**TRABAJO DE GRADO**
**Opción Seminario-Diplomado.**

# Diseño Y Despliegue De Una Red EC2 Multi-Sistema Con Acceso Público Y Servicios Web Dockerizados


Corporación Universitaria Remington.
Nombre de la facultad: Facultad de Ingenierías
Nombre del programa académico: Ingeniería de Sistemas

Nombres de los estudiantes autores del trabajo de grado.
Harold Stiven Valencia Rosero
Juan Sebastian Rodriguez Fajardo
Nombre del Tutor del trabajo de grado (docente del seminario o diplomado).
Juan Pablo Berrío López


Opción de Trabajo de grado Seminario-Diplomado.
2025

**Tabla de Contenidos**

**Portada** ..... **1**
**Tabla De Contenido** ..... **2**
**Resumen** ..... **4**
**Palabras Clave** ..... **4**
**Marco Conceptual Y Contextual** ..... **5**
**Diseño Y Despliegue De Una Red Ec2 Multi-Sistema Con Acceso Público Y Servicios Web Dockerizados** ..... **8**
**Objetivo General** ..... **8**
**Diagrama De Arquitectura** ..... **8**
**Descripción De La Arquitectura** ..... **9**
**Breve Explicación De La Red Creada** ..... **9**
**Tipo De Instancias Usadas (Linux: Amazon Linux, Ubuntu, Etc. | Windows: Versión De Windows Server)** ..... **9**
**Justificación De Las Configuraciones De Red (Por Ejemplo, Uso De VPC, Subredes Públicas, Internet Gateway)** ..... **10**
**Configuraciones Realizadas** ..... **11**
**Pasos Para Crear La Instancia EC2** ..... **11**
**Detalles De Los Grupos De Seguridad (Puertos Abiertos: RDP, SSH, HTTP).** ..... **19**
**Asignación De Ips Públicas Y Privadas** ..... **20**
**Procedimiento De Acceso** ..... **20**
**Cómo Acceder A Cada Servidor (Cliente RDP Para Windows, SSH Para Linux)** ..... **20**
**Consideraciones De Seguridad (Por Ejemplo, Uso De Llaves PEM, Contraseñas Seguras)** ..... **32**
**Configuración Del Servidor Web** ..... **33**
**Pasos Seguidos Para Instalar IIS En Windows Server.** ..... **33**
**Pasos Para Instalar Apache O Nginx En Linux.** ..... **33**
**Pruebas Básicas Para Verificar Que Los Servidores Web Son Accesibles Desde Internet (Captura De Pantallas Del Navegador)** ..... **34**
**Pruebas De Conectividad** ..... **35**
**Desde La Instancia Windows Hacer *Ping* A La IP Privada De La Instancia Linux Y Viceversa** ..... **35**
**Documentar Si Hay Necesidad De Habilitar ICMP En Los Grupos De Seguridad Para Permitir Ping.** ..... **36**
**Validación De Acceso Web** ..... **37**
**Acceder Desde El Navegador Local Al Sitio Web De La Instancia Windows** ..... **37**
**Acceder Desde El Navegador Local Al Sitio Web De La Instancia Linux** ..... **37**
**Implementar El Servicio De Docker** ..... **38**
**Instalar Docker En La Máquina Virtual.** ..... **38**
**Ver El Estado Del Servicio De Docker.** ..... **38**
**Activar El Sistema De Docker.** ..... **39**
**Instalar La Imagen De Apache En Docker.** ..... **39**

**Creamos La Primera Aplicación A Partir De Un Servidor Web En Docker**. .....**39**
**Verificamos El Docker ID**.....**39**
**Agregar El Puerto Al SG De La Instancia En Este Caso El Puerto 8080**. .....**40**
**Verificar Que El Contenedor Esta Trabajando**. .....**40**
**Cambiamos El It Works Por El Sitio Que Queremos Subir Al Contenedor**. .....**40**
**Verificar Si El Contenedor Puede Acceder Al Sitio**. .....**41**
**De Esa Forma Podemos Crear Más Contenedores Con Un Sitio Estático**. .....**41**
**Pruebas De Estrés A La Instancia Free Tier**.....**41**
**Creamos Un Contenedor Que Consuma 1 Vcpu**. .....**41**
**Podemos Ver El Estrés De La Instancia Con "Docker Stats"**.....**42**
**Conclusiones**.....**43**
**Referencias**.....**44**

## Resumen

El presente trabajo expone, paso a paso, el proceso de creación, configuración e implementación de recursos en la nube utilizando Amazon Web Service (AWS). Se inicia con el registro de una cuenta en la plataforma de AWS para acceder a sus servicios, una vez dentro del entorno, el paso a seguir es, tener claridad en cuál será la estructura de red que se va a implementar en Amazon Web Service (ASW). Luego de esto se explica cómo es, la elaboración, creación y configuración de máquinas virtuales o como se conocen en Amazon Web Service (AWS) instancias, este proceso se debe realizar en la pestaña de EC2 dentro de Amazon Web Service (AWS), seguidamente vamos a poder observar y configurar la red virtual (VPC), junto con las subredes públicas y privadas y el Internet Gateway, que es el que permite la salida a internet desde las instancias, este proceso lo realiza automáticamente Amazon Web Service (AWS). También se configuran los grupos de seguridad, que definen las reglas de acceso para habilitar la comunicación adecuada entre los recursos y el exterior. Amazon Web Service (AWS) asigna automáticamente direccionamiento IP, lo que facilita la conexión remota a las instancias. Se describe el proceso de conexión a instancias Windows mediante RDP y a instancias Linux por SSH, utilizando llaves PEM generadas con la opción de Key Pair, que garantizan la autenticación segura. Estas llaves permiten establecer sesiones protegidas con los servidores alojados en la nube, para validar la conectividad y la correcta implementación, se instalaron servidores web: IIS en Windows Server y Nginx en Linux, con el fin de realizar pruebas básicas de acceso desde dispositivos locales. Esto permitió verificar el funcionamiento de los servicios y la comunicación entre las instancias, realizando pruebas como el uso de ping entre instancias y accediendo desde a los servicios web localmente. Finalmente, se aborda la integración de Docker en Amazon Web Service (AWS), lo cual permite el despliegue de múltiples contenedores para ejecutar servidores web de forma aislada y eficiente. Esta práctica ofrece una alternativa moderna y escalable en la administración de aplicaciones dentro de la nube.

## Palabras clave

1. VPC
2. Instancias
3. Key Pair
4. Grupo de seguridad
5. Docker

## Marco conceptual y contextual

Este trabajo es desarrollado para dar a conocer los primeros pasos en la inmersión hacia los avances tecnológicos de los que estamos siendo partícipes nosotros, como ingenieros de sistemas.

En primer lugar, se abordará un tema fundamental: el Cloud Computing o computación en la nube. Este nuevo desarrollo tecnológico se ha venido fortaleciendo a lo largo del tiempo, con un gran impacto y auge en los sistemas de información. Otorga a empresas y usuarios soluciones que permiten una mayor flexibilidad, reducción de costos y multiplicación de beneficios. Un claro ejemplo de esto es la posibilidad de compartir archivos a través de internet o contar con espacios de almacenamiento en la nube, como Dropbox, Google Drive, Microsoft Office 365, entre otros, que son herramientas ampliamente utilizadas con estos fines (De Trabajo & Moreno, 2015)

Todos nosotros hemos sido beneficiados por este concepto de Cloud Computing en algún momento. Esta tecnología está presente, por ejemplo, en los correos electrónicos que utilizamos a diario. Solo es necesario acceder a servicios, aplicaciones o incluso infraestructuras que residen en internet, lo que nos exonera de tener servidores físicos en nuestras casas, oficinas o universidades. Esto significa que los recursos pueden estar ubicados y funcionando desde cualquier parte del mundo.(Fernando et al., n.d.)

En segundo lugar, presentamos los proveedores de nube: IBM, Google, Microsoft, AT&T, Apache, EMC, Cisco, Amazon, Salesforce.com, Enomaly, CapGemini, RightScale, Vordel. Estas empresas ofrecen múltiples servicios y cuentan con infraestructuras locales. Sin embargo, el beneficio para las organizaciones es que se puede acceder a ellos de manera remota, es decir, desde cualquier lugar con conexión a internet, ya sea desde nuestros hogares, universidades o lugares de trabajo.

Estos proveedores tienen un impacto significativo en las empresas, ya que modifican la necesidad del uso de equipos locales y centros de datos propios. En lugar de invertir en infraestructura propia, las organizaciones pueden arrendar el uso de servicios y recursos que ofrece el proveedor. Es este proveedor quien se encarga del mantenimiento, actualización, disponibilidad y seguridad de los servicios, mientras que el cliente solo debe enfocarse en el manejo de su información, mantenerla actualizada y realizar copias de seguridad. Ya no es necesario preocuparse por equipos obsoletos o limitaciones de espacio físico, pues basta con contar con un computador con acceso a internet para utilizar los servicios contratados. Este es el impacto que reciben las empresas con estas nuevas formas de trabajo y almacenamiento de la información.(Ortiz Clavijo et al., 2018)

En tercer lugar, vamos a enfocarnos y adentrarnos un poco más en el conocimiento del proveedor de nube Amazon Web Services (AWS). Este ofrece múltiples servicios a sus usuarios. Su uso radica, básicamente, en crear una cuenta para obtener un usuario y contraseña; luego de este paso, ya podremos ingresar a su plataforma de gestión.
Tenemos la posibilidad, inicialmente, de conocer en la opción de manuales el uso de la aplicación y el correcto funcionamiento e implementación de los servicios que ofrece, para crear cualquier tipo de sistema con las especificaciones más básicas, o si se desea, llegar a crear un gran data center, implementando una estructura de red (esto de forma automática

por AWS) y usando solamente los dispositivos necesarios que se requieran para dichas implementaciones.
Esto permite usar únicamente lo que se necesita, ni más ni menos. Es importante aclarar que este es un servicio que tiene un costo, el cual depende del uso de las herramientas, el almacenamiento y el tiempo de uso.
AWS ofrece protección de datos, gestión de accesos seguros, regula el acceso a los recursos en la nube, se integra fácilmente con los sistemas actuales de gestión de acceso, cumple con estrictas normativas, maneja flexibilidad, rentabilización, servicios y plataformas backend seguras.
Proporciona además un aumento en la productividad, disponibilidad de escalabilidad, no patrocina la piratería, usa tecnologías avanzadas, ofrece recuperación ante desastres, almacenamiento múltiple, entre muchas más herramientas y dispositivos para uso doméstico, universitario y profesional. (Mukherjee, n.d.)

En cuarto lugar, se realiza la presentación de los componentes de Amazon Web Services (AWS) que se utilizan, y se indica cómo es la instalación de estos. Por otro lado, se documenta cuáles son los servicios implementados. Para realizar la configuración e interconexión entre cada uno de los elementos, se procede con la configuración de red, lo cual dentro de AWS se conoce como VPC (Virtual Private Cloud).
Dentro de este proceso se inicia con la elaboración de los data centers, los cuales son responsables de contener las máquinas virtuales (instancias) creadas en EC2. En estas se modifican los grupos de seguridad, se les asignan IP públicas o privadas, y AWS, por defecto, crea las tablas de enrutamiento, las cuales se conectan al Internet Gateway para otorgar permisos a las IP públicas y que estas puedan salir a Internet. Por el contrario, si se configuran como IP privadas, no tendrán salida ni acceso desde y hacia Internet.
De igual manera, dentro de estos data centers se almacena la creación de los contenedores Docker, lo que permite obtener dentro de una misma máquina varios contenedores. En cada uno de estos se puede desplegar un servicio distinto, lo que permite lograr alta disponibilidad: si un contenedor falla o se pierde la conexión, se utilizan los otros contenedores disponibles.
Para realizar las respectivas pruebas de funcionalidad, inicialmente se debe confirmar la conexión y acceso a las máquinas virtuales (instancias). Dentro de ellas se despliegan servidores web, que permiten verificar su funcionamiento desde el localhost de cada máquina. Luego, usando la IP pública, se realizan pruebas desde los equipos locales, ingresando la dirección en el navegador para verificar que el servidor web instalado en cada sistema operativo (Windows o Linux) se despliegue de manera normal. Finalmente, se ejecuta un ping entre las máquinas para validar la conectividad.(Jiménez Candela & Fornes Juan, 2024)
Y por último, para la validación del entorno Docker, se realiza una prueba de estrés al sitio web, con el fin de confirmar que los contenedores responden correctamente a cada una de las peticiones, evaluar su comportamiento y finalizar con las respectivas conclusiones.(Martel, n.d.)

La implementación y el uso de estas herramientas generan beneficios a los usuarios que las adquieren. Estos beneficios radican en que el gasto será únicamente por el arrendamiento de componentes, lo cual resulta más económico que tenerlos en un lugar físico. Esto se

debe a que un espacio físico conlleva gastos de mantenimiento del lugar, servicios públicos, daños locativos, y desactualización de dispositivos que requieren ser reemplazados por obsolescencia o bajos recursos.
La administración de estos recursos en la nube reduce todos estos costos y genera tranquilidad frente a esos temas, ya que son los proveedores quienes deben preocuparse por ellos.
Existen varios tipos de servicios en la nube, como, por ejemplo: IaaS (Infrastructure as a Service), que significa infraestructura como servicio, y provee al usuario de máquinas virtuales (EC2), almacenamiento (S3), espacio en servidores, redes, almacenamiento y sistemas operativos.
Por otro lado, están los servicios PaaS (Platform as a Service), que significa plataforma como servicio, y ofrecen un entorno de desarrollo ya listo.
Otro servicio es el SaaS (Software as a Service), que significa software como servicio, y permite a los usuarios acceder a aplicaciones completas directamente desde Internet, sin necesidad de instalar nada.(De Trabajo & Moreno, 2015)

Este trabajo es el desarrollo de un seminario, el cual tiene como propósito enseñar un poco del contenido del entorno de Amazon Web Services (AWS). Se basa en practicar dentro de su plataforma, logueados con una cuenta, y por medio de esta poder realizar las configuraciones de los componentes que ofrece en sus servicios AWS. En esta oportunidad: máquinas virtuales, configuración de VPC, asignación de grupos de seguridad, visualización de creación de las Route Table para las diferentes conexiones, permitiendo hacer pruebas de uso de estos servicios.

¿Cómo? Realizando creaciones de servidores web e ingresando localmente desde nuestros equipos y desde las máquinas virtuales creadas, permitiéndonos poder ver en ejecución estos procesos. Abriendo el entendimiento a que los recursos en la nube no solamente otorgan beneficios económicos, sino que también nos muestran la forma en cómo los servicios tienen alta disponibilidad, redundancia, poca probabilidad al fallo y alta disponibilidad de los servicios que se creen o almacenen en esta nueva tecnología.

Esto nos ofreció la enseñanza de cuáles son las ventajas y desventajas que se obtienen al implementar estas nuevas tecnologías en empresas que aún no hacen parte de las migraciones a estos entornos digitales, quizá por miedo o desconocimiento de la gran cantidad de beneficios que trae, sobre todo el ahorro que genera el adaptar sus procesos a este mundo del cloud computing, utilizando el proveedor de internet de Amazon Web Services (AWS).

## DISEÑO Y DESPLIEGUE DE UNA RED EC2 MULTI-SISTEMA CON ACCESO PÚBLICO Y SERVICIOS WEB DOCKERIZADOS

### 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.

# 1. Diagrama de arquitectura

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

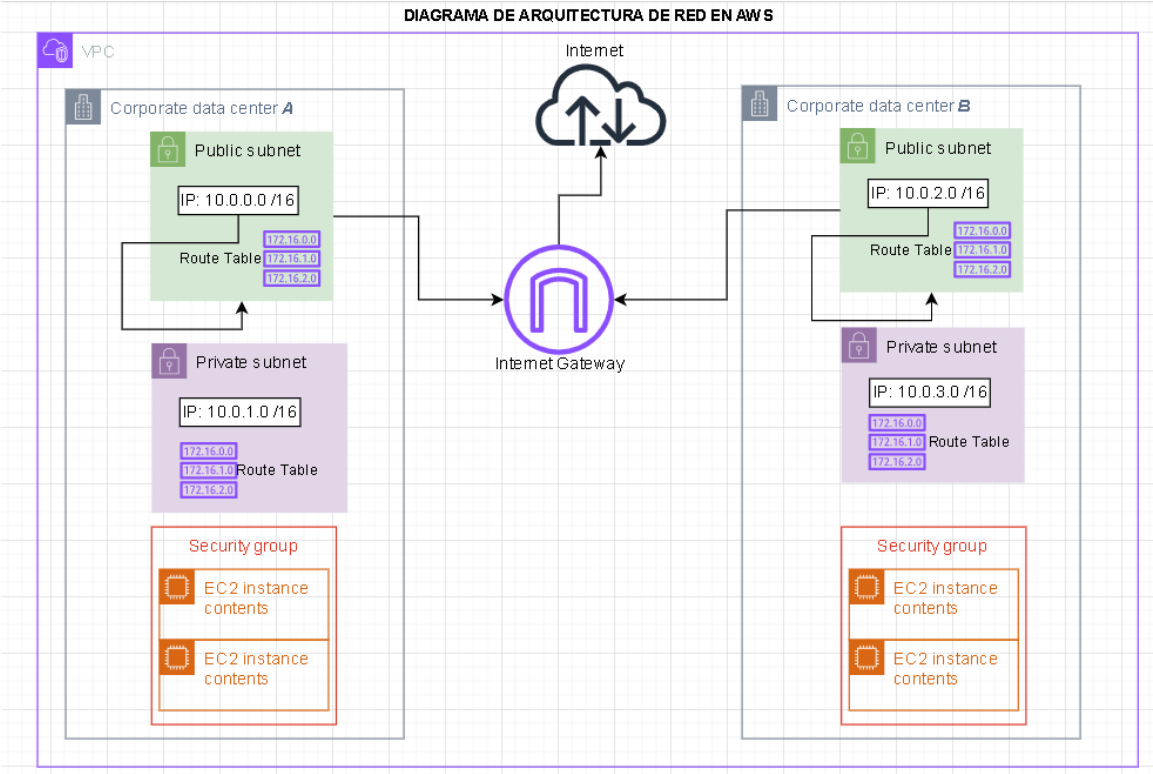



*Figura 1Diagrama de Arquitectura de Red en AWS*

# 2. Descripción de la arquitectura

## 2.1 Breve explicación de la red creada.

En este proyecto, se implementará una red utilizando el servicio de Amazon web Service, aquí haremos uso de un espacio donde vamos a crear un VPC (Virtual Private Cloud), con el nombre Seminario-AWS-vpc. Dentro de esta VPC se realizará la configuración de dos subredes públicas, cada una de estas se ubicará en una zona de disponibilidad distinta datacenter A y data center B, para mejorar la disponibilidad y redundancia. Estas subredes están asociadas a una tabla de rutas (Route table) que permite la salida a Internet mediante un Internet Gateway, esta es una configuración que realiza internamente AWS, el cual

estará dentro de la VPC. En estas subredes se desplegarán dos instancias EC2: una con Windows Server 2016 Base y otra con Amazon Linux 2023 AMI, ambas tienen la configuración de direcciones IP públicas asignadas para permitir el acceso remoto por medio del aplicativo Conexión a Escritorio Remoto y la disponibilidad de poder acceder a los servicios web que se despliegan desde estas mismas máquinas a través de Internet.

