TRABAJO DE GRADO
Opción Seminario-Diplomado.

Amazon Web Services (AWS)

Oscar Iván Rivero Henao
Daniel Mira Sepúlveda
Yoleison Valencia Lemos

Corporación Universitaria Remington
Facultad de Ingenierías
Ingeniería de Sistemas

Nombre del Tutor Seminario: Juan Pablo Berrio López
Opción de Trabajo de grado Seminario
Año 2024

Tabla de Contenidos

Índice de Figuras 3
Índice de Tablas 6
Resumen 7
Marco conceptual y contextual 9
Desarrollo e implementación del aprendizaje 11
Conclusiones 88
Referencias 89

Índice de Figuras

Figura 1 Línea de tiempo 16
Figura 2 Línea de tiempo 16
Figura 3 Crear VPC 23
Figura 4 Instancia 24
Figura 5 Máquina Virtual Linux 24
Figura 6 Llave 25
Figura 7 Connect 26
Figura 8 SSH Client 27
Figura 9 WSL 28
Figura 10 MV Linux Creada 28
Figura 11 Crear Docker 29
Figura 12 Configuración puertos 30
Figura 13 Ingreso ruta local 30
Figura 14 Configuración IP 31
Figura 15 Crear Directorio nginx 31
Figura 16 Codificación 32
Figura 17 Prueba contenedor 1 33
Figura 18 Prueba Contenedor 2 34
Figura 19 Crear Bucket 35
Figura 20 Resumen Creación Bucket 36
Figura 21 Cargue archivo DEFINICIONES 36
Figura 22 Instalación AWS CLI 37
Figura 23 Uso de Comandos 38
Figura 24 Correcta Instalación AWS CLI 38
Figura 25 Creación usuario CLI 39
Figura 26 Asignación contraseña usuario CLI 39
Figura 27 Creación grupo 40
Figura 28 Asignación de permisos usuario 40
Figura 29 Asignación permisos grupo 41
Figura 30 Autenticación de doble factor 42
Figura 31 Copiar variables de acceso 42
Figura 32 Uso comando s3 ls 43
Figura 33 Cargue archivo al bucket 44
Figura 34 Resumen cargue bucket 44
Figura 35 Conexión por RDS a base de datos 45
Figura 36 Creación BD MySQL 45
Figura 37 Versión motor y capa 46
Figura 38 Configuración BD 46
Figura 39 Asignación clave BD 47
Figura 40 Selección disco y CPU 47
Figura 41 Selección VPC y subred 48
Figura 42 Selección tipo acceso a BD 48

Figura 43 Resumen BD creadas 49
Figura 44 Copia punto enlace BD 49
Figura 45 Ingreso a grupo 50
Figura 46 Uso aplicación Heidi 51
Figura 47 Conexión a BD creada desde Heidi 51
Figura 48 Creación BD "prueba1" 52
Figura 49 Resumen conexión 52
Figura 50 Creación instancia 53
Figura 51Selección sistema operativo 54
Figura 52 Uso clave 54
Figura 53 Selección VPC 55
Figura 54 Resumen instancia "Servidor1" 56
Figura 55 Archivo escritorio remoto 56
Figura 56 Acceso a conexión remota 57
Figura 57 Ingreso a server manager 58
Figura 58 Habilitar servicio internet information server 58
Figura 59 Cargue página de inicio 59
Figura 60 Configuración grupo de seguridad 60
Figura 61 Conexión desde navegador 60
Figura 62 Modificación página inicio "Servidor1" 61
Figura 63 Creación instancia 62
Figura 64 Creación snapshot 62
Figura 65 Crear imagen a partir de Snapshot 63
Figura 66 Resumen imagen creada 63
Figura 67 Creación segunda instancia "server2" 64
Figura 68 Clave instancia 65
Figura 69 Selección VPC y subred 65
Figura 70 Resumen y estado de instancias 66
Figura 71 Conexión a "server2" 66
Figura 72 Conexión desde IP pública 67
Figura 73 Creación balanceador de carga 68
Figura 74 Asignación nombre balanceador 68
Figura 75 Selección VPC 69
Figura 76 Creación Target Group 70
Figura 77Configuración Target Group 70
Figura 78 Selección instancia a incluir en TG 71
Figura 79 Creación grupo TG-Remington 72
Figura 80 Balanceador con 2 instancias creadas 72
Figura 81 DNS 73
Figura 82Asignación reglas 73
Figura 83 Prueba de conexión local Servidor1 74
Figura 84 Prueba de conexión local Servidor2 74
Figura 85 Creación grupo Auto Scaling 75
Figura 86 Creación plantilla de lanzamiento 75

Figura 87 Seleccionar VPC y subredes 76
Figura 88 Asociar balanceador y TG 77
Figura 89 Selección imagen a usar 77
Figura 90 Selección tipo CPU y clave 78
Figura 91 Selección grupo de seguridad 79
Figura 92 Asignación de IP para conexión remota 80
Figura 93 Selección plantilla de lanzamiento 80
Figura 94 Selección VPC 81
Figura 95 Asociar balanceador y TG 82
Figura 96 Selección de capacidades en auto scaling 83
Figura 97 Configuración política de escalado 84
Figura 98 Resumen de instancias 84
Figura 99 Uso Silver Bench para estresar CPU 85
Figura 100 Resumen de instancias en modo estrés 86
Figura 101 Valores escalado automático 86
Figura 102 Estado de funcionamiento escalado automático 87

Índice de Tablas

Tabla 1 18

Resumen

El presente trabajo de grado se centra en el estudio de la plataforma Amazon Web Services (AWS), teniendo en cuenta que es una de las plataformas lideres a nivel mundial de servicios en la nube, resaltando la importancia que tiene en el desarrollo tecnológico de las empresas ofreciendo servicios que ayudan a estas a optimizar sus recursos, mantener segura la información y tener una alta disponibilidad de sus servicios informáticos.

En primera instancia se abordaron los orígenes de la virtualización y la computación en la nube, realización de una línea de tiempo con los sucesos más importantes que contribuyeron desde sus inicios hasta la actualidad al desarrollo de estas tecnologías, además se muestra en una tabla comparativa los servicios más importantes ofrecidos por las tres principales plataformas de servicios en la nube en la actualidad como son: Amazon web Services (AWS), Microsoft Azure y Google Cloud Platform (GCP).

El estudio de la plataforma AWS se realizó elaborando un paso a paso con imágenes descriptivas de cómo se implementan algunos de los servicios más importantes tales como:

1. Creación de Virtual Private Cloud (VPC), Subredes, Instancias EC2, Docker o contenedores dentro de la instancia y configuración para acceder desde un equipo local conectado a Internet de manera directa a través de la instancia a la página de inicio del servidor web de cada contenedor por su nombre de dominio.

2. Conexión a través del Subsistema de Windows para Linux (WSL) a una instancia Linux en AWS, creación de un Bucket en el cual se realiza el cargue de archivos S3 y conexión a una base de datos RDS MySQL.

3. Creación de un balanceador de cargas con 2 Instancias y configuración de un grupo de Autoescalado para que cree o elimine automáticamente las Instancias dependiendo del promedio del uso de CPU de estas.

Palabras clave

Amazon Web Services, computación en la nube, instancias, contenedor, máquina virtual, base de datos.

Marco conceptual y contextual

Identity and Access Management (IAM)

Es un servicio web que lo ayuda a controlar de forma segura el acceso a los recursos de AWS. Con IAM, puede administrar de forma centralizada los permisos que controlan a qué recursos de AWS pueden acceder los usuarios. Utilice IAM para controlar quién está autenticado (ha iniciado sesión) y autorizado (tiene permisos) para utilizar recursos. (Amazon, 2024)

Amazon S3 (Simple Storage Service)

Es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento líderes del sector. Los clientes de todos los tamaños y sectores pueden utilizar Amazon S3 para almacenar y proteger cualquier cantidad de datos para diversos casos de uso, tales como lagos de datos, sitios web, aplicaciones móviles, copia de seguridad y restauración, archivado, aplicaciones empresariales, dispositivos IoT y análisis de big data. Amazon S3 proporciona funciones de gestión para que pueda optimizar, organizar y configurar el acceso a sus datos para satisfacer sus requisitos empresariales, organizativos y de conformidad específicos. (Amazon, 2024)

Amazon RDS (Relational Database Service)

Es un servicio de bases de datos relacionales fácil de administrar, optimizado para el costo total de propiedad. Es fácil de configurar, operar y escalar según la demanda. Amazon RDS automatiza las tareas indiferenciadas de administración de bases de datos, como el

aprovisionamiento, la configuración, las copias de seguridad y la aplicación de revisiones. (Amazon, 2024)

Amazon EC2 (Elastic Compute Cloud)

Proporciona capacidad de computación escalable bajo demanda en la nube de Amazon Web Services (AWS). El uso de Amazon EC2 reduce los costos de hardware para que pueda desarrollar e implementar aplicaciones con mayor rapidez. Puede usar Amazon EC2 para lanzar tantos servidores virtuales como necesite, configurar la seguridad y las redes, y administrar el almacenamiento. Puede agregar capacidad (escalar verticalmente) para gestionar tareas que requieren mucha computación, como los procesos mensuales o anuales, o los picos de tráfico del sitio web. (Amazon, 2024)

Amazon EBS (Elastic Block Store)

Se refiere a almacenamiento en bloque persistente diseñado para proporcionar al espacio de datos persistentes para instancias de Amazon EC2. Ofrece diferentes tipos de volúmenes los cuales poseen características específicas de rendimiento y precio como volúmenes de propósito general. (Amazon, 2024)

Desarrollo e implementación del aprendizaje

El desarrollo de la temática del seminario se basó en la entrega de varias actividades con diferentes temáticas explicadas en cada una de las clases, las cuales se evidencian a continuación:

Entrega 1

Investigar cuáles fueron los orígenes de la virtualización y la computación en la nube, haga una línea de tiempo con los diferentes hitos para que hoy en día exista el nivel tecnológico que hemos alcanzado en la nube.

Realizar una comparación de nombres de servicios entre AWS, Azure y GCP.

1. Origen de la virtualización y la computación en la nube.

¿Qué es la virtualización?

La virtualización es un proceso que permite crear múltiples entornos de procesamiento simulado desde un pool de recursos de sistema físicos. En general, se usa para ejecutar múltiples sistemas operativos en el mismo sistema de hardware al mismo tiempo. (Hpe, 2024)

Historia de la virtualización

El origen de la virtualización data de los días de los grandes PC mainframe en la década de los 60, cuando cada una de estas masivas piezas de hardware solo podía funcionar en un

proceso a la vez. Con el tiempo, los clientes comenzaron a exigir que estas importantes inversiones pudieran admitir a más de un usuario o proceso a la vez. A fines de la década de los 60, IBM desarrolló el sistema operativo CP-67, un hipervisor inicial que introdujo la memoria virtual a la familia de servidores System 360 de la empresa. Sin embargo, se desarrollaron otras soluciones para que múltiples usuarios trabajen en un solo servidor, y la virtualización languideció como tecnología de nicho durante varias décadas. (Hpe, 2024)

En la década de los 90, cuando muchas empresas tenían el desafío de mantenerse al día con pilas de TI de un proveedor y aplicaciones heredadas, se dieron cuenta de la necesidad de usar mejor sus recursos de servidores, que con frecuencia no se utilizaban lo suficiente. Al adoptar la virtualización, no solo particionaron su infraestructura de servidor de manera más eficiente, sino que ejecutaron sus aplicaciones heredadas en tipos y versiones de OS diferentes. Debido a su gran red compuesta de muchos tipos diferentes de PC ejecutando diferentes sistemas operativos, el crecimiento de Internet ayudó a impulsar la adopción de la virtualización. A medida que la virtualización se empezó a utilizar con mayor frecuencia, redujo la dependencia del proveedor para servidores y fue la base para el desarrollo de la computación en la nube. (Hpe, 2024)

¿Cómo funciona la virtualización?

La virtualización es posible gracias a una capa de software llamada hipervisor. Este software abstrae los recursos de su sistema host (sea una CPU, GPU, memoria, espacio de almacenamiento, o ancho de banda de red) y los asigna dinámicamente a una serie de máquinas virtuales (VM) que se ejecutan en el sistema según las solicitudes de recursos que reciba. Esta

máquina virtual funciona como un archivo de datos en el sistema host y puede moverse fácilmente de un sistema a otro, o incluso en la nube, y funciona igual cuando se abre de nuevo. (Hpe, 2024)

Tipos de virtualización

A medida que la tecnología de virtualización sigue evolucionando, se aplica de más y más maneras.

- La virtualización de servidores es la aplicación de tecnología de virtualización más común en el mercado actual. Como los servidores están diseñados para procesar un alto volumen de tareas, particionarlas para que los componentes puedan usarse de manera más eficiente para múltiples funciones en todo el sistema puede beneficiar enormemente a muchas organizaciones. (Hpe, 2024)

- La virtualización del almacenamiento consta de un grupo de servidores gestionados por un sistema de almacenamiento virtual. Este sistema gestiona el almacenamiento desde múltiples fuentes y lo considera como un grupo de almacenamiento, aunque haya diferencias de hardware entre los sistemas host. Esta virtualización hace más fáciles las tareas de copia de seguridad, archivo y recuperación. (Hpe, 2024)

- La virtualización de aplicaciones desacopla la aplicación del sistema operativo y el hardware en el que se ejecuta. El usuario final en general accede a aplicaciones virtualizadas en un cliente ligero, mientras la aplicación misma se ejecuta en un servidor del centro de datos

conectado por Internet. Esto puede facilitar la ejecución de aplicaciones que requieren versiones anteriores del OS o puede poner en riesgo otros recursos del sistema. (Hpe, 2024)

- La virtualización de escritorios, también conocida como infraestructura de escritorios virtuales (VDI), refleja un entorno de escritorio del usuario en un sistema basado en software al que se puede acceder de forma remota por Internet. Todos los elementos del espacio de trabajo físico se almacenan en el servidor, y los usuarios finales tienen una experiencia similar, con cualquier dispositivo que usen. Todos los datos y programas del usuario existen en el servidor host, no en el dispositivo del usuario final. (Hpe, 2024)

- La virtualización de red separa las redes virtuales de su hardware subyacente. Los conmutadores virtuales se ocupan de toda la gestión de redes. La virtualización de red hace más fácil para los administradores asignar y distribuir recursos y tener un mayor y más estable rendimiento de la red. (Hpe, 2024)

Los comienzos de la computación en la nube

Históricamente, el concepto de computación en la nube se les atribuye a múltiples autores, siendo uno de ellos Joseph Carl Robnett Licklider, quien pretendía que todo el mundo estuviera interconectado y pudiera acceder a la información desde cualquier lugar. De esa forma nació la computación en la nube, con el propósito de descentralizar los recursos de cómputo y almacenamiento para distribuirlos alrededor del mundo y facilitar el acceso a la información.

Esta idea llevó a Licklider a introducir en 1969 el concepto de redes intergalácticas de computación. Estas redes fueron una de las primeras propuestas de redes informáticas globales, lo que hoy denominamos Internet, y son el pilar de la computación en la nube. Sin embargo, en esta época, el concepto de redes intergalácticas de Licklinder se enfrentó a múltiples inconvenientes relacionados con la lentitud de la innovación en tecnología y con el ancho de banda disponible, lo cual solo pudo resolverse en la década de 1990, cuando la infraestructura de telecomunicaciones evolucionó. Esto permitió que en 1999 se ofrecieran por primera vez servicios de software as a service (SaaS), que revolucionaron así el modelo de licenciamiento del software, de manera que los usuarios ya no tendrían la necesidad de preocuparse por la instalación de actualizaciones o por la compra de licencias. (Ardila, 2019)

Amazon y la nube pública

A partir del año 2000, Amazon comenzó a desempeñar un papel fundamental en el desarrollo de la computación en la nube. Partiendo de las necesidades tecnológicas particulares de Amazon como plataforma de comercio electrónico, se lograron consolidar servicios como Simple Queue System (SQS), que fue el primer servicio SaaS en ser ofrecido por Amazon al público en el año 2004. Luego de esto, en el año 2006, la empresa Amazon Web Services (AWS) fue constituida como la primera nube pública, y en el año 2010 toda la infraestructura tecnológica de Amazon había sido desplegada utilizando servicios de AWS. Además, en el año 2007, Google, IBM y múltiples universidades en Estados Unidos iniciaron un proyecto de investigación que permitió consolidar la primera API que permitía la interacción entre las nubes privadas y la nube pública de AWS, lo que dio lugar al concepto de nube híbrida. Por otro lado, esto permitió la dinamización del mercado de nubes públicas, con la aparición de jugadores muy

importantes en este mercado, como Google Cloud Platform (GCP), lanzada en el 2008, Microsoft Azure, lanzada en el 2010, e IBM Cloud, lanzada en el 2011. (Ardila, 2019)

2. Línea de tiempo con los diferentes hitos para que hoy en día exista el nivel tecnológico.

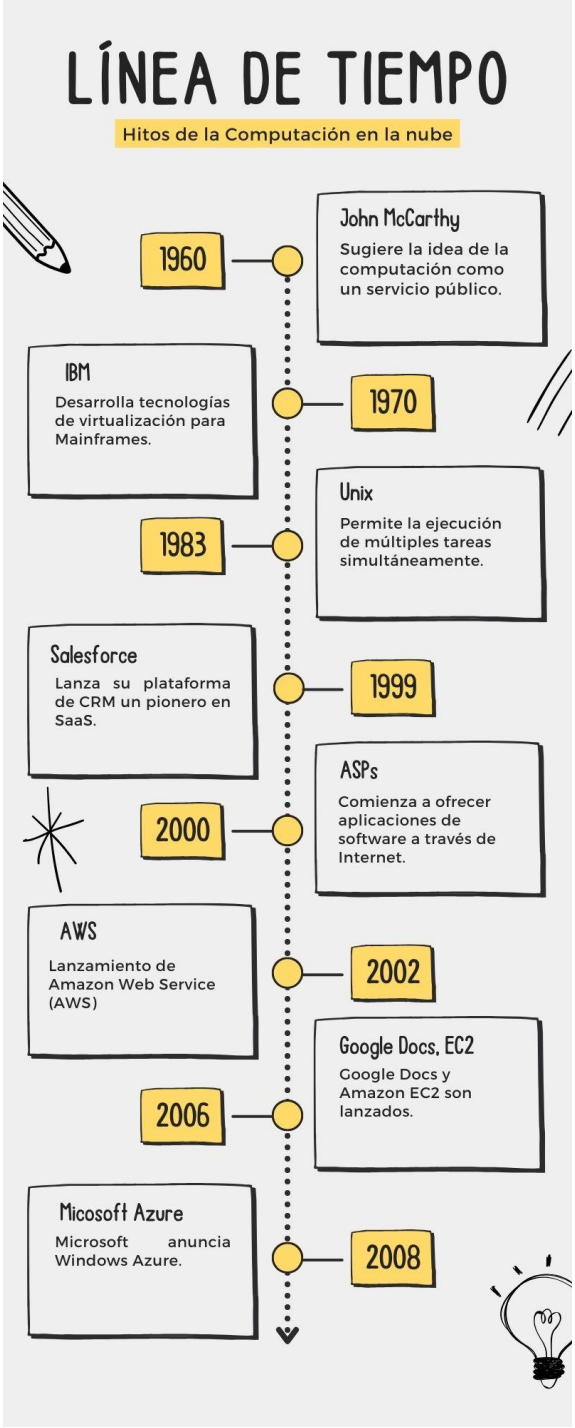


Figura 1 Línea de tiempo

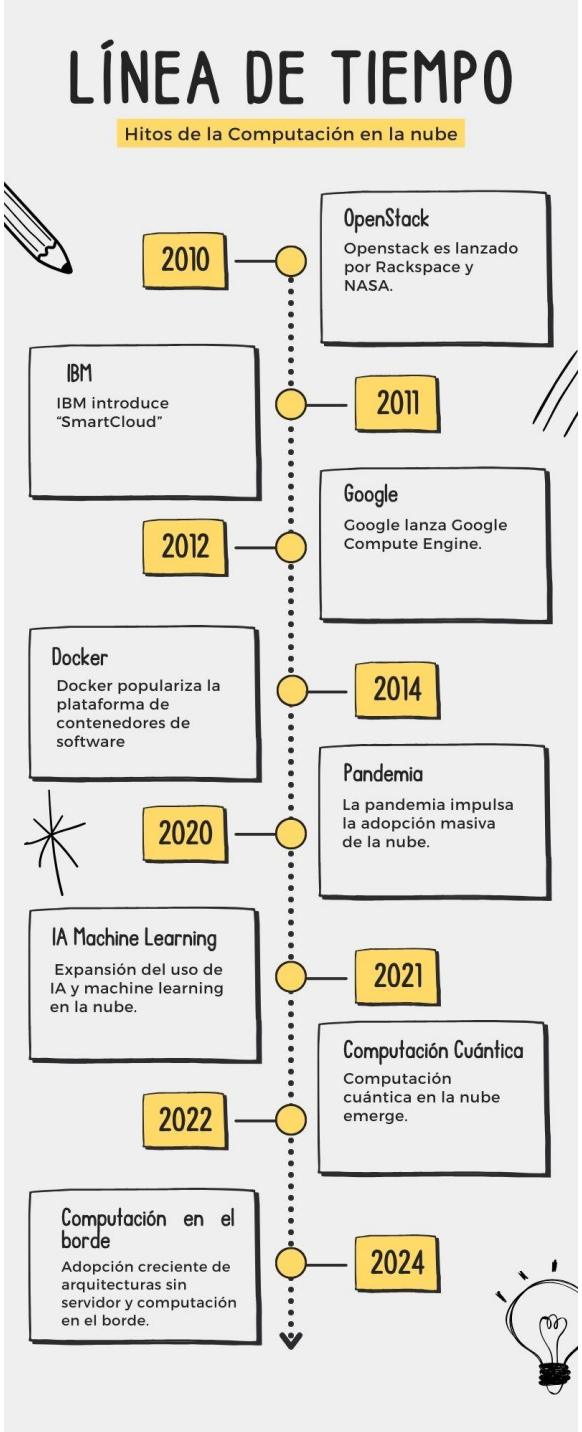


Figura 2 Línea de tiempo

3. Comparación de nombres de servicios entre AWS, Azure y GCP.

A continuación, se describe un cuadro comparativo de servicios ofrecidos por las tres plataformas de la nube más importantes a nivel mundial AWS (Amazon Web Services), Microsoft Azure y GCP (Google Cloud Platform).

Tabla 1

Servicio	AWS	AZURE	GCP
IaaS - Despliegue, gestión y mantenimiento de servidores virtuales	Elastic Compute Cloud (EC2)	Azure Virtual Machines	Compute Engine
Platform-as-a-Service	Elastic Beanstalk	App Service Cloud Services	App Engine Standard Environment App Engine Flexible Environment
Servidores Virtuales Privados	LightSail	Virtual Machine images	No tiene el servicio
Servicio Gestionado de Contenedores	EC2 Container Service (ECS)	No tiene el servicio	No tiene el servicio
Servicio gestionado de Kubernetes	Elastic Container Service para Kubernetes (EKS)	Azure Kubernetes Service (AKS)	Kubernetes Engine
Registro de contenedores Docker	EC2 Container Registry (ECR)	Azure Container Registry	Container Registry
Contenedores serverless, sin tener que gestionar clústeres ni servidores.	Fargate	Container Instances	No tiene el servicio
Servicio gestionado para desplegar y orquestar aplicaciones basadas en microservicios	Service Fabric	App Engine	No tiene el servicio
Serverless	Lambda	Functions	Cloud Functions
Computación por lotes	AWS Batch	Azure Batch	
Escalado automático de instancias	AWS Auto Scaling	Virtual Machine Scale Sets App Service Scale Capability (PAAS) Autoscaling	Instance Groups
Infraestructura de la cloud en local	AWS Outposts	No tiene el servicio	GKE On-Prem
Almacenamiento de Objetos	Simple Storage Services (S3)	Blob Storage	Google Cloud Storage
Almacenamiento para archivado	S3 Infrequent Access (IA) Glacier	Storage (Cool) Storage (Archive)	Nearline Coldline
Disco para instancias	Elastic Bloc Store (EBS)	Disk Storage	Persistent Disk
Almacenamiento de ficheros	Elastic Block Store (EBS)	Disk Storage	Cloud Filestore

Servicio	AWS	AZURE	GCP
Soluciones para transferir grandes cantidades de datos hacia la nube	AWS DataSync Snowball Edge Snowmobile	Import/Export Azure Data Box	Storage transfer Service
Backup	Glacier Storage Gateway	Azure Backup	Coldline
Almacenamiento híbrido	Storage Gateway	StorSimple	No tiene el servicio
Servicio de Disaster Recovery	Site Recovery		No tiene el servicio
Bases de datos relacionales	RDS Amazon Aurora	SQL Database	Cloud SQL Cloud Spanner
NoSQL: almacenamiento de documentos	Dynamo DB	Azure Cosmos DB	Cloud Datastore Cloud Bigtable
NoSQL: almacenamiento de clave-valor	Dynamo DB Simple DB	Table Storage	Cloud Datastore
Almacenamiento en caché	ElastiCache	Azure Redis Cache	Cloud Memorystore
Migración de bases de datos	Database Migration Service	Azure Database Migration Service	No tiene el servicio
Data warehouse gestionado	RedShift	SQL Data Warehouse	Big Query
Bases de datos basada en grafos	Neptune	Azure Cosmos DB	No tiene el servicio
Entornos virtuales de red aislados	Virtual Private Cloud	Virtual Network	Virtual Private Cloud
Conexión con entornos on-prem	AWS Managed VPN	VPN Gateway	Cloud VPN
DNS administrado para gestionar nombres y registros	Route 53	Azure DNS	Google Cloud DNS
Redirección del tráfico entrante para mejorar el rendimiento y la fiabilidad	No tiene el servicio	Traffic Manager	No tiene el servicio
Red de entrega de contenido global	CloudFront	Content Delivery Network	Cloud CDN
Red dedicada, privada para conectar la nube y los entornos locales	Direct Connect	ExpressRoute	Cloud Interconnect
Balanceo de carga: Distribuye automáticamente el tráfico entrante	Elastic Load Balancing	Load Balancer	Cloud Load Balancing
Capacidades de asesoramiento Cloud	Trusted Advisor	Azure Advisor	Cloud Platform Security
Aprovisionamiento de recursos y orquestación	OpsWorks (Chef-based) CloudFormation	Azure Automation Resource Manager VM extensions	Cloud Deployment Manager

Servicio	AWS	AZURE	GCP
Monitorización y administración de recursos Cloud	CloudWatch X-Ray Management Console	Azure Portal Azure Monitor Azure Application Insights	Stackdriver Monitoring Cloud Shell Debugger Trace Error Reporting
Facturación	Usage and Billing	Azure Billing API	Stackdriver Monitoring Cloud Billing
Administración	Application Discovery Service Systems Manager Personal Health Dashboard	Log Analytics Operations Management Suite Resources Health Storage Explorer	Cloud Console
Seguimiento de la actividad de los usuarios y el uso de las APIs	CloudTrail	Log Analytics Audit logging dentro de Azure Monitor	Cloud Audit Logging
Evalúa el estado de las cargas de trabajo en comparación con las prácticas recomendadas sobre arquitectura de AWS	Well-Architected Tool	No tiene el servicio	No tiene el servicio
Autenticación y autorización	Identity and Access Management (IAM) Organizations	Active Directory Active Directory Premium	Cloud IAM Cloud Identity-Aware Proxy
Protección de la información	No tiene el servicio	Azure Information Protection	No tiene el servicio
Cifrado	AWS Key Management Service CloudHSM	Key Vault	Cloud Key Management Service
Firewall	Web Application Firewall	Application Gateway	No tiene el servicio
Evaluación de la seguridad	Inspector	Security Center	No tiene el servicio
Administración de certificados	Certificate Manager	App Service Certificates	No tiene el servicio
Servicios de directorio	AWS Directory Service	Active Directory Domain Services	No tiene el servicio
Administración de identidades	Cognito	Azure Active Directory B2C	No tiene el servicio
Autenticacion multifactor	Multi-Factor Authentication	Multi-Factor Authentication	Multi-Factor Authentication
Detección de amenazas y actividades anómalas.	Guard Duty AWS Macie	Azure Advanced Threat Protection	Cloud SecurityCommand Center (Cloud Security Scanner, Cloud Data Loss Prevention API)
Cumplimiento normativo	Artifact	Service Trust Portal	No tiene el servicio
Protección frente a ataques por denegación de servicio distribuido (DDoS)	AWS Shield	DDoS Protection Service	Cloud Armor

Nota. (Juan María, V. M. C. (s/f). *AWS vs Azure vs GCP, 2019)*

Entrega 2

Investigue e implemente, como hacer para que, al ejecutar dos contenedores, cada uno responda solamente a un nombre de dominio especifico, la implementación se hace con una instancia EC2 y 2 contenedores. La consulta de los dominios debe hacerse sin puertos. La ejecución de la prueba puede hacerla localmente.

