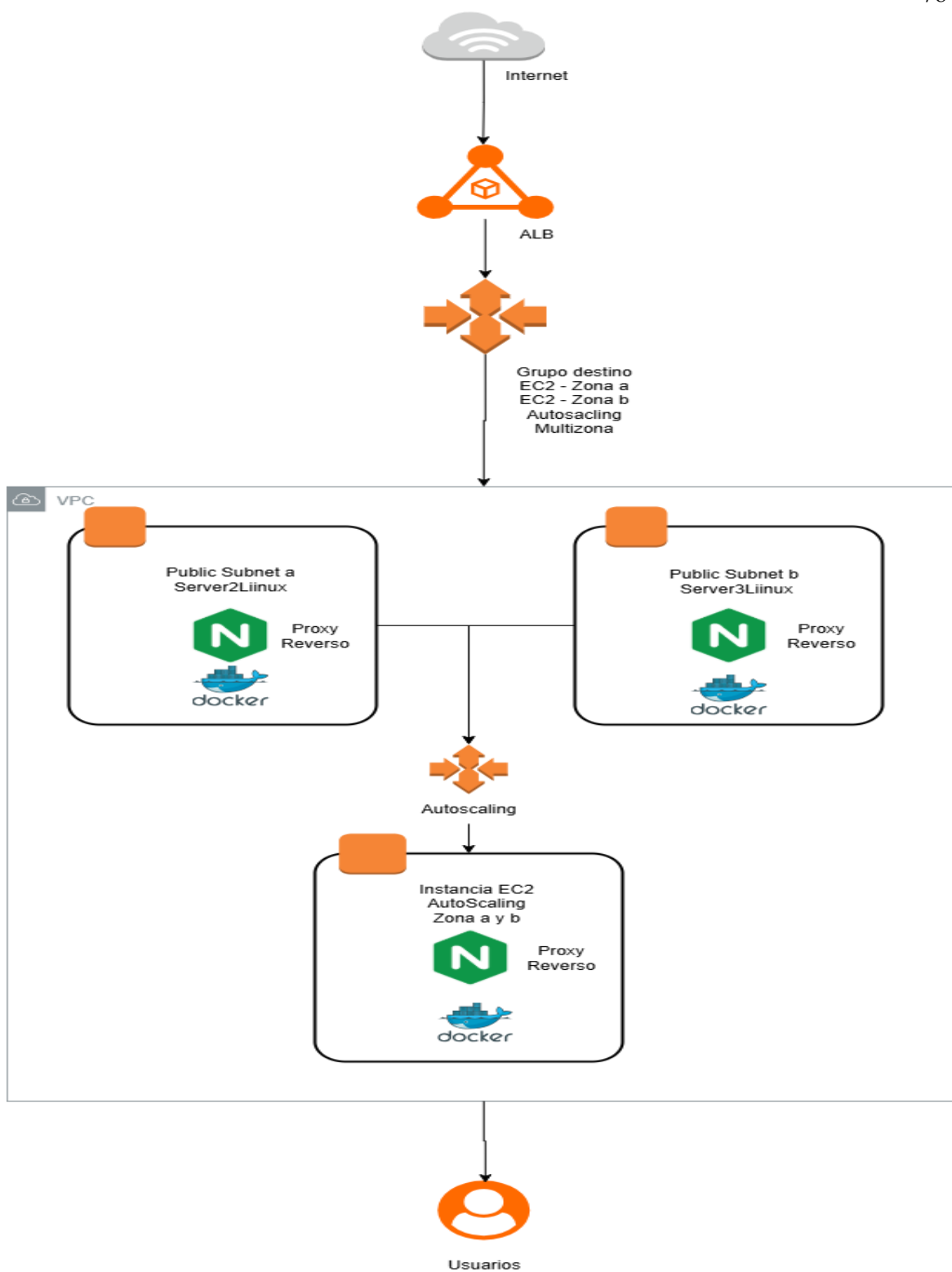


Ilustración 0.49

CONCLUSIONES

Amazon EC2 se consolidó como un servicio versátil que facilita la implementación de servidores virtuales personalizados. La experiencia mostró que su configuración inicial, junto con las políticas de seguridad adecuadas, permite levantar entornos funcionales en minutos, accesibles desde cualquier lugar del mundo.

La creación de una VPC personalizada permitió comprender de manera profunda la segmentación de redes en AWS. Al implementar subredes públicas y controlar el acceso mediante grupos de seguridad, se logró una infraestructura segura y eficiente, validando el principio de mínimo privilegio.

El uso de Application Load Balancer (ALB) resultó clave para garantizar la disponibilidad del servicio web. La posibilidad de distribuir el tráfico entre múltiples instancias no solo aumentó la resiliencia del sistema, sino que también mejoró los tiempos de respuesta bajo condiciones variables de carga.

Las pruebas con Auto Scaling Groups permitieron comprobar que AWS puede escalar automáticamente los recursos según la demanda. Esta capacidad se vuelve esencial en aplicaciones con tráfico fluctuante, ya que ayuda a mantener el rendimiento sin incurrir en costos innecesarios.

La incorporación de contenedores Docker, junto con Nginx como proxy reverso, demostró ser una solución práctica y eficiente para desplegar aplicaciones. Docker facilitó la portabilidad y consistencia del entorno, mientras que Nginx mejoró la gestión del tráfico interno en cada instancia.

En conjunto, los servicios estudiados y aplicados durante el proyecto evidencian que AWS ofrece una plataforma robusta y flexible, que permite construir arquitecturas modernas con enfoque en escalabilidad, seguridad, automatización y alta disponibilidad. Esta práctica dejó en

claro cómo cada servicio se articula con los demás, formando soluciones completas que pueden adaptarse a distintos contextos empresariales.

Referencias

<https://docs.aws.amazon.com/ec2/>

<https://docs.aws.amazon.com/elasticloadbalancing/>

<https://docs.docker.com/>

<https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/>

<https://docs.aws.amazon.com/autoscaling/>

<https://stackoverflow.com/questions/14972792/nginx-nginx-emerg-bind-to-80-failed-98-address-already-in-use>