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

Windows Server 2016 Base:

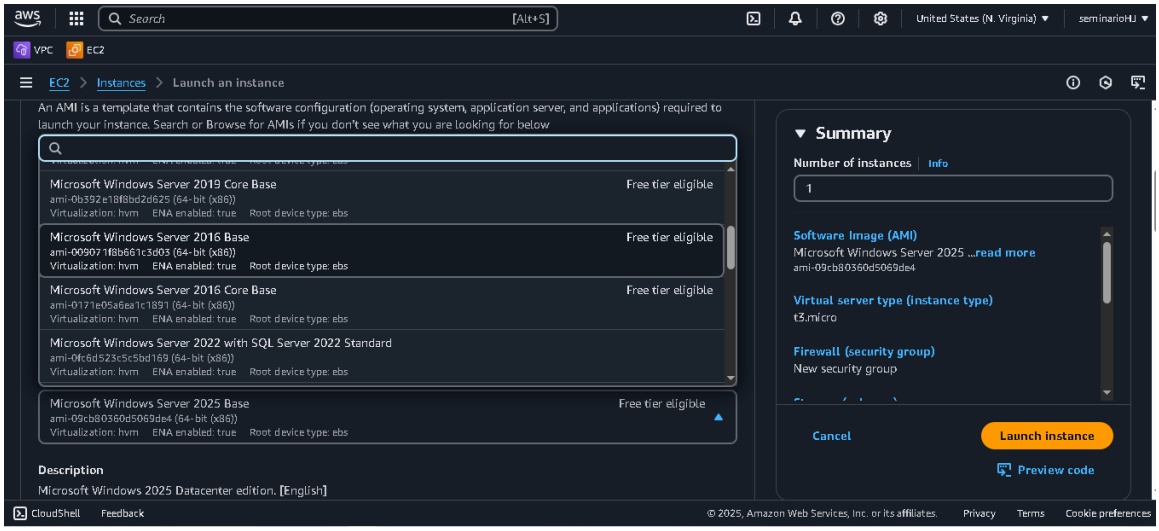



*Figura 2 Entorno de AWS para crear la instancia de Windows*

Amazon Linux 2023 AMI:

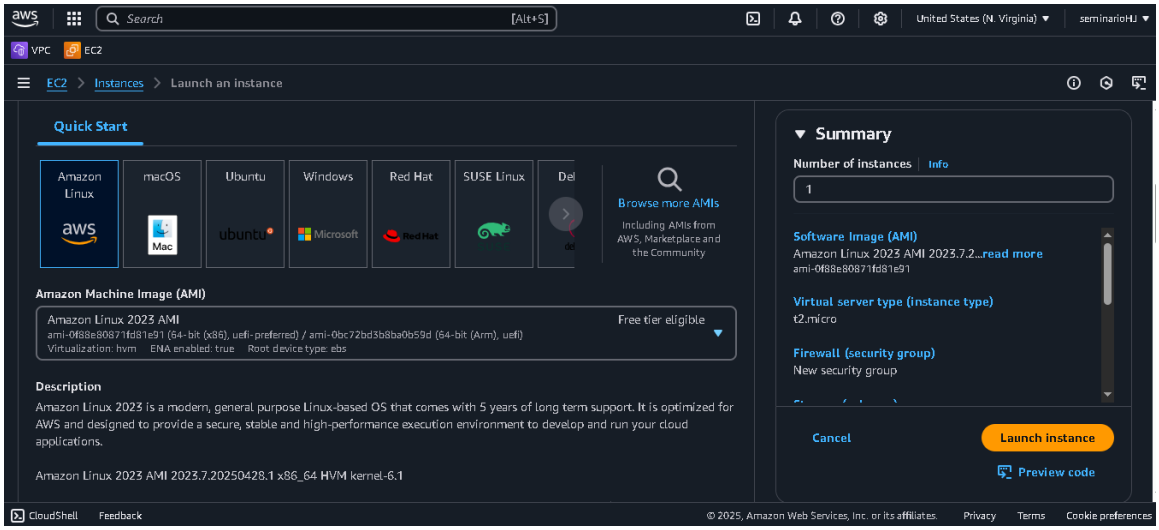



*Figura 3 Entorno de AWS para la instancia de Linux*

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

Inicialmente para nuestro proyecto empezamos con la creación de la VPC (Virtual Private Cloud) que sirve para crear un espacio de trabajo en AWS, esto se puede representar también como un Data Center, esto lo que permite es que todas las instancias, bases de datos, contenedores, etc, que se creen van a pertenecer a un red para realizar su interconexión y comunicación entre ellos, en AWS el direccionamiento IP que realiza por defecto utiliza la siguiente segmentación de red 10.0.0.0 / 16, partiendo de esta información, nosotros lo que debemos hacer es, validar que se realice la correcta creación de las instancias y luego por medio de estas, que se crean con ayuda de EC2, visualizar la información de la opción **Public IPv4 address**, aquí es donde se evidencia nuestra IP pública que ha asignado por defecto AWS, estas IP públicas son las que nos van a servir a nosotros para realizar las respectivas validaciones de acceso tanto a las instancias como a los servicios que estas estén ejecutando dentro de estas; adicionalmente a esto, debemos tener claro un punto importante para que exista conexión hacía internet y desde internet, utilizando el internet Gateway (este término, es como lo conoce AWS, para hacer referencia al Router) que tiene como segmentación de red la 0.0.0.0 / 0, entonces inicialmente nuestras instancias deben estar dentro de una red pública y seguido a esto, deben tener asignada una IP pública, de resto el funcionamiento es interno que es el que realiza la tabla de enrutamiento o Route Table, esta es la que se encarga de recibir las peticiones salientes, que son las que se generan desde la instancia hacía internet o las peticiones entrantes que son las peticiones que se reciben desde el navegador de los clientes hacía las instancias, entonces el internet Gateway con ayuda de la Route Table se sincronizan y se comunican para llevar la información al destino según corresponda.

## 3. Configuraciones realizadas

### 3.1 Pasos para crear la instancia EC2

Para crear instancias lo primero que debemos hacer es ingresar a EC2, aquí es donde se crean las instancias (máquinas virtuales)

**3.1.1.1** Dar click en Launch Instances o lanzar instancia para crear la de Windows

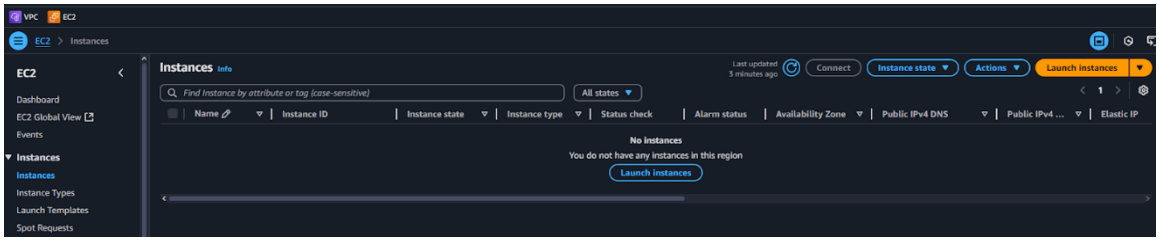



*Figura 4 Entorno de AWS para Visualizar la Opción de Launch Instances*

**3.1.1.2** Colocarle un nombre a nuestra instancia


*Figura 5 Entorno de AWS para Colocar el Nombre a la Instancia*

**3.1.1.3** Elegir un Sistema Operativo

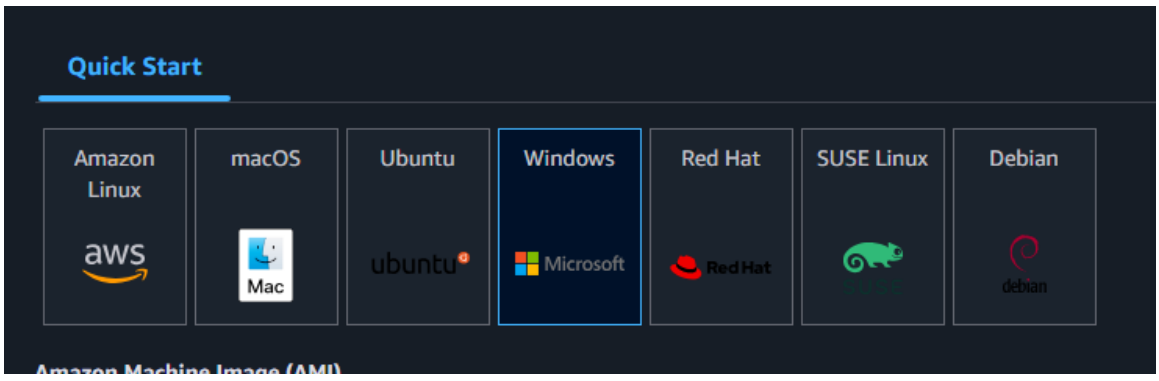



*Figura 6 Entorno de AWS para Elegir el Sistema Operativo*

**3.1.1.4** Tener en cuenta que tipo de instancia se va a usar, porque de esta depende la CPU, RAM y la Marca de la CPU

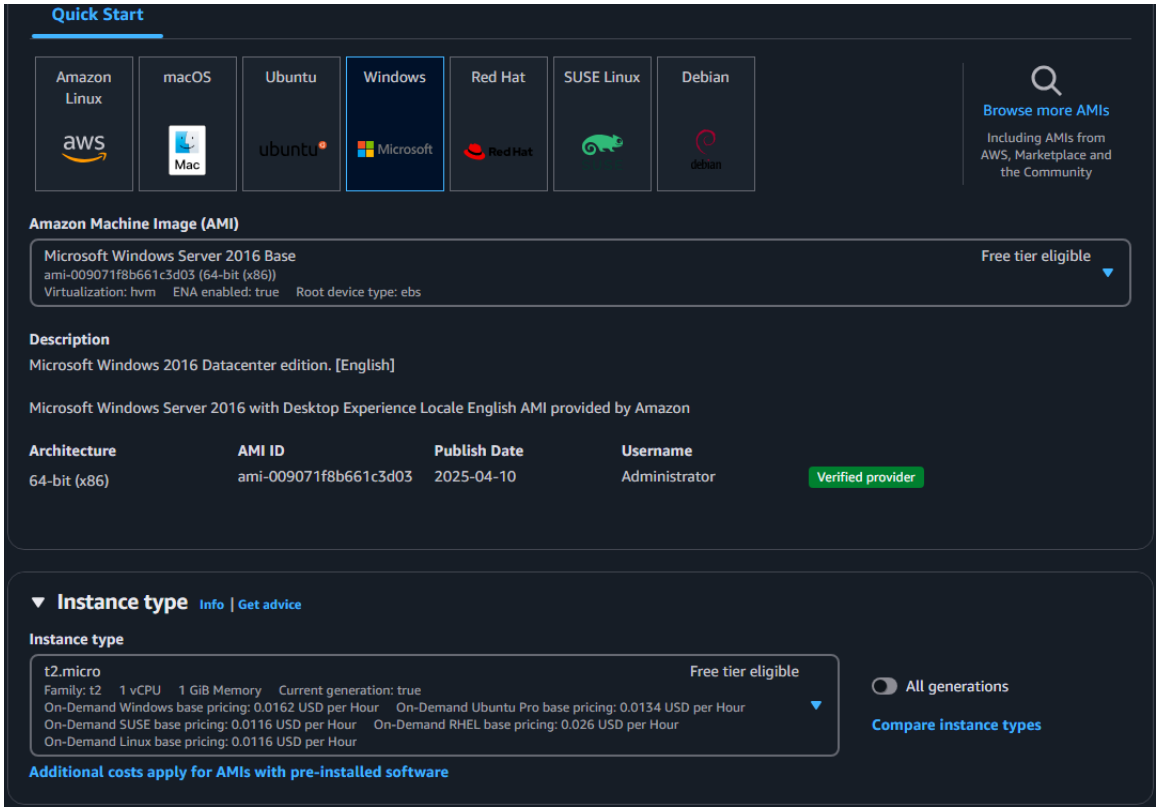



*Figura 7 Entorno de AWS para Elegir las Opciones de CPU, RAM, Marca CPU*

**3.1.1.5** Elegir la versión del SO


*Figura 8 Entorno de AWS para Elegirla Versión del Sistema Operativo*

**3.1.1.6** Configuración del Key pair o clave de seguridad para acceder a nuestras instancias, la key se descarga en un archivo tipo pem, el cual está encriptado.

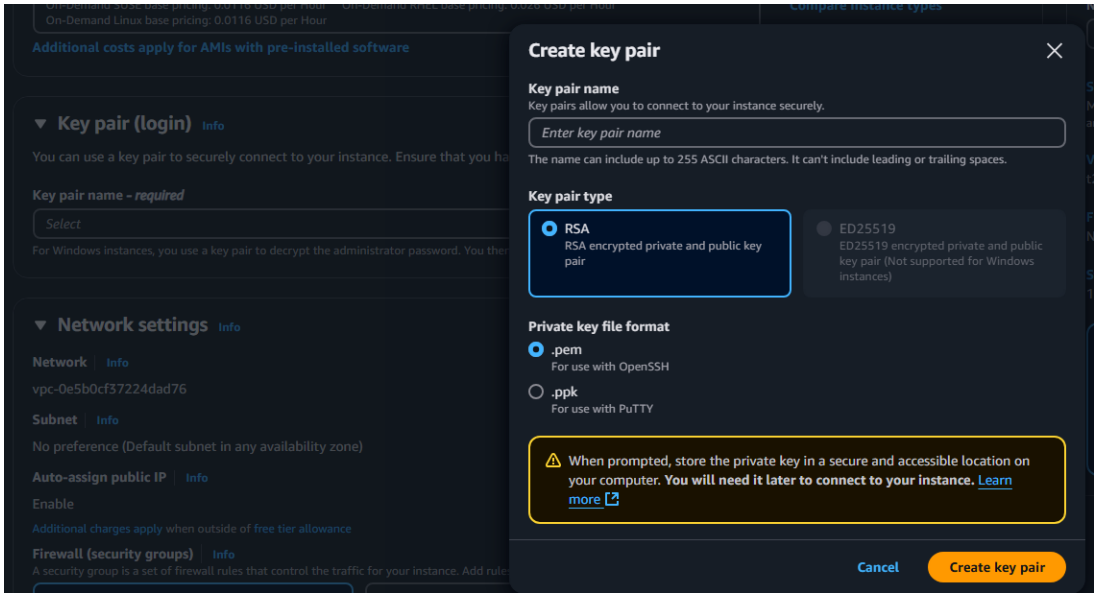



*Figura 9 Entorno de AWS para Configurar la Clave Key Pair*

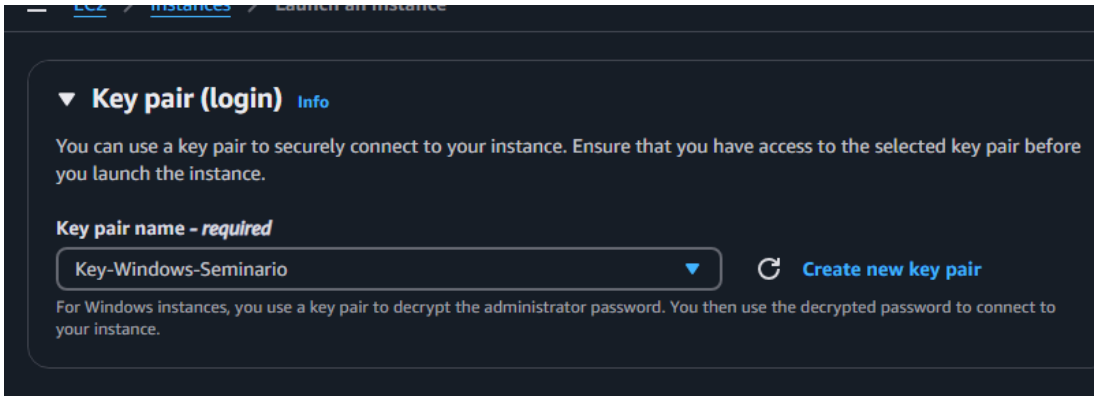



*Figura 10 Entorno de AWS para Crear la Clave Key Pair*

**3.1.1.7** Configuraciones de Red o Network settings

3.1.1.7.1 Configuración del VPC

3.1.1.7.2 Asignación del direccionamiento IP puede ser público o privado

3.1.1.7.3 Autoasignar public IP, esto debe estar habilitado (Enable)

3.1.1.7.4 Configuración del grupo de seguridad del Firewall (cada instancia tiene una para agregar reglas de seguridad para garantizar que pase o no los datos hacía la instancia, aquí se crea el Security Group)