1. A continuación, enunciamos algunos conceptos utilizados para el desarrollo de la actividad.

AWS: Amazon Web Services, cuyas siglas son AWS, es la plataforma de servicios en la nube que lanzó la compañía estadounidense de comercio electrónico en el 2006, ofreciendo servicios de infraestructuras de Tecnologías de la Información (TI) para empresas en forma de lo que hoy conocemos como Cloud Computing, facilitando su crecimiento y escalabilidad. (Deloitte, 2024)

EC2: Amazon Elastic Compute Cloud (Amazon EC2) proporciona capacidad de computación escalable bajo demanda en la nube de Amazon Web Services (AWS). El uso de Amazon EC2 reduce los costos de hardware para que pueda desarrollar e implementar aplicaciones con mayor rapidez. (Amazon, 2024)

Docker: Docker es una plataforma de software que le permite crear, probar e implementar aplicaciones rápidamente. Docker empaqueta software en unidades estandarizadas

llamadas contenedores que incluyen todo lo necesario para que el software se ejecute, incluidas bibliotecas, herramientas de sistema, código y tiempo de ejecución. (Amazon, 2023)

Máquina virtual: Una máquina virtual (VM) es un entorno informático que funciona como un sistema aislado con su propia CPU, memoria, interfaz de red y almacenamiento, el cual se crea a partir de un conjunto de recursos de hardware. Mediante un sistema de software denominado hipervisor, se aíslan los recursos informáticos necesarios y se crean y gestionan las máquinas virtuales. (Redhat, 2023)

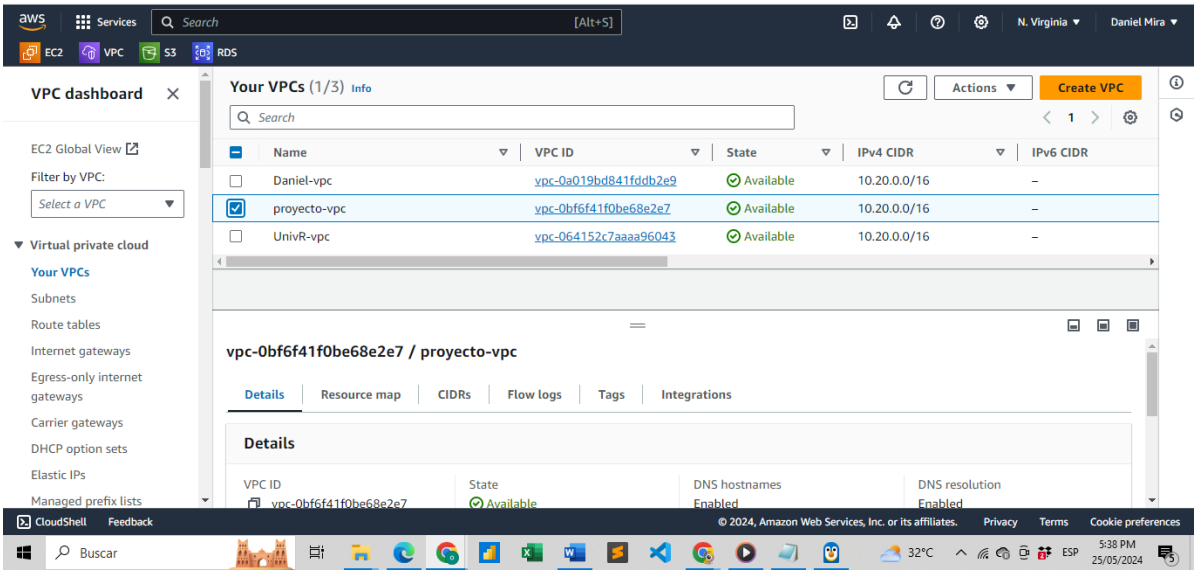
WSL: Subsistema de Windows para Linux (WSL) es una característica de Windows que permite ejecutar un entorno Linux en la máquina Windows, sin necesidad de una máquina virtual independiente ni de arranque dual. WSL está diseñado para proporcionar una experiencia perfecta y productiva para los desarrolladores que quieren usar Windows y Linux al mismo tiempo. (Microsoft, 2023)

VPC: Con Amazon Virtual Private Cloud (Amazon VPC), puede lanzar recursos de AWS en una red virtual aislada de manera lógica que haya definido. Esta red virtual es muy similar a la red tradicional que usaría en su propio centro de datos, pero con los beneficios que supone utilizar la infraestructura escalable de AWS. (Amazon, 2024)

2. A continuación, se describe el paso a paso para el desarrollo de la actividad. Para crear los componentes se debe previamente ingresar a la plataforma de AWS, realizando la creación de usuario y contraseña.

El primer paso es crear un VPC opción que se encuentra en la parte superior izquierda, en este caso con el nombre “proyecto” como lo ilustra la siguiente imagen:

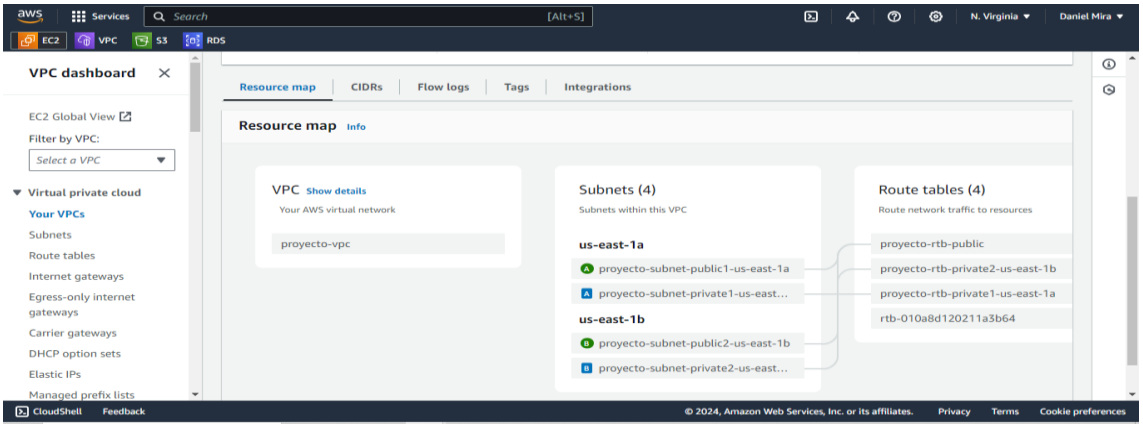
Figura 3 Crear VPC



Fuente: Elaboración propia

Con la creación del VPC la plataforma automáticamente crea las subredes que el usuario establezca en la configuración creando 2 Datacenter en los cuales por cada uno se crean una subred privada y una pública como lo ilustra la siguiente imagen.

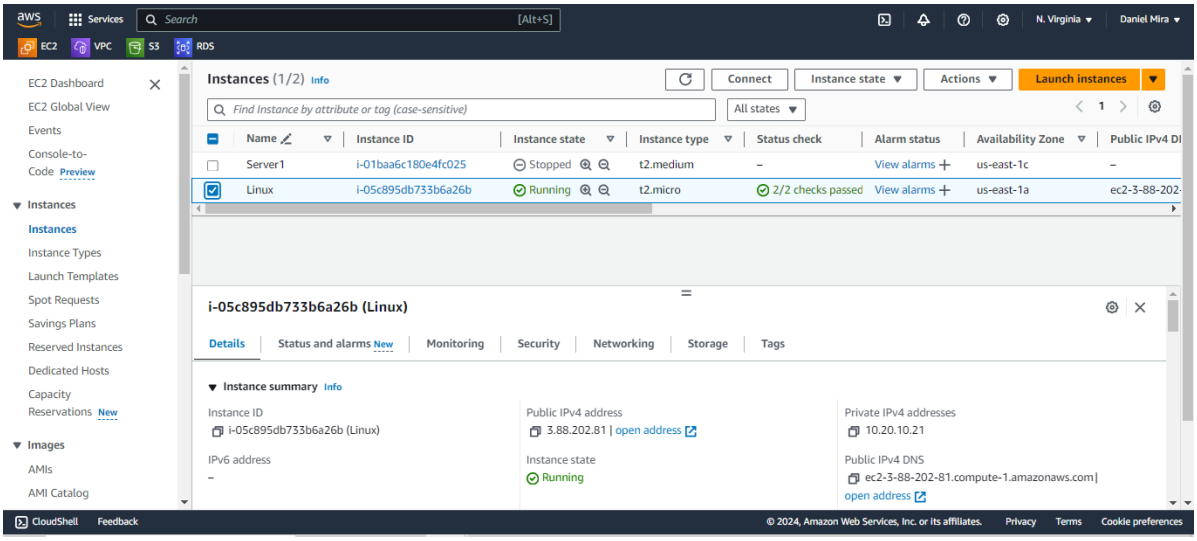
Figura 4 Instancia



Fuente: Elaboración propia

Luego dentro del VPC se realiza la creación de la Instancia por la opción ubicada en la parte superior izquierda EC2, para este caso crearemos una máquina virtual de Linux con requerimientos mínimos para su funcionamiento cuya dirección ip pública es 3.88.202.81.

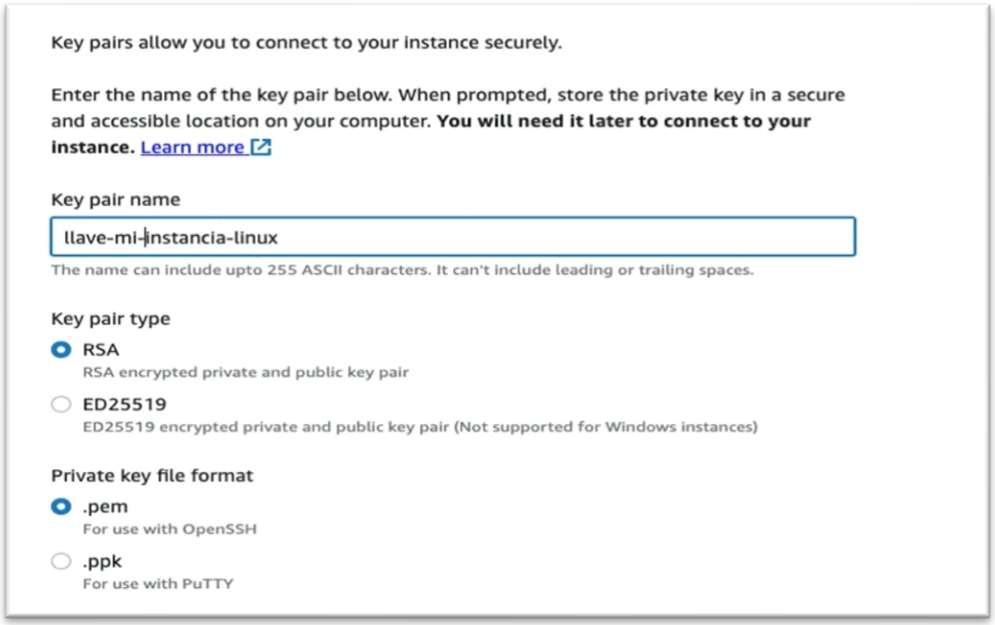
Figura 5 Máquina Virtual Linux



Fuente: Elaboración propia

Es importante mencionar que en el proceso de creación de la máquina virtual se debe generar una llave virtual, la cual es un archivo que permite la conexión a la instancia, para el caso de Linux se genera **.pem** y es un archivo que se descarga y se debe tener cuidado de no perder porque es el único que permite la conexión a dicha instancia.

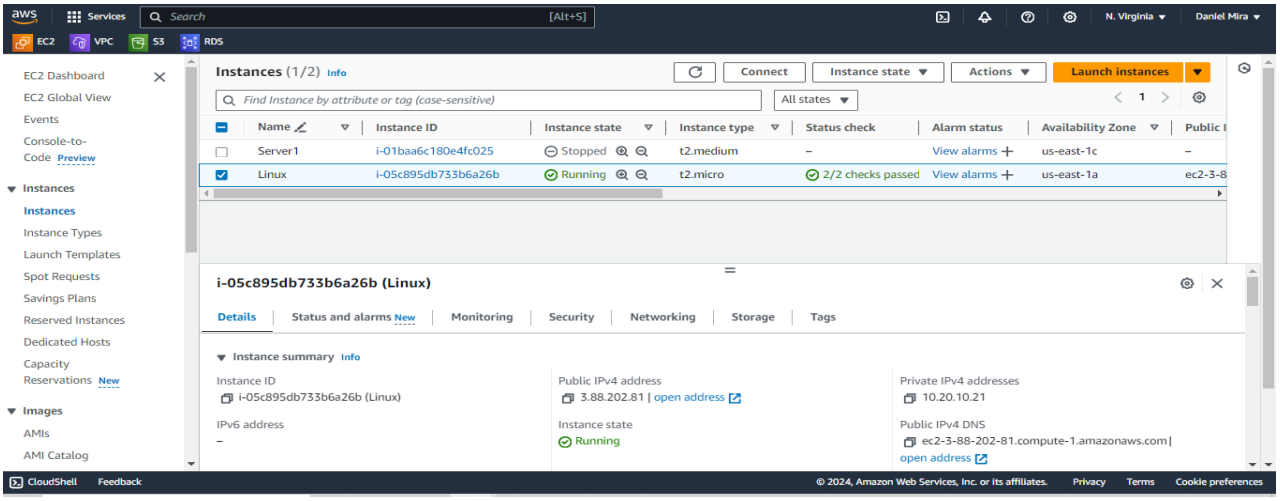
Figura 6 Llave



Fuente: Elaboración propia

El paso siguiente es la conexión a la máquina virtual de Linux, para lo cual damos clic en la opción ubicada en la parte superior derecha “Connect”

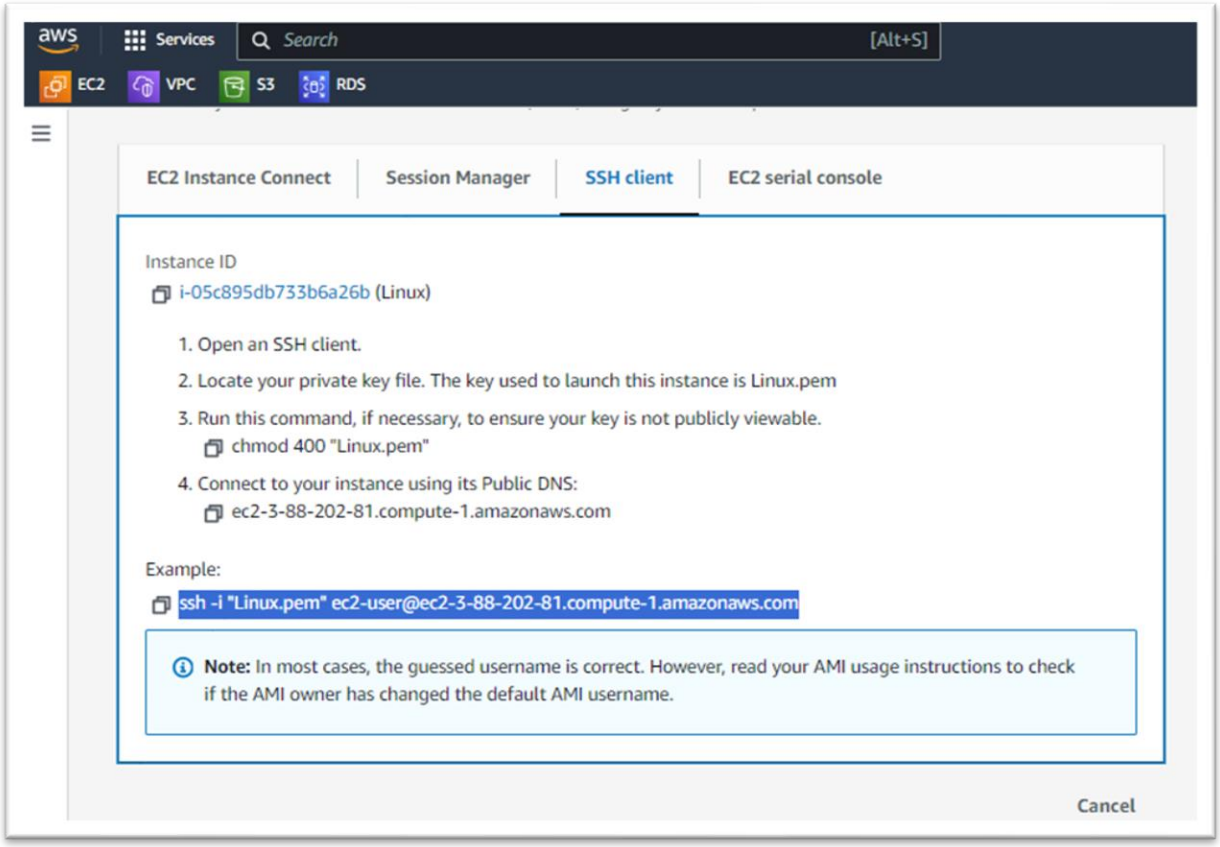
Figura 7 Connect



Fuente: Elaboración propia

Seguidamente nos ubicamos en la opción “SSH Client” y copiamos la ruta que se encuentra debajo de Example como lo ilustra la imagen.

Figura 8 SSH Client

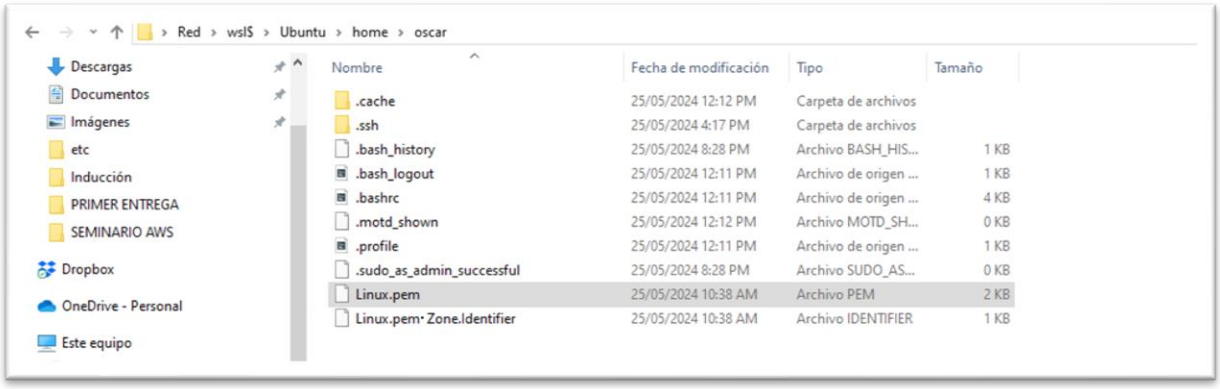


Fuente: Elaboración propia

Para la conexión es necesario instalar WSL, el cual nos permite conectarnos a Linux desde un sistema operativo Windows, es importante que en la ruta donde se instala WSL se copie el archivo de la llave virtual generada al momento de crear la máquina virtual para que nos permita la conexión.

Para llegar a la ruta en la barra del explorador de Windows digitamos \\wsl$

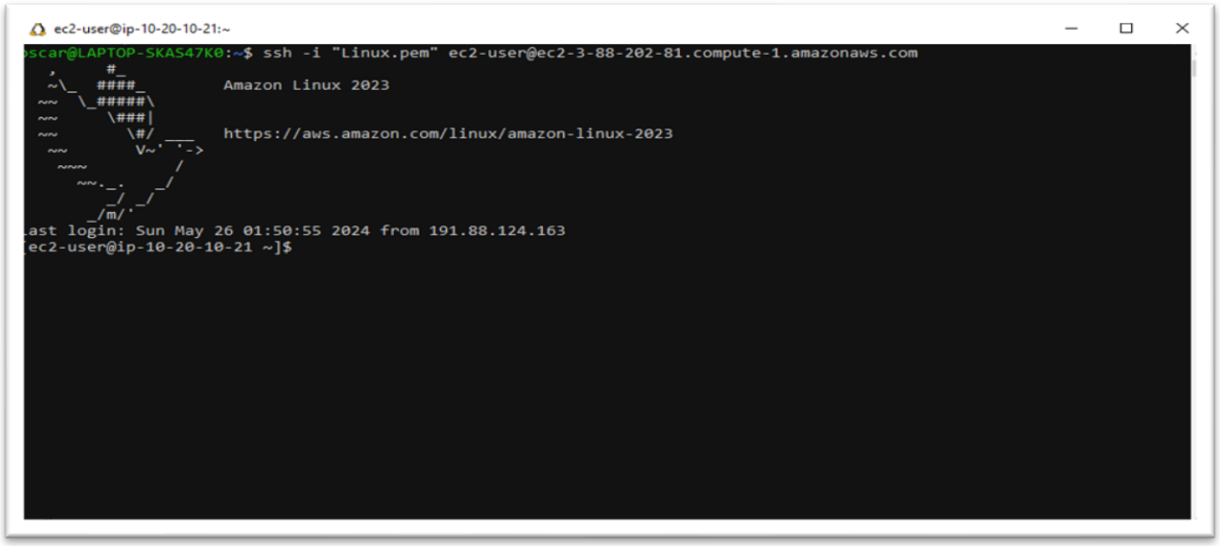
Figura 9 WSL



Fuente: Elaboración propia

El siguiente paso es abrir wsl y digitar la ruta copiada en la imagen anterior de conexión en la plataforma, con lo cual ya estamos conectados a la máquina virtual de Linux creada en AWS.

Figura 10 MV Linux Creada



Fuente: Elaboración propia

Ahora procedemos a crear los Docker o contenedores y les asignamos un nombre y un puerto por el cual se va a acceder a cada uno.

Figura 11 Crear Docker

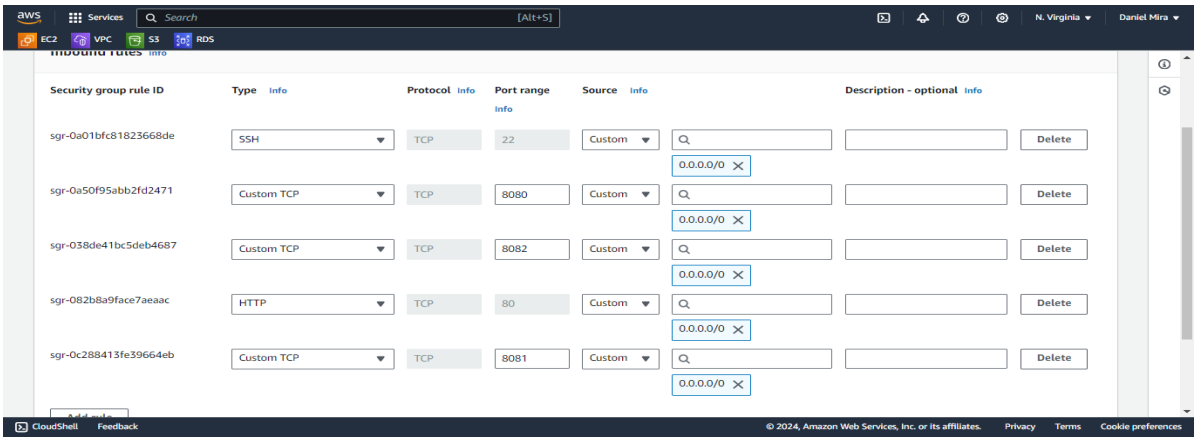
```
oscar@LAPTOP-SKAS47K0:~$ ssh -i "Linux.pem" ec2-user@ec2-3-88-202-81.compute-1.amazonaws.com
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~   \_#####\
  ~~      \###|
  ~~        \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~        V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Sun May 26 00:34:59 2024 from 191.88.124.163
[ec2-user@ip-10-20-10-21 ~]$ sudo su
[root@ip-10-20-10-21 ec2-user]# systemctl start docker
[root@ip-10-20-10-21 ec2-user]# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED       STATUS       PORTS                                      NAMES
271fcd8171a9   nginx     "/docker-entrypoint.…"   5 hours ago   Up 5 hours   0.0.0.0:80->80/tcp, :::80->80/tcp          nginx-proxy
4df4451d22c0   httpd     "httpd-foreground"       6 hours ago   Up 6 hours   0.0.0.0:8082->80/tcp, :::8082->80/tcp      contenedor3
bd24833d94c3   httpd     "httpd-foreground"       7 hours ago   Up 7 hours   0.0.0.0:8081->80/tcp, :::8081->80/tcp      contenedor2
54e1835960ea   httpd     "httpd-foreground"       7 hours ago   Up 7 hours   0.0.0.0:8080->80/tcp, :::8080->80/tcp      contenedor1
[root@ip-10-20-10-21 ec2-user]#
```

Fuente: Elaboración propia

Luego por la opción de Security Groups en la máquina virtual en la plataforma de AWS se configuran los puertos de entrada a través del puerto 80 que para este caso configuraremos los puertos 8080 y 8081 y se agrega el protocolo HTTP.

Al ingresar la dirección 0.0.0.0/0 permitimos la entrada desde cualquier dirección ip de Internet.

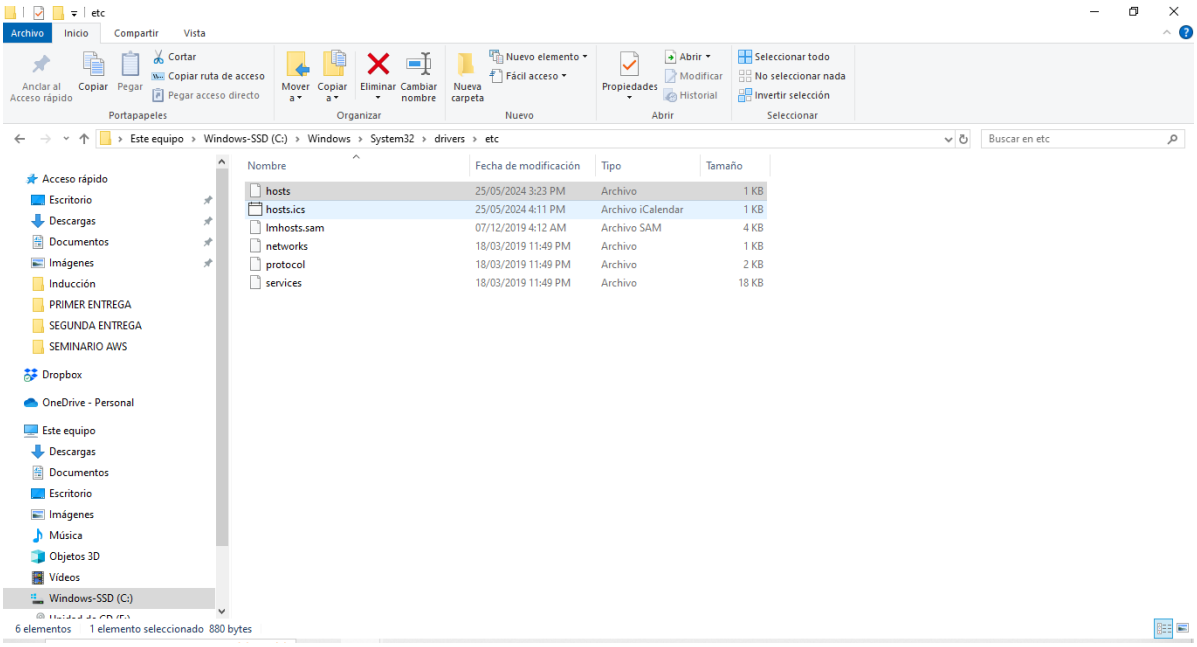
Figura 12 Configuración puertos



Fuente: Elaboración propia

El siguiente paso es ingresar en el equipo local a la ruta C:\Windows\System32\drivers\etc y abrimos el archivo hosts.

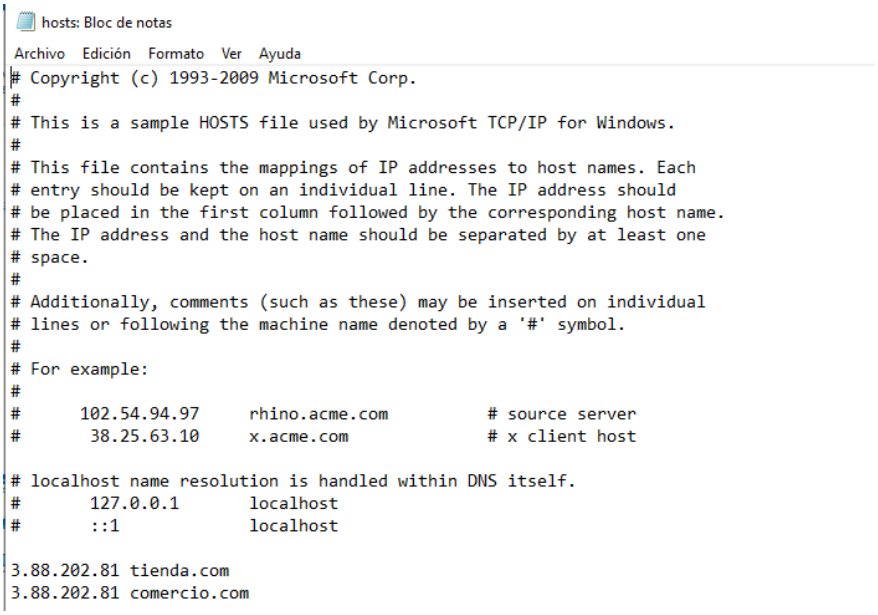
Figura 13 Ingreso ruta local



Fuente: Elaboración propia

Luego se agregan 2 filas al final del archivo indicando la ip de la máquina virtual y el dominio, para nuestro caso 3.88.202.81 tienda.com y 3.88.202.81 para comercio.com.

Figura 14 Configuración IP



```
hosts: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost

3.88.202.81 tienda.com
3.88.202.81 comercio.com
```

Fuente: Elaboración propia

El siguiente paso es crear el directorio nginx e ingresar al archivo nginx.conf.

Figura 15 Crear Directorio nginx

```sh
mkdir ~/nginx
nano ~/nginx/nginx.conf
```

Fuente: Elaboración propia

En el cual debemos ingresar el siguiente código y guardar.

Figura 16 Codificación

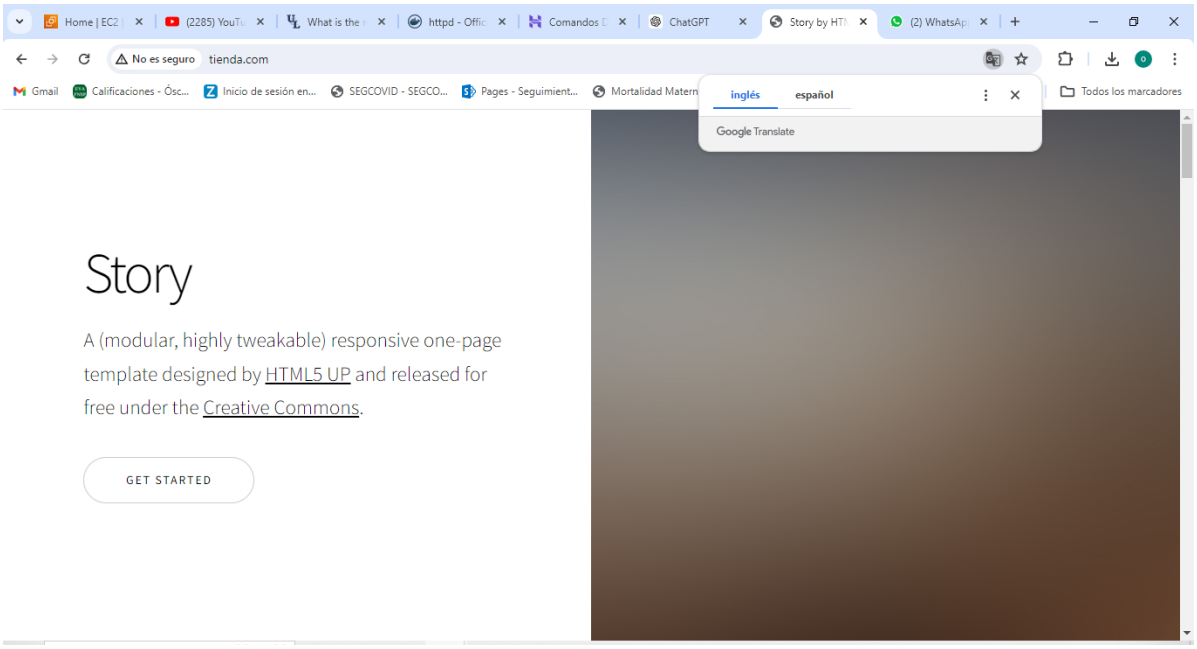
```
    server {
        listen 80;
server_name tienda.com;

        location/{
            proxy_pass http://localhost:8080/;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Proto $scheme;

        }
}
server{
listen 80;
server_name comercio.com;

        location/{
            proxy_pass http://localhost:8081/;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Proto $scheme;


    }
}
```

Fuente: Elaboración propia

Es importante mencionar que al momento de crear los contenedores se crea una carpeta para cada uno, en la cual se ingresan los archivos correspondientes a la página web que se va a cargar. Importante el archivo Index.html.

Por último, en el navegador en la barra de búsqueda ingresamos tienda.com y debe cargarnos la página contenida en la carpeta de ese contenedor.

Figura 17 Prueba contenedor 1



Fuente: Elaboración propia

También comercio.com carga la página del otro contenedor.

Figura 18 Prueba Contenedor 2

Fuente: Elaboración propia

Entrega 3

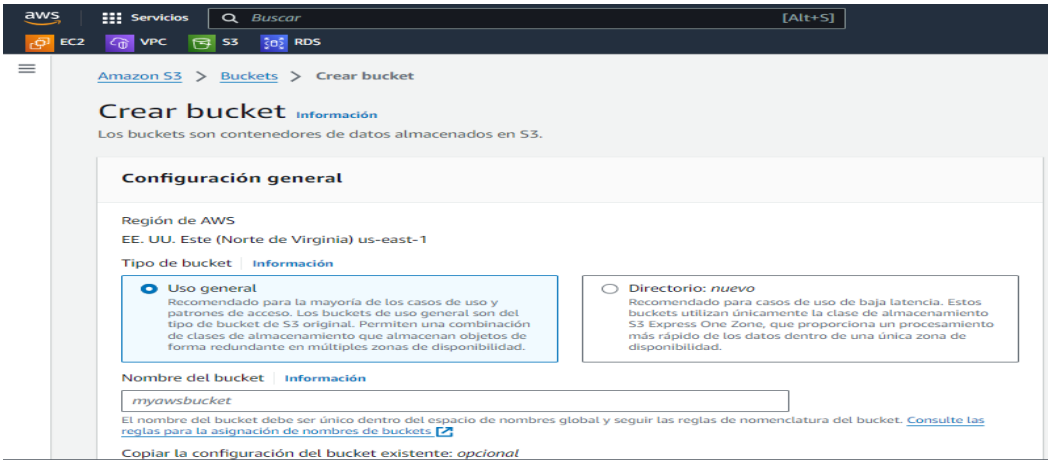
Explicar y adjuntar evidencia de cómo se hace la conexión de una instancia AWS en Linux para que pueda cargar archivos en S3 y además pueda conectarse a una base de datos RDS MySQL.

A continuación, se describen los pasos a seguir para el cargue de los archivos S3.

Lo primero es ingresar a AWS con la cuenta previamente creada y escogemos en el menú superior izquierdo la opción S3, bucket, crear bucket.

Le ponemos un nombre al bucket y escogemos las opciones para la creación y le damos clic en crear.

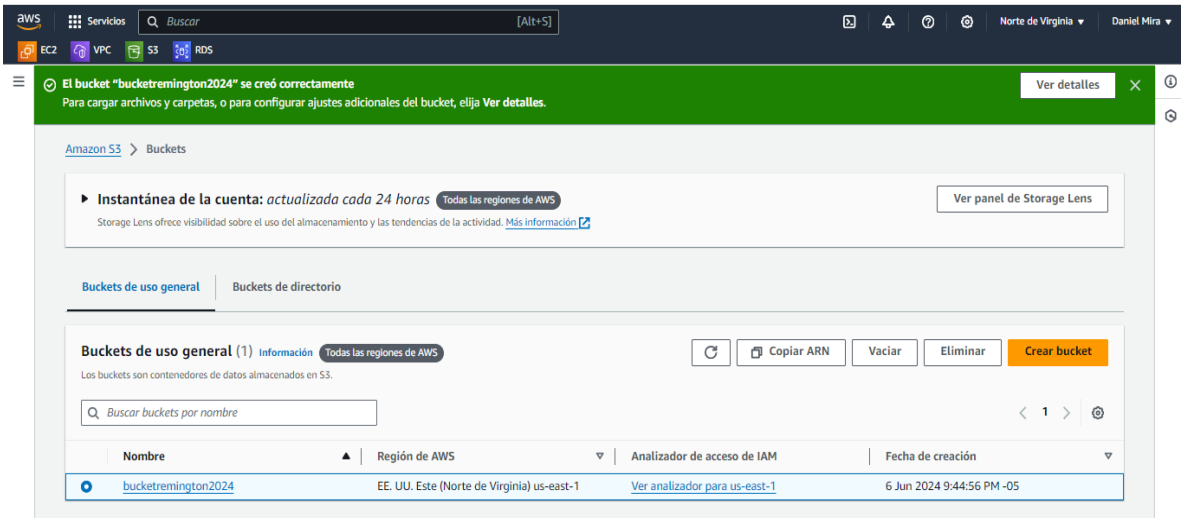
Figura 19 Crear Bucket



Fuente: Elaboración propia

En la siguiente imagen nos aparece creado el bucket “bucketremington2024”.

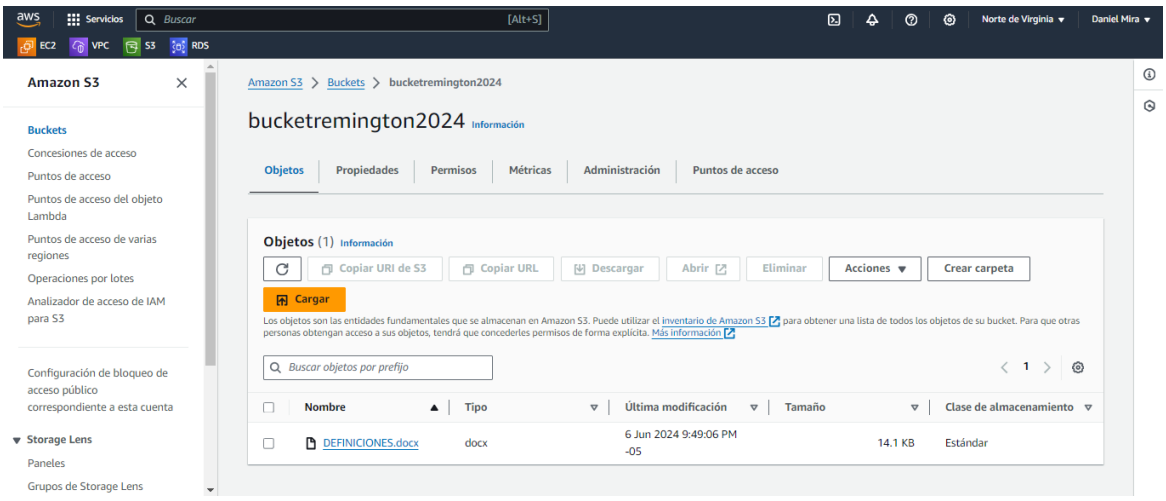
Figura 20 Resumen Creación Bucket



Fuente: Elaboración propia

Podemos desde la interfaz de AWS cargar un archivo desde nuestro pc al bucket, en este caso cargamos el archivo Word “DEFINICIONES.docx”.

Figura 21 Cargue archivo DEFINICIONES



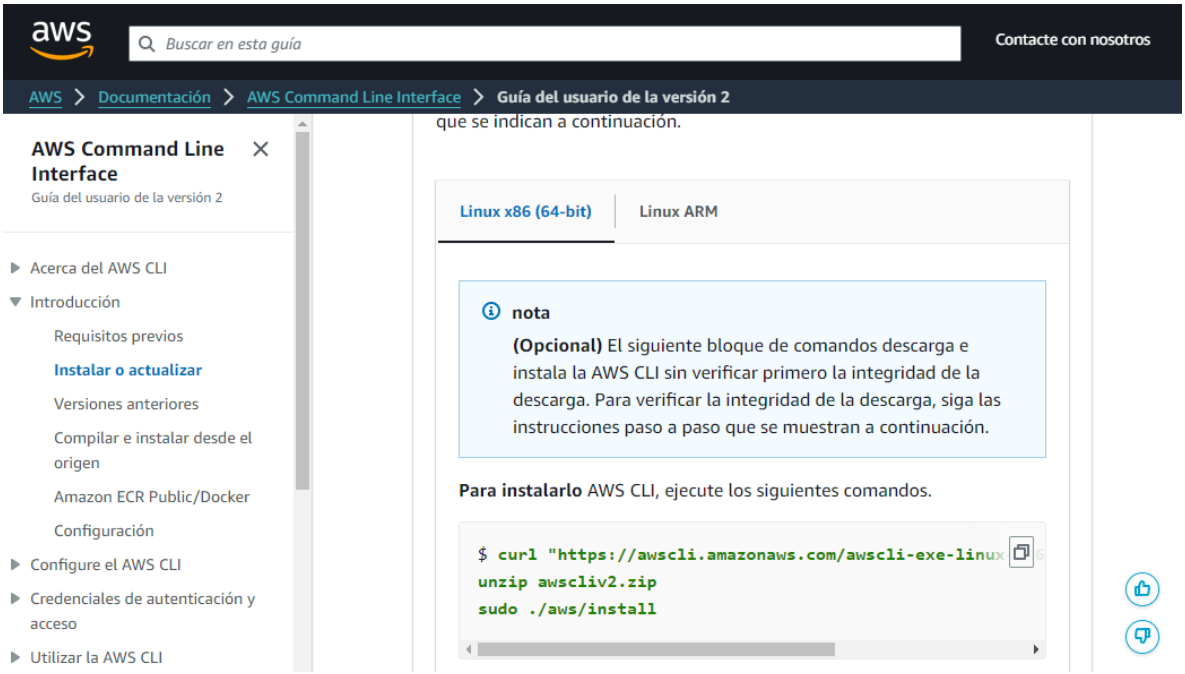
Fuente: Elaboración propia

A continuación, vamos a instalar en la consola WSL AWS CLI, para que nos permita a través de la consola conectarnos a las instancias creadas en AWS.

Para Instalar AWS CLI ingresamos a la siguiente página.

https://docs.aws.amazon.com/es_es/cli/latest/userguide/getting-started-install.html

Figura 22 Instalación AWS CLI



Fuente: Elaboración propia

Copiamos los comandos para realizar la instalación en la consola WSL, los cuales descargan el archivo, luego se descomprime, nos pide que instalemos la utilidad para descomprimir, se hace con el comando “sudo apt install unzip” y se ejecuta.

Figura 23 Uso de Comandos

```
oscar@LAPTOP-SKAS47K0: ~
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.146.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


This message is shown once a day. To disable it please create the
/home/oscar/.hushlogin file.
oscar@LAPTOP-SKAS47K0:~$ which aws
oscar@LAPTOP-SKAS47K0:~$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 57.8M  100 57.8M    0     0  46547      0  0:21:43  0:21:43 --:--:-- 35512
oscar@LAPTOP-SKAS47K0:~$ unzip awscliv2.zip
Command 'unzip' not found, but can be installed with:
sudo apt install unzip
oscar@LAPTOP-SKAS47K0:~$ sudo apt install unzip
[sudo] password for oscar:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  zip
The following NEW packages will be installed:
  unzip
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 174 kB of archives.
After this operation, 385 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 unzip amd64 6.0-26ubuntu3.1 [174 kB]
Fetched 174 kB in 3s (52.5 kB/s)
Selecting previously unselected package unzip.
(Reading database ... 24208 files and directories currently installed.)
Preparing to unpack .../unzip_6.0-26ubuntu3.1_amd64.deb ...
Unpacking unzip (6.0-26ubuntu3.1) ...
Setting up unzip (6.0-26ubuntu3.1) ...
Processing triggers for man-db (2.10.2-1) ...
oscar@LAPTOP-SKAS47K0:~$ ls
Linux.pem  Linux.pem:Zone.Identifier  awscliv2.zip
oscar@LAPTOP-SKAS47K0:~$ unzip awscliv2.zip
Archive:  awscliv2.zip
   creating: aws/
   creating: aws/dist/
  inflating: aws/THIRD_PARTY_LICENSES
```

Fuente: Elaboración propia

En esta imagen nos muestra que AWS CLI ya fue instalado.

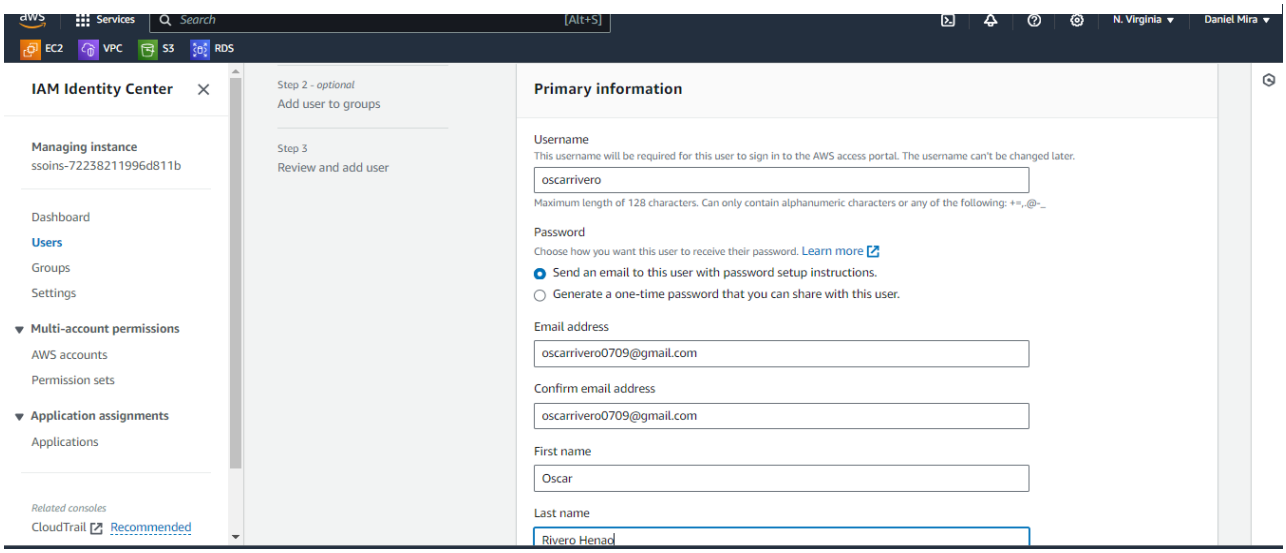
Figura 24 Correcta Instalación AWS CLI

```
oscar@LAPTOP-SKAS47K0: ~
  inflating: aws/dist/docutils/writers/s5_html/themes/default/opera.css
  inflating: aws/dist/docutils/writers/s5_html/themes/default/slides.css
  inflating: aws/dist/docutils/writers/s5_html/themes/default/slides.js
  inflating: aws/dist/docutils/writers/s5_html/themes/default/framing.css
  inflating: aws/dist/docutils/writers/s5_html/themes/big-white/pretty.css
  inflating: aws/dist/docutils/writers/s5_html/themes/big-white/framing.css
  inflating: aws/dist/docutils/writers/s5_html/themes/medium-white/framing.css
  inflating: aws/dist/docutils/writers/s5_html/themes/medium-white/pretty.css
  inflating: aws/dist/docutils/writers/s5_html/themes/small-black/__base__
  inflating: aws/dist/docutils/writers/s5_html/themes/small-black/pretty.css
  inflating: aws/dist/docutils/writers/s5_html/themes/medium-black/__base__
  inflating: aws/dist/docutils/writers/s5_html/themes/medium-black/pretty.css
  inflating: aws/dist/docutils/writers/s5_html/themes/big-black/framing.css
  inflating: aws/dist/docutils/writers/s5_html/themes/big-black/__base__
  inflating: aws/dist/docutils/writers/s5_html/themes/big-black/pretty.css
  inflating: aws/dist/docutils/writers/s5_html/themes/small-white/framing.css
  inflating: aws/dist/docutils/writers/s5_html/themes/small-white/pretty.css
  inflating: aws/dist/docutils/writers/html5_polyglot/minimal.css
  inflating: aws/dist/docutils/writers/html5_polyglot/responsive.css
  inflating: aws/dist/docutils/writers/html5_polyglot/tuftig.css
  inflating: aws/dist/docutils/writers/html5_polyglot/template.txt
  inflating: aws/dist/docutils/writers/html5_polyglot/plain.css
  inflating: aws/dist/docutils/writers/html5_polyglot/math.css
  inflating: aws/dist/docutils/writers/pep_html/pep.css
  inflating: aws/dist/docutils/writers/pep_html/template.txt
  inflating: aws/dist/docutils/writers/latex2e/titlepage.tex
  inflating: aws/dist/docutils/writers/latex2e/xelatex.tex
  inflating: aws/dist/docutils/writers/latex2e/titlingpage.tex
  inflating: aws/dist/docutils/writers/latex2e/default.tex
  inflating: aws/dist/docutils/writers/latex2e/docutils.sty
  inflating: aws/dist/docutils/writers/html4css1/html4css1.css
  inflating: aws/dist/docutils/writers/html4css1/template.txt
  inflating: aws/dist/docutils/writers/odf_odt/styles.odt
oscar@LAPTOP-SKAS47K0:~$ ls
Linux.pem  Linux.pem:Zone.Identifier  aws  awscliv2.zip
oscar@LAPTOP-SKAS47K0:~$ sudo ./aws/install
You can now run: /usr/local/bin/aws --version
oscar@LAPTOP-SKAS47K0:~$ 123
123: command not found
oscar@LAPTOP-SKAS47K0:~$ /usr/local/bin/aws --version
aws-cli/2.16.3 Python/3.11.8 Linux/5.15.146.1-microsoft-standard-WSL2 exe/x86_64.ubuntu.22
oscar@LAPTOP-SKAS47K0:~$
```

Fuente: Elaboración propia

Para conectarnos a través de la consola de WSL una vez instalado AWS CLI,vamos a la página de AWS y desde el usuario raiz de nuestra cuenta creamos un usuario.

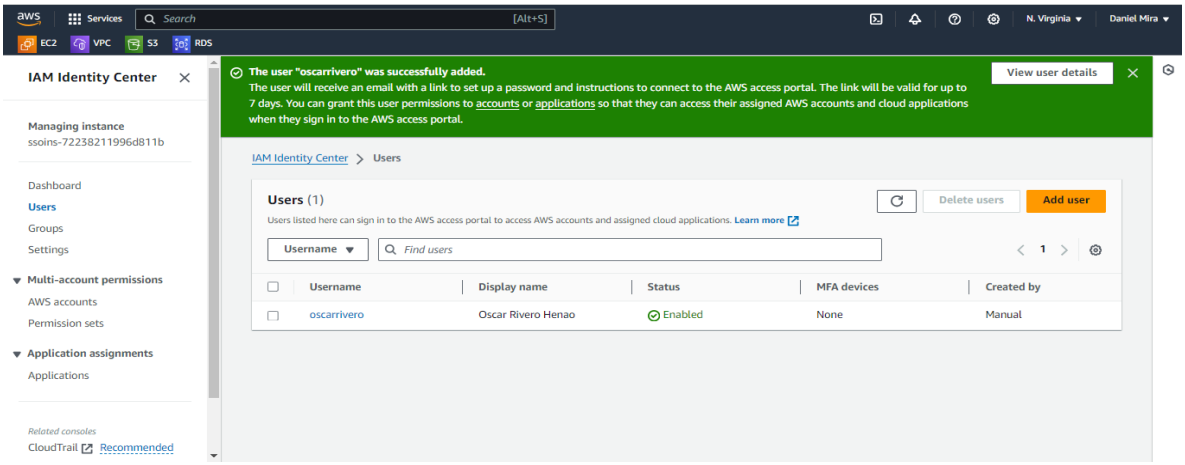
Figura 25 Creación usuario CLI



Fuente: Elaboración propia

Para este caso el usuario creado es oscarrivero, el cual se le establece una contraseña.

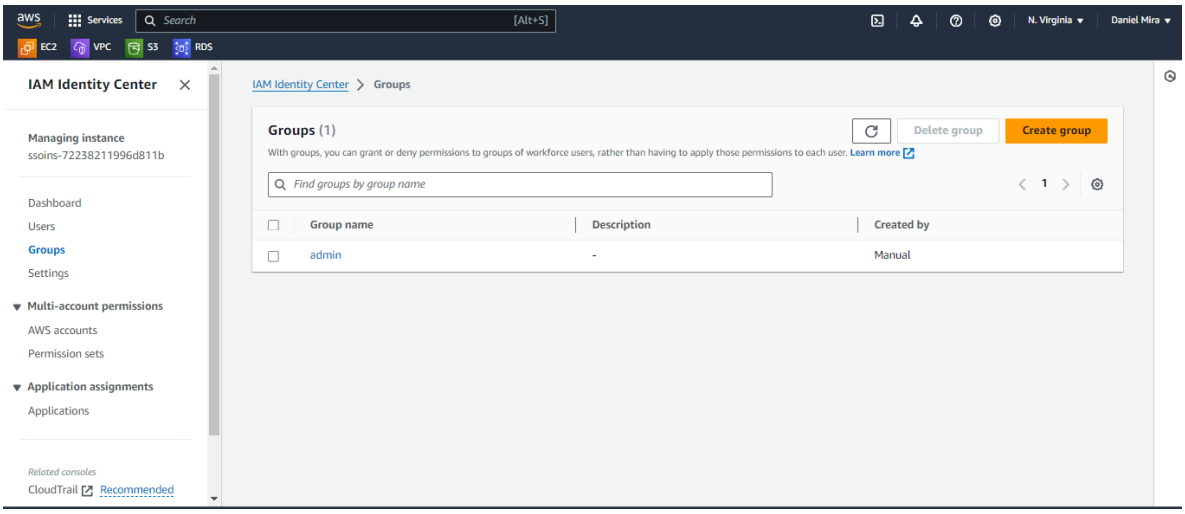
Figura 26 Asignación contraseña usuario CLI



Fuente: Elaboración propia

A continuación, creamos un grupo que para este caso es “admin”.

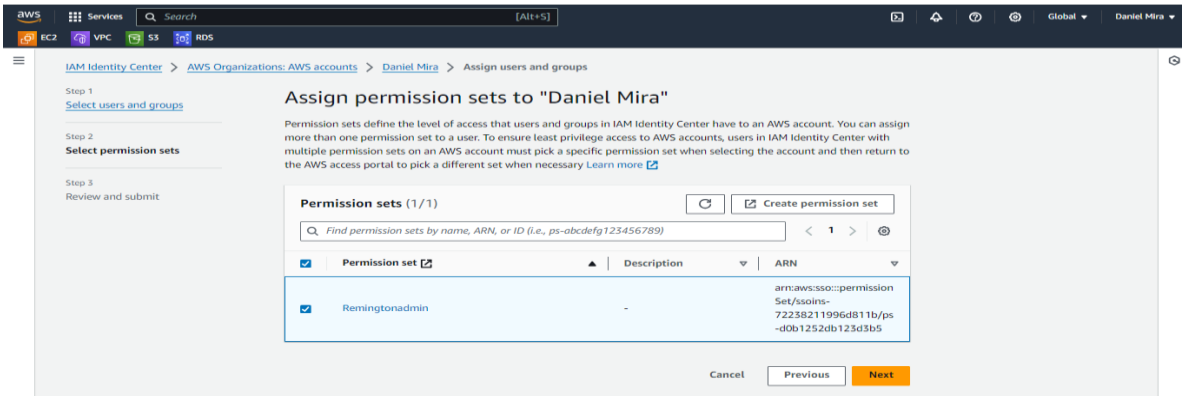
Figura 27 Creación grupo



Fuente: Elaboración propia

Luego creamos los permisos los cuales vienen predefinidos por AWS, en este caso los permisos van a ser de usuario administrador.

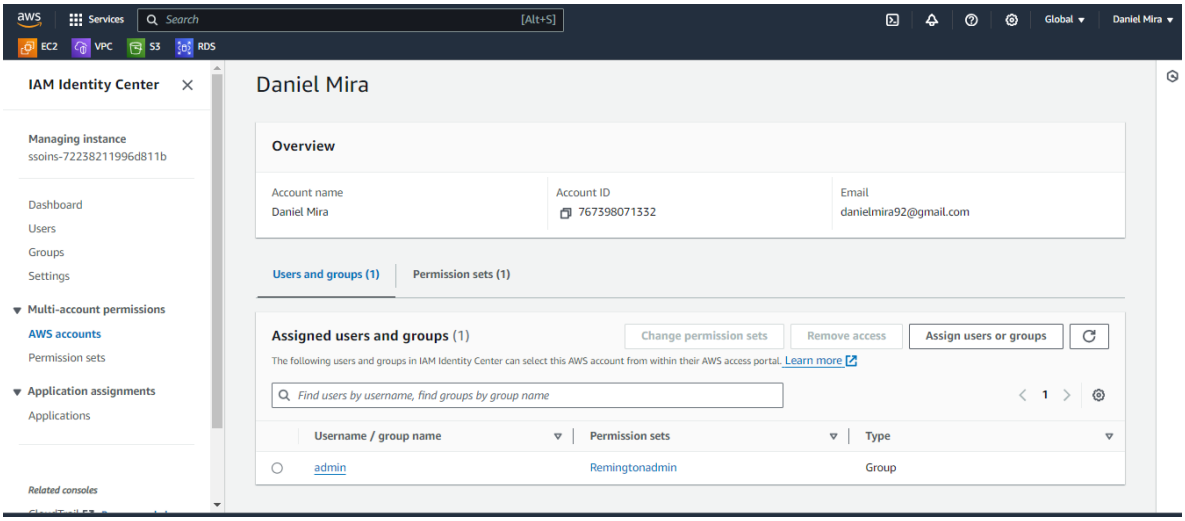
Figura 28 Asignación de permisos usuario



Fuente: Elaboración propia

Asignamos los permisos al grupo admin, el cual es el grupo al que pertenece el usuario “oscarrivero” que tiene los permisos de administrador.

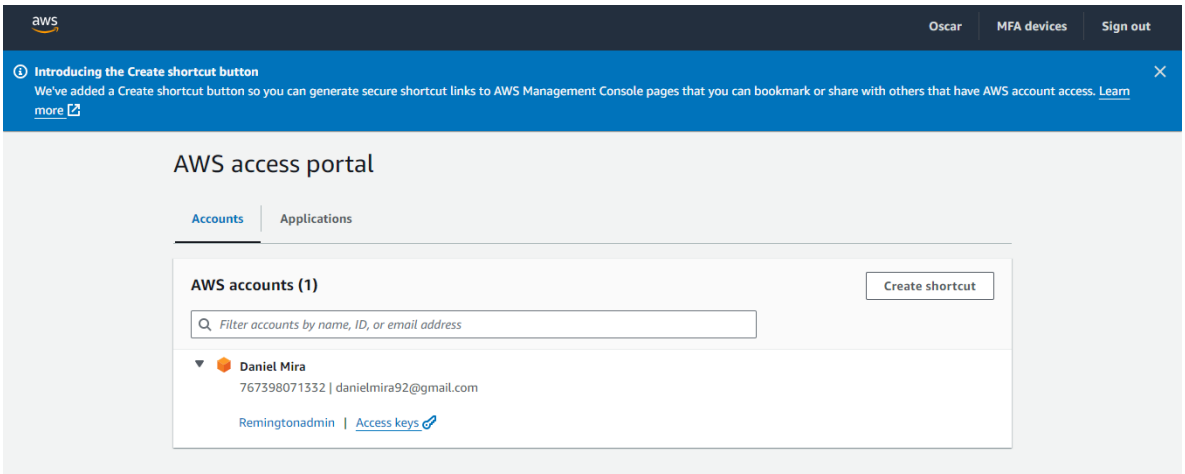
Figura 29 Asignación permisos grupo



Fuente: Elaboración propia

Al correo del usuario registrado llega el enlace para ingresar a AWS, con su usuario, contraseña y además pide autenticación doble para lo cual se debe instalar en el celular Google Authenticator, escanear el código QR, la aplicación le genera un código que introduce y finalmente le muestra la siguiente pantalla para ingresar a AWS.

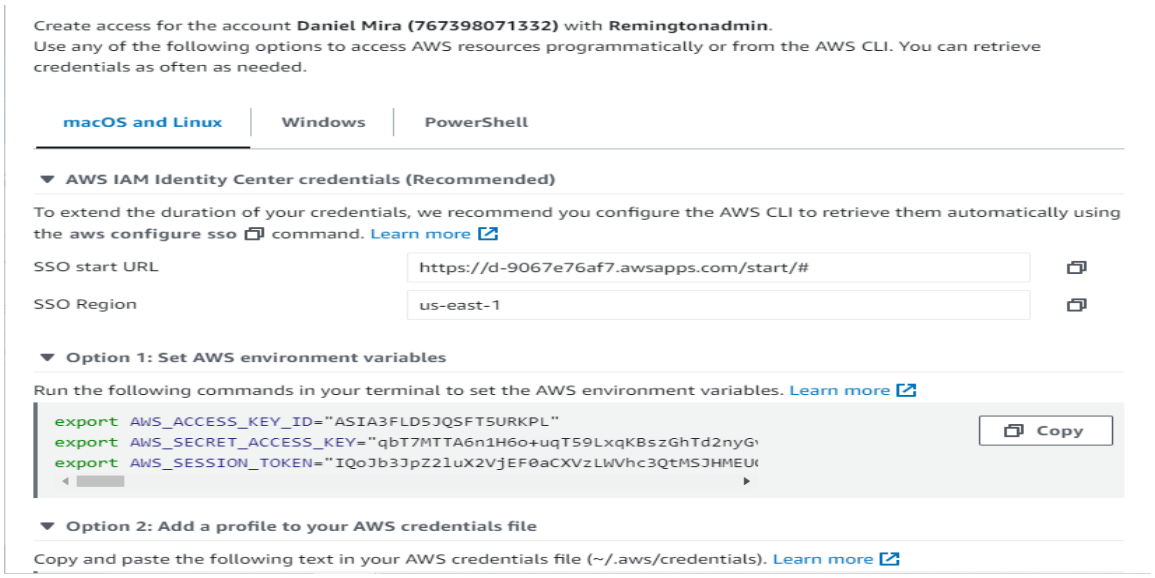
Figura 30 Autenticación de doble factor



Fuente: Elaboración propia

Debemos ingresar por la opción Access keys, la opción macOS and Linux y copiar las variables de acceso de la opción 1.