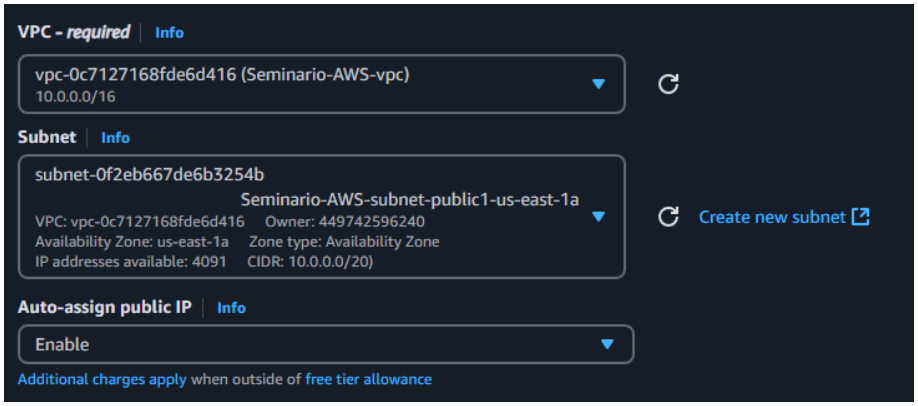



*Figura 11 Entorno de AWS para las Configuraciones de Red*

3.1.1.7.5 Crear grupo de seguridad

3.1.1.7.6 Crear reglas y dejar la que aparece por defecto para la conexión RDP por el puerto 3389

**3.1.1.8** Configuración del almacenamiento o configure storage, donde se realiza la configuración de los discos duros lo dejamos por defecto

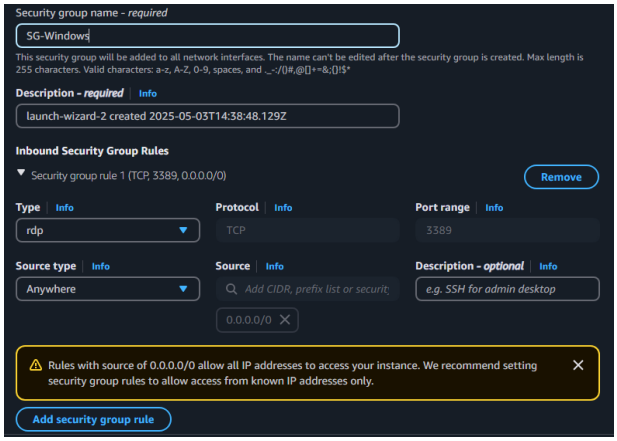



*Figura 12 Entorno de AWS para Configurar Reglas de seguridad, Conexión y Almacenamiento*

**3.1.1.9** Por último elegimos la opción de Launch Instance para su respectiva creación

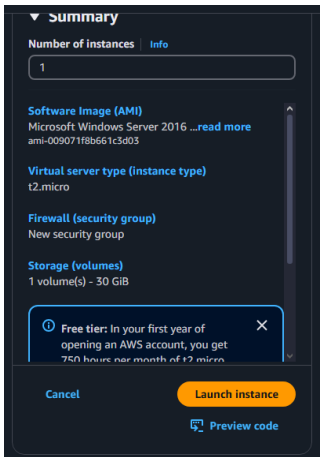



*Figura 13 Entorno de AWS que Muestra la Opció de Launch instance para Culminar la Creación de la Instancia*

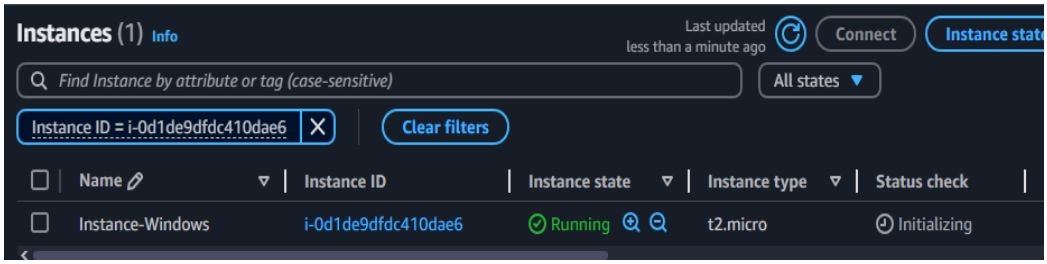



*Figura 14 Entorno de AWS para Confimar la Creación de la Instancia*

### 3.1.2 Pasos para crear la instancia EC2

Para crear instancias lo primero que debemos hacer es ingresar a EC2, aquí es donde se crean las instancias (máquinas virtuales)

**3.1.2.1** Dar click en Launch Instances o lanzar instancia para crear la de Linux
**3.1.2.2** Colocarle un nombre a nuestra instancia
**3.1.2.3** Elegir un Sistema Operativo

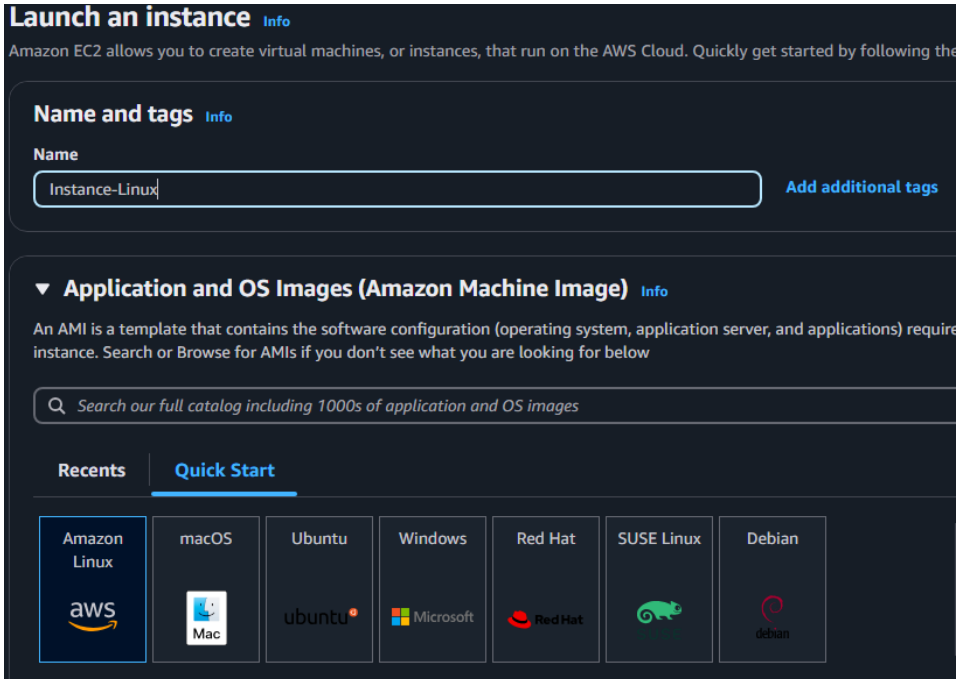



*Figura 15 Entorno de AWS para crear la instancia de Linux*

**3.1.2.4** Tener en cuenta que tipo de instancia se va a usar, porque de esta depende la CPU, RAM y la Marca de la CPU
**3.1.2.5** Elegir la versión del SO
**3.1.2.6** Configuración del Key pair o clave de seguridad para acceder a nuestras instancias

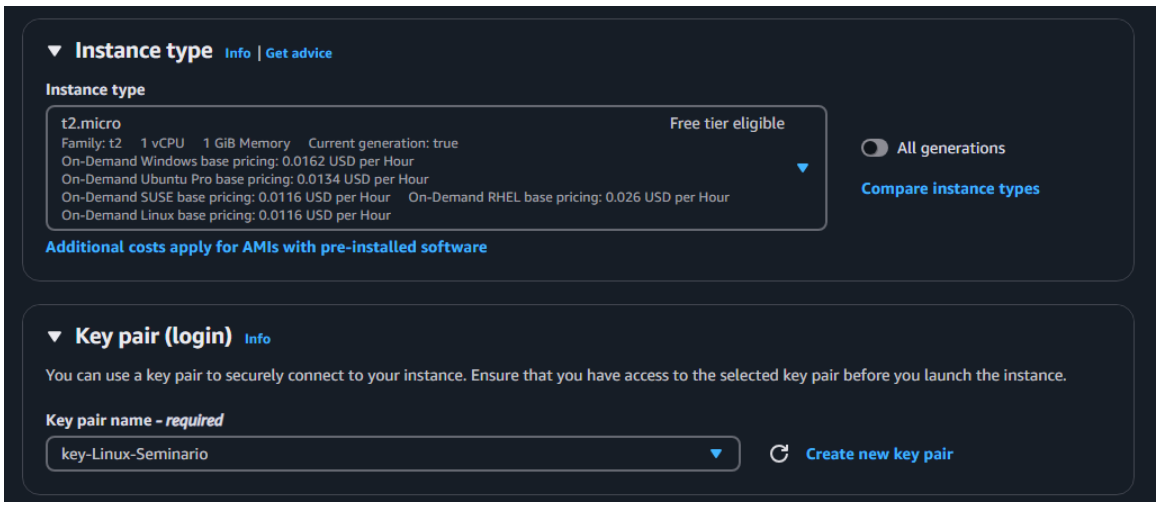



*Figura 16 Entorno de AWS para Elegir las Opciones de CPU, RAM, Marca CPU y Clave Key Pair*

**3.1.2.7** Configuraciones de Red o Network settings

3.1.2.7.1 Configuración del VPC

3.1.2.7.2 Asignación del direccionamiento IP puede ser público o privado

3.1.2.7.3 Autoasignar public IP, esto debe estar habilitado (Enable)

3.1.2.7.4 Configuración del grupo de seguridad del Firewall (cada instancia tiene **una** para agregar reglas de seguridad para garantizar que pase o no los datos hacía la instancia, aquí se crea el Security Group)

3.1.2.7.5 Crear grupo de seguridad

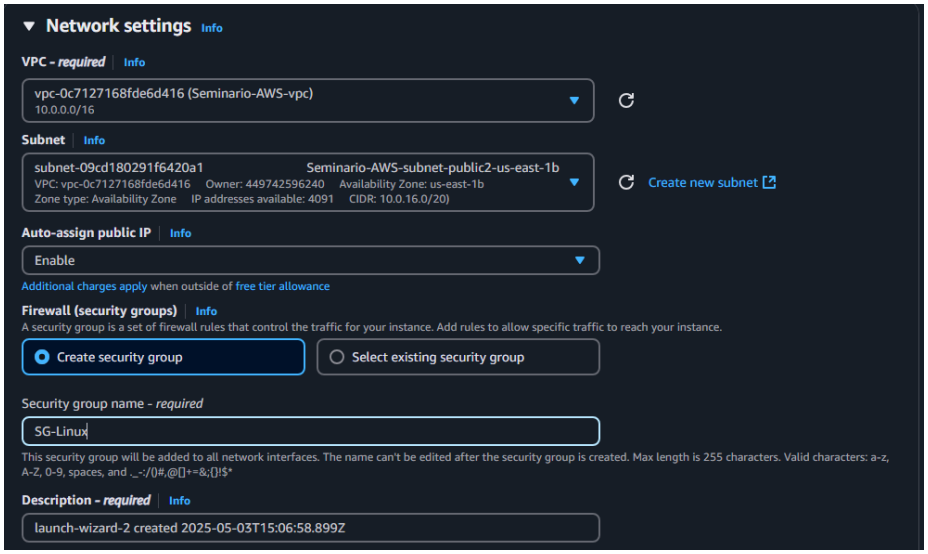



*Figura 17Entorno de AWS para Elegir las Opciones de Red y Grupo de Seguridad*

**3.1.2.7.6** Crear reglas y dejar la que aparece por defecto para la conexión SSH por el puerto 22

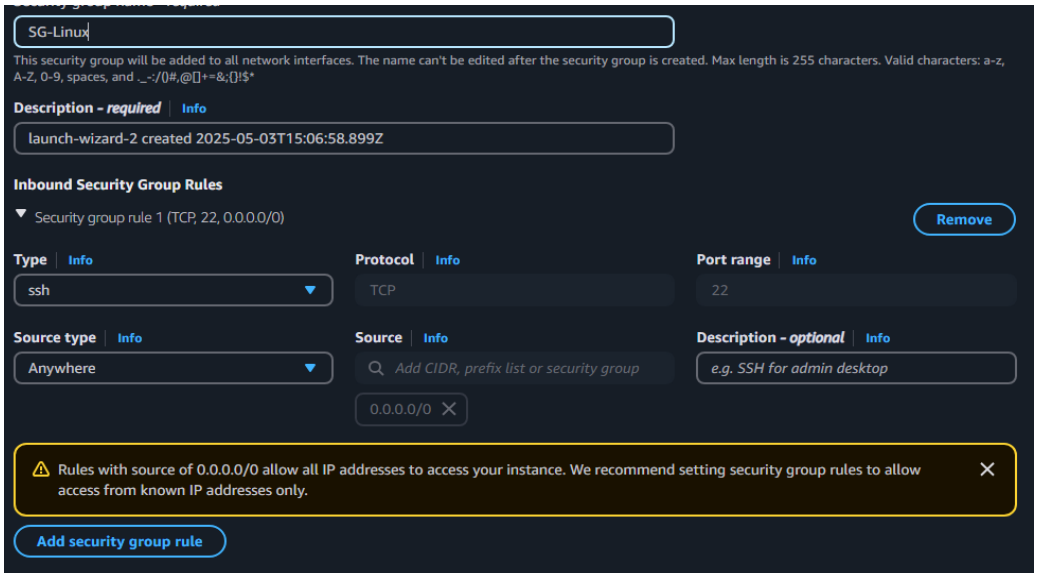



*Figura 18 Entorno de AWS para Elegir las Opciones de Conexión por SSH*

**3.1.2.8** Configuración del almacenamiento o configure storage, donde se realiza la configuración de los discos duros lo dejamos por defecto

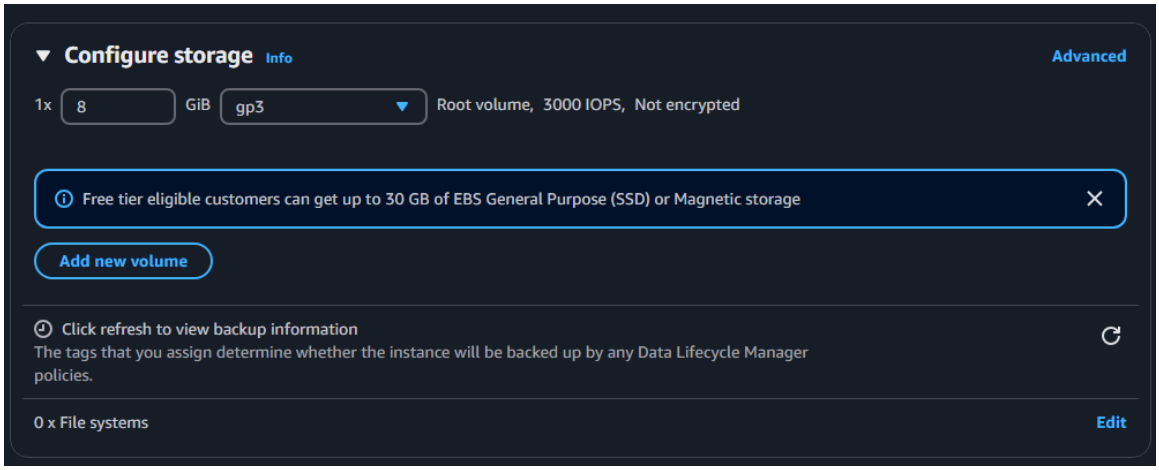



*Figura 19 Entorno de AWS para Elegir las Opción de Almacenamiento de la Instancia*

**3.1.2.9** Por último elegimos la opción de Launch instance para su respectiva creación y confirmamos su creación

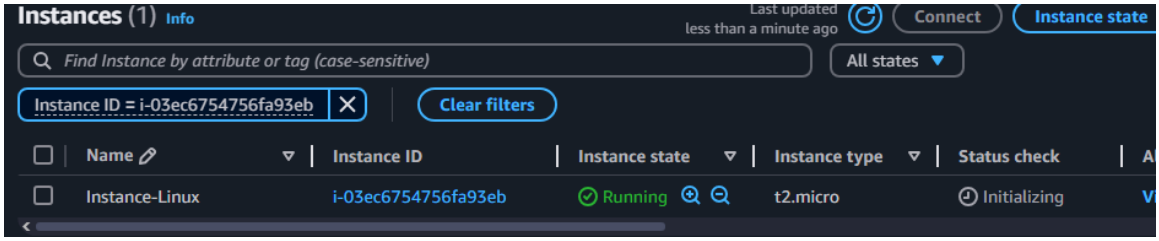



*Figura 20 Entorno de AWS para Visualizar la Instacia Creada*

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

**3.2.1** Cuando estamos modificando el Firewall de la instancia de Windows se crean las reglas de seguridad se debe dejar la que aparece por defecto que es para la conexión RDP por el puerto 3389
Grupo de seguridad configurado para poder ingresar al sitio de prueba desde cualquier parte. Habilitando el puerto 80

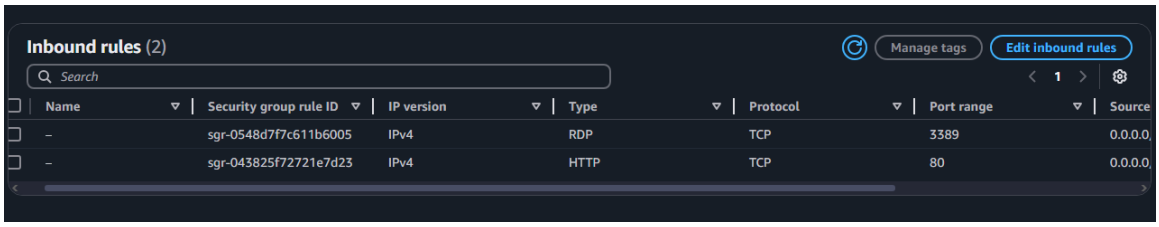



*Figura 21 Entorno de AWS para Ver los Detalles de los Grupos de Seguridad*

**3.2.2** Cuando estamos modificando el Firewall de la instancia de Linux se crean las reglas de seguridad se debe dejar la que aparece por defecto que es para la conexión SSH por el puerto 22
HTTP (puerto 80) abierto a todo Internet (0.0.0.0/0) para ambos.