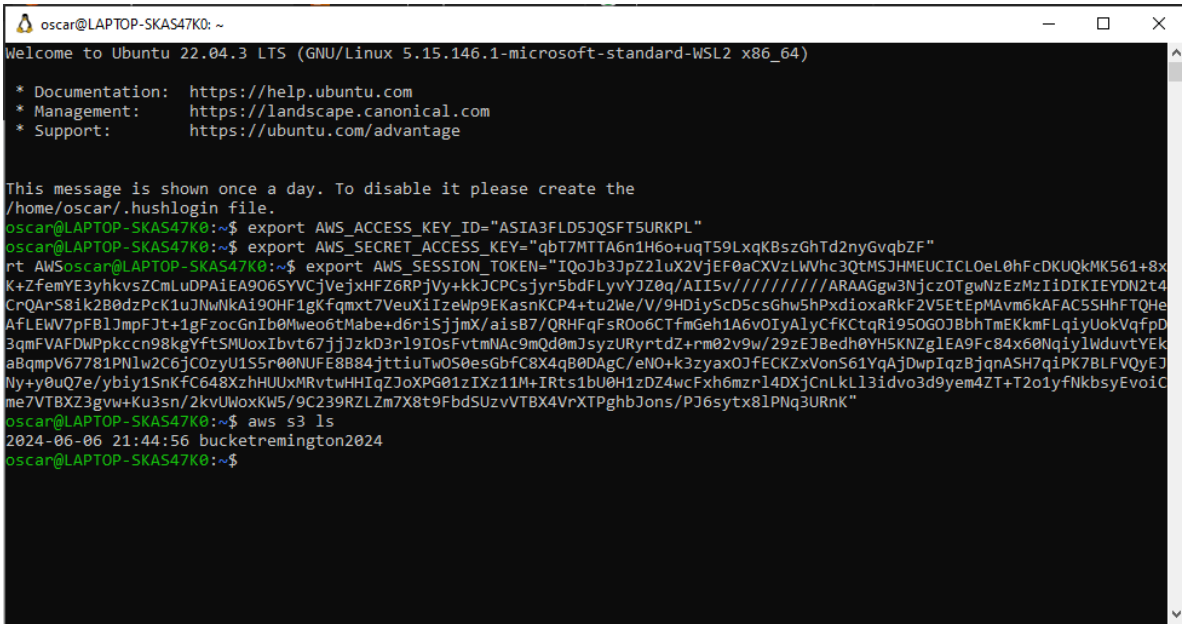
Figura 31 Copiar variables de acceso



Fuente: Elaboración propia

Pegamos las variables de acceso en la consola de WSL, con lo cual ingresa a AWS logueado con el usuario creado, luego con el comando “s3 ls” nos confirma que estamos conectados y nos muestra el Bucket creado inicialmente “bucketremington2024”.

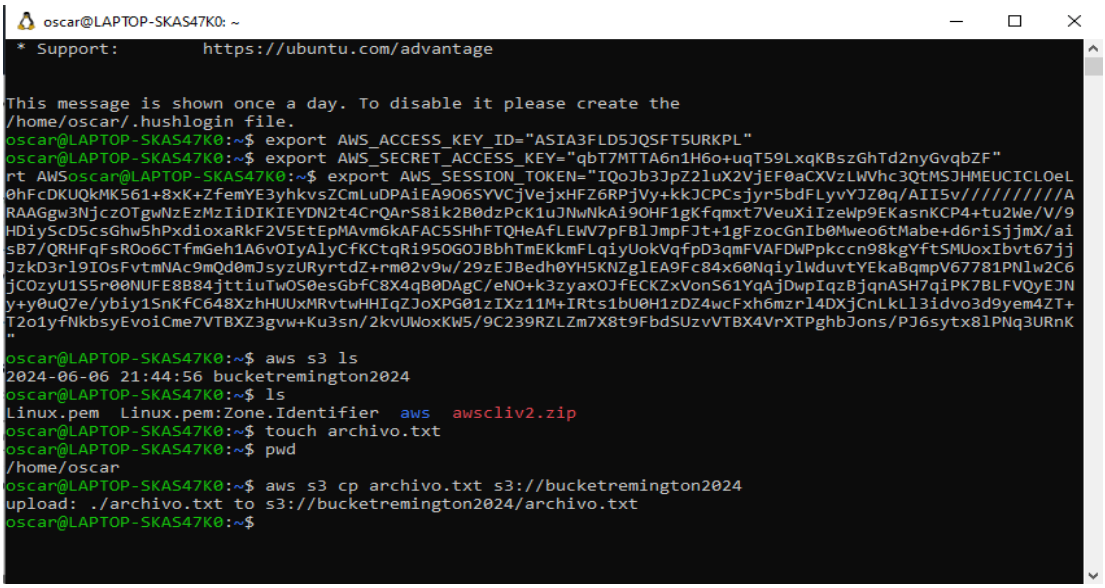
Figura 32 Uso comando s3 ls



Fuente: Elaboración propia

Con el comando “touch archivo.txt” creamos un archivo en la carpeta raíz de nuestro pc local y con el comando “aws s3 cp archivo.txt s3://bucketremington2024” donde s3://bucketremington2024 es la ruta del bucket la cual podemos copiar al momento de su creación. Finalmente cargamos el archivo al Bucket creado en AWS.

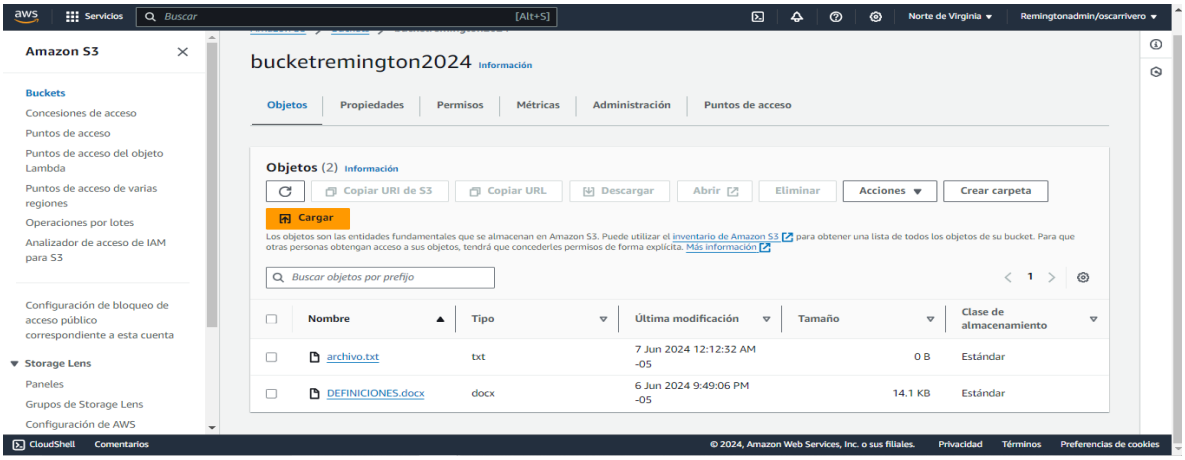
Figura 33 Cargue archivo al bucket



Fuente: Elaboración propia

En la siguiente imagen vemos el contenido del Bucket para verificar que el “archivo.txt” que cargamos desde la consola de WSL efectivamente se encuentra cargado en AWS.

Figura 34 Resumen cargue bucket

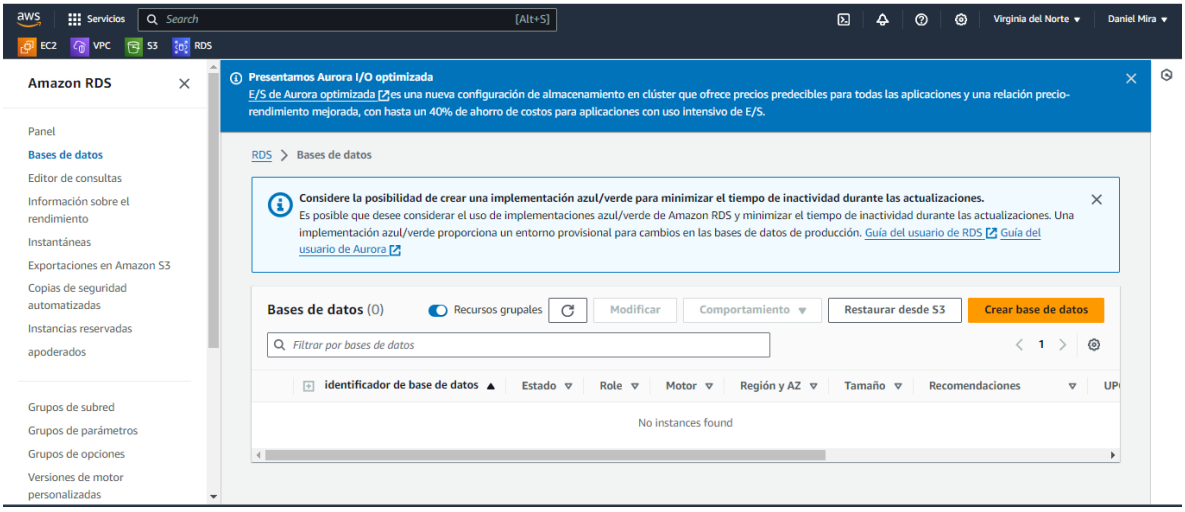


Fuente: Elaboración propia

A continuación, se describen los pasos para conectarnos por RDS a una base de datos creada en AWS.

Ingresamos por la opción RDS, Bases de datos, Crear base de datos.

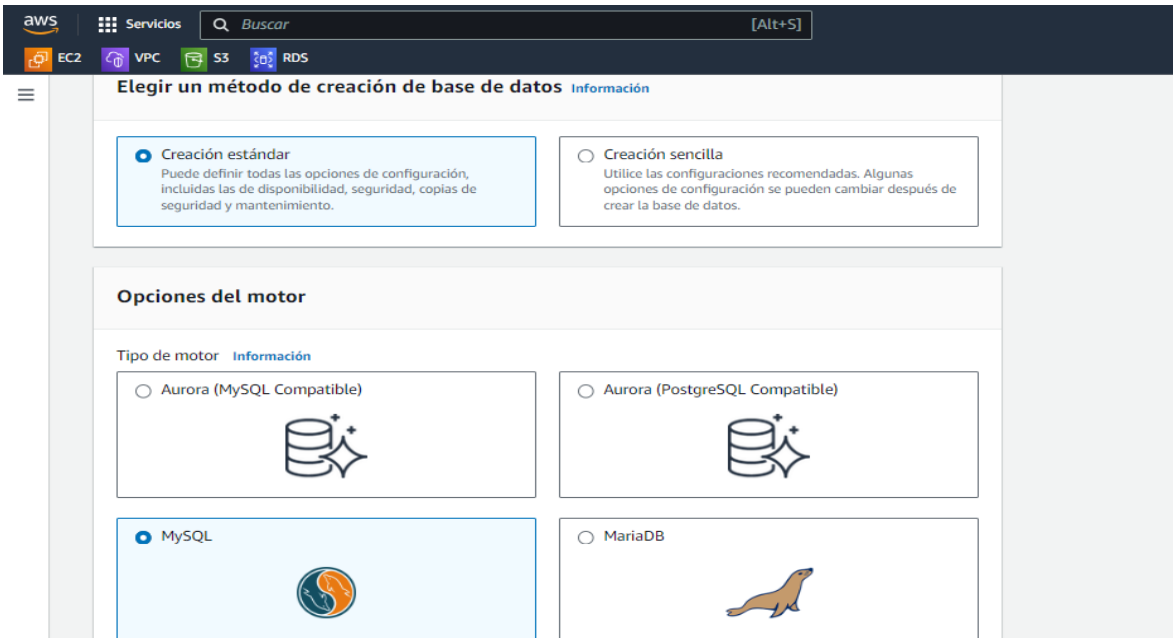
Figura 35 Conexión por RDS a base de datos



Fuente: Elaboración propia

Para este caso escogemos creación estándar, motor de base de datos MySQL.

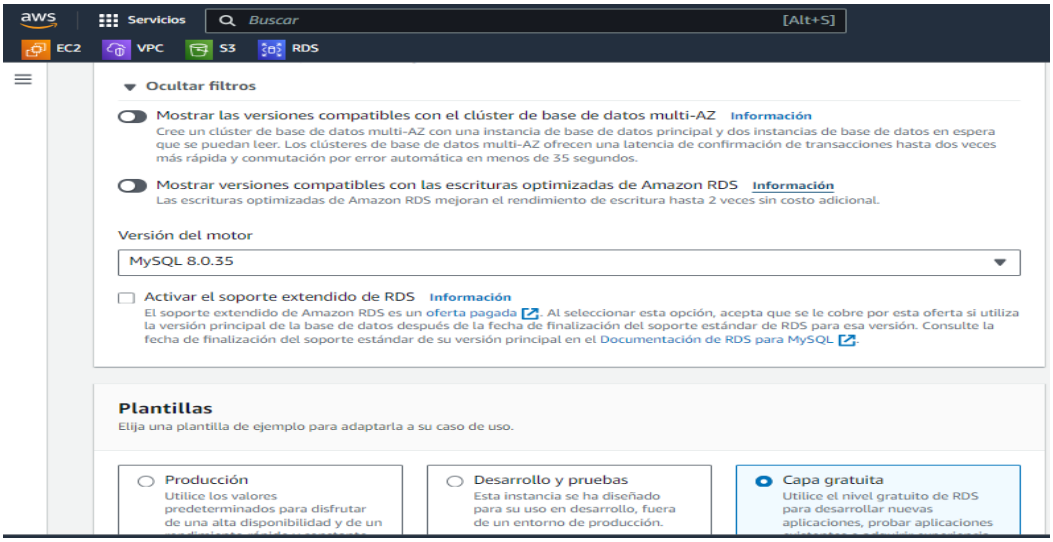
Figura 36 Creación BD MySQL



Fuente: Elaboración propia

Versión del motor de la base de datos y capa gratuita.

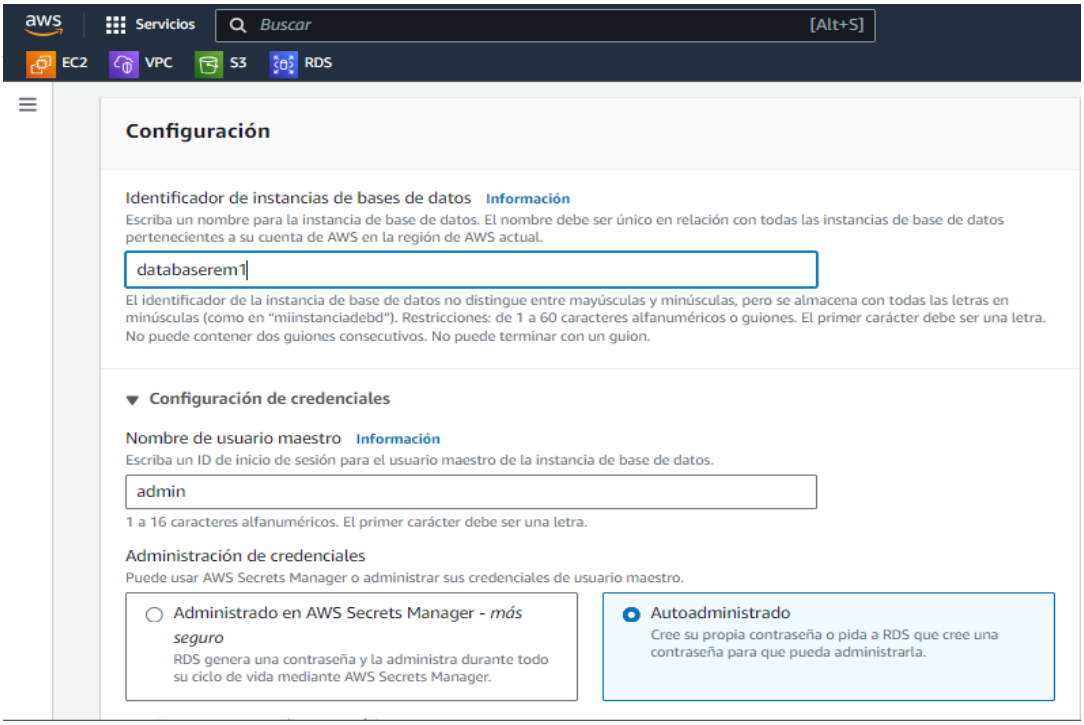
Figura 37 Versión motor y capa



Fuente: Elaboración propia

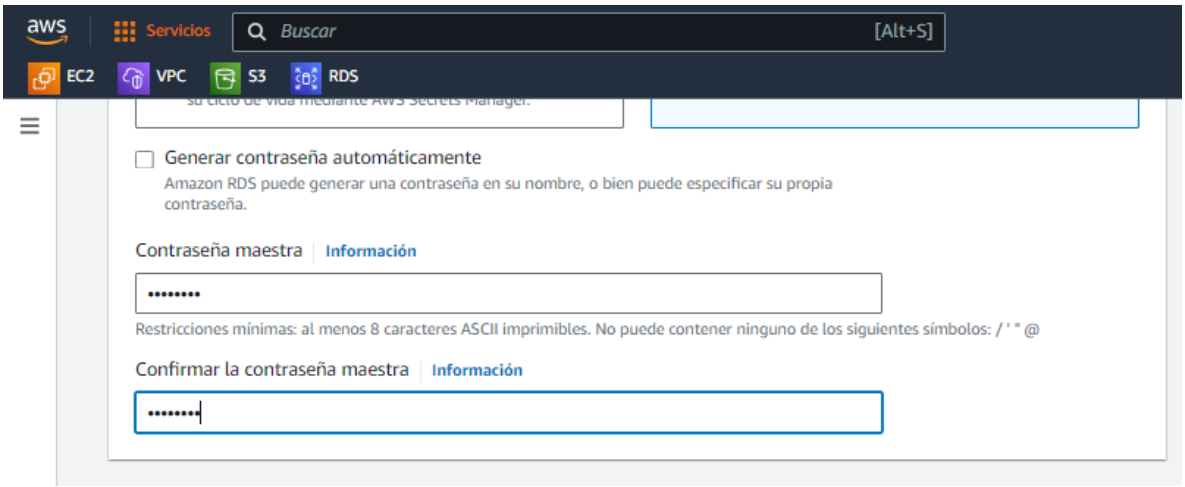
Establecemos el nombre de la base de datos, usuario y la opción autoadministrado.

Figura 38 Configuración BD



Fuente: Elaboración propia

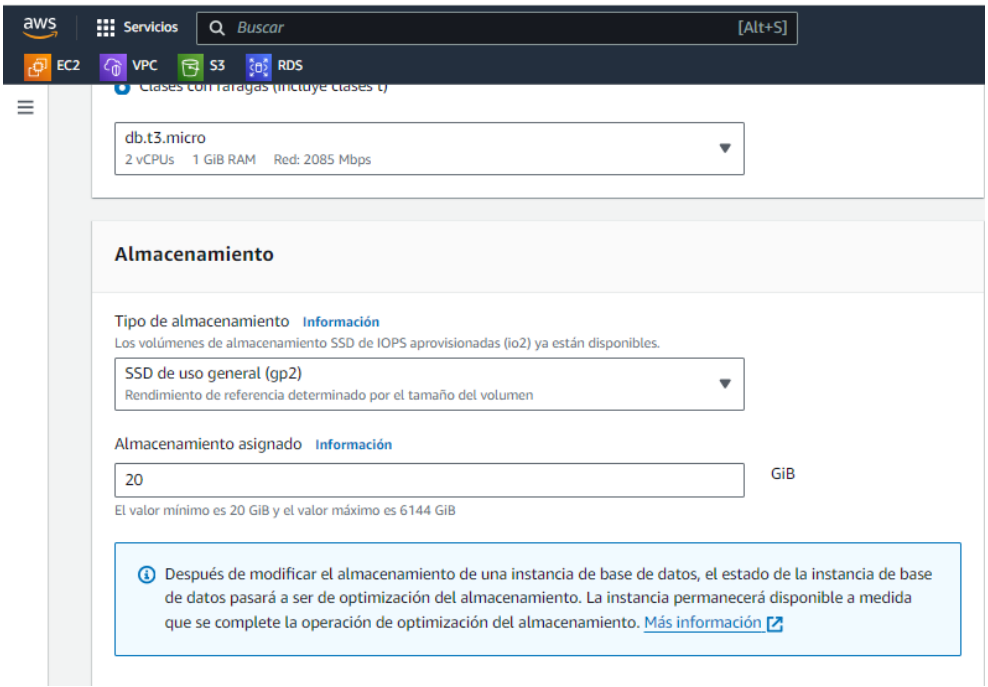
Figura 39 Asignación clave BD



Fuente: Elaboración propia

Escogemos la CPU, el tipo de disco duro y la cantidad de espacio de almacenamiento.

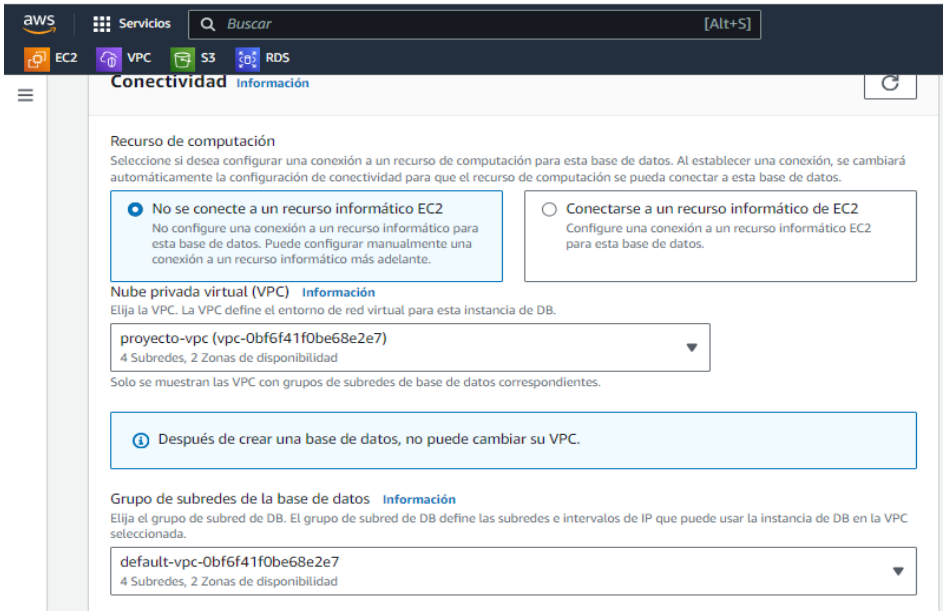
Figura 40 Selección disco y CPU



Fuente: Elaboración propia

Escogemos a que VPC y subred va a pertenecer nuestra base de datos.

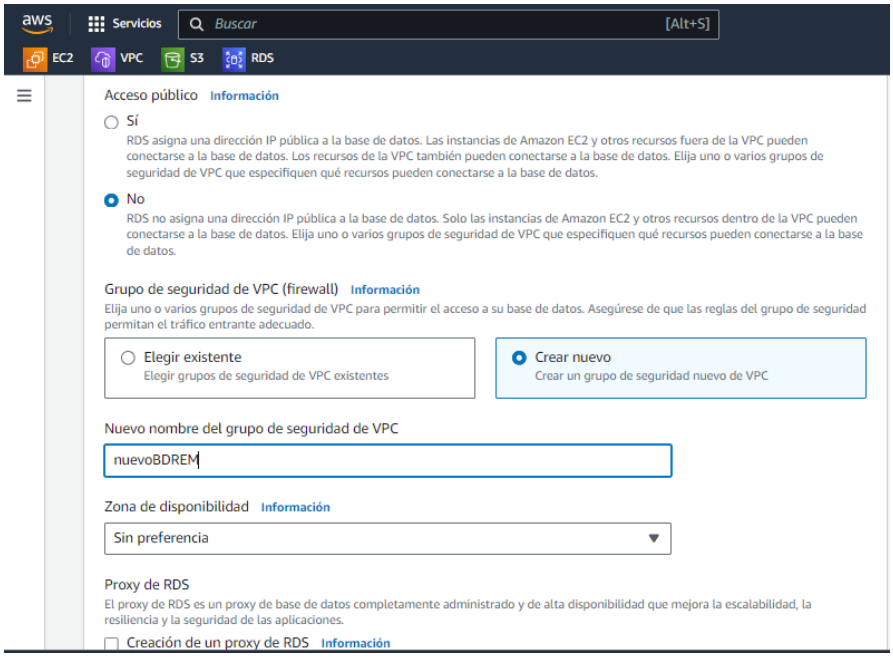
Figura 41 Selección VPC y subred



Fuente: Elaboración propia

Escogemos el tipo de acceso a la base de datos y el grupo de seguridad “nuevoBDREM” el cual posteriormente se modificará para permitir el acceso.

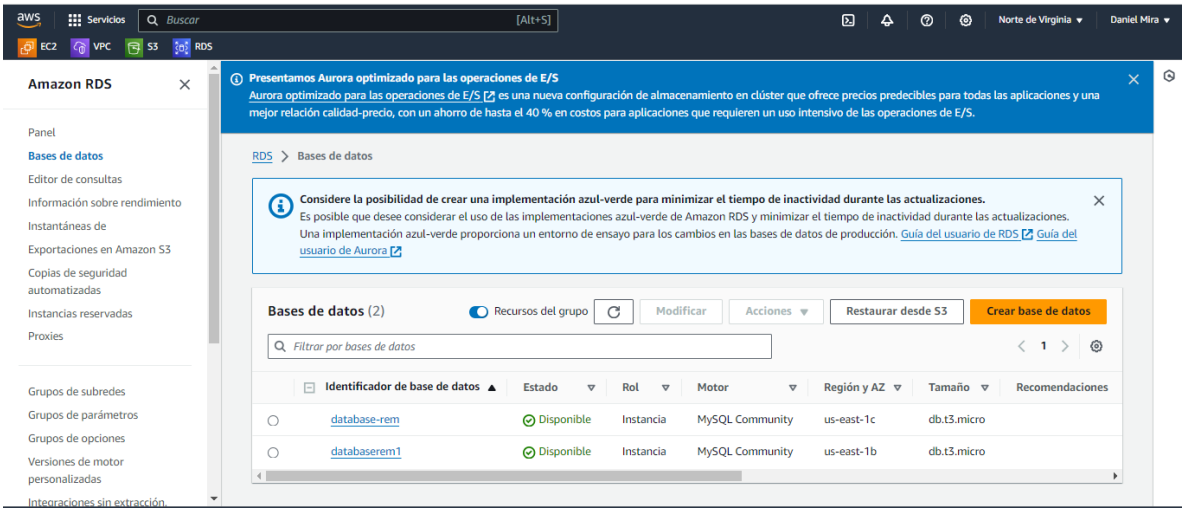
Figura 42 Selección tipo acceso a BD



Fuente: Elaboración propia

Creamos la base de datos y al volver a la opción base de datos muestra las ya creadas.

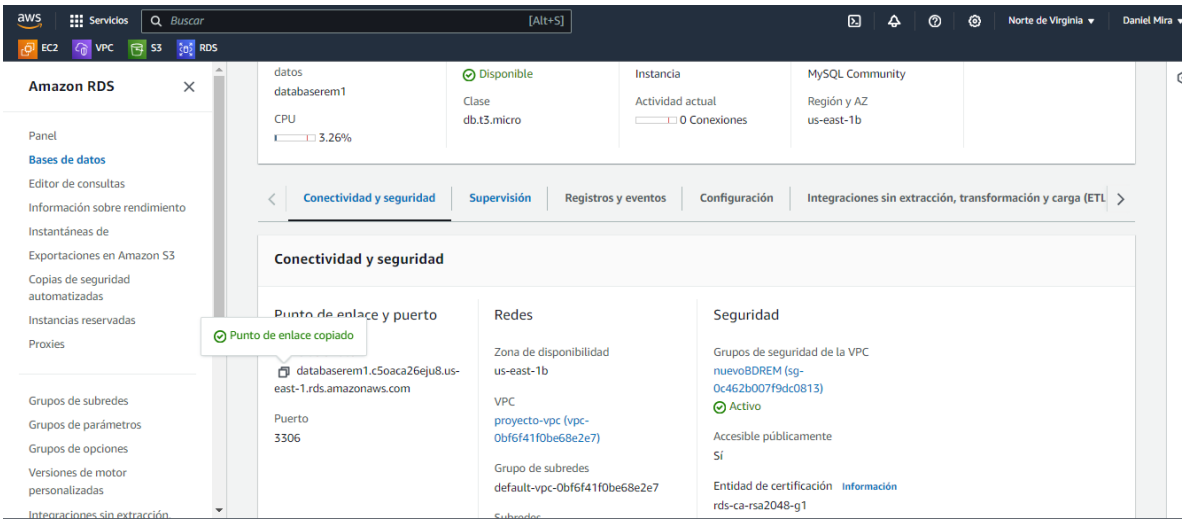
Figura 43 Resumen BD creadas



Fuente: Elaboración propia

Ingresamos a la base de datos y copiamos el punto de enlace con el cual nos conectaremos posteriormente.

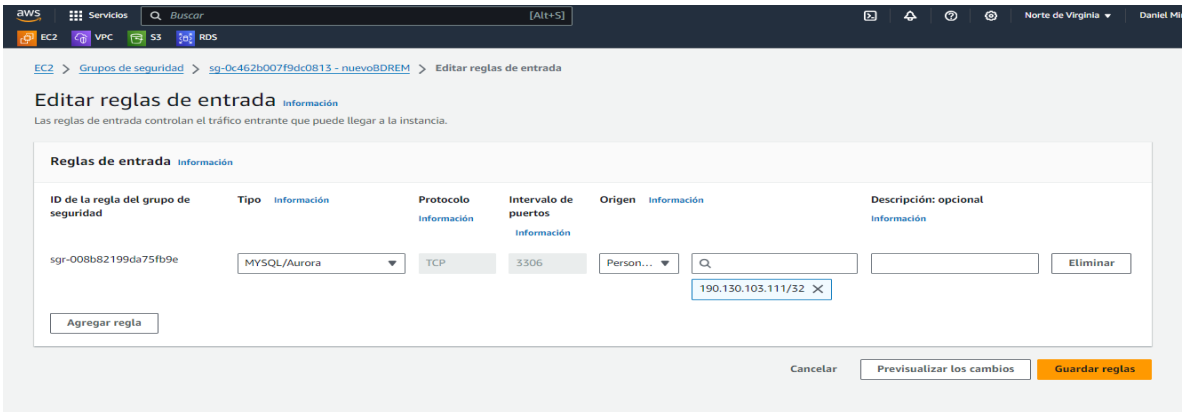
Figura 44 Copia punto enlace BD



Fuente: Elaboración propia

Ingresamos al grupo creado que fue asignado a la base de datos, el cual AWS automáticamente le asigna acceso solo a la ip fija del pc local donde se realizó la configuración.