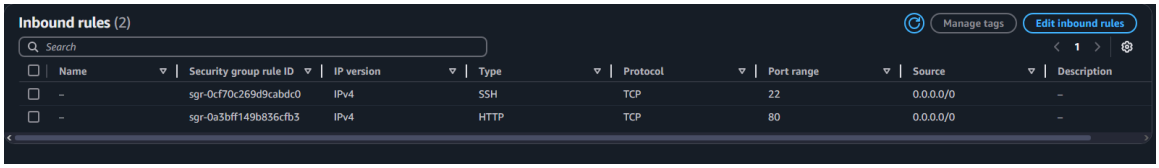



*Figura 22 Entorno de AWS para Ver los Detalles de los Grupos de Seguridad*

### 3.3 Asignación de IPs públicas y privadas.

**3.3.1** Para realizar la asignación de las IP públicas, basta con mirar el paso a paso y ver lo que se realiza en este, más exactamente en el punto 7.2, aquí es donde se le indica a AWS

que quiero que se cree la instancia como pública o privada (para nuestro caso va hacer pública) y en el paso 7.3 es donde se habilita que el proceso de asignación de IP pública se haga automáticamente al crear la instancia seleccionando la opción de **Enable** de **Auto-assign public IP**

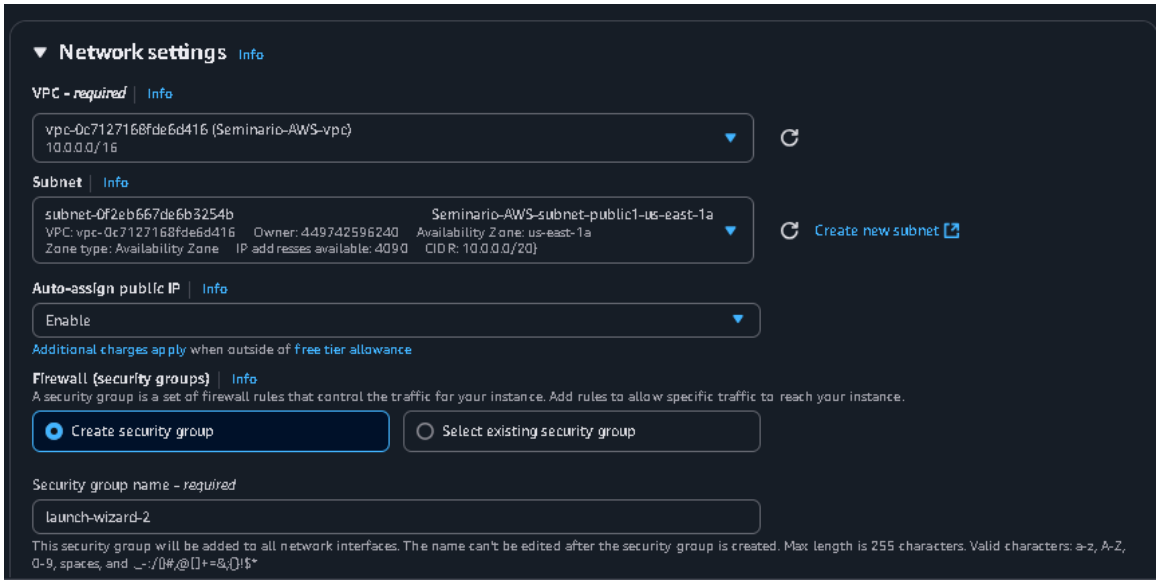



*Figura 23 Entorno de AWS para Realizar la Asignación de IP Publica o Privada*

## 4. Procedimiento de acceso

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

Para acceder a los dos diferentes servidores (instancias), creadas en nuestro caso lo que debemos realizar es lo siguiente:

### 4.1.1 Conexión a cliente RDP para Windows:

**4.1.1.1** Para acceder al servidor Windows por medio de RDP, nos debemos dirigir a la página principal de EC2, a la opción de la parte izquierda **Instances**, aquí podemos evidenciar las dos Instancias creadas, en este caso vamos a seleccionar la instancia que lleva por nombre **Instance-Windows**:

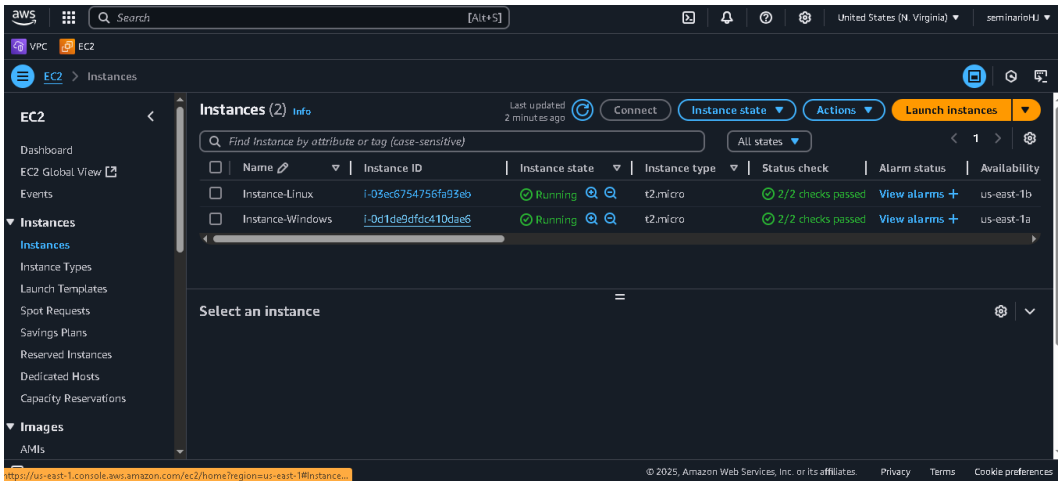



*Figura 24 Entorno de AWS Donde se Visualiza las Instancias Creadas Windows y Linux*

**4.1.1.2** Ingresamos a esta instancia y aquí podemos evidenciar su IP pública en este caso es 54.91.97.149, la cual vamos a utilizar para hacer la conexión

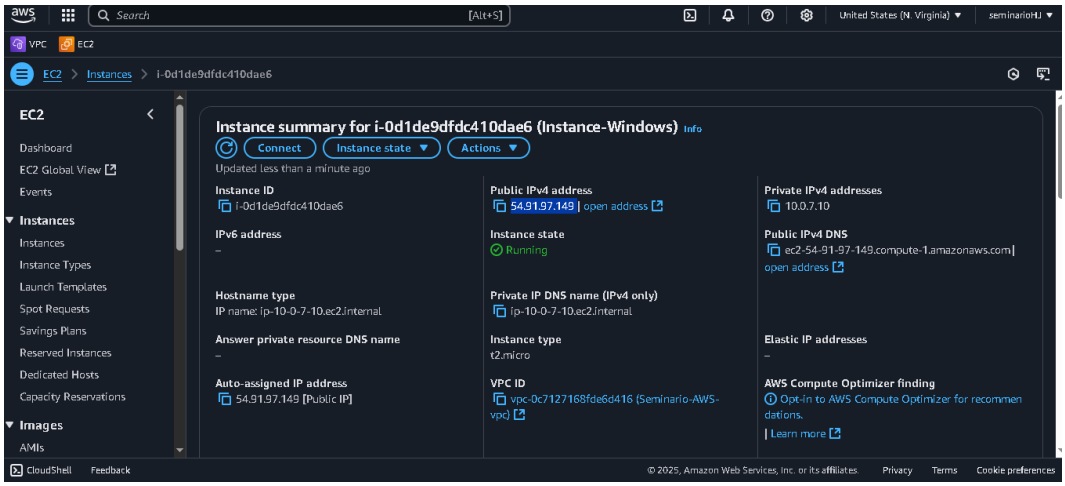



*Figura 25 Entorno de AWS para Visualizar la Información a Detalle de las Instancias*

**4.1.1.3** Una vez copeada esta IP publica 54.91.97.149 la vamos a utilizar para copearla en nuestro aplicativo Conexión a Escritorio Remoto de Windows para acceder a este servidor

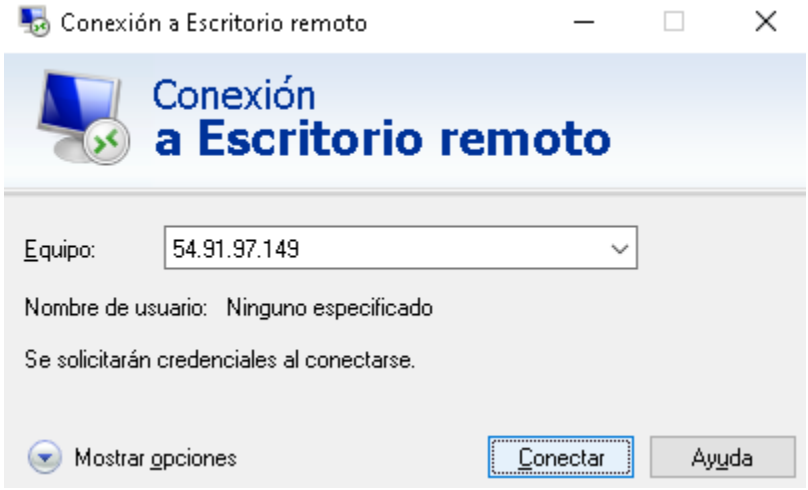



*Figura 26 Imagen de la Aplicación de Windows Conexión a Escritorio Remoto*

**4.1.1.4** Para conocer cuales son las credenciales de acceso, debemos irnos a la parte superior izquierda que dice Connect

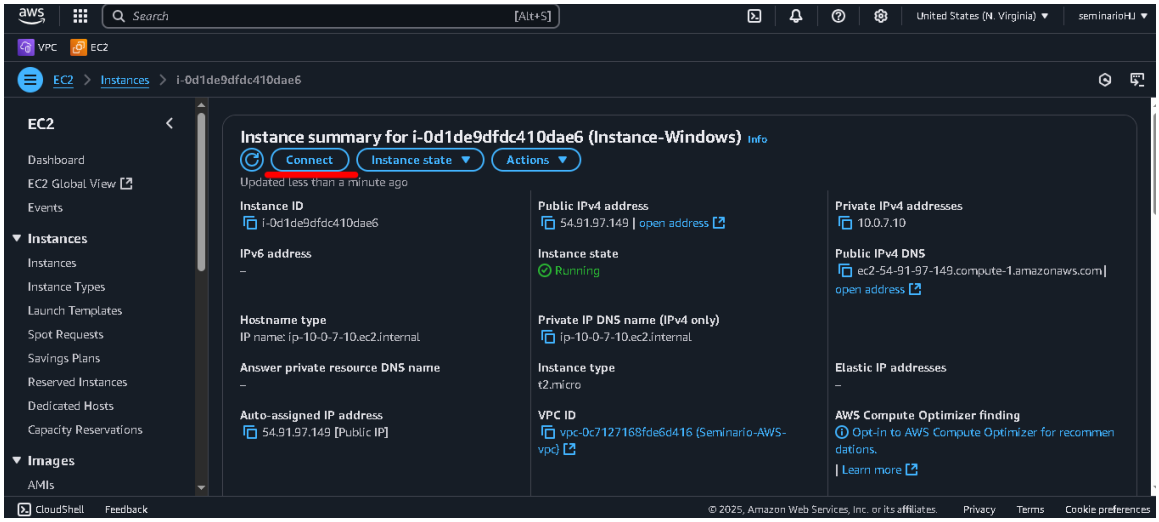



*Figura 27 Entorno de AWS para Ver los Detalles de Conexion a la Instancia*

**4.1.1.5** Cuando carga la información, debemos elegir la opción de RDP Client aquí podemos cargar la clave Key que se genero al crear la instancia y automáticamente se nos muestra el password que se genera y al lado derecho se evidencia el usuario que debemos utilizar para este caso es el Administrator

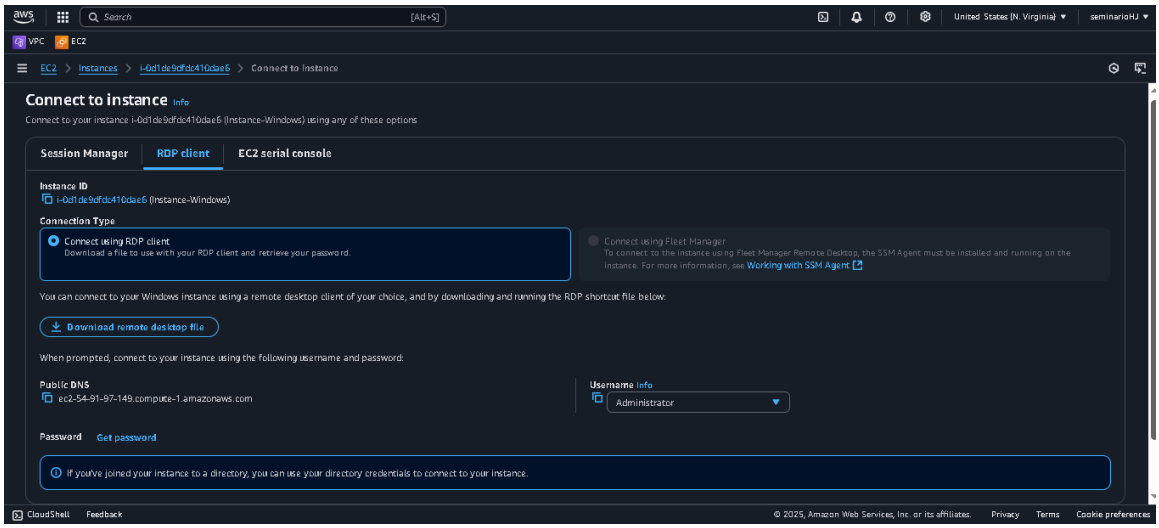



*Figura 28 Entorno de AWS para Realizar el Proceso de Obtener el Usuario y la Clave para Acceder a la Instancia*

**4.1.1.6** Aquí realizamos el cargue del Key password y se debe elegir la opción de desencriptar el password

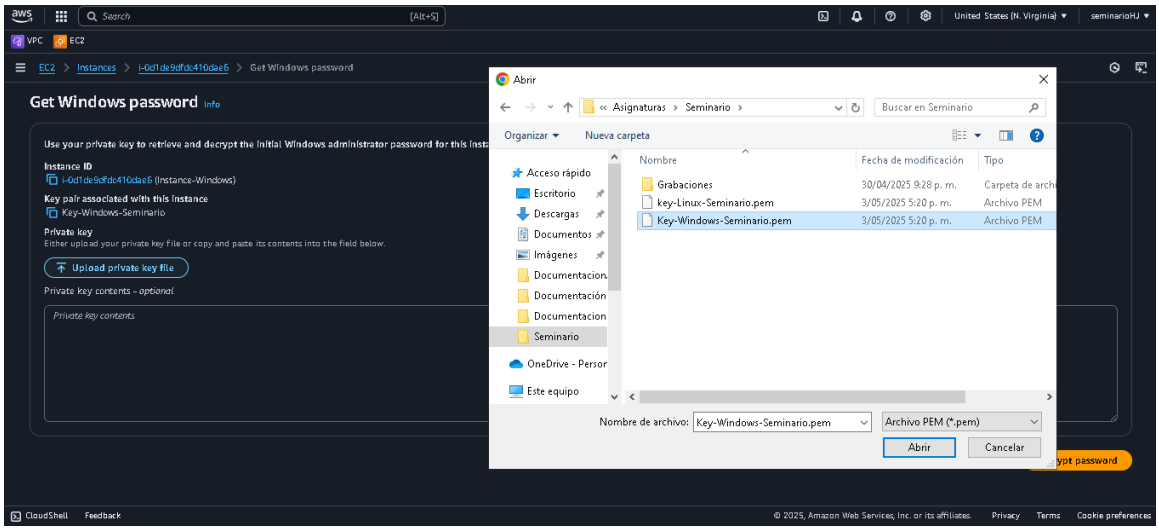



*Figura 29 Visualización del Explorador de Archivos del Pc Donde se Aloja la Clave Key*

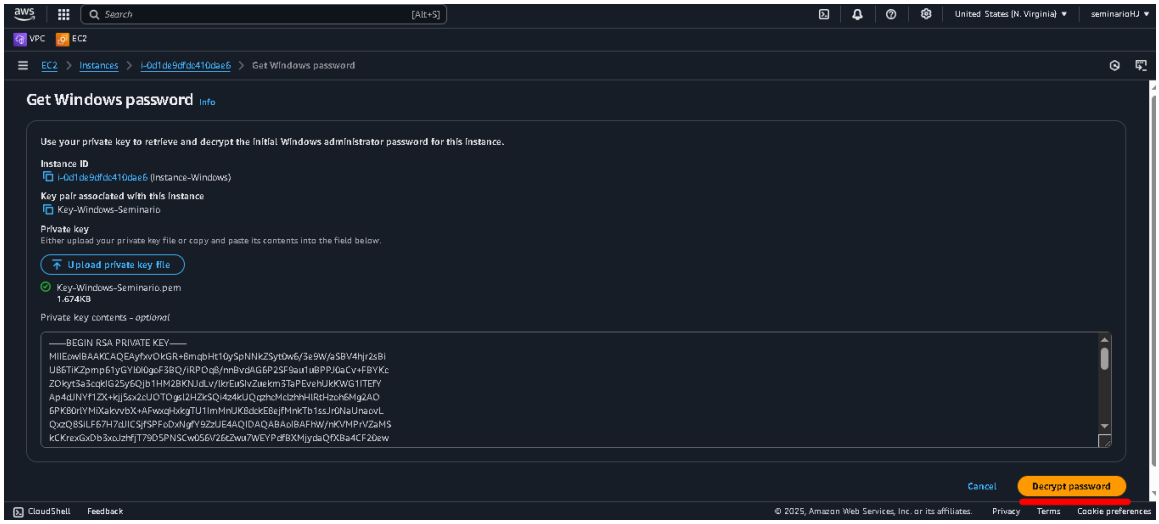



*Figura 30 Entorno de AWS para Descomprimir la Clave*

**4.1.1.7** Aquí vemos que se genera la clave que nos asigna el sistema y esta es la que vamos a usar para poder concluir con la conexión al servidor