Figura 45 Ingreso a grupo

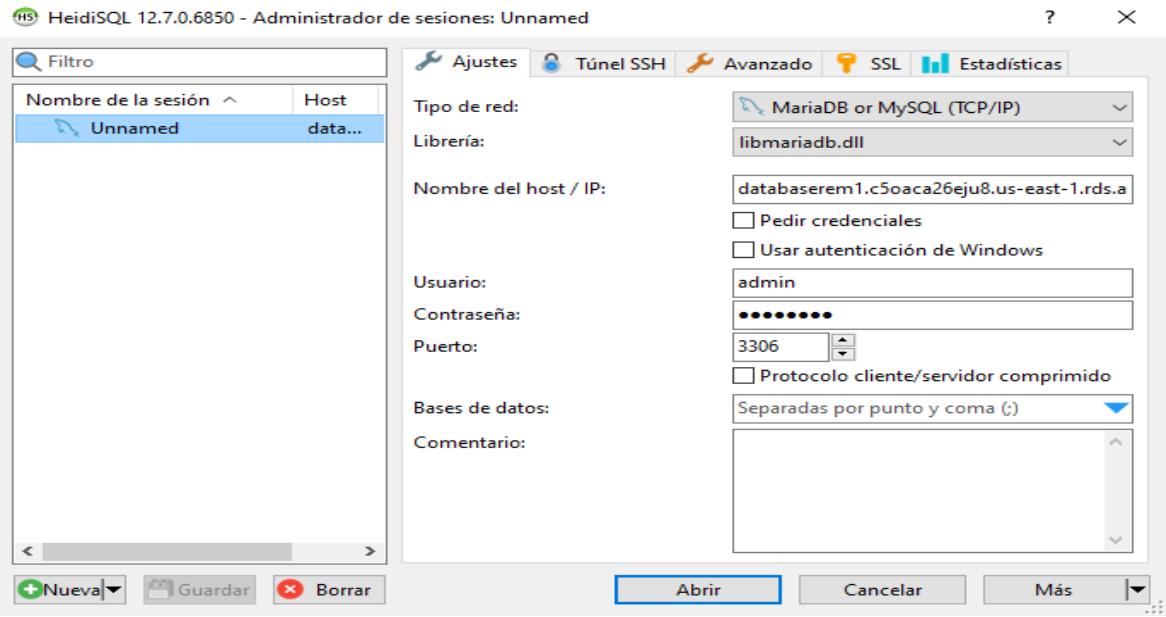


Fuente: Elaboración propia

El siguiente paso es descargar la aplicación heidiSQL de la página https://www.heidisql.com/download.php?download=portable-64

Esta aplicación nos permitirá conectarnos a la base de datos, debemos copiar la ruta de la base de datos la cual se genera al momento de crear la base de datos, usuario, contraseña y el puerto 3306.

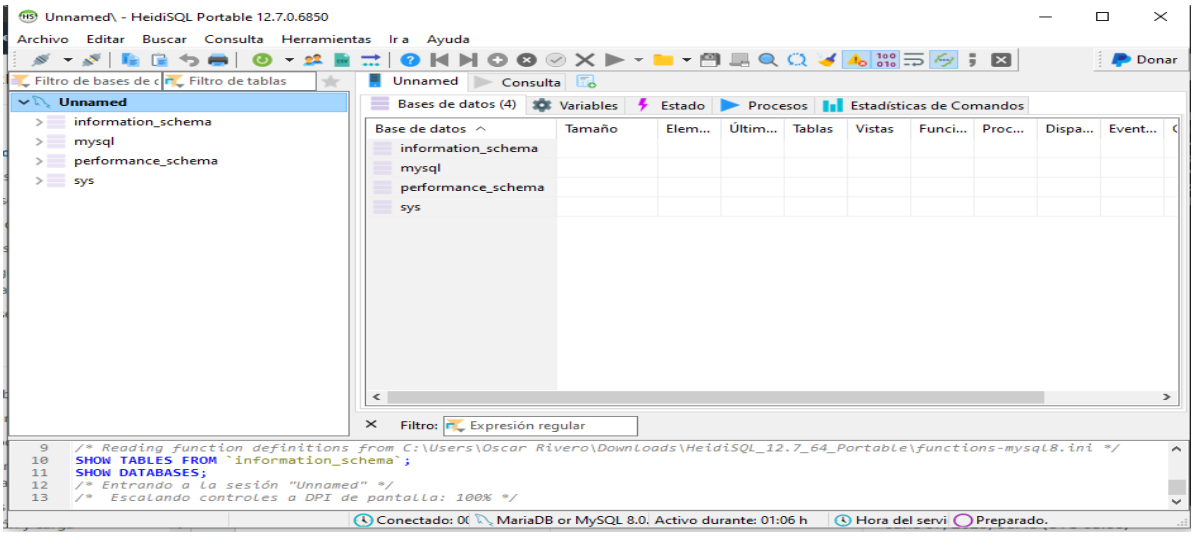
Figura 46 Uso aplicación Heidi



Fuente: Elaboración propia

Clic en abrir y establecemos la conexión a la base de datos creada en AWS.

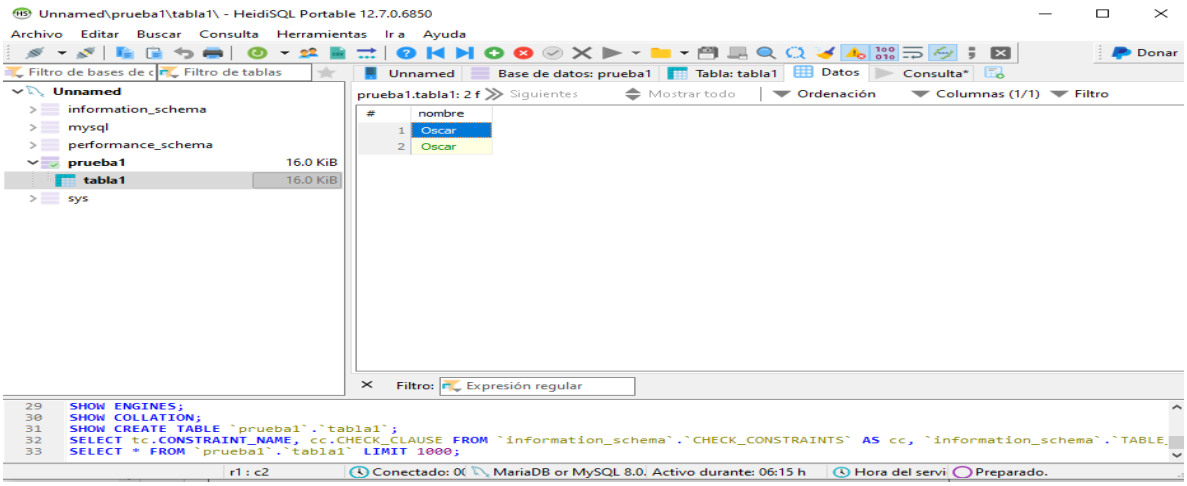
Figura 47 Conexión a BD creada desde Heidi



Fuente: Elaboración propia

Creamos la base de datos “prueba1” dentro la “tabla1” e insertamos un valor al campo nombre creado.

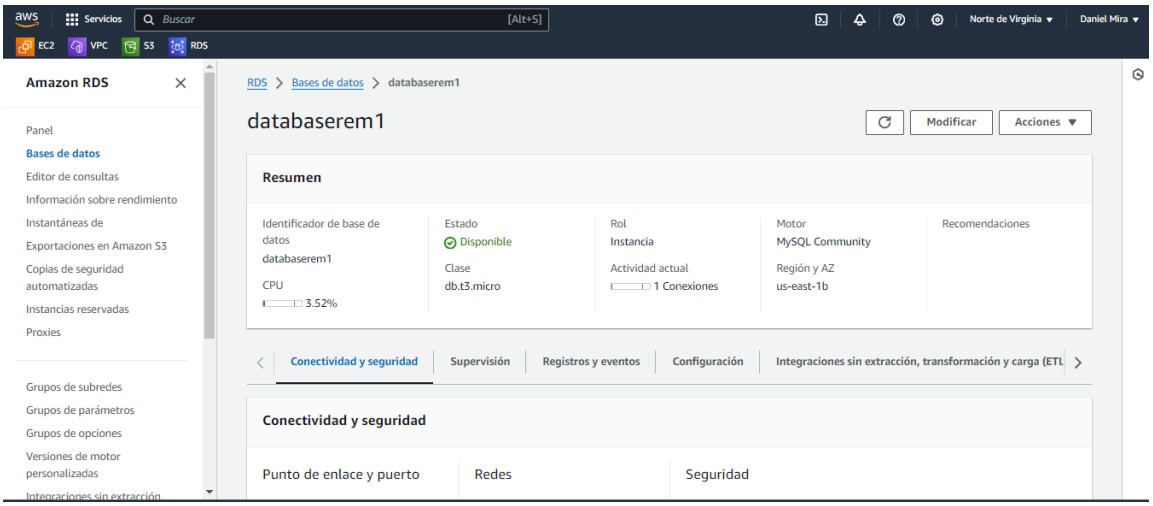
Figura 48 Creación BD "prueba1"



Fuente: Elaboración propia

Finalmente, al ingresar a la plataforma de AWS nos muestra que tenemos 1 conexión a la base de datos.

Figura 49 Resumen conexión



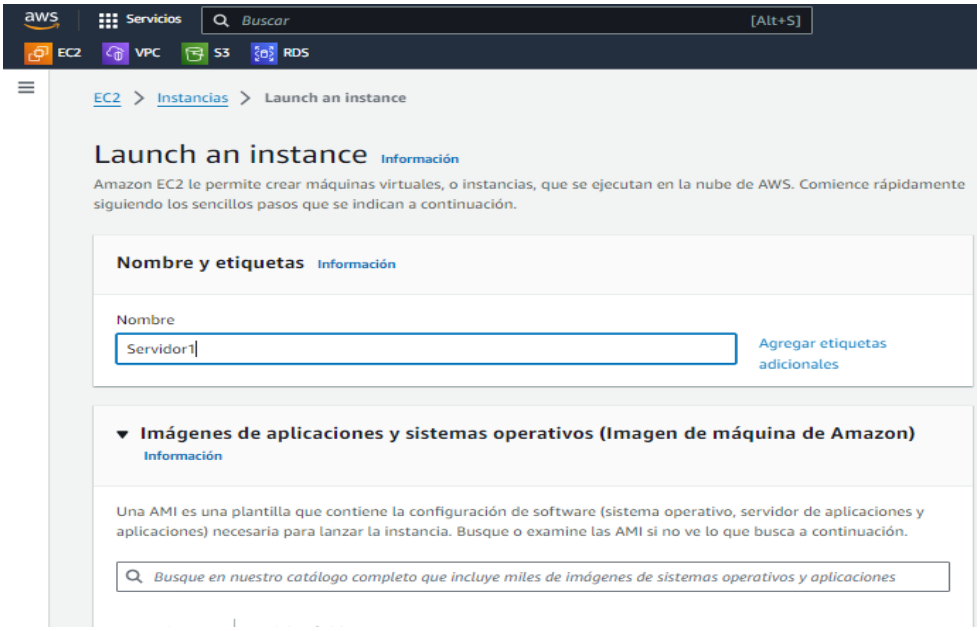
Fuente: Elaboración propia

Entrega 4

A continuación se describe el proceso para crear 2 instancias, configurar en cada una de ellas un servidor web, probar a través de acceso remoto la conexión a ambas instancias, luego las conectaremos a un balanceador de cargas y por último crearemos un grupo de Autoscaling.

Lo primero es la creación de las Instancias, en este caso nombraremos la primera instancia "Servidor1"

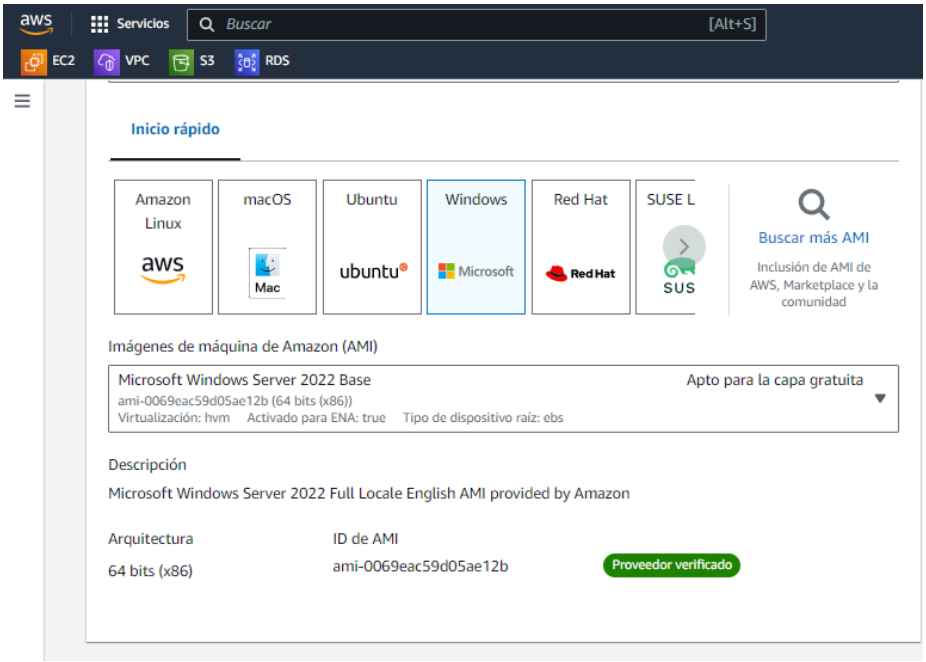
Figura 50 Creación instancia



Fuente: Elaboración propia

Escogemos para la instancia el sistema operativo Windows Server 2022.

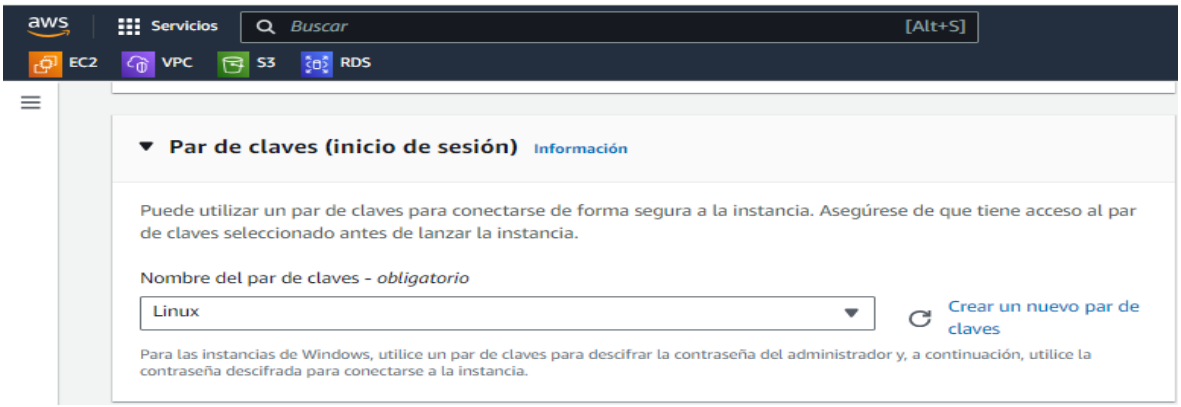
Figura 51Selección sistema operativo



Fuente: Elaboración propia

La clave es “Linux.pem” la cual ya tenemos creada y descargada.

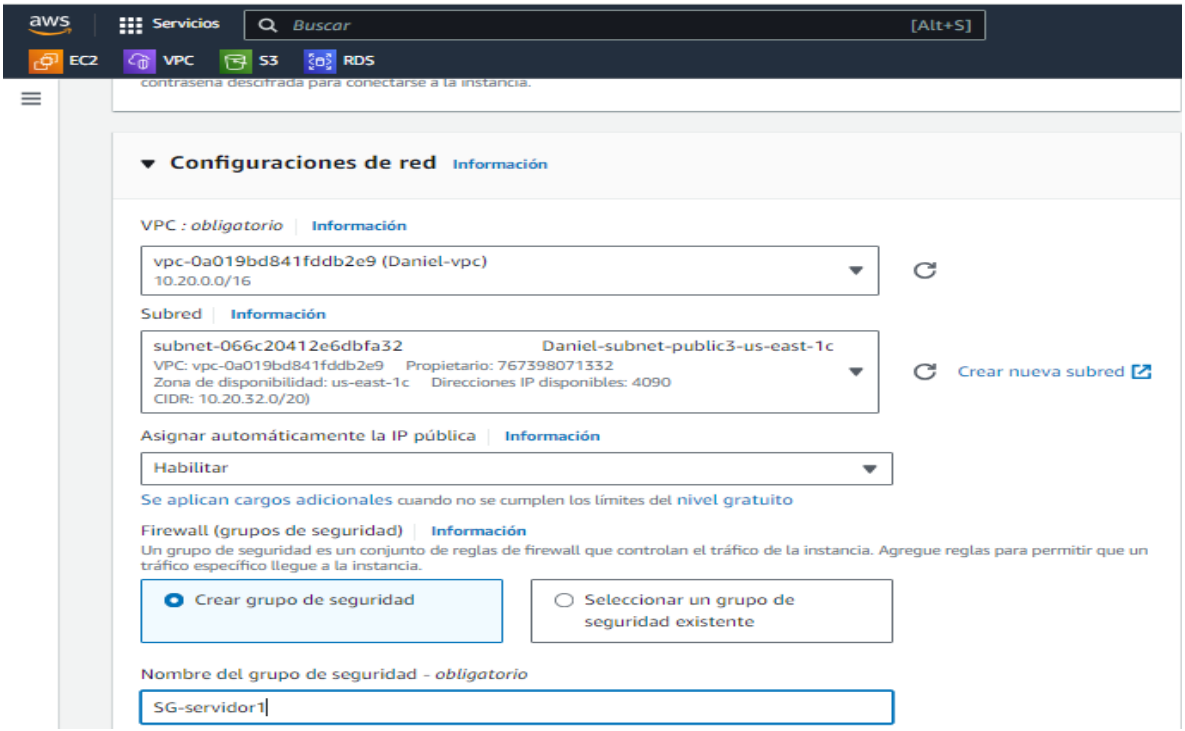
Figura 52 Uso clave



Fuente: Elaboración propia

En configuración de red escogemos el VPC “Daniel-vpc” el cual habíamos creado con anterioridad y la subred “Publi3 1c”, habilitamos la asignación de una dirección Ip pública para que podamos conectarnos desde cualquier equipo con conexión a internet y creamos el grupo de seguridad “SG-Servidor1”.

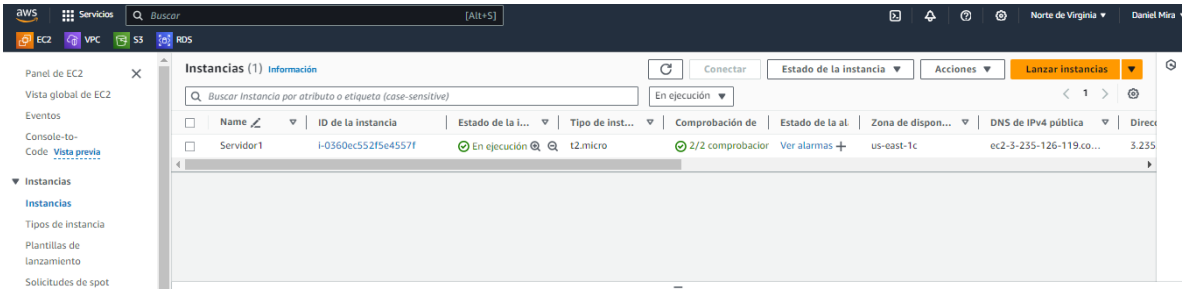
Figura 53 Selección VPC



Fuente: Elaboración propia

Como se muestra a continuación hemos creado la primera instancia “Servidor1”.

Figura 54 Resumen instancia "Servidor1"



Fuente: Elaboración propia

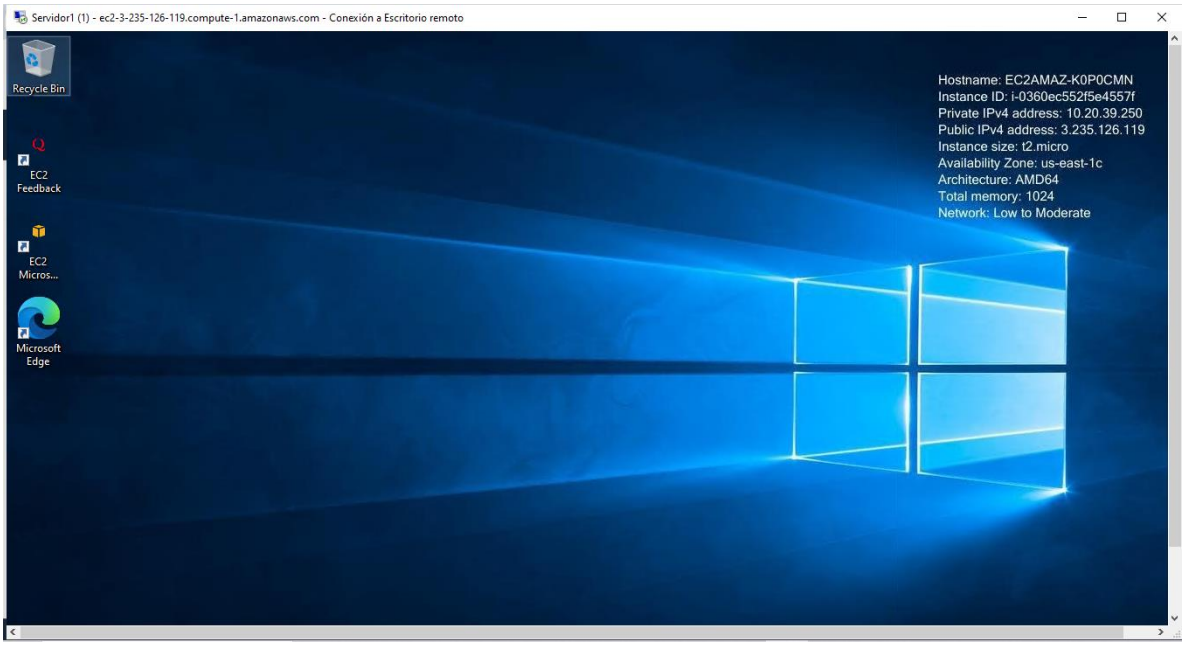
Teniendo seleccionada la instancia damos clic en Conectar, Cliente de RDP, cargando la clave nos genera la contraseña para acceder a la instancia, para lo cual damos clic en la opción Descargar archivo de escritorio remoto.

Figura 55 Archivo escritorio remoto

Fuente: Elaboración propia

Ingresamos la contraseña y nos abre la conexión remota a la instancia “Servidor1”.

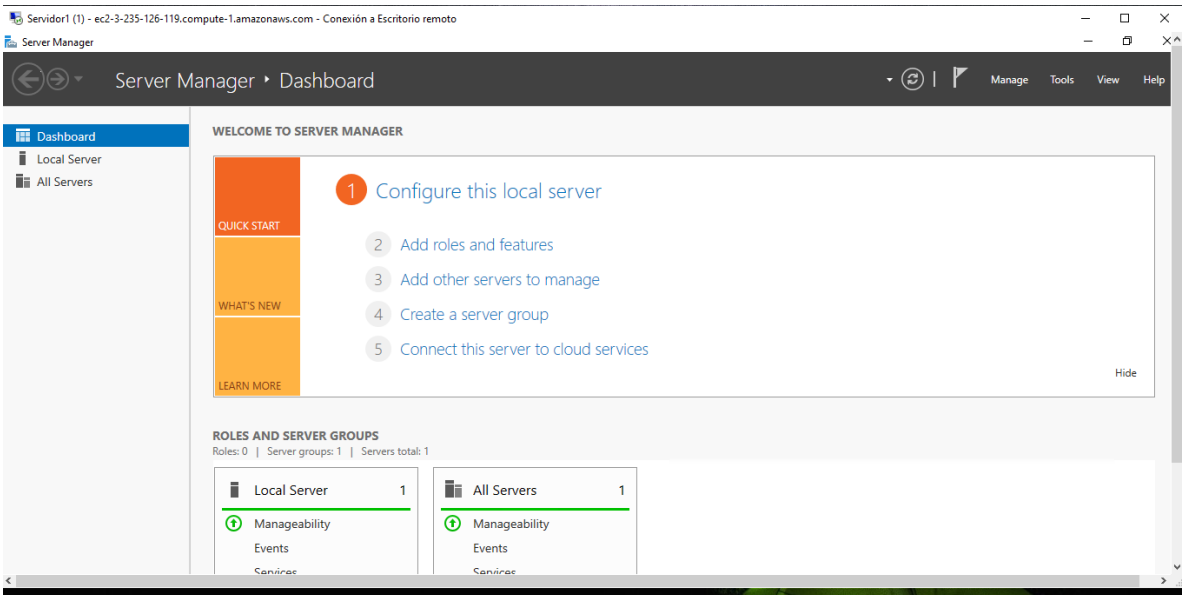
Figura 56 Acceso a conexión remota



Fuente: Elaboración propia

A continuación, ingresamos a la opción server manager donde vamos a habilitar el servidor web de la instancia.

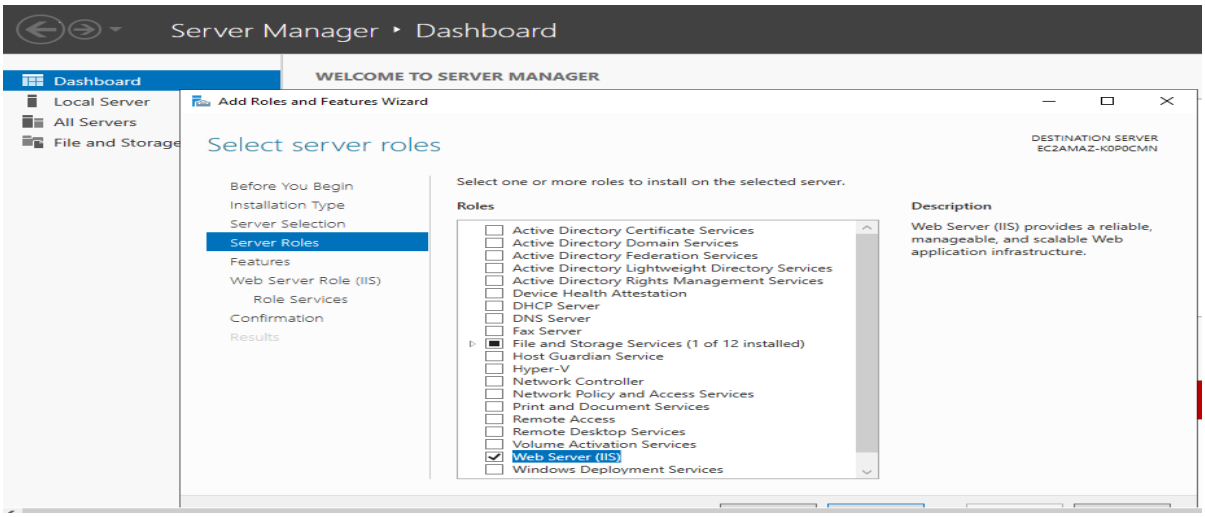
Figura 57 Ingreso a server manager



Fuente: Elaboración propia

Habilitamos el servicio Internet Information Server.

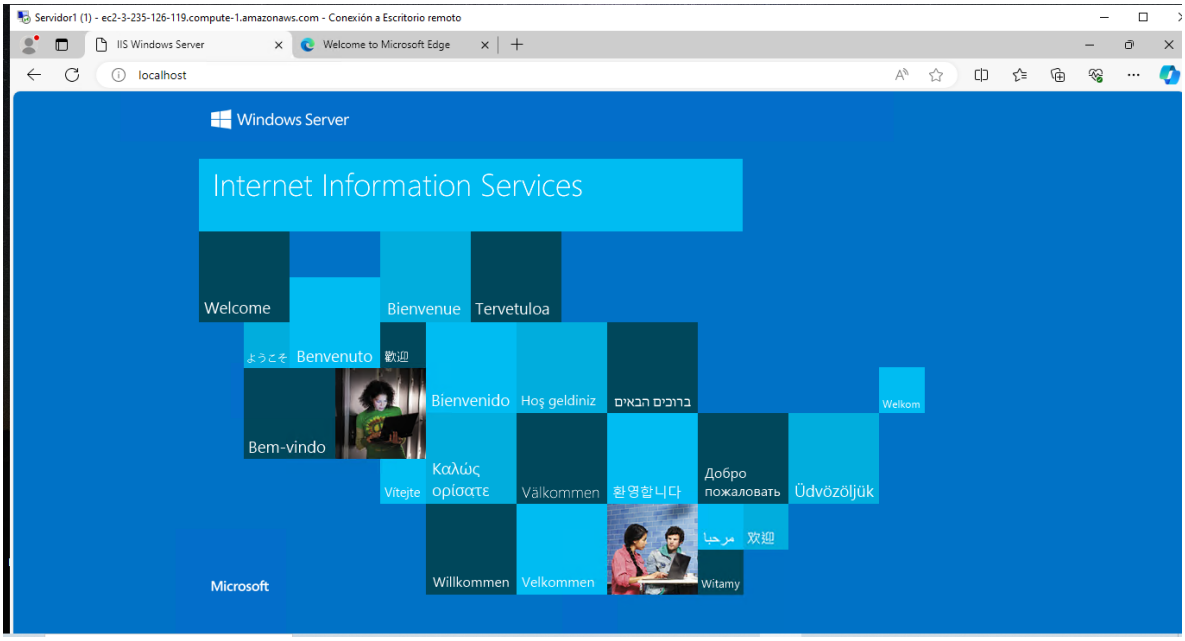
Figura 58 Habilitar servicio internet information server



Fuente: Elaboración propia

Abrimos el navegador y en la barra de búsqueda ingresamos “localhost”, con lo cual nos carga la página de inicio del servidor web.

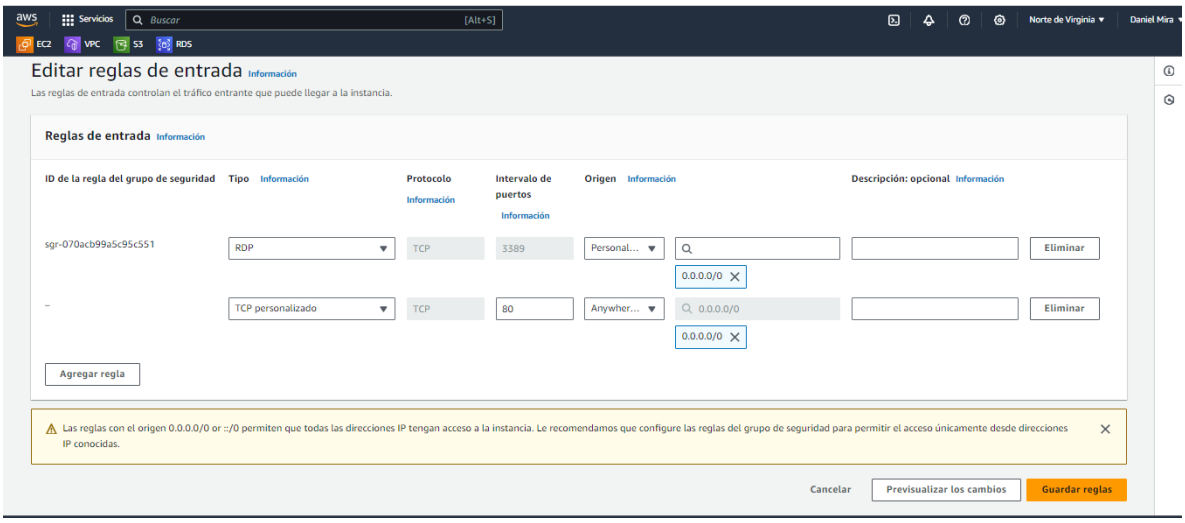
Figura 59 Cargue página de inicio



Fuente: Elaboración propia

El paso siguiente es conectarnos a la Instancia desde el navegador de nuestro computador conectado a internet, ya que establecimos una dirección Ip pública para la Instancia debemos editar el grupo de seguridad asociada la instancia “SG-servidor1” agregándole una regla de entrada que habilite el puerto 80 para cualquier conexión desde internet.

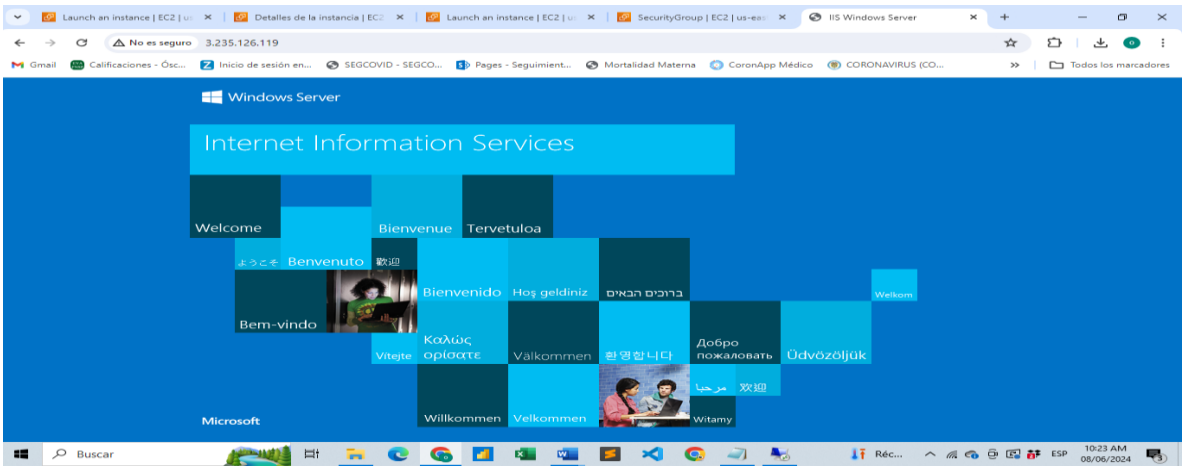
Figura 60 Configuración grupo de seguridad



Fuente: Elaboración propia

En la imagen a continuación se muestra la conexión desde nuestro navegador local a través de la Ip pública de la instancia 3.325.126.119 la cual ingresa directamente al servidor web configurado en el puerto 80.

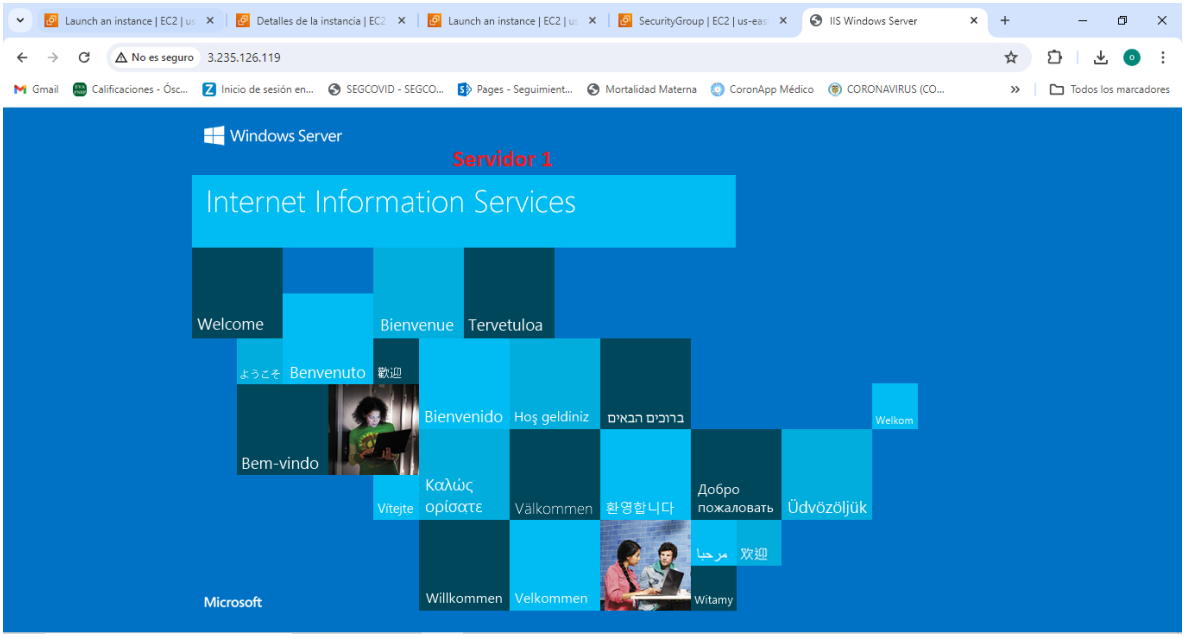
Figura 61 Conexión desde navegador



Fuente: Elaboración propia

Se realiza la modificación de la página de inicio del servidor web de la instancia para identificar que se trata de “Servidor1”.

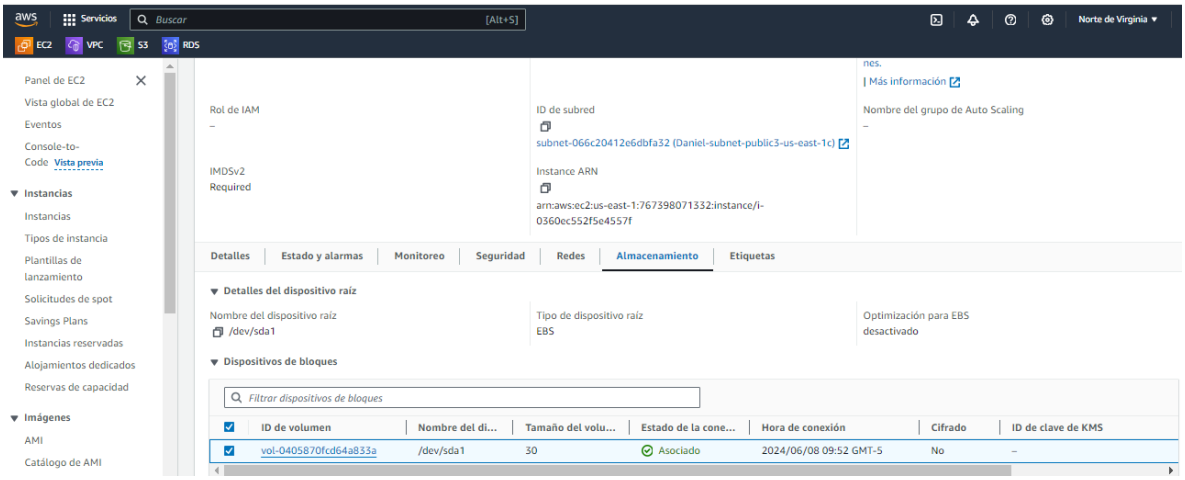
Figura 62 Modificación página inicio "Servidor1"



Fuente: Elaboración propia

El siguiente paso es crear otra instancia, pero lo vamos a hacer a partir de la instancia ya creada para lo cual ingresamos a la opción almacenamiento y accedemos al volumen, el cual es el disco asociado a la instancia.

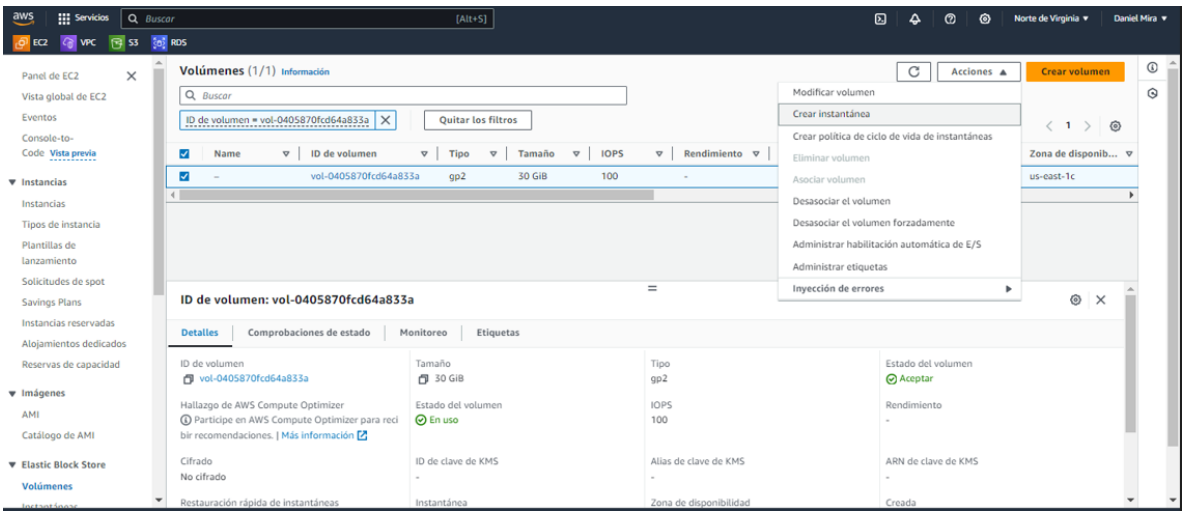
Figura 63 Creación instancia



Fuente: Elaboración propia

El paso siguiente es crear una instantánea o Snapshot que es una copia exacta de la configuración de la instancia Servidor1.

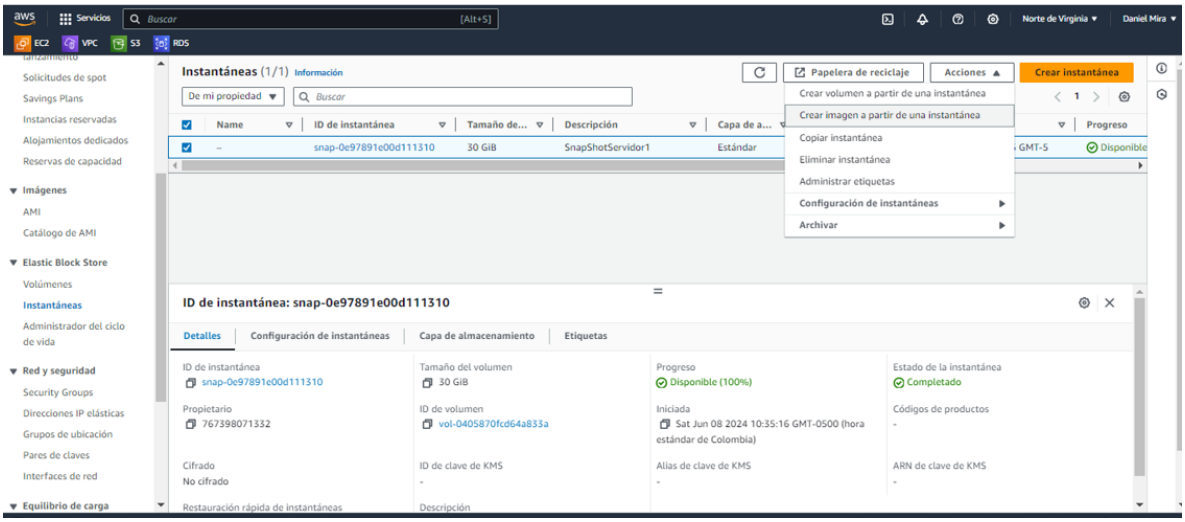
Figura 64 Creación snapshot



Fuente: Elaboración propia

A continuación, dentro del Snapshot damos clic en la opción Crear imagen a partir de una instantánea.

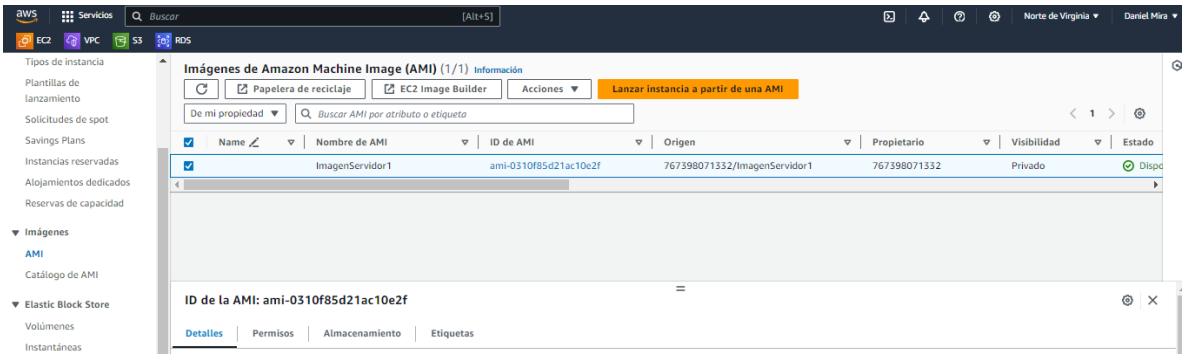
Figura 65 Crear imagen a partir de Snapshot



Fuente: Elaboración propia

Hemos creado la imagen “imagenServidor1”, la cual nos servirá para crear una segunda Instancia.

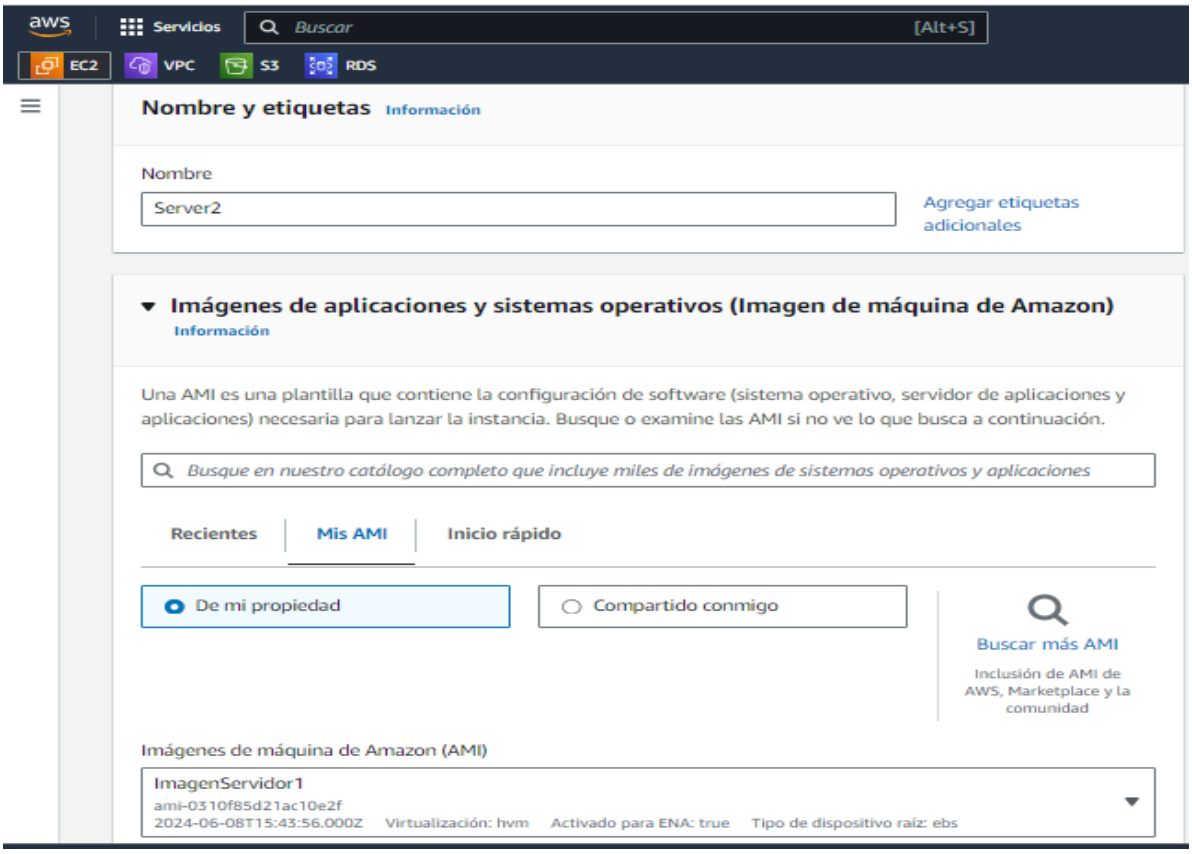
Figura 66 Resumen imagen creada



Fuente: Elaboración propia

Creación de la segunda Instancia la cual llamaremos “Server2”, escogemos la imagen creada en el paso anterior “ImagenServidor1”.

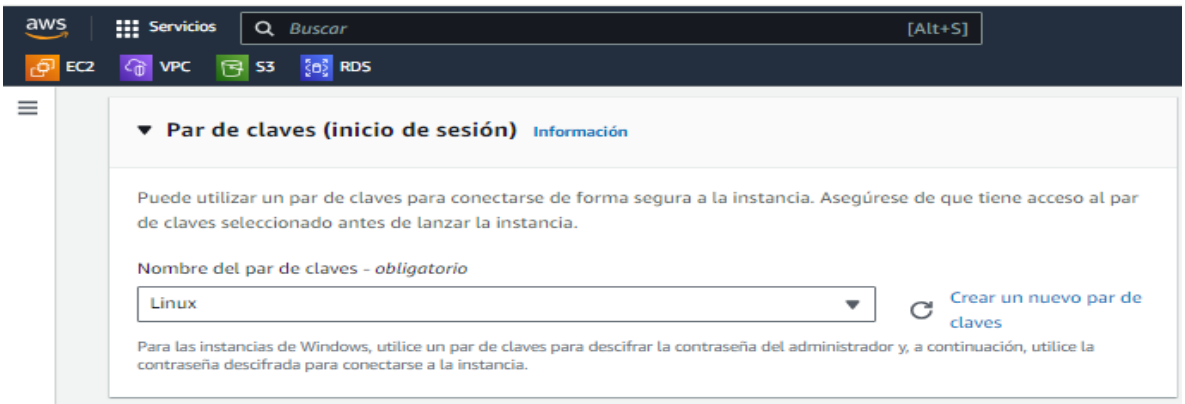
Figura 67 Creación segunda instancia "server2"



Fuente: Elaboración propia

La clave de esta segunda instancia será la misma Linux.pem.

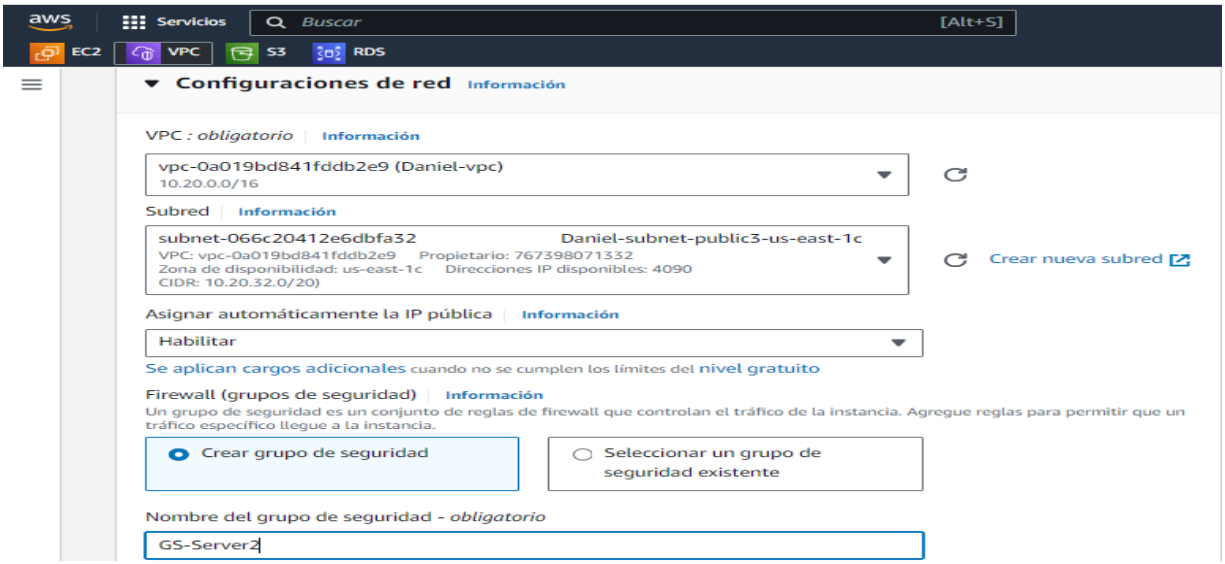
Figura 68 Clave instancia



Fuente: Elaboración propia

Escogemos el VPC, la subred la cual debe ser la misma en la que creamos la primera Instancia y creamos el grupo de seguridad el cual llamaremos “GS-Server2”

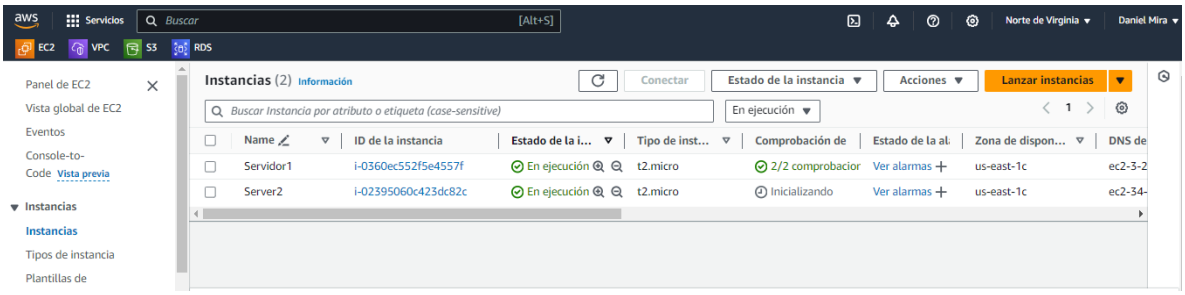
Figura 69 Selección VPC y subred



Fuente: Elaboración propia

A continuación, nos muestra que tenemos 2 instancias creadas, las cuales se encuentran en ejecución.

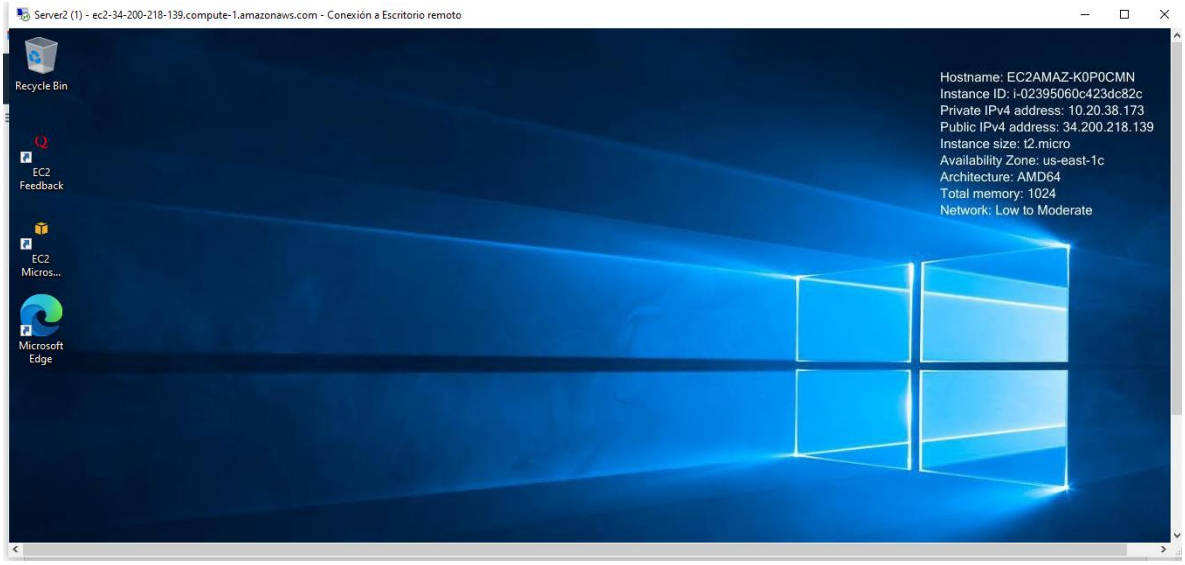
Figura 70 Resumen y estado de instancias



Fuente: Elaboración propia

Nos conectamos a “Server2” desde el escritorio remoto, con la misma clave de “Servidor1”, ya que se trata de una copia exacta.

Figura 71 Conexión a "server2"

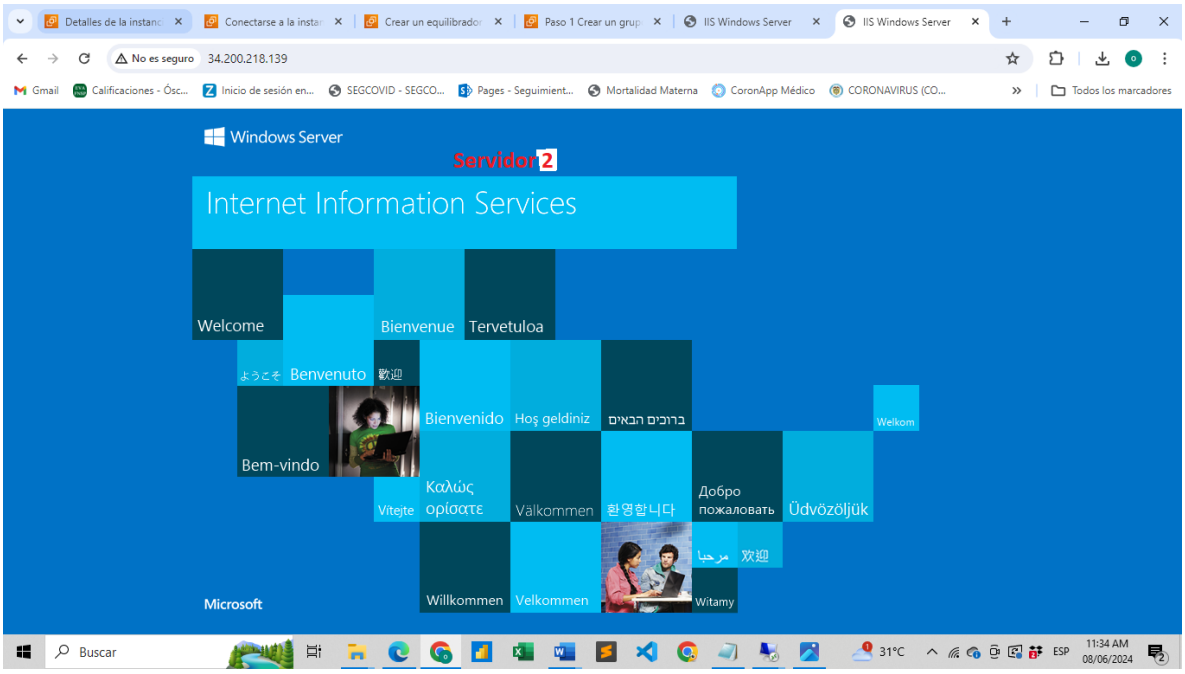


Fuente: Elaboración propia

Realizamos la modificación en la página de inicio del servidor web “Server2” y nos conectamos desde nuestro equipo local con la Ip Pública asignada al momento de crear la instancia.

En este momento tenemos las 2 instancias creadas con conexión a través de Ip pública desde internet al servidor web configurado.

Figura 72 Conexión desde IP pública



Fuente: Elaboración propia

El siguiente paso es la creación del Balanceador de carga, el cual tendrá conexión directa con las instancias, recibirá las solicitudes realizadas por los usuarios y las direccionará de forma equitativa a cada una.

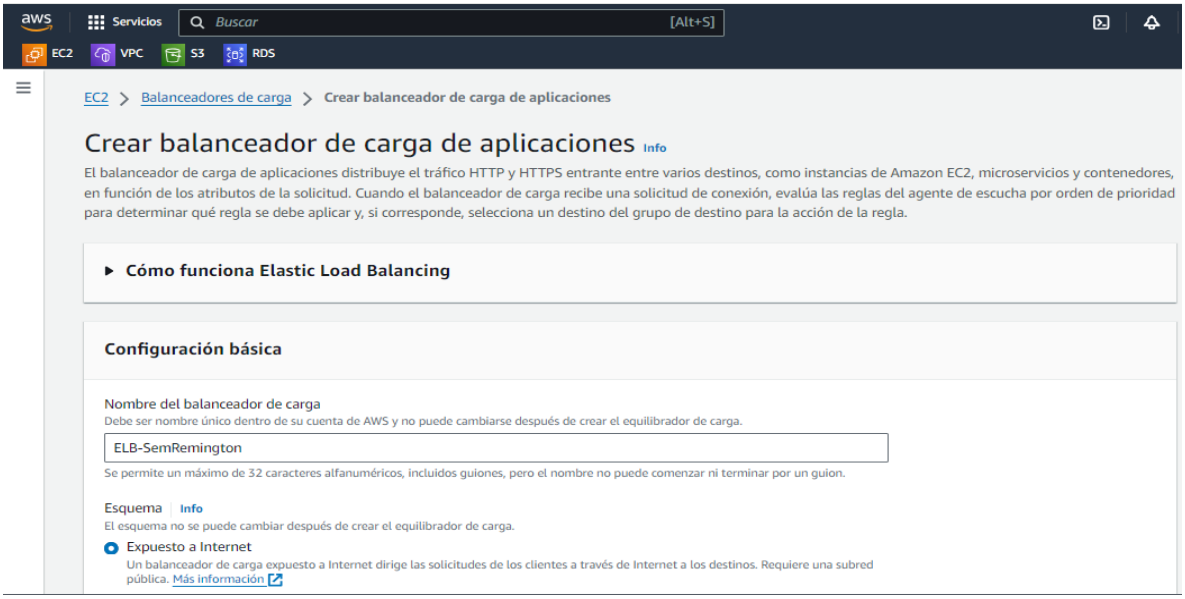
Escogemos la opción Balanceador de carga de aplicaciones.

Figura 73 Creación balanceador de carga

Fuente: Elaboración propia

Nombre del Balanceador de carga “ELB-SemRemington”.