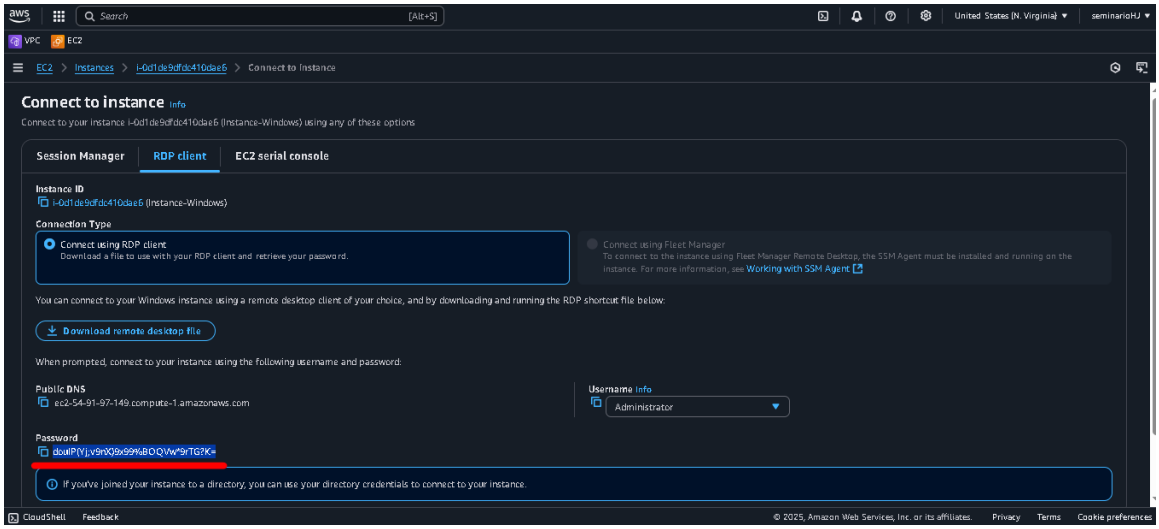



*Figura 31 Entorno de AWS que nos Muestra la Clave que Asigna y el Usuario*

**4.1.1.8** Una vez tenemos los datos, en nuestro aplicativos Conexión a Escritorio Remoto, seleccionamos la opción de conectar y nos va a solicitar el usuario y la contraseña y usamos los que se indican anteriormente y damos en Aceptar

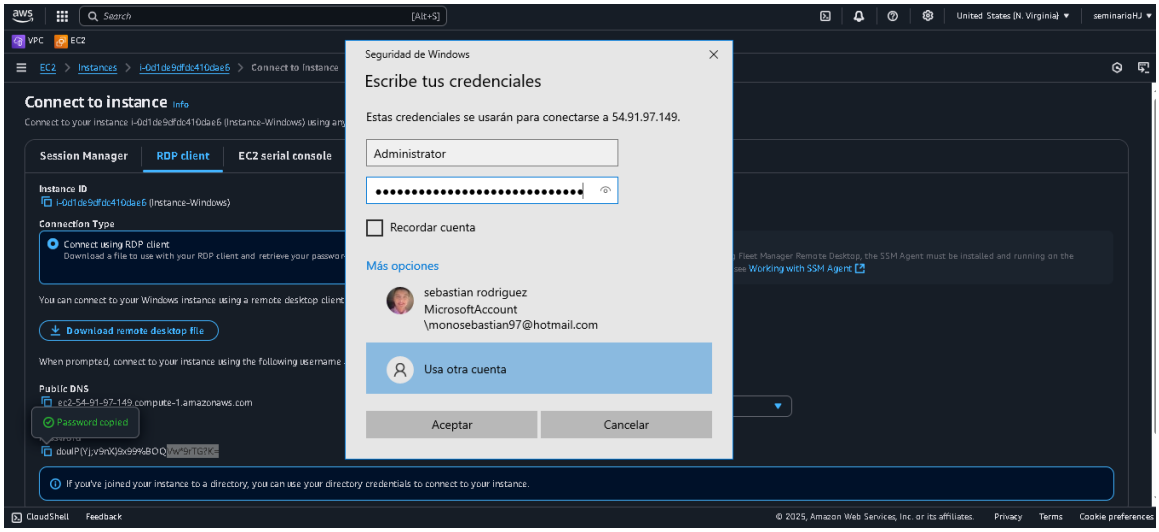



*Figura 32 Imagen del Aplicativo de Escritorio Remoto con el Usuario y la Clave Diligenciados*

**4.1.1.9** Se nos muestra el certificado de seguridad

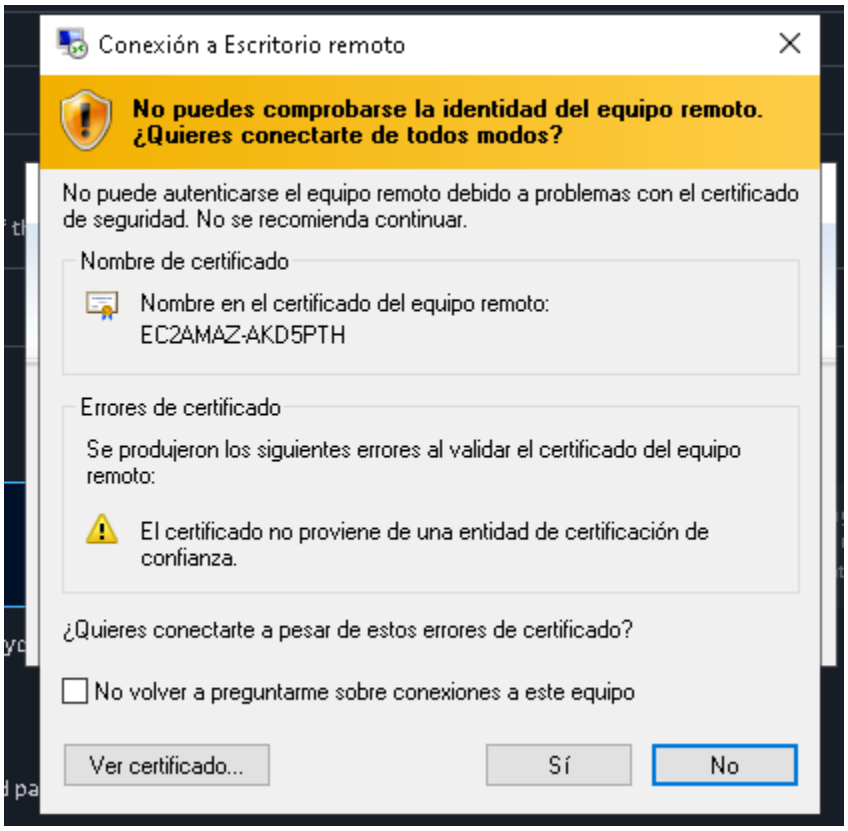



*Figura 33 Imagen Posterior que es un Certificado de confiabilidad*

**4.1.1.10** Tenemos iniciada la máquina virtual.


*Figura 34 Imagen del Escritorio de la Instancia que se Crea en los Servicios de AWS*

## 4.1.2 Conexión a cliente SSH para Linux:

**4.1.2.1** Para acceder al servidor Linux por medio de SSH, nos debemos dirigir a la página principal de EC2, a la opción de la parte izquierda Instances, aquí podemos evidenciar las dos Instancias creadas, en este caso vamos a seleccionar la instancia que lleva por nombre Instance-Linux:

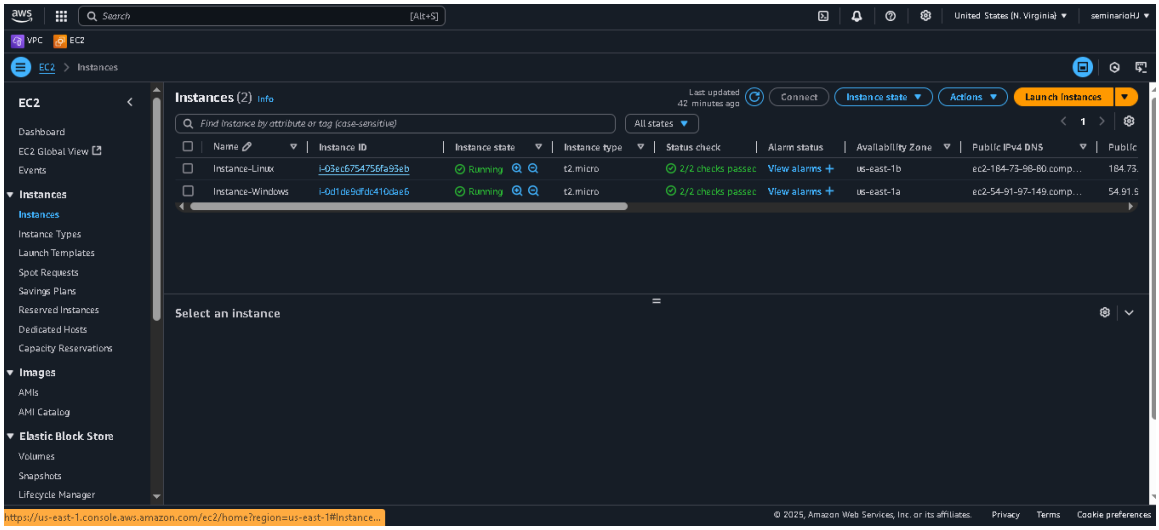



*Figura 35 Entorno de AWS Donde se Visualiza las Instancias Creadas Windows y Linux*

**4.1.2.2** Ingresamos a esta instancia y aquí podemos evidenciar su IP pública en este caso es 184.73.98.80, la cual vamos a utilizar para hacer la conexión

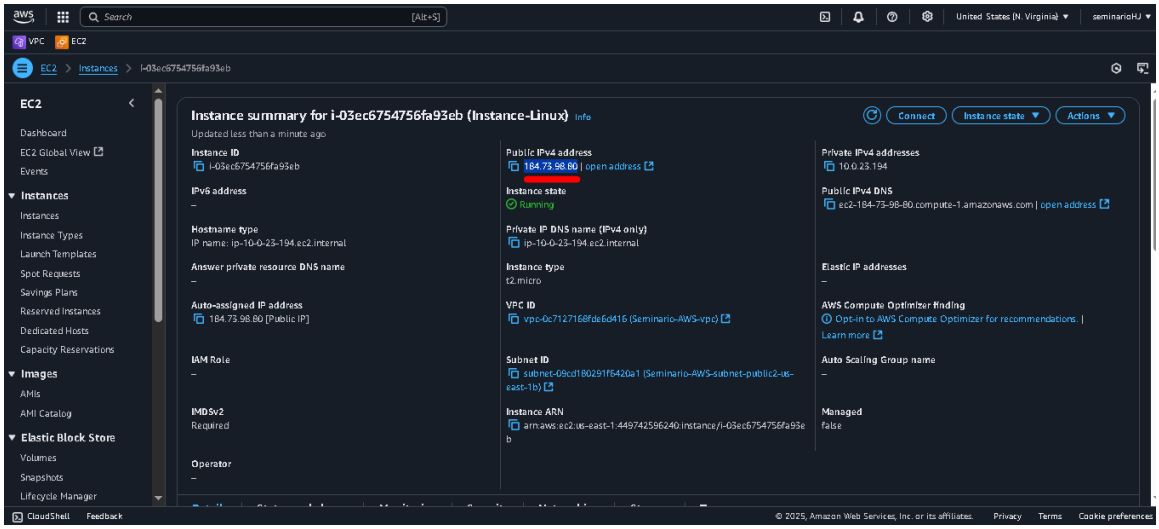



*Figura 36 Entorno de AWS para Ver los Detalles de Conexion a la Instancia*

**4.1.2.3** Una vez estamos en la anterior pantalla vamos a ir a la opción de Connect en la parte superior derecha

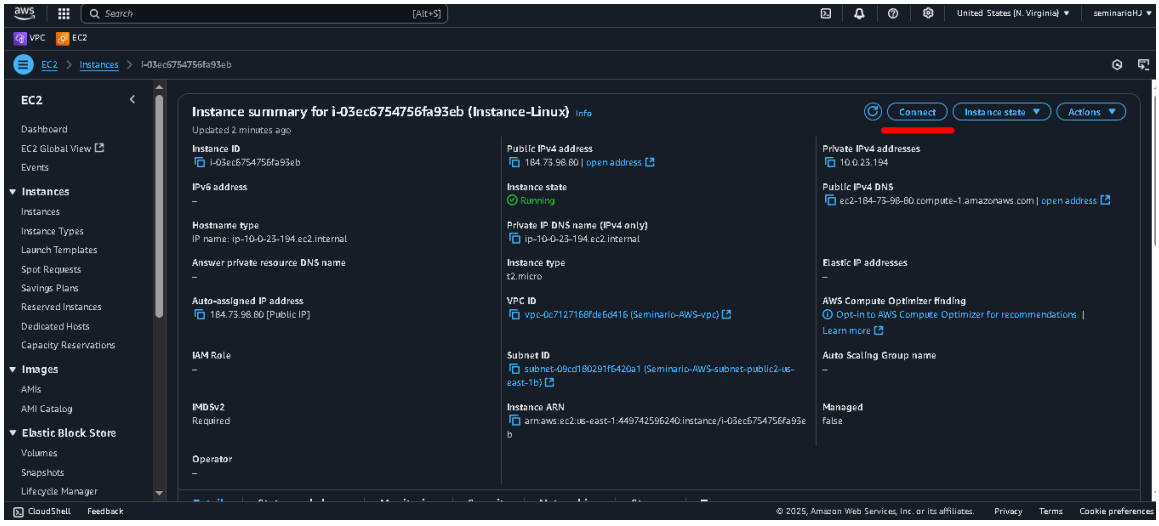



*Figura 37 Entorno de AWS para Ver los Detalles de Conexion a la Instancia*

**4.1.2.4** Aquí se despliega la información de la opción de Connect vamos a ir a la pestaña que dice SSH client

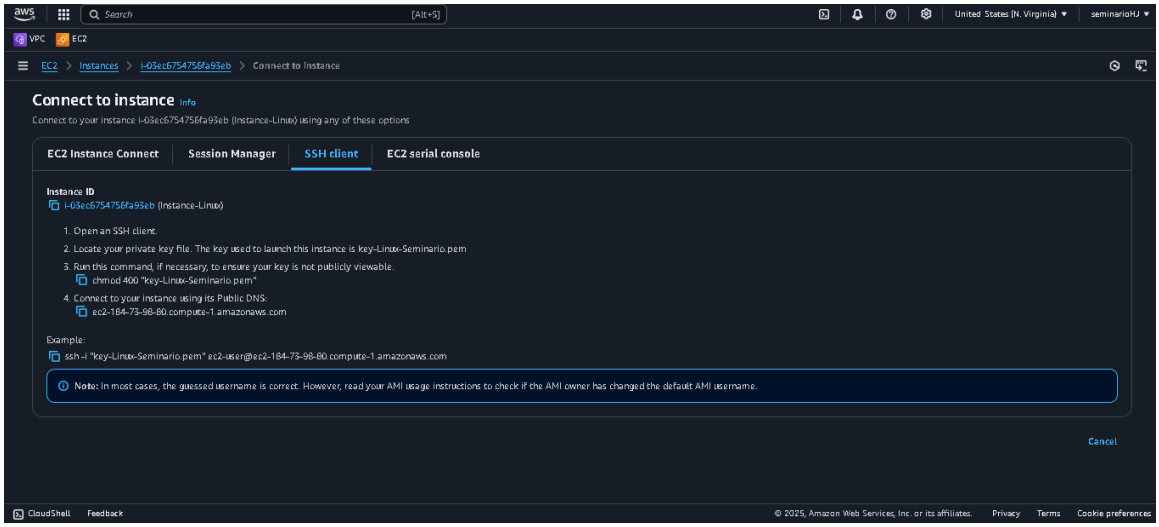



*Figura 38 Entorno de AWS para Ver la Información de Conexión por SSH*

**4.1.2.5** Para podernos conectar, podemos realizarlo de dos maneras, utilizan la IP Pública que se generó automáticamente (184.73.98.80) o también usando la Public IPv4 DNS (ec2-184-73-98-80.compute-1.amazonaws.com) que genera automáticamente el AWS y esta conexión la vamos a realizar por una de tantas aplicaciones que sirven para hacer esta conexión, la que nosotros elegimos es la MobaXterm

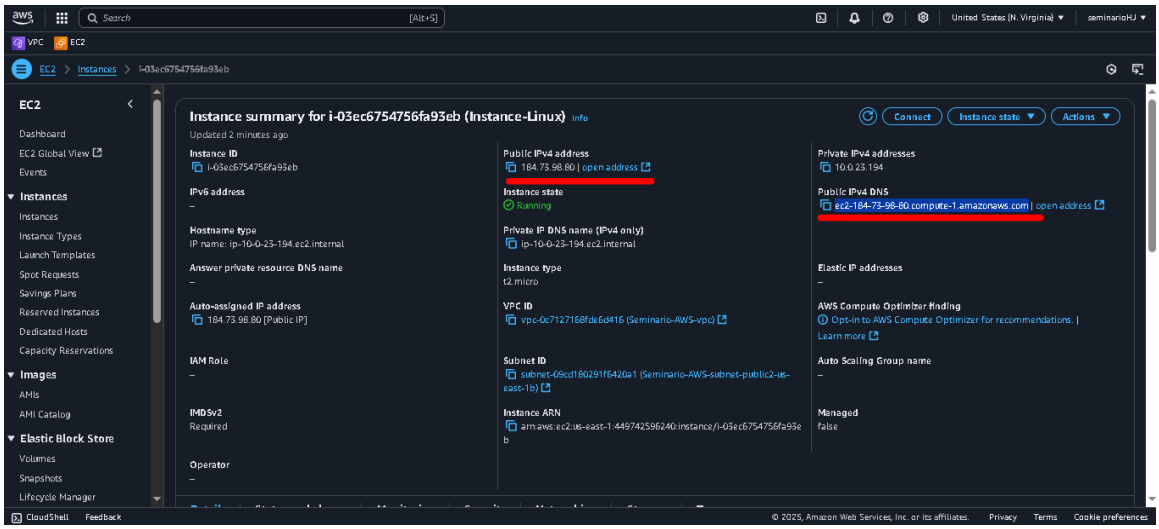



*Figura 39 Entorno de AWS para Ver las Diferentes Opciones de Ip que Usamos para Conectarnos a la Instancia*

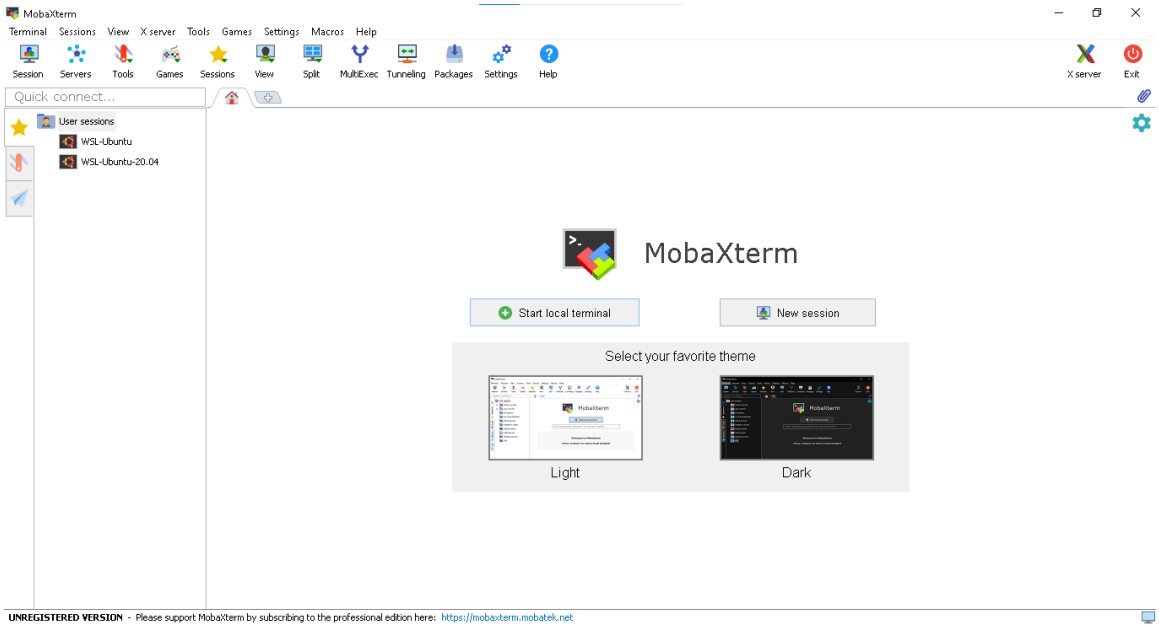



*Figura 40 Imagen del Aplicativo Moba Xterm*

**4.1.2.6** Vamos a ir a la parte que dice Session, damos click en la opción que dice SSH, diligenciamos la opción que dice Remote host y aquí vamos a colocar la IP o el nombre del DNS de la máquina, también diligenciamos la opción que dice Specify username el usuario que vamos a colocar es el de Amazon Linux que es el ec2-user, si se utiliza otra distribución diferente, se debe consultar, cual es el usuario que está utilizando esa distribución por defecto en AWS, para finalizar se diligencia el puerto 22. Adicionalmente se debe seleccionar la opción de configuración avanzada de SSH o Advanced SSH settings y aquí lo que vamos hacer es utilizar nuestro certificado de seguridad o clave key directamente para acceder, entonces marcamos con un check la opción de Use private key y luego damos click en el botón del lado derecho que tiene como icono de una lupa y se nos despliega los archivos de nuestro equipo para seleccionar el certificado de seguridad o clave key de Linux que creamos al inicio de creación de la máquina para finalizar le damos en OK

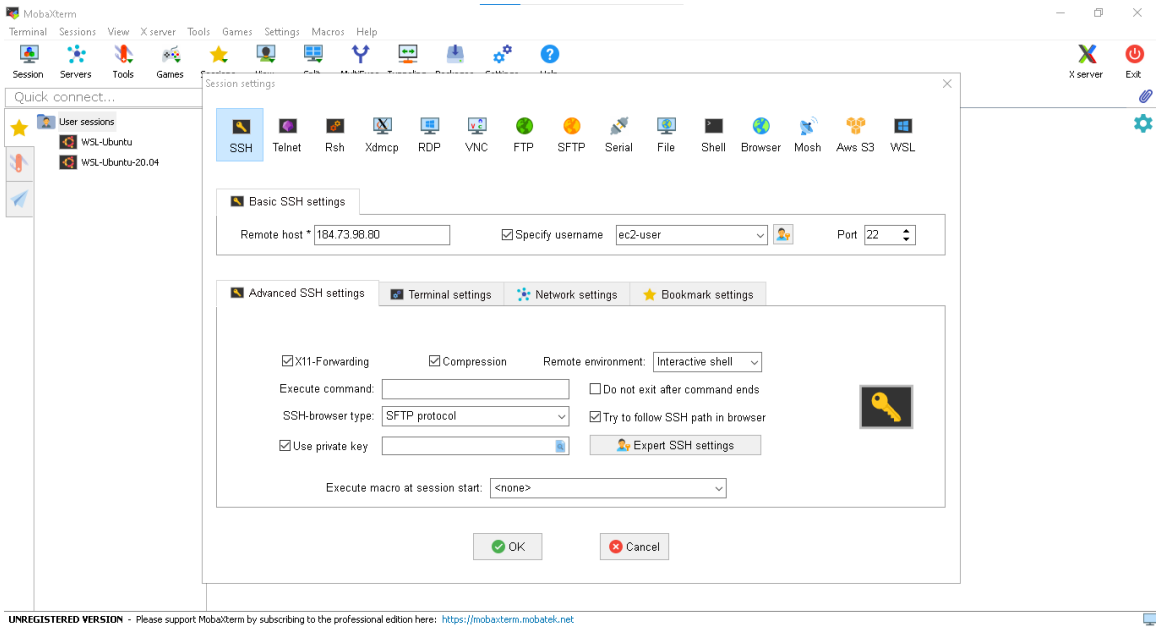



*Figura 41 Imagen de las Opciones a Diligenciar en el Aplicativo Moba Sterm*

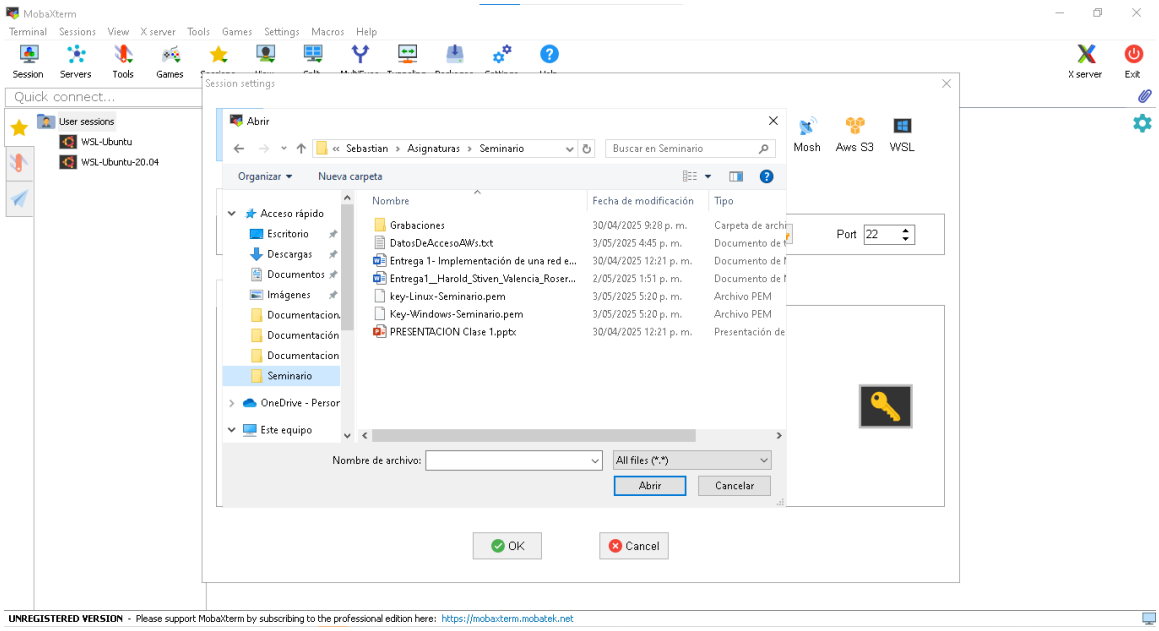



*Figura 42 Imagen del Explorador de Archivos de Donde se Carga la Clave Key Pair*

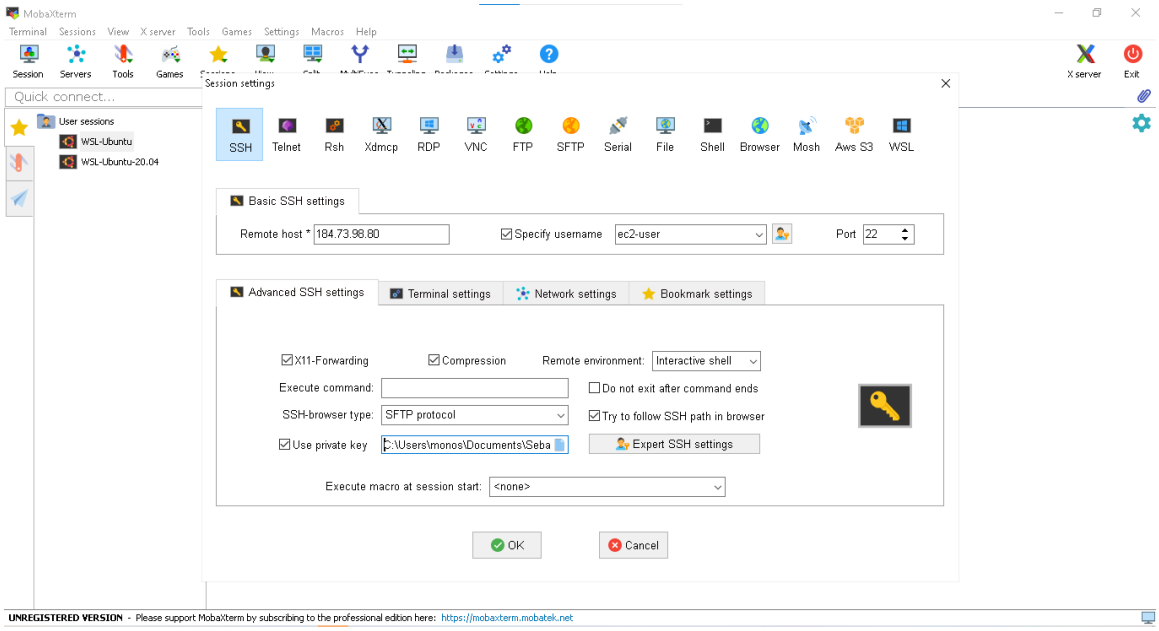



*Figura 43 Imagen del Aplicativo Moba Xtterm con Toda la Información Diligenciada*

### **4.1.2.7** Confirmamos el acceso al servidor Linux

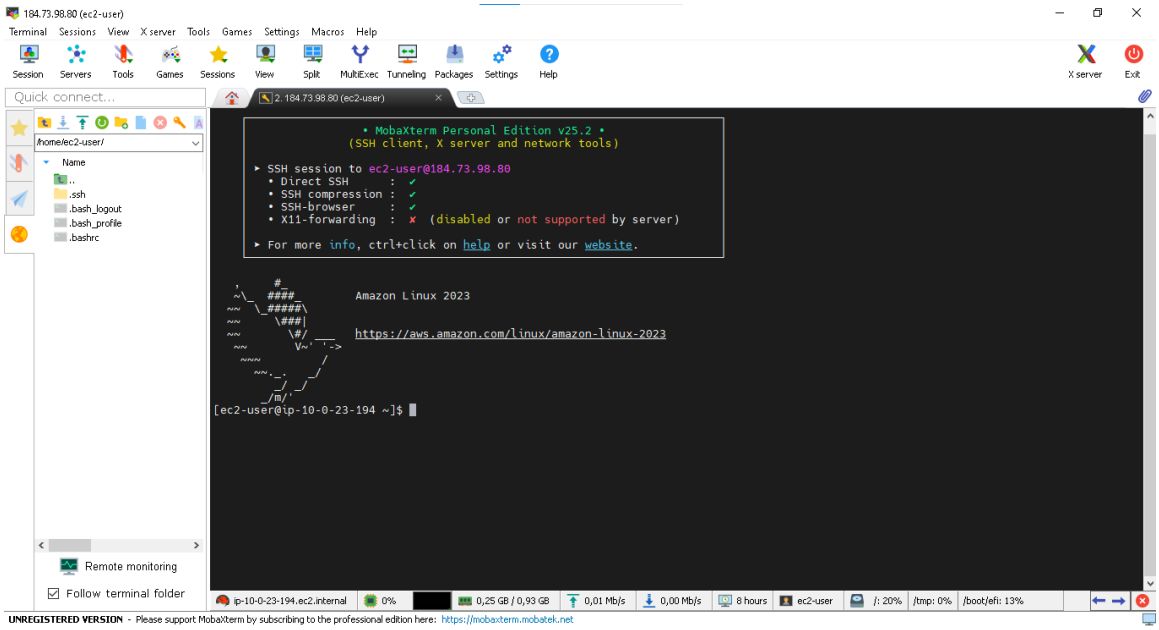



*Figura 44 Imagem de la Conexión por SSH al Servidor Linux desde Moba Xterm*

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

### 4.2.1 Uso de Llaves PEM (clave privada)

- Para acceder de forma segura a la instancia EC2 con sistema operativo Linux, se utiliza un archivo .pem, que corresponde a la clave privada de un par de llaves generado por AWS.
- Este archivo permite la autenticación mediante SSH sin necesidad de utilizar una contraseña.
- La clave pública asociada se almacena automáticamente en la instancia EC2 durante su creación.
- Es fundamental mantener la clave .pem en un lugar seguro, ya que quien posea este archivo puede acceder al servidor.

**Buenas prácticas:**

- Asignar permisos restringidos al archivo, por ejemplo, ejecutar chmod 400 nombre_archivo.pem en sistemas Linux.
- No compartir la clave ni almacenarla en repositorios públicos o sistemas en la nube sin protección.
- En caso de pérdida del archivo .pem, no es posible recuperarlo; se recomienda crear un nuevo par de claves y asociarlo mediante una AMI o una nueva instancia.

### 4.2.2 Contraseñas Seguras (servidor Windows)

- Para acceder al servidor Windows a través de RDP (Remote Desktop Protocol), AWS permite descifrar la contraseña del administrador utilizando el archivo .pem.
- Una vez dentro del sistema, es recomendable cambiar la contraseña del usuario administrador por una más segura.

**Recomendaciones para contraseñas seguras:**

- Al menos 12 caracteres de longitud.
- Incluir letras mayúsculas, minúsculas, números y símbolos.
- Evitar palabras comunes, fechas o datos personales.
- Cambiar periódicamente la contraseña y no reutilizar antiguas.

### 4.2.3 Reglas de Seguridad en el Grupo de Seguridad (Security Group)

- Los grupos de seguridad en AWS actúan como firewalls virtuales que controlan el tráfico hacia y desde las instancias.
- Para minimizar riesgos, se deben configurar reglas de acceso de forma restringida y segura.