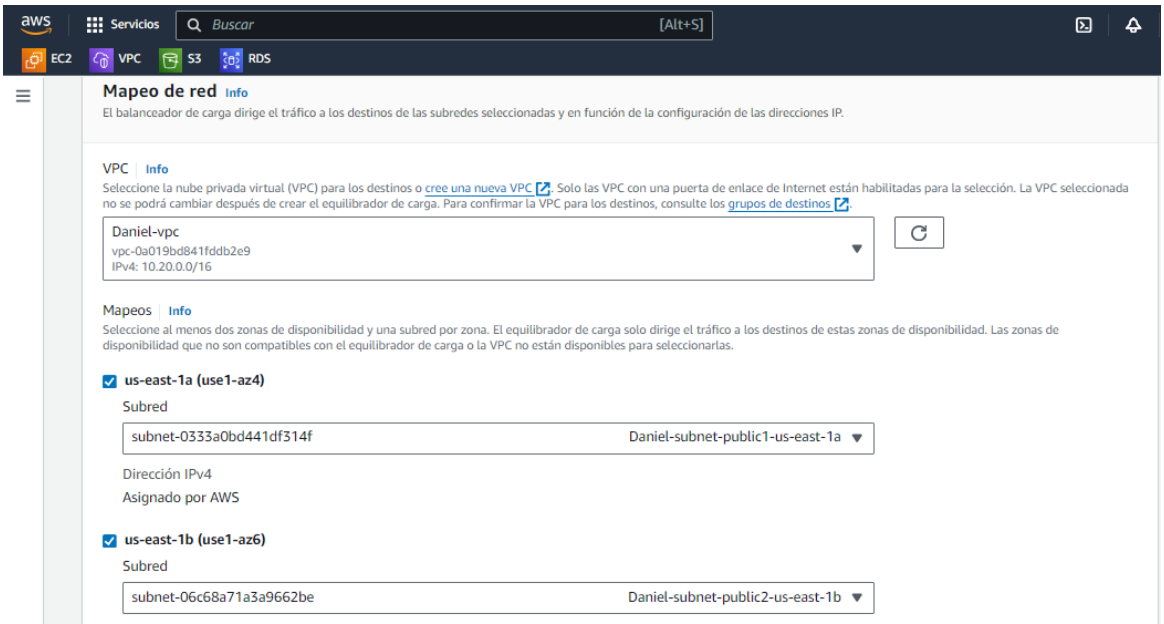
Figura 74 Asignación nombre balanceador



Fuente: Elaboración propia

Escogemos el VPC y las subredes, las cuales deben ser las mismas a las que se encuentran conectadas las Instancias.

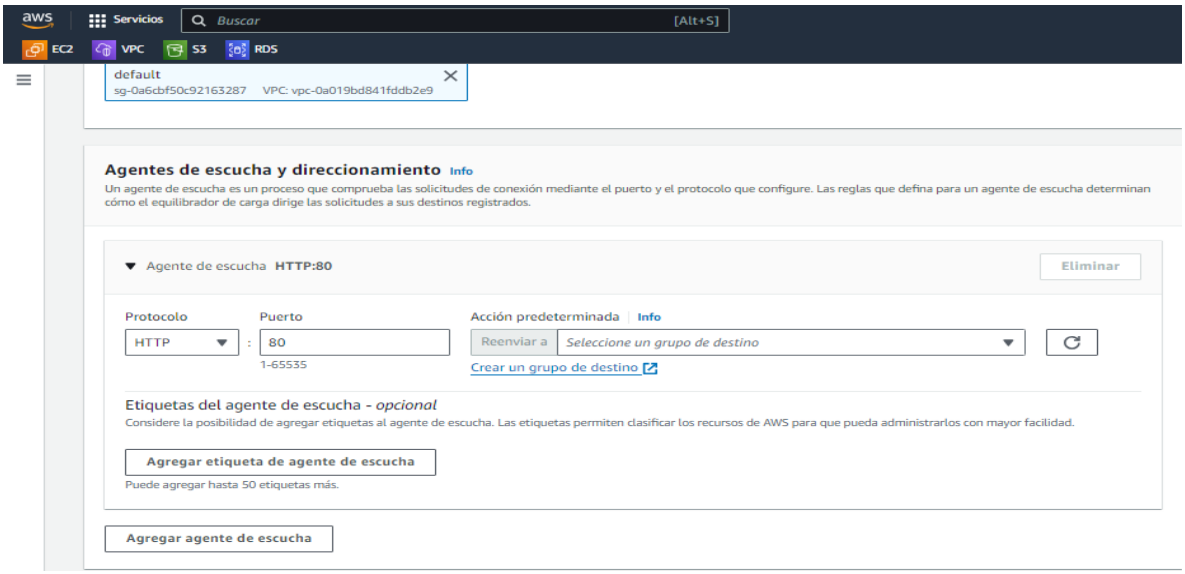
Figura 75 Selección VPC



Fuente: Elaboración propia

A continuación, vamos a crear el “Target Group” o Grupo de destino.

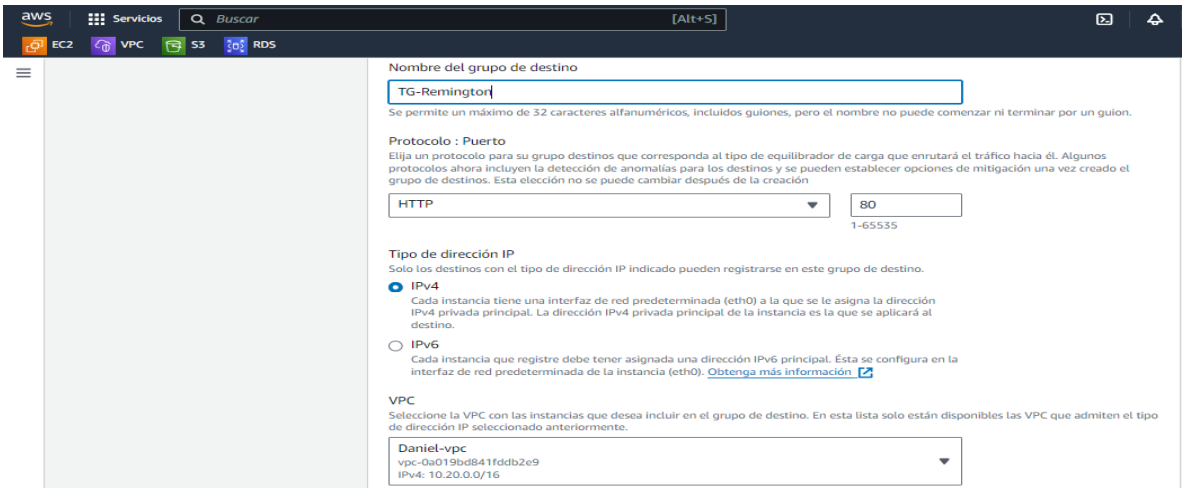
Figura 76 Creación Target Group



Fuente: Elaboración propia

El grupo de destino lo nombraremos “TG-Remington”, establecemos la conexión a través del puerto 80, IPV4 y escogemos el mismo VPC en el cual estamos trabajando.

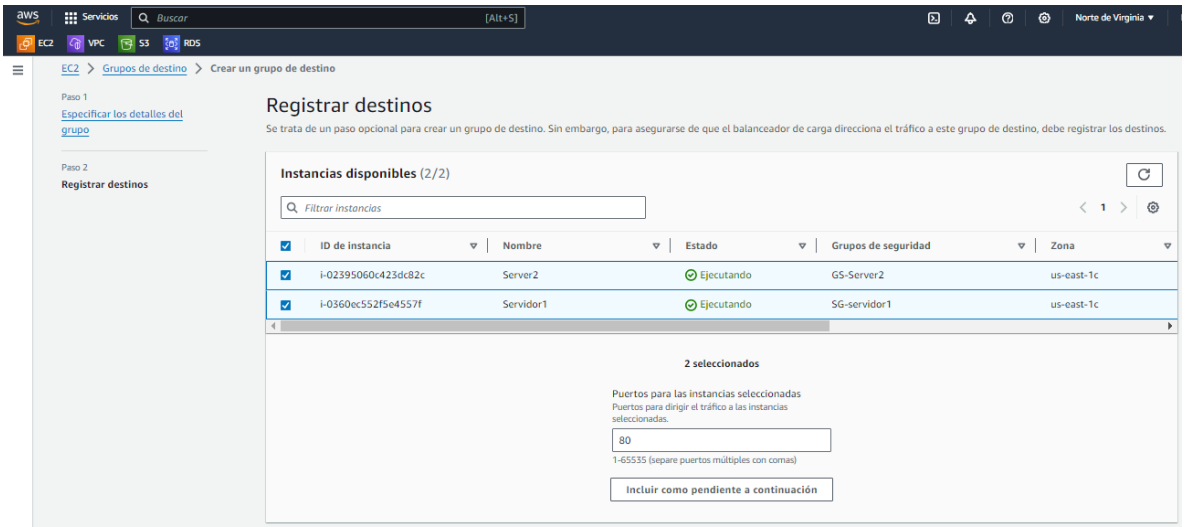
Figura 77Configuración Target Group



Fuente: Elaboración propia

Seleccionamos las instancias que vamos a incluir en el Target Group que para este caso son Servidor1 y Server2.

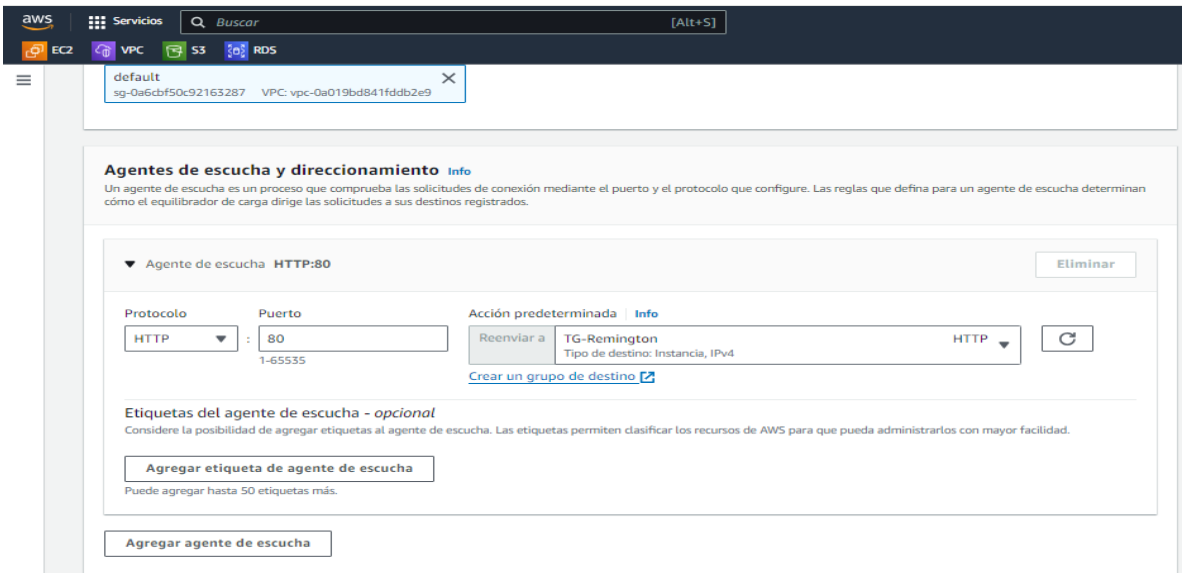
Figura 78 Selección instancia a incluir en TG



Fuente: Elaboración propia

Volvemos a la creación del Balanceador de carga y seleccionamos el grupo “TG-Remington” y le damos en crear.

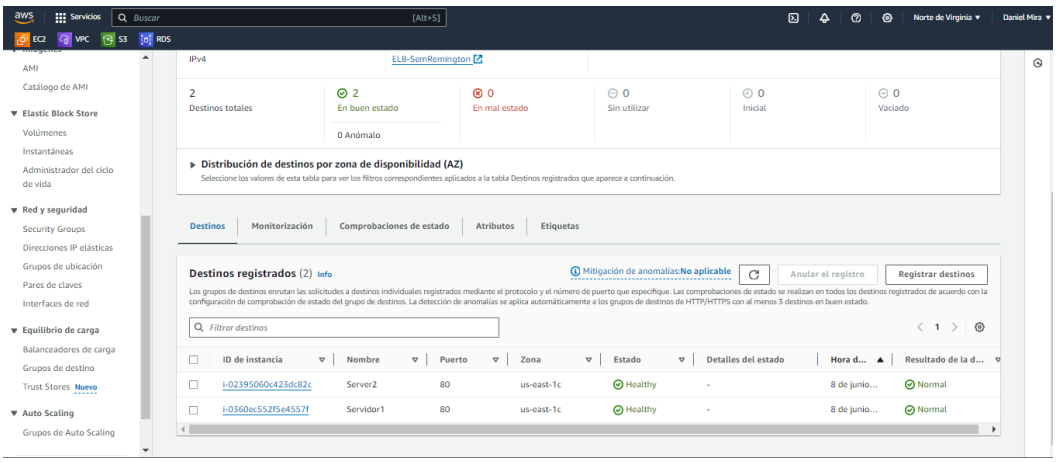
Figura 79 Creación grupo TG-Remington



Fuente: Elaboración propia

En la siguiente imagen vemos el balanceador de carga “ELB-SemRemington” que acabamos de crear con las 2 instancias asociadas.

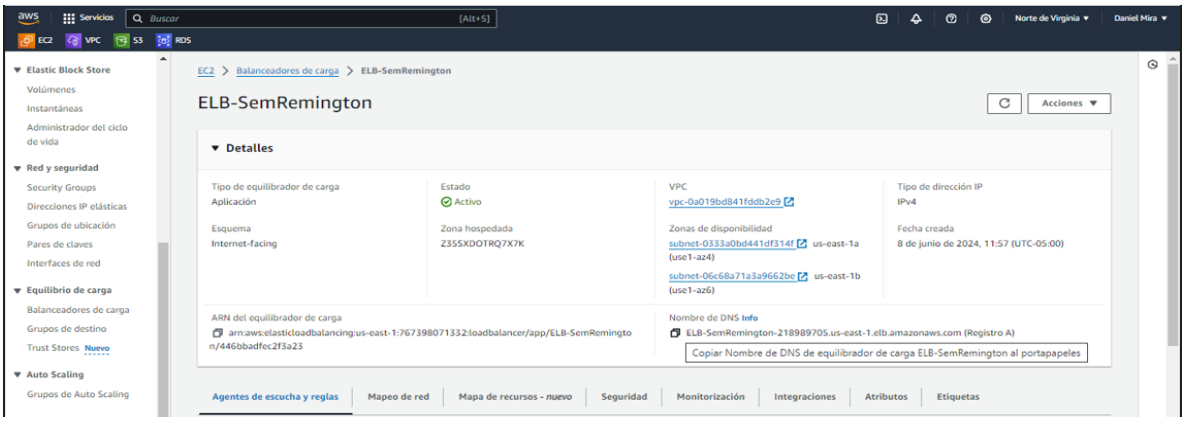
Figura 80 Balanceador con 2 instancias creadas



Fuente: Elaboración propia

Copiamos el nombre del DNS asociado al Balanceador de carga, el cual nos va a permitir la conexión a las instancias desde cualquier equipo conectado a Internet.

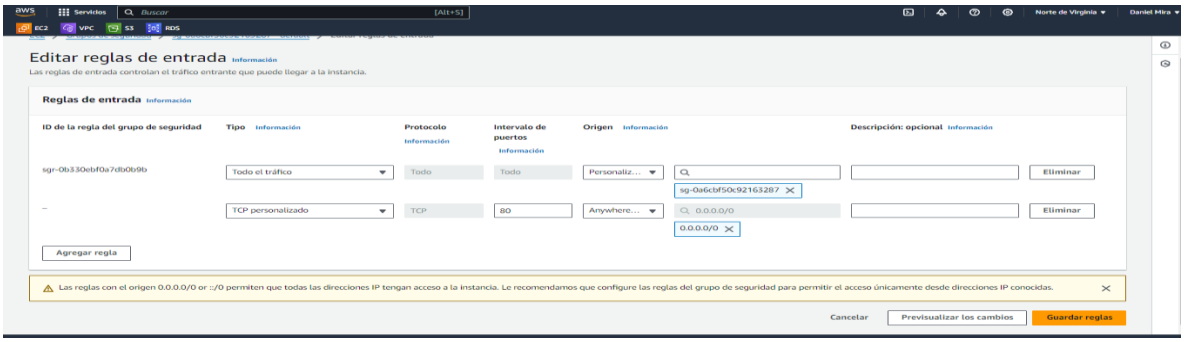
Figura 81 DNS



Fuente: Elaboración propia

Para que nos permita la conexión debemos agregar en las reglas de entrada del grupo de seguridad asociado al Balanceador de carga el puerto 80, esto para permitir la conexión desde cualquier equipo conectado a Internet.

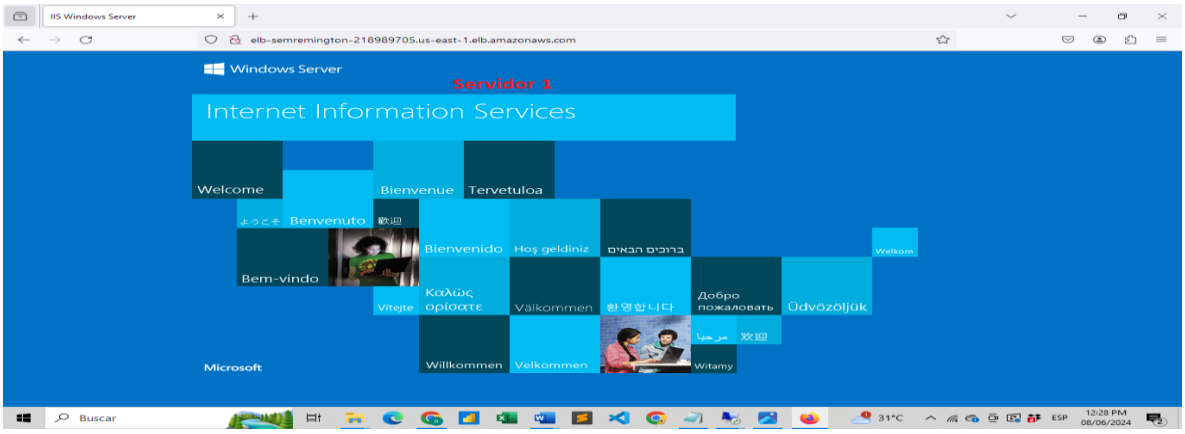
Figura 82Asignación reglas



Fuente: Elaboración propia

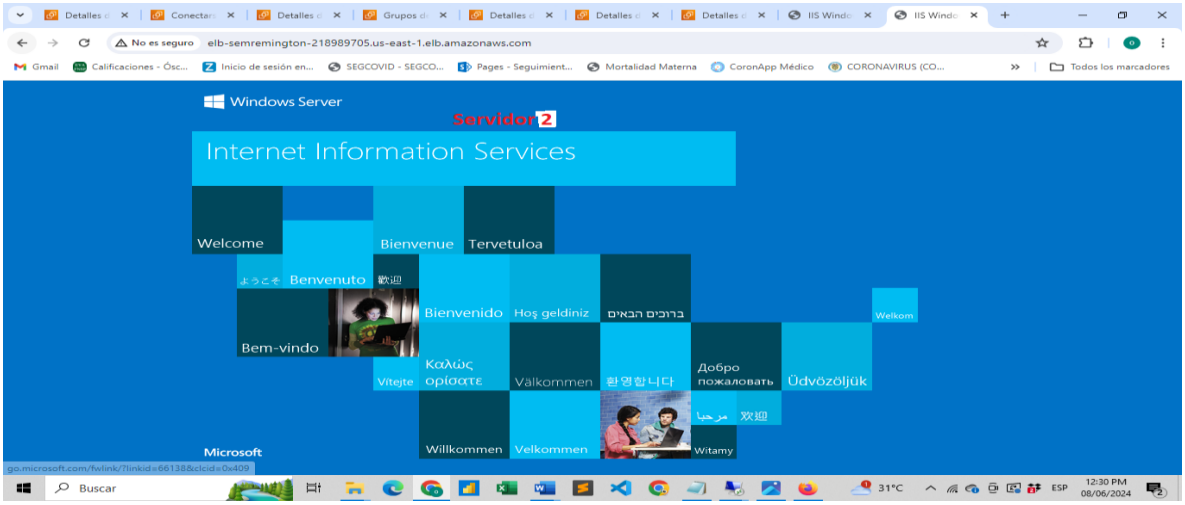
Luego verificamos que a través de la dirección del balanceador de carga podamos ingresar de manera local y nos responda unas veces el Servidor1 y otras el Servidor2.

Figura 83 Prueba de conexión local Servidor1



Fuente: Elaboración propia

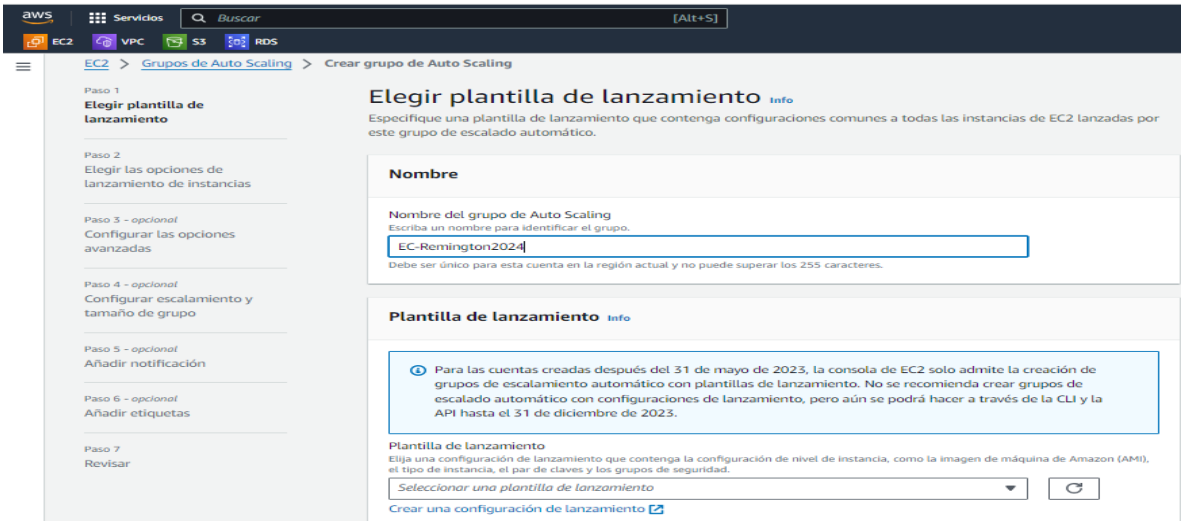
Figura 84 Prueba de conexión local Servidor2



Fuente: Elaboración propia

Creación del grupo de Auto Scaling para este caso lo llamaremos “EC-Remington2024”.

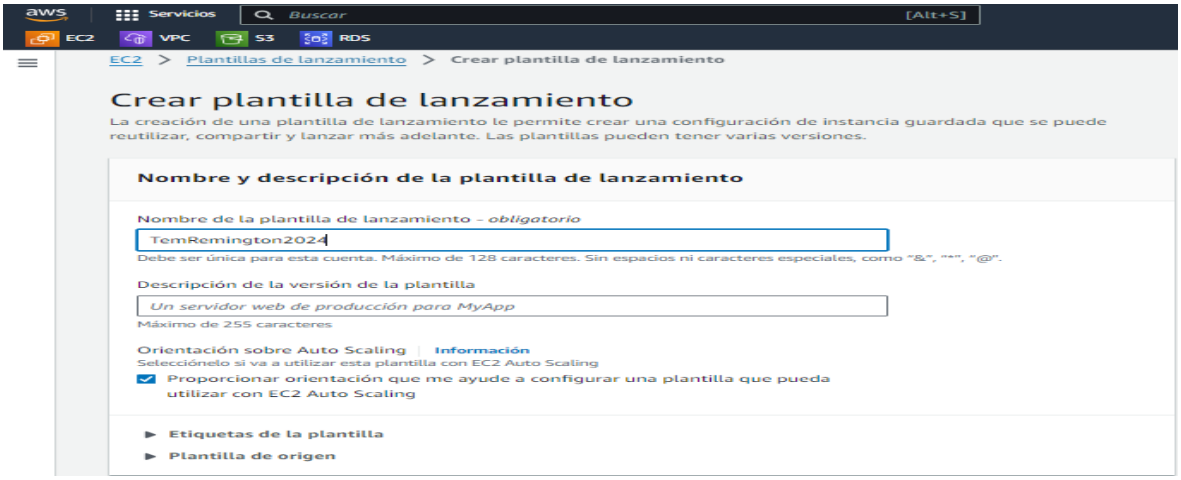
Figura 85 Creación grupo Auto Scaling



Fuente: Elaboración propia

Debemos crear una plantilla de lanzamiento, la cual nombraremos “TemRemington2024”.

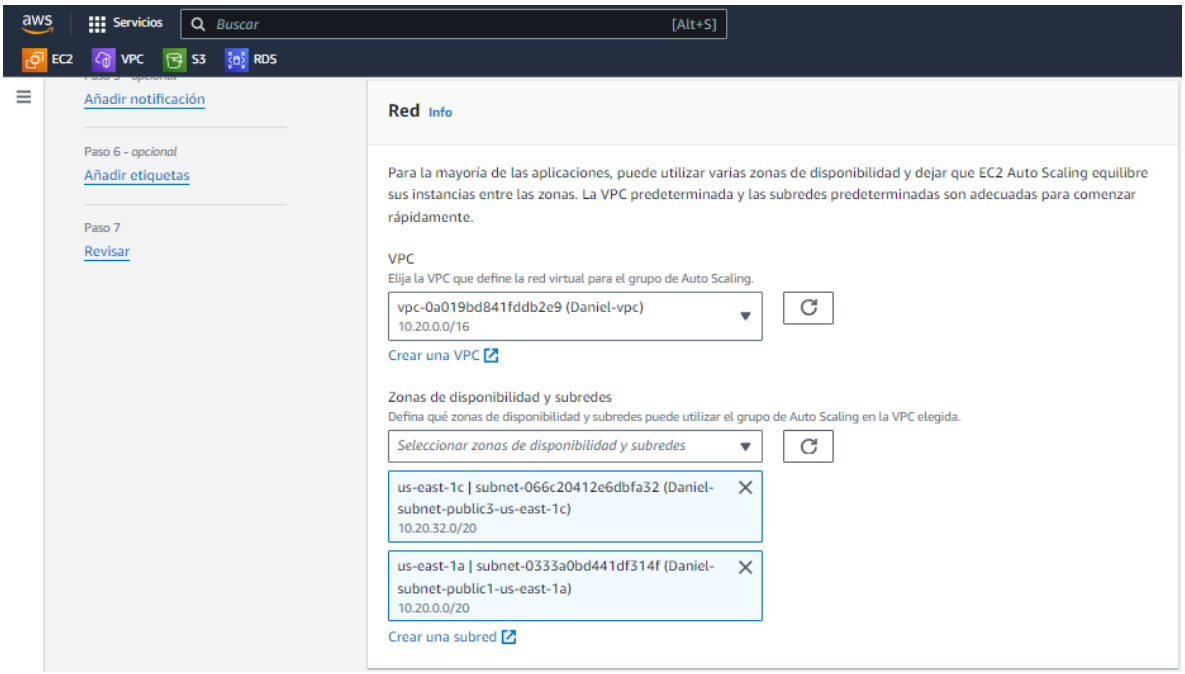
Figura 86 Creación plantilla de lanzamiento



Fuente: Elaboración propia

En la configuración de red debemos escoger la VPC y las subredes donde se encuentran las 2 Instancias creadas Servidor1 y Server2.

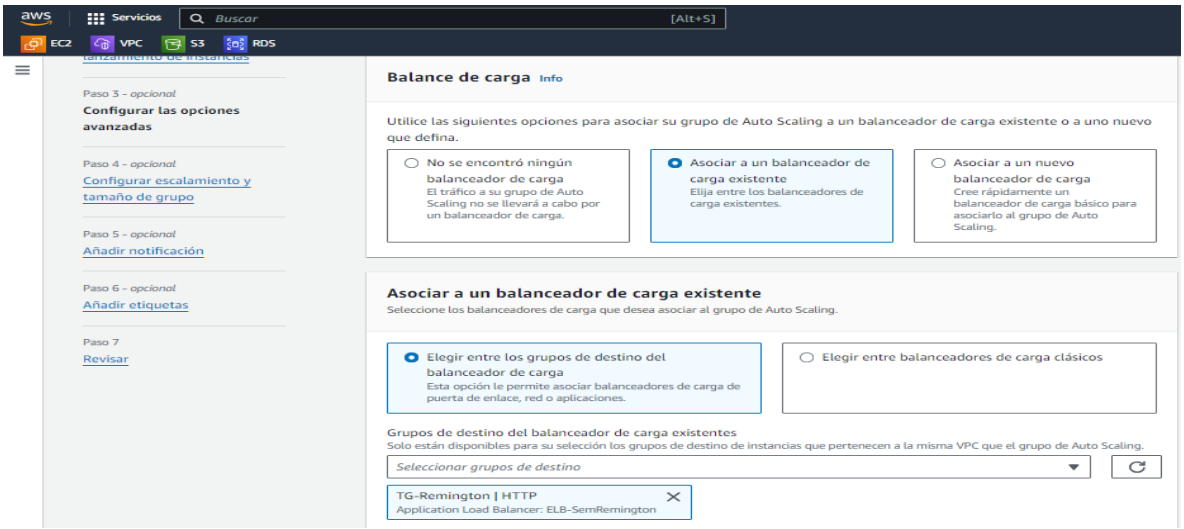
Figura 87 Seleccionar VPC y subredes



Fuente: Elaboración propia

Escogemos la opción Asociar a un balanceador de carga existente y asociamos en Target Group creado “TG-Remington”.

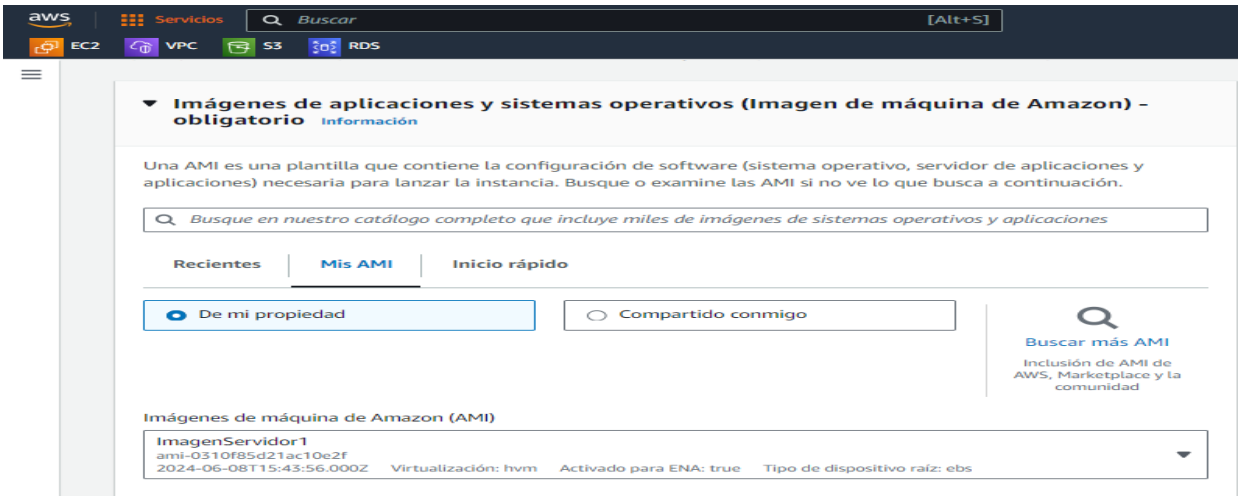
Figura 88 Asociar balanceador y TG



Fuente: Elaboración propia

Luego indicamos que la imagen que se usará para crear de forma automática las nuevas instancias será “ImagenServidor1”, la cual contiene una copia exacta de toda la configuración y los archivos de la Instancia “Servidor1”.