# 5. Configuración del servidor web

## 5.1 Pasos seguidos para instalar IIS en Windows Server.

### 5.1.1 Configurar el Rol web server (IIS)

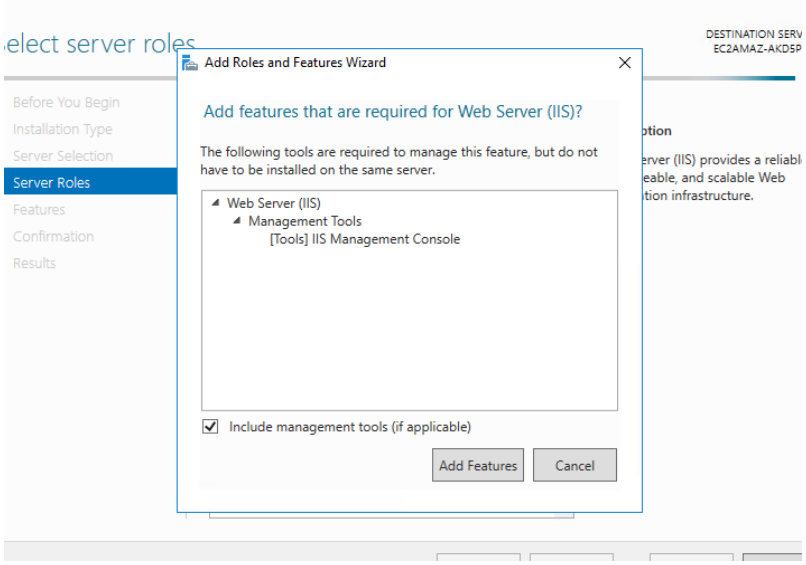



*Figura 45 Imagen de la Configuración del Servidor Web que se Aloja en la Instacia que se Creo, Windows*

## 5.2 Pasos para instalar Apache o Nginx en Linux.

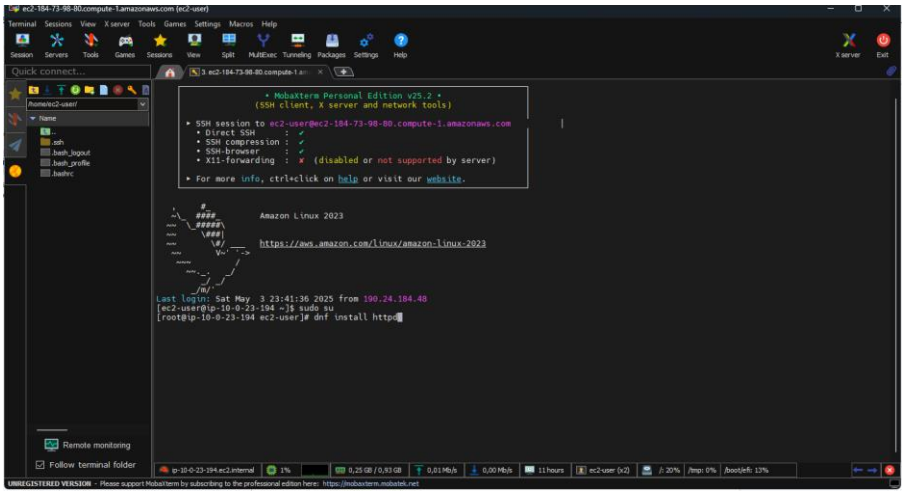



*Figura 46 Imagen del Servidor de Linux Donde se Isntala Nginx*

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

**5.3.1.1** Servidor de windows desde la máquina virtual:

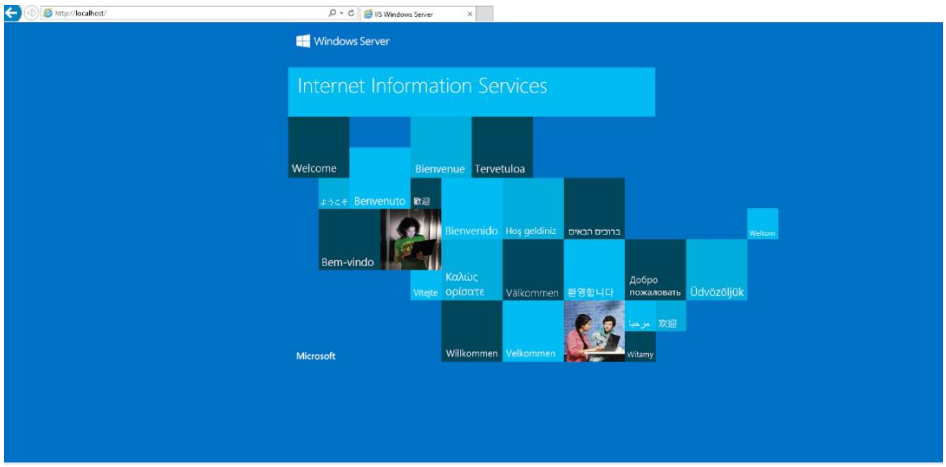



*Figura 47 Imagen de la Funcionalidad del Servidor Web Desde la Instancia Creada, Servidor de Windows*

**5.3.1.2** Servidor de windows fuera de la máquina virtual:

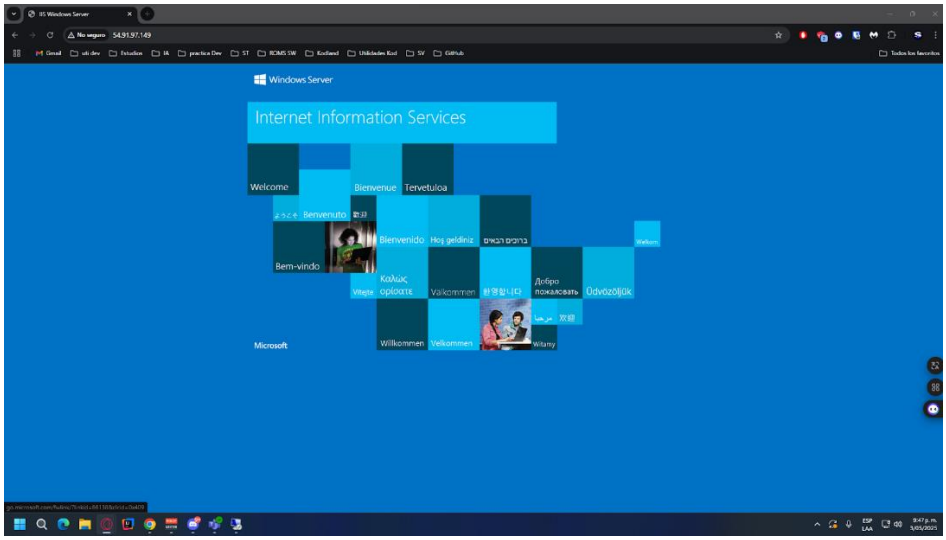



*Figura 48 Imagen de la Funcionalidad del Servidor Web de Windows Desde el Equipo Local*

**5.3.2.1** Servidor de Linux fuera de la máquina virtual:

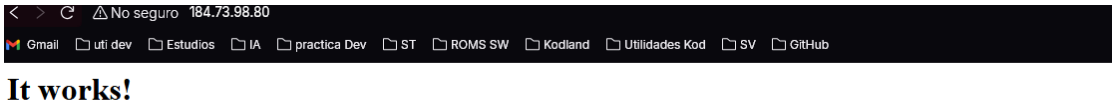



*Figura 49 Imagen de la Funcionalidad del Servidor Web de Linux Desde el Equipo Local*

## 6. Pruebas de conectividad

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

**6.1.1** Desde Linux a windows

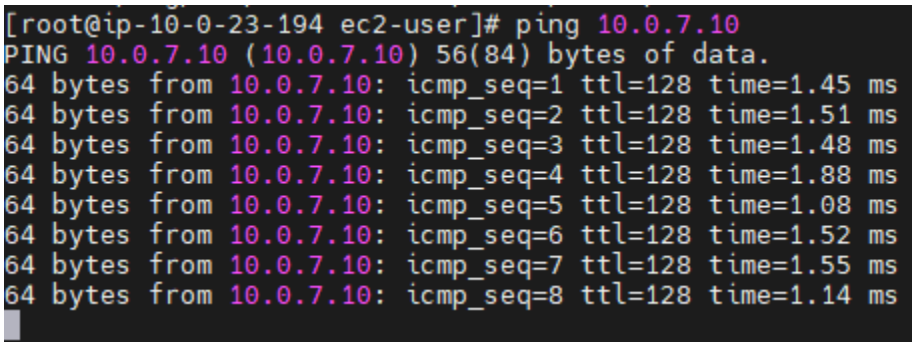



*Figura 50 Imagen del Ping que se Realiza de Linux a Windows*

**6.1.2** Desde windows  a Linux

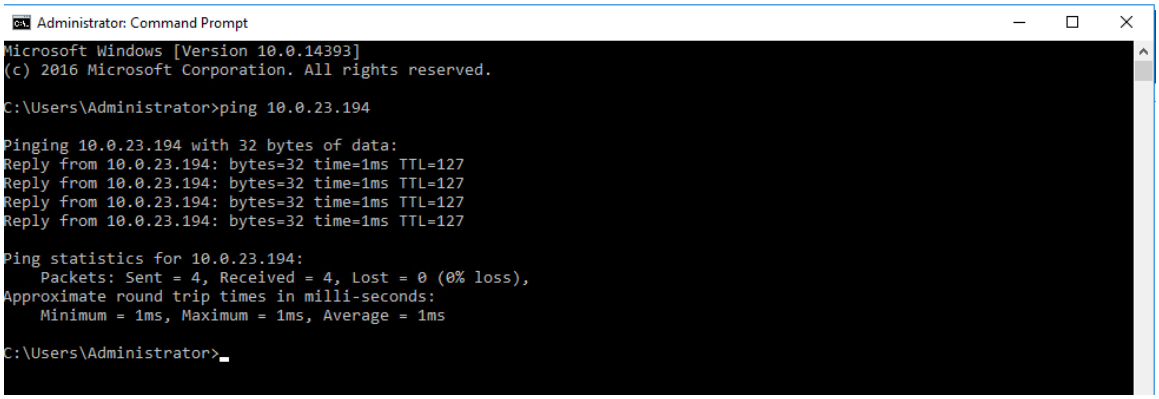



*Figura 51 Imagen del Ping que se Realiza de Windows a Linux*

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

**6.2.1** En el grupo de seguridad se grego otra regla que permita la consulta ICMP

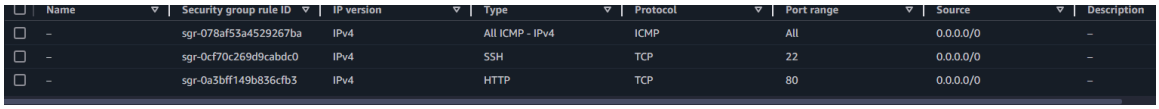



*Figura 52 Imagen del Grupo de Seguridad para las Conexiones*

**6.2.2** Además de activar la regla desde windows :

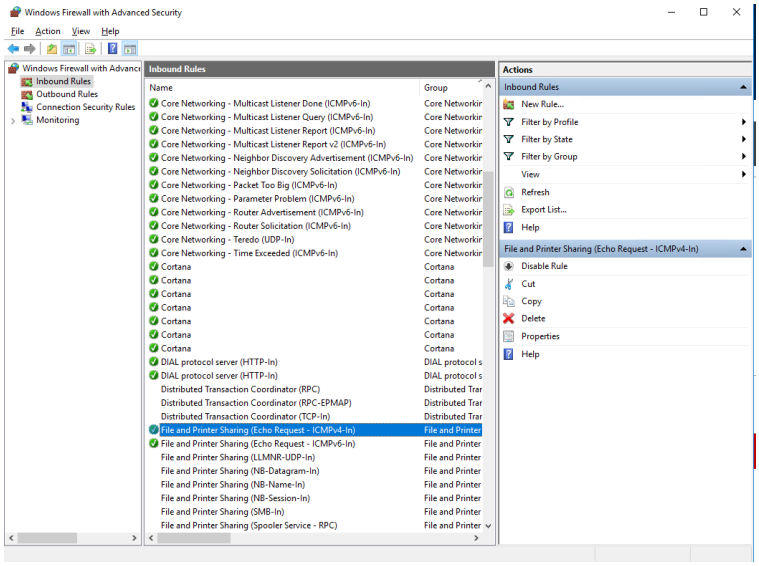



*Figura 53 Imagen de la Configuración de las Reglas del Firewall de la Instancia Windows*

## 7. Validación de acceso web

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

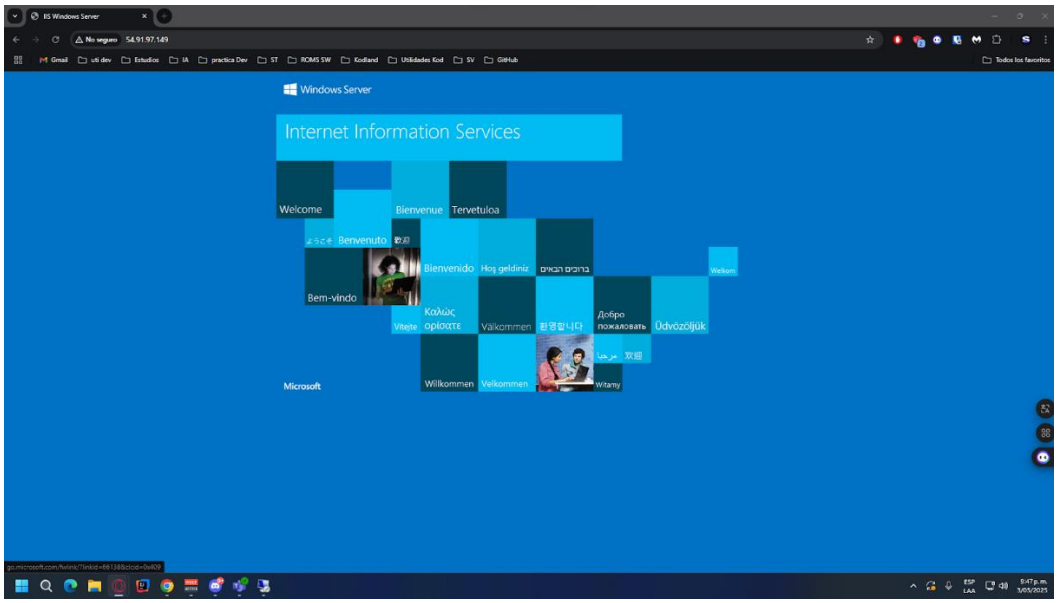
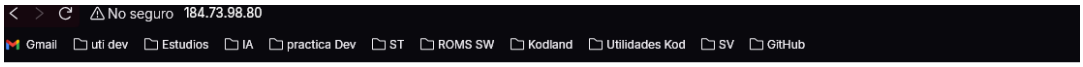



*Figura 54 Imagen de la Funcionalidad del Servidor Web de Windows Desde el Equipo Local*

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




**It works!**

*Figura 55 Imagen de la Funcionalidad del Servidor Web de Linux Desde el Equipo Loca*

## 8. Implementar el servicio de Docker

**8.1**. Instalar Docker en la máquina virtual.

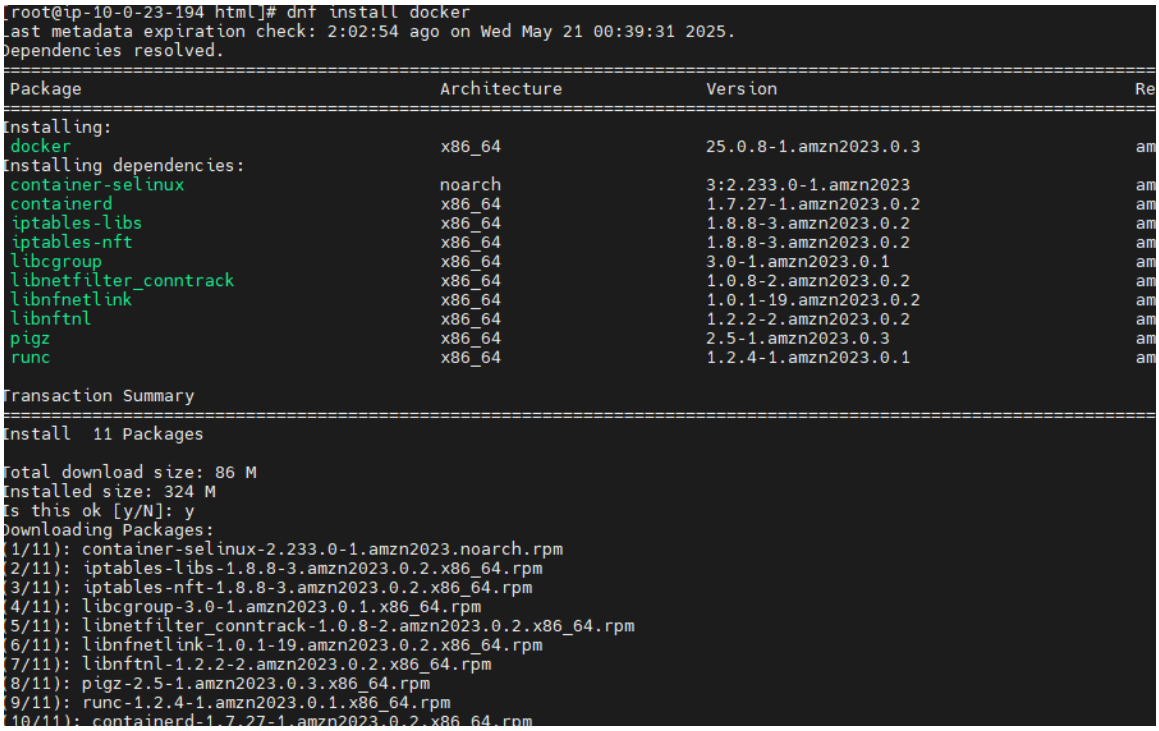



*Figura 56 Imagen de la instalación de Docker en máquina virtual*

**8.2.** Ver el estado del servicio de Docker.

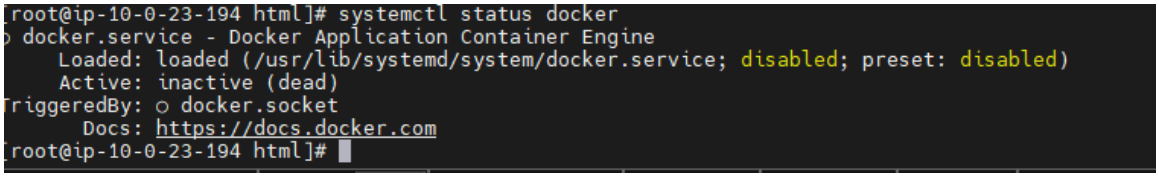



*Figura 57 Imagen del estado de Docker.*

**8.3**. Activar el sistema de Docker

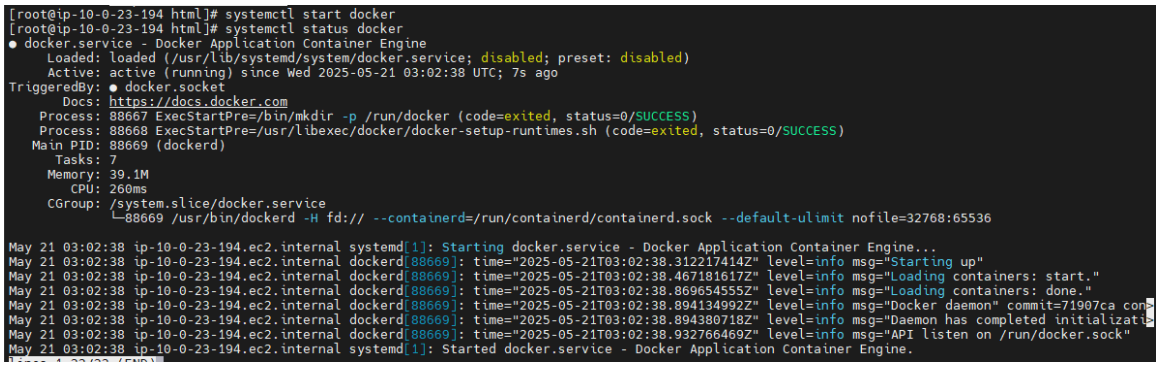



*Figura 58 Imagen de Docker activado y corriendo.*

**8.4** Instalar la imagen de apache en Docker

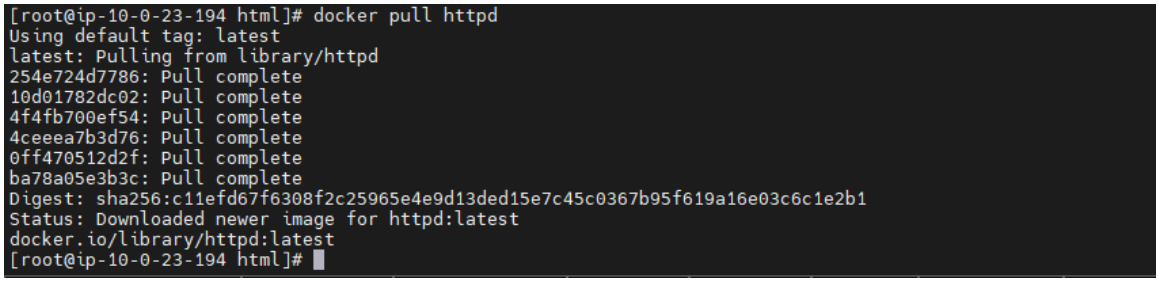



*Figura 59 Imagen de instalación imagen apache.*

**8.5** Creamos la primera aplicación a partir de un servidor web en Docker.

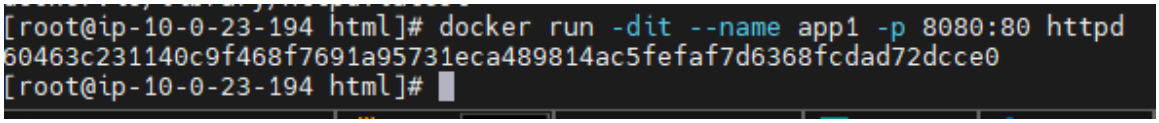



*Figura 60 Imagen de creación de primera aplicación de Docker.*

**8.6** Verificamos el Docker ID

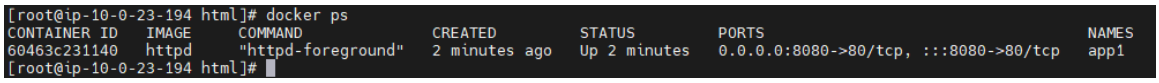



*Figura 61 Imagen de verificación de Docker ID.*

**8.7** Agregar el puerto al SG de la instancia en este caso el puerto 8080, todo depende del puerto que se ponga en la creación del contenedor.

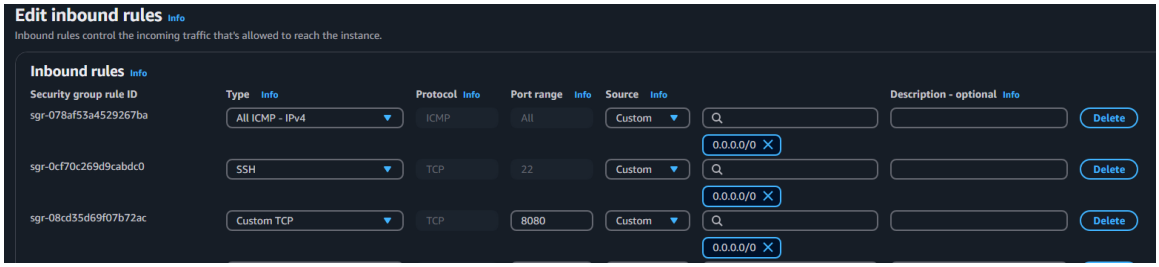



*Figura 62 Imagen de Security Group.*

**8.8** Verificar que el contenedor esta trabajando y se puede acceder desde el puerto.

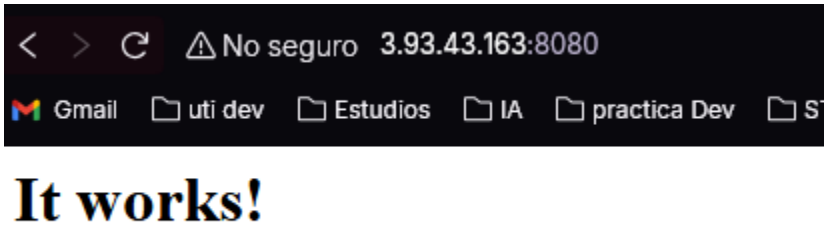



*Figura 63 Imagen de verificación del contenedor.*

**8.9** Cambiamos el It Works por el sitio que queremos subir al contenedor, para eso podemos crear varios contenedores y apuntar a la misma ruta (depende de la imagen que se utiliza nos da la ruta de búsqueda para el sitio). En este caso cree otro contenedor con una nueva configuración para el sitio.

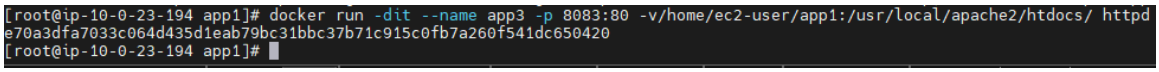



*Figura 64 Imagen de configuración del sitio.*

**8.10** Verificar si el contenedor puede acceder al sitio.

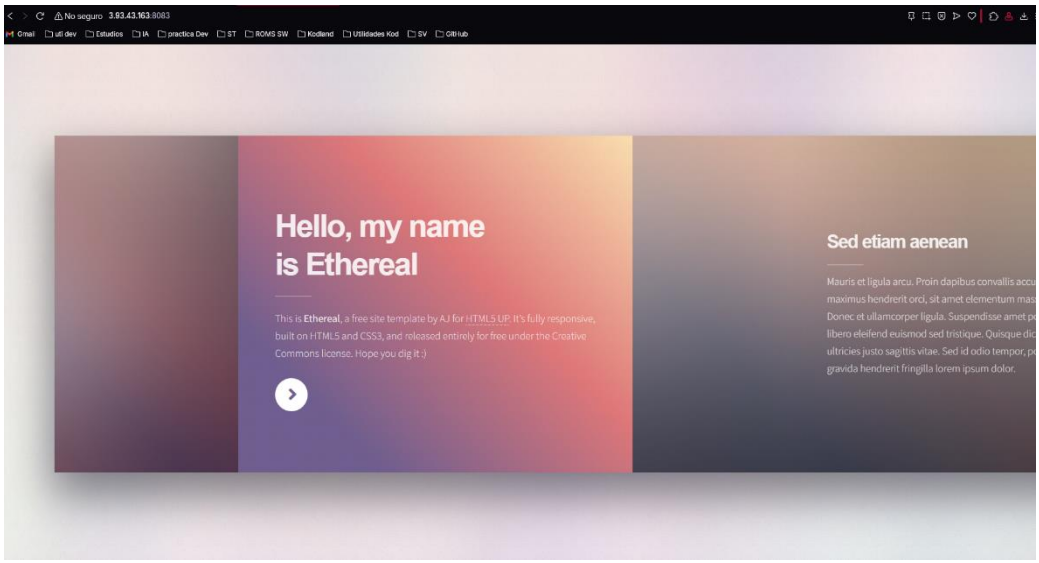



*Figura 65 Imagen de verificación del sitio.*

**8.11** De esa forma podemos crear más contenedores con un sitio estático y podemos ver los que están corriendo.

```
[root@ip-10-0-23-194 app1]# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                                   NAMES
f95d0155d7a9   httpd     "httpd-foreground"   5 minutes ago   Up 5 minutes   0.0.0.0:8084->80/tcp, :::8084->80/tcp   app4
8f929d36a379   httpd     "httpd-foreground"   5 minutes ago   Up 5 minutes   0.0.0.0:8082->80/tcp, :::8082->80/tcp   app2
afad440d2bbc   httpd     "httpd-foreground"   5 minutes ago   Up 5 minutes   0.0.0.0:8080->80/tcp, :::8080->80/tcp   app1
e70a3dfa7033   httpd     "httpd-foreground"   2 hours ago     Up 2 hours     0.0.0.0:8083->80/tcp, :::8083->80/tcp   app3
[root@ip-10-0-23-194 app1]#
```

*Figura 66 Imagen de verificación de los contenedores funcionando.*

## 9. Pruebas de estrés a la instancia Free Tier

**9.1** Creamos un contenedor que consuma 1 vCPU para monitorear su consumo.

```
[root@ip-10-0-23-194 app1]# docker run -dit --name stress1 progrium/stress --cpu 1
87cb0698bcc2af2928ce6f3cf7fb68af6ca0f84d4c62721c820c5b34a0ac8e91
```

*Figura 67 Imagen de creación de contenedor de estres.*

**9.2** Podemos ver el estrés de la instancia con “docker stats”, vemos que contenedor es el que más consume y como afecta a su vCPU.

```
[root@ip-10-0-23-194 app1]# docker stats
CONTAINER ID   NAME      CPU %     MEM USAGE / LIMIT     MEM %     NET I/O         BLOCK I/O        PIDS
87cb0698bcc2   stress1   0.50%     2.41MiB / 949.4MiB    0.25%     726B / 0B       1.99MB / 0B      2
f95d0155d7a9   app4      0.00%     6.422MiB / 949.4MiB   0.68%     25.4kB / 695kB  12.3kB / 4.1kB   82
8f929d36a379   app2      0.00%     7.195MiB / 949.4MiB   0.76%     25.1kB / 695kB  872kB / 4.1kB    82
afad440d2bbc   app1      0.00%     9.086MiB / 949.4MiB   0.96%     37.5kB / 896kB  4.79MB / 4.1kB   82
e70a3dfa7033   app3      0.00%     7.172MiB / 949.4MiB   0.76%     28.6kB / 697kB  774kB / 4.1kB    82
CONTAINER ID   NAME      CPU %     MEM USAGE / LIMIT     MEM %     NET I/O         BLOCK I/O        PIDS
87cb0698bcc2   stress1   0.50%     2.41MiB / 949.4MiB    0.25%     726B / 0B       1.99MB / 0B      2
f95d0155d7a9   app4      0.00%     6.422MiB / 949.4MiB   0.68%     25.4kB / 695kB  12.3kB / 4.1kB   82
8f929d36a379   app2      0.00%     7.195MiB / 949.4MiB   0.76%     25.1kB / 695kB  872kB / 4.1kB    82
afad440d2bbc   app1      0.00%     9.086MiB / 949.4MiB   0.96%     37.5kB / 896kB  4.79MB / 4.1kB   82
e70a3dfa7033   app3      0.00%     7.172MiB / 949.4MiB   0.76%     28.6kB / 697kB  774kB / 4.1kB    82
CONTAINER ID   NAME      CPU %     MEM USAGE / LIMIT     MEM %     NET I/O         BLOCK I/O        PIDS
87cb0698bcc2   stress1   97.14%    2.41MiB / 949.4MiB    0.25%     726B / 0B       1.99MB / 0B      2
f95d0155d7a9   app4      0.00%     6.422MiB / 949.4MiB   0.68%     25.4kB / 695kB  12.3kB / 4.1kB   82
8f929d36a379   app2      0.00%     7.195MiB / 949.4MiB   0.76%     25.1kB / 695kB  872kB / 4.1kB    82
afad440d2bbc   app1      0.00%     9.086MiB / 949.4MiB   0.96%     37.5kB / 896kB  4.79MB / 4.1kB   82
e70a3dfa7033   app3      0.00%     7.172MiB / 949.4MiB   0.76%     28.6kB / 697kB  774kB / 4.1kB    82
CONTAINER ID   NAME      CPU %     MEM USAGE / LIMIT     MEM %     NET I/O         BLOCK I/O        PIDS
87cb0698bcc2   stress1   97.14%    2.41MiB / 949.4MiB    0.25%     726B / 0B       1.99MB / 0B      2
f95d0155d7a9   app4      0.00%     6.422MiB / 949.4MiB   0.68%     25.4kB / 695kB  12.3kB / 4.1kB   82
8f929d36a379   app2      0.00%     7.195MiB / 949.4MiB   0.76%     25.1kB / 695kB  872kB / 4.1kB    82
afad440d2bbc   app1      0.00%     9.086MiB / 949.4MiB   0.96%     37.5kB / 896kB  4.79MB / 4.1kB   82
e70a3dfa7033   app3      0.00%     7.172MiB / 949.4MiB   0.76%     28.6kB / 697kB  774kB / 4.1kB    82
CONTAINER ID   NAME      CPU %     MEM USAGE / LIMIT     MEM %     NET I/O         BLOCK I/O        PIDS
87cb0698bcc2   stress1   99.97%    2.41MiB / 949.4MiB    0.25%     726B / 0B       1.99MB / 0B      2
f95d0155d7a9   app4      0.00%     6.422MiB / 949.4MiB   0.68%     25.5kB / 695kB  12.3kB / 4.1kB   82
8f929d36a379   app2      0.01%     7.195MiB / 949.4MiB   0.76%     25.1kB / 695kB  872kB / 4.1kB    82
afad440d2bbc   app1      0.00%     9.086MiB / 949.4MiB   0.96%     37.5kB / 896kB  4.79MB / 4.1kB   82
e70a3dfa7033   app3      0.00%     7.172MiB / 949.4MiB   0.76%     28.6kB / 697kB  774kB / 4.1kB    82
CONTAINER ID   NAME      CPU %     MEM USAGE / LIMIT     MEM %     NET I/O         BLOCK I/O        PIDS
87cb0698bcc2   stress1   99.97%    2.41MiB / 949.4MiB    0.25%     726B / 0B       1.99MB / 0B      2
```

*Figura 68 Imagen de estadísticas de consumo de vCPU.*

## Conclusiones

- Entender cómo se realiza la estructura de red en un ambiente de nube permite ampliar la perspectiva sobre cómo los avances tecnológicos facilitarán los procesos en

medianas y pequeñas empresas, permitiendo optimizar sus ganancias y ahorrar en dispositivos innecesarios.

- Se logró una implementación exitosa del despliegue de instancias en AWS para comprobar su accesibilidad desde la red. Estas instancias fueron de sistemas operativos Windows y Linux.
- Creamos una VPC con la configuración adecuada en las instancias para poder acceder mediante los protocolos SSH, RDP y HTTP. También nos aseguramos de configurar la seguridad mediante un Security Group, el cual permite el acceso a los puertos establecidos.
- Se realizaron pruebas de conectividad a los servicios de Amazon Web Services (AWS) desde nuestros equipos locales, lo cual generó un sentimiento de satisfacción al ver en acción estas nuevas tecnologías.
- Instalamos contenedores Docker en la instancia de Linux, lo cual nos permitió poner en marcha un servidor web escalable y práctico con un sitio estático y poco demandante.
- Se utilizaron las instancias Free Tier, las cuales son gratuitas en la plataforma de AWS y hacen posible la práctica de este ejercicio. Observamos que estas máquinas virtuales no están destinadas a un uso riguroso, sino únicamente a un nivel académico y con propósito de estudio.
- Se realizaron pruebas de disponibilidad con los componentes de Docker para alojar un servidor web y confirmar que, si uno fallaba, el otro podía brindar respaldo.
- La computación en la nube o Cloud Computing que maneja Amazon Web Services (AWS) actualmente ofrece muchas garantías a los usuarios en cuanto a disponibilidad y seguridad.
- Se crearon balanceadores de carga que ayudan precisamente a mantener la estabilidad cuando nuestra página web comienza a recibir muchas solicitudes. Esto evita que se sature y previene una posible caída.
- En la actualidad se evidencia que los cambios tecnológicos están transformando la forma en que se mueve el mundo de la tecnología en los campos empresariales. Las industrias, si quieren ahorrar dinero, deben dejar de invertir en dispositivos físicos que con el tiempo se volverán obsoletos, y migrar sus servicios hacia tecnologías de vanguardia que permitan mantener una ventaja competitiva mediante una alta disponibilidad.
- Este trabajo es una introducción a lo que se avecina en el futuro de los data centers. Estas infraestructuras, que antes eran motivo de orgullo para las empresas, pasan ahora a un segundo plano, pues lo más conveniente es optar por la renta de espacios y componentes de última tecnología, que ofrecen mejores rendimientos sin los gastos ni mantenimientos que generan los dispositivos físicos.

## Referencias

De Trabajo, D., & Moreno, M. S. (2015). *UNIVERSIDAD DEL CEMA Buenos Aires Argentina Serie*. www.cema.edu.ar/publicaciones/doc_trabajo.html

Fernando, C., Pérez, V., Enrique, J., Cleves, P., & Pallares, L. (n.d.). *Volumen especial-E-ISSN: 2248-762X Universidad Distrital Francisco José de Caldas INFORMACIÓN Y LA COMUNICACIÓN CLOUD COMPUTING: A NEW PARADIGM IN INFORMATION AND COMMUNICATION TECHNOLOGIES*. http://revistas.udistrital.edu.co/ojs/index.php/REDES/index

Jiménez Candela, M., & Fornes Juan, J. DE. (2024). *OPTIMIZACIÓN DE INFRAESTRUCTURA CLOUD: MIGRACIÓN DE EC2 A UN ENTORNO ECS EN AWS*.

Martel, C. S. (n.d.). *Despliegue de una aplicación Ruby on Rails utilizando las tecnologías de virtualización Docker y CoreOS en la nube pública de Amazon Web Services*.

Mukherjee, S. (n.d.). *Benefits of AWS in Modern Cloud*.

Ortiz Clavijo, L. F., Fernández Ledesma, J. D., Cadavid Nieto, S., & Gallego Duque, C. J. (2018). Computación en la Nube: Estudio de herramientas orientadas a la Industria 4.0. *Lámpsakos*, *20*, 68–75. https://doi.org/10.21501/21454086.2560

Docker Inc. (2023). *Documentación de Docker*. Recuperado de https://docs.docker.com/

Progrium. (2016). *Docker image stress para pruebas de carga*. Docker Hub. Recuperado de https://hub.docker.com/r/progrium/stress/