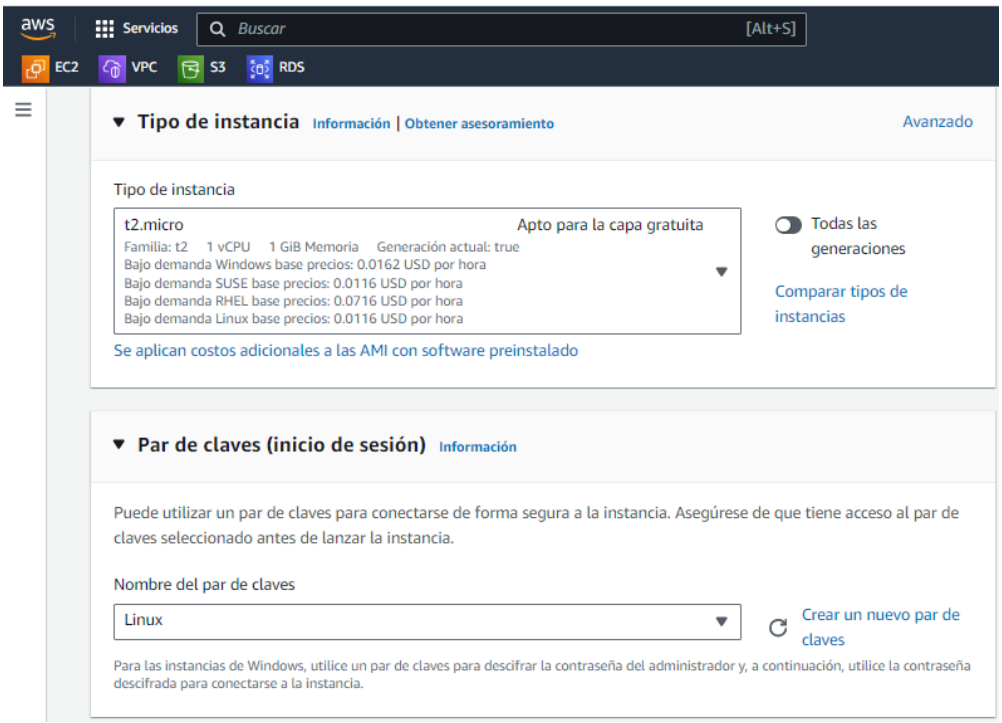
Figura 89 Selección imagen a usar



Fuente: Elaboración propia

Escogemos el tipo de CPU asignada a las nuevas instancias y la clave, la cual será la misma que hemos estado usando “Linux.pem”.

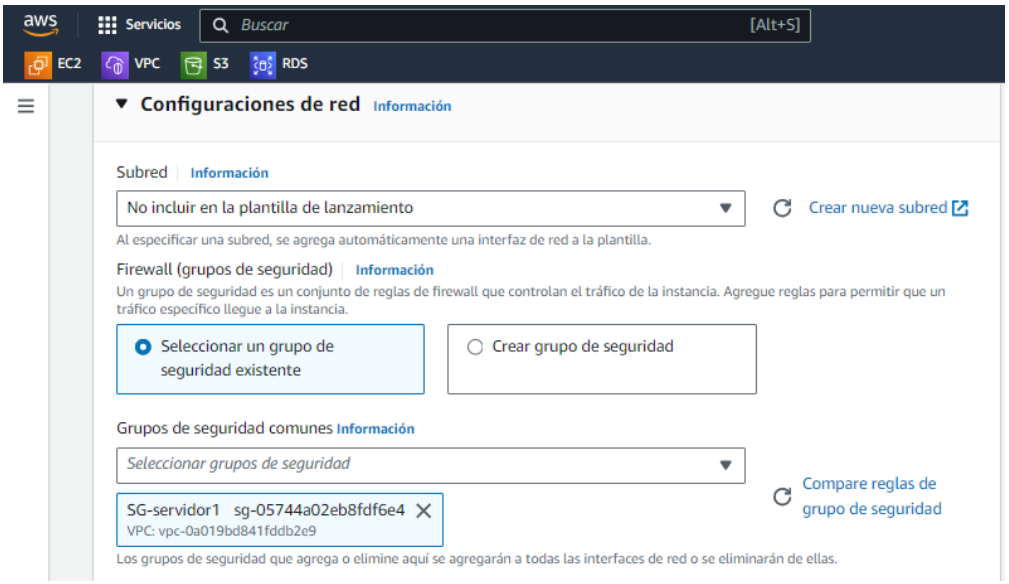
Figura 90 Selección tipo CPU y clave



Fuente: Elaboración propia

A continuación, seleccionamos un grupo de seguridad existente “SG-servidor1”, el cual es el mismo asociado a la Instancia Servidor1.

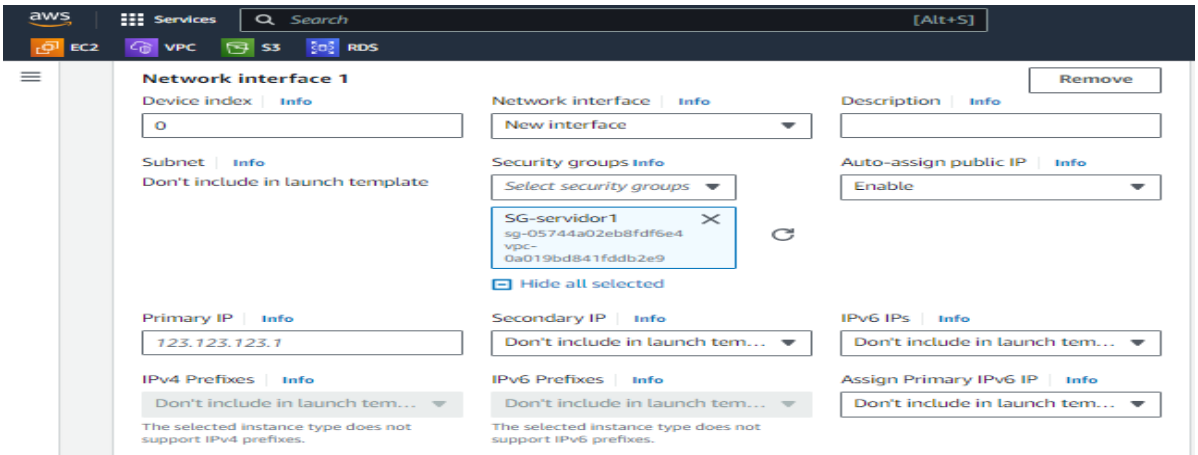
Figura 91 Selección grupo de seguridad



Fuente: Elaboración propia

Igualmente, en la configuración de red escogemos el grupo de seguridad “SG-servidor1” y habilitamos la asignación de una Ip pública para que las instancias creadas por el Auto Scaling permitan el acceso por escritorio remoto.

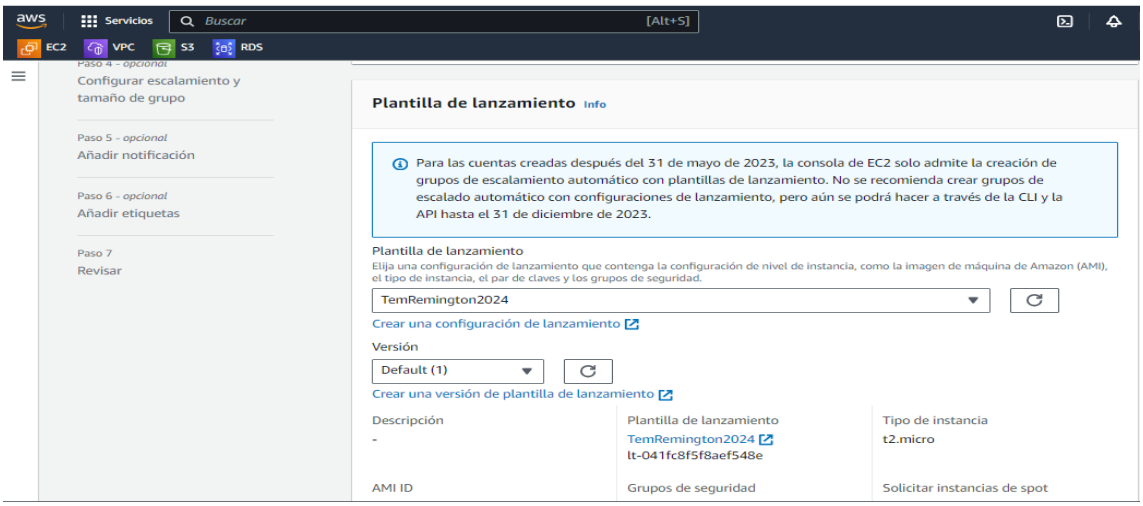
Figura 92 Asignación de IP para conexión remota



Fuente: Elaboración propia

Volvemos a la creación del grupo Auto Scaling y escogemos la plantilla de lanzamiento "TemRemington2024" que acabamos de crear.

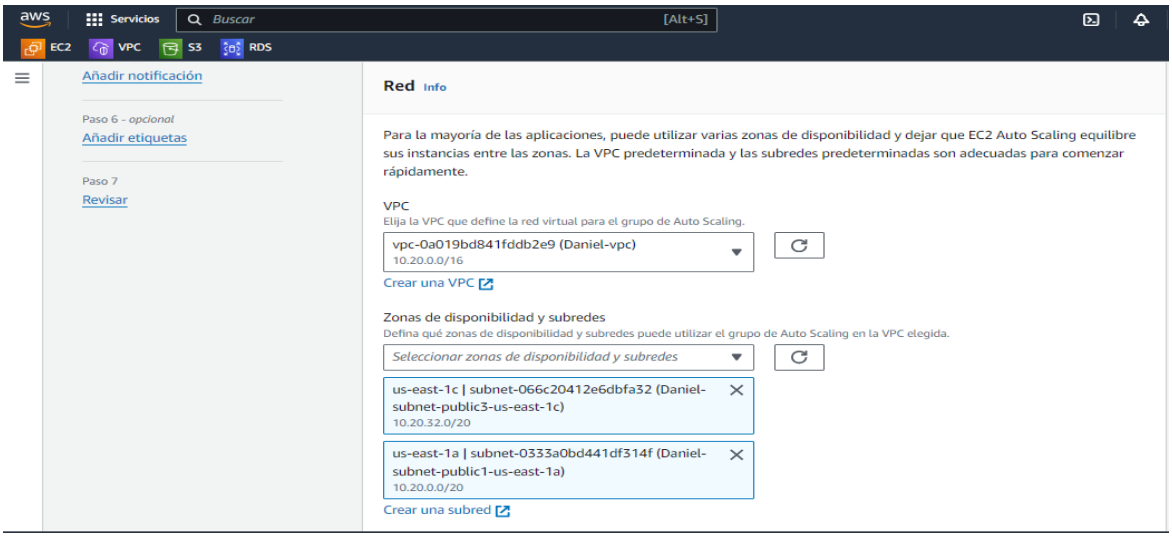
Figura 93 Selección plantilla de lanzamiento



Fuente: Elaboración propia

En la configuración de red escogemos el VPC que estamos usando “Daniel-vpc” y las subredes donde se encuentran las Instancias creadas.

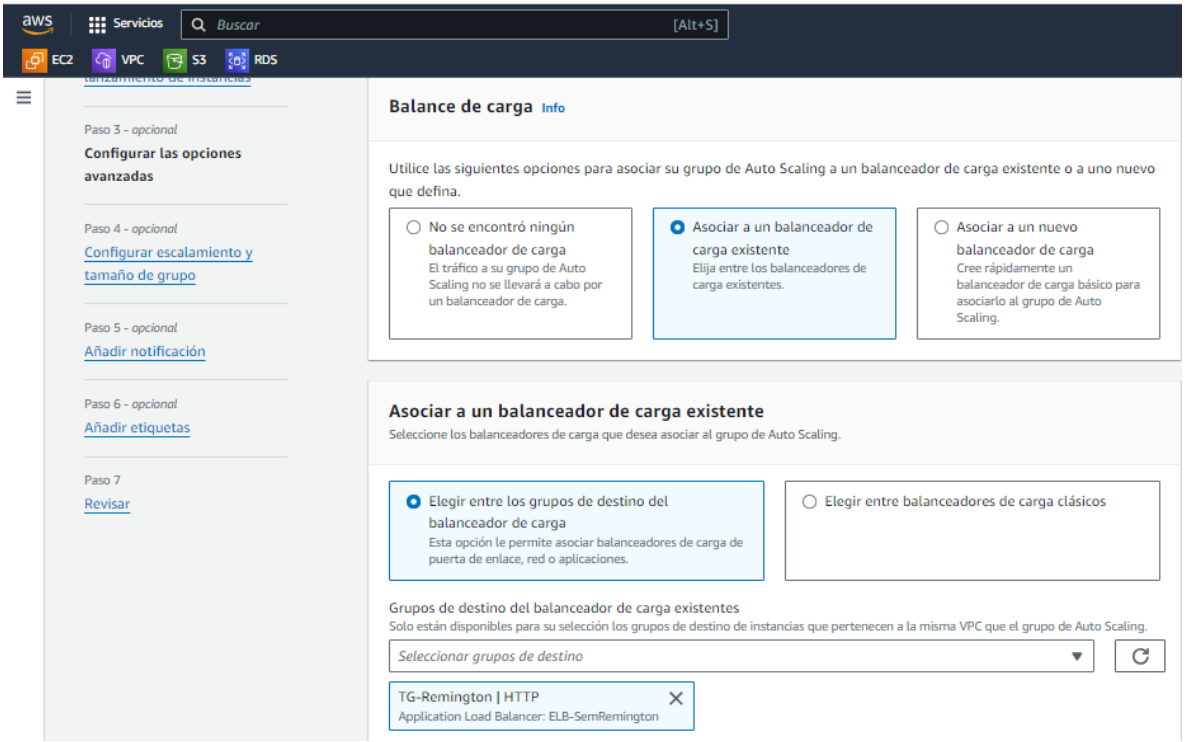
Figura 94 Selección VPC



Fuente: Elaboración propia

Luego escogemos la opción Asociar a un balanceador de carga existente y escogemos el Target Group asociado al balanceador de carga.

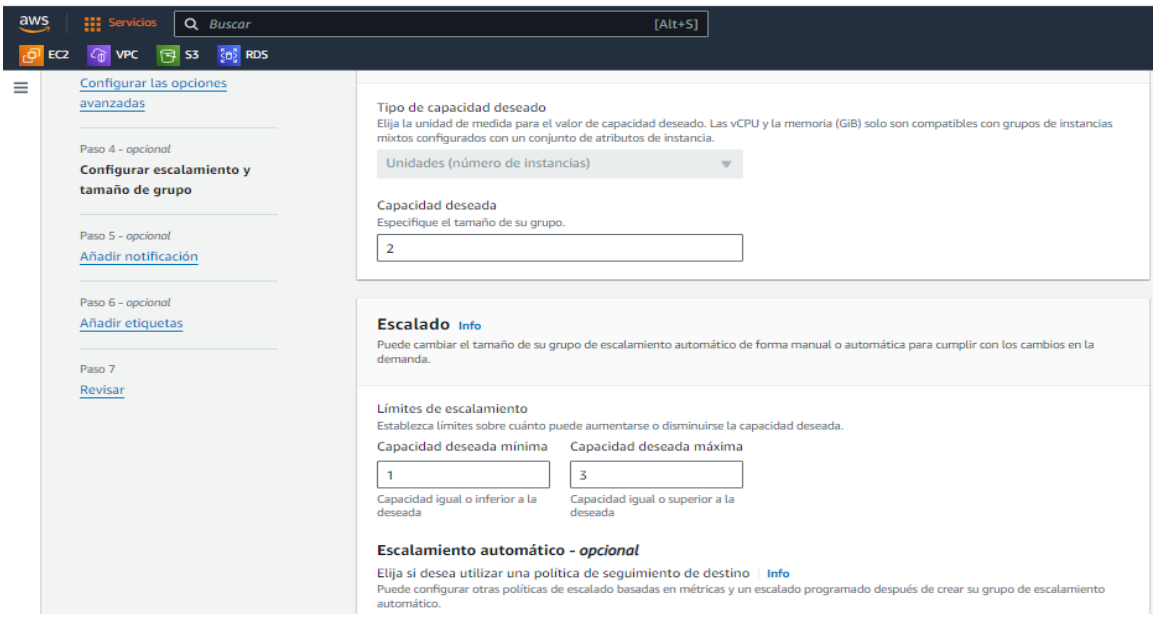
Figura 95 Asociar balanceador y TG



Fuente: Elaboración propia

Indicamos la capacidad deseada, la capacidad deseada mínima y la capacidad deseada máxima. Estos valores los tendrá en cuenta el Auto Scaling para el lanzamiento automático de las Instancias.

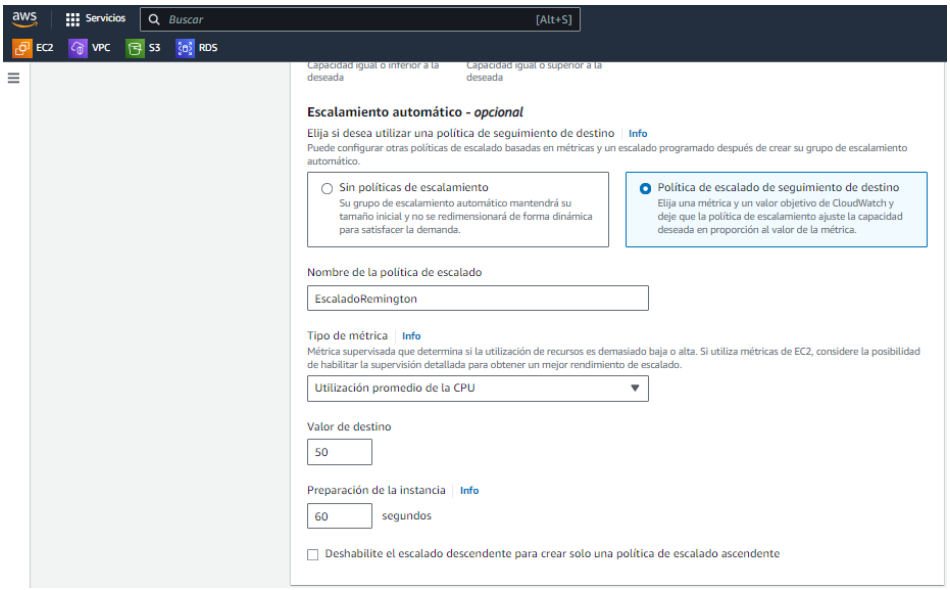
Figura 96 Selección de capacidades en auto scaling



Fuente: Elaboración propia

En el escalamiento automático, escogemos la opción de política de escalado de seguimiento de destino, le ponemos un nombre y le establecemos que cuando el porcentaje de uso de la CPU en las instancias creadas sea superior a 50% cree una nueva instancia y cuando sea inferior a 50% elimine una o varias instancias teniendo en cuenta el valor máximo que establecimos.

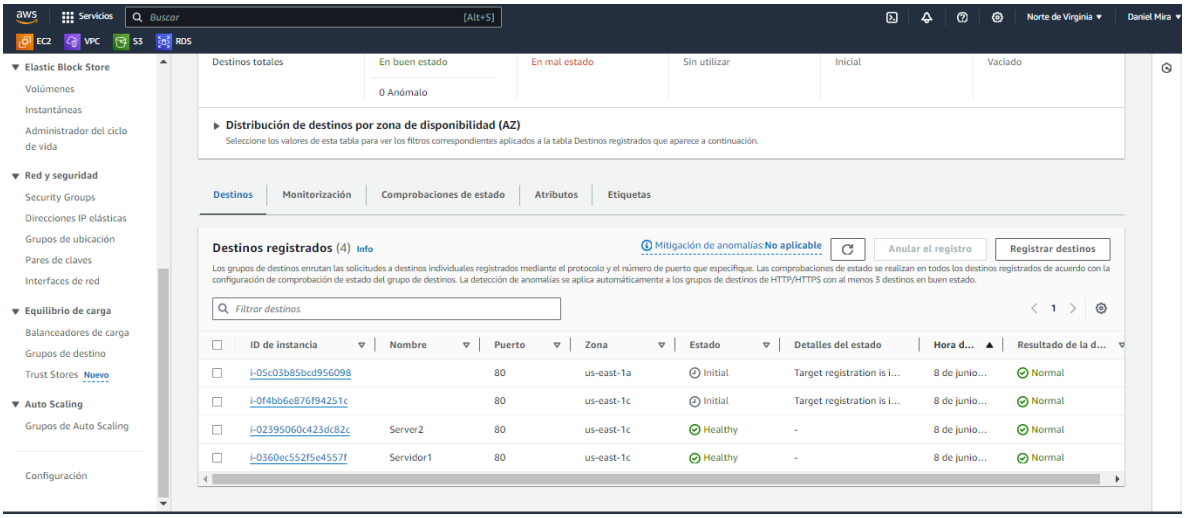
Figura 97 Configuración política de escalado



Fuente: Elaboración propia

En la siguiente imagen vemos que se están iniciando 2 Instancias adicionales a las que fueron creadas de forma manual.

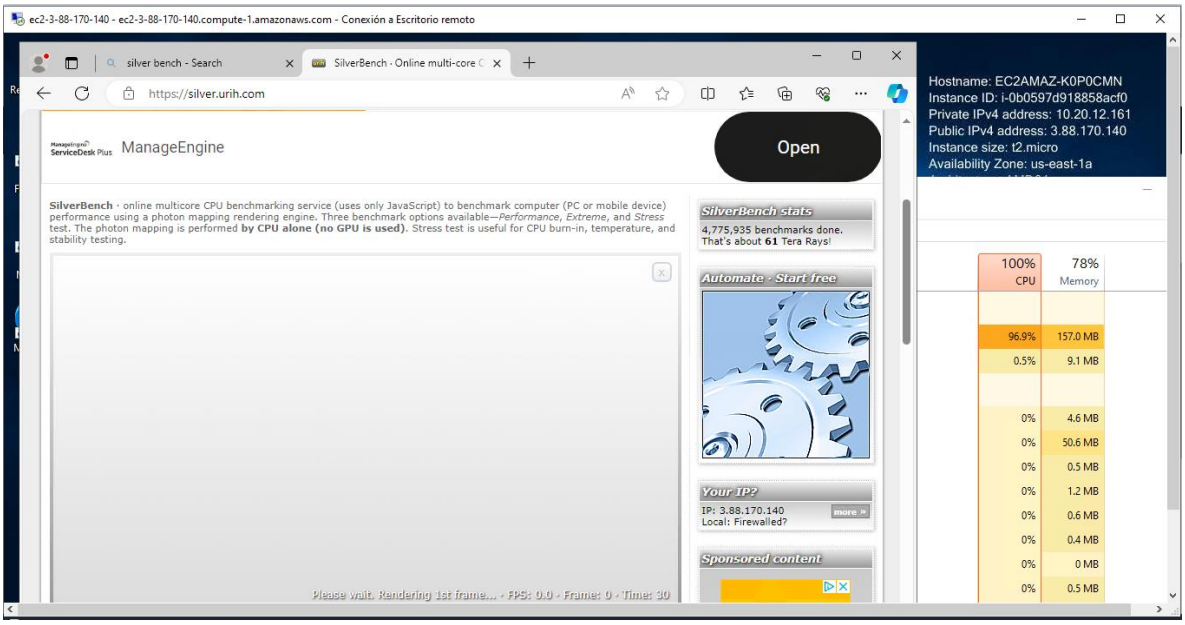
Figura 98 Resumen de instancias



Fuente: Elaboración propia

Para someter a estrés la CPU de una de las instancias creada automáticamente por el Auto Scaling, nos conectamos a través del escritorio remoto y abrimos la página de Silver Bench, la cual como se puede ver en la imagen aumenta el uso de CPU a 100%.

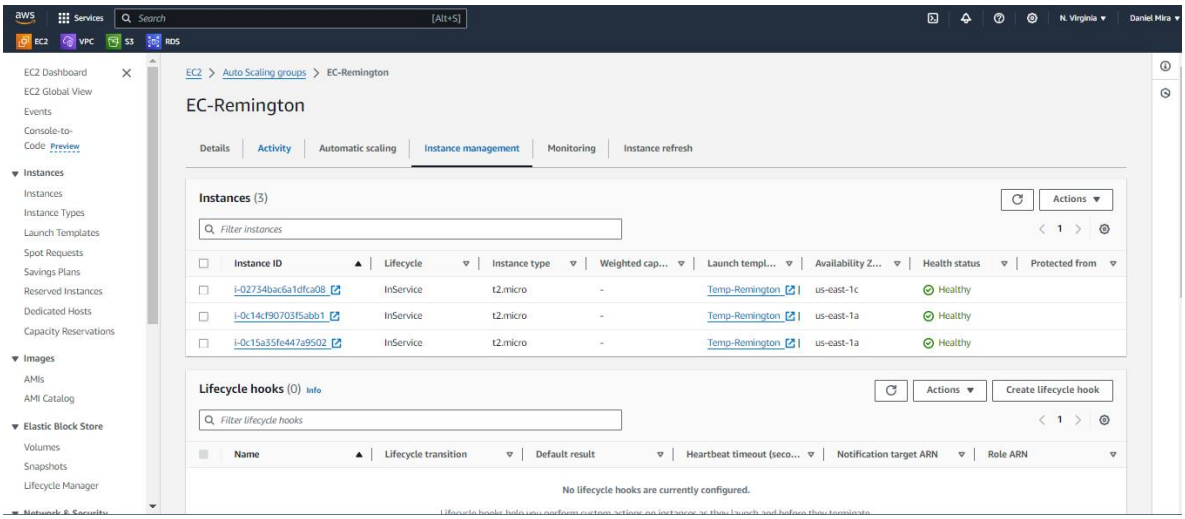
Figura 99 Uso Silver Bench para estresar CPU



Fuente: Elaboración propia

En la siguiente imagen vemos 3 Instancias funcionando, ya que al someter a una de ellas a estrés de la CPU alcanzando el 100% el grupo de Auto Scaling lanza automaticamente una instancia más.

Figura 100 Resumen de instancias en modo estrés

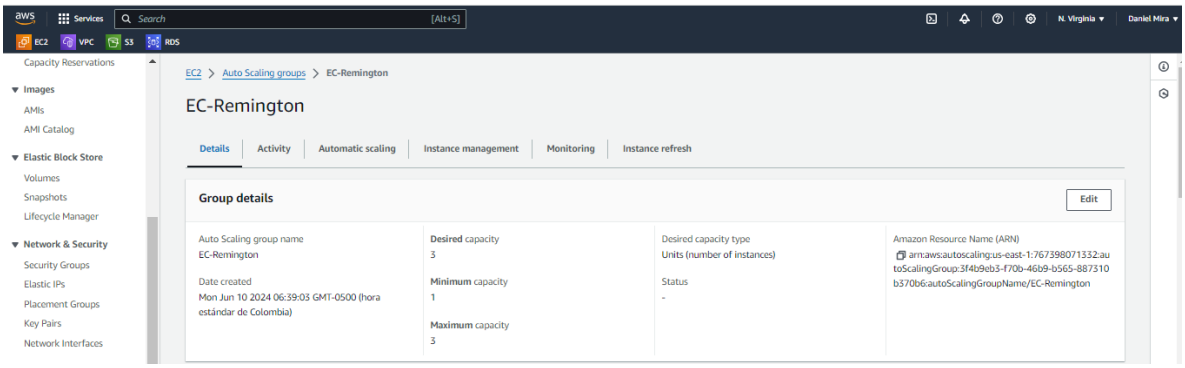


Fuente: Elaboración propia

Se pudo evidenciar que el escalado automático cambia los valores deseados de instancia y elimina o crea las instancias dependiendo del uso de CPU.

Para la imagen siguiente la capacidad deseada es 3 cuando habíamos establecido 2.

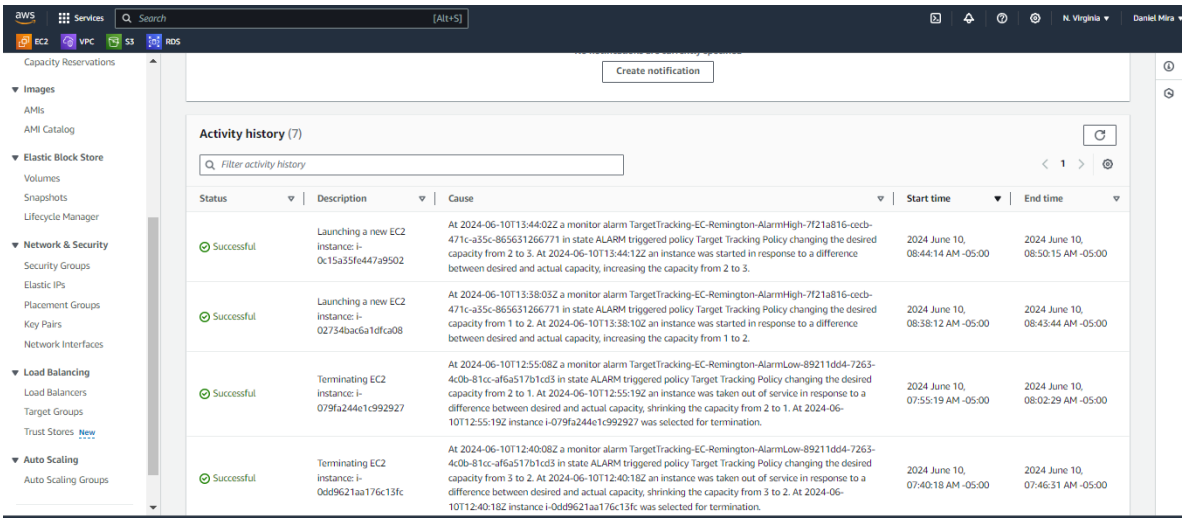
Figura 101 Valores escalado automático



Fuente: Elaboración propia

En la siguiente imagen podemos evidenciar a través de la actividad, que las instancias se han eliminado o creado dependiendo del uso de CPU, como lo muestra en “AlarmHigh” y “AlarmLow”, lo que demuestra que la política de escalado automático funciona.

Figura 102 Estado de funcionamiento escalado automático



Fuente: Elaboración propia

Conclusiones

En conclusión, este trabajo examina la relevancia de Amazon Web Services (AWS) como líder mundial en servicios en la nube, destacando su impacto en el desarrollo tecnológico empresarial. Se exploraron los orígenes de la virtualización y la computación en la nube, proporcionando una línea de tiempo de los hitos clave y una comparación de los servicios principales ofrecidos por AWS, Microsoft Azure y Google Cloud Platform (GCP), subrayando cómo estas plataformas ayudan a las empresas a optimizar recursos, asegurar información y garantizar la disponibilidad de servicios informáticos. También tuvimos la oportunidad de interactuar directamente con la consola de AWS a través de la creación de una cuenta para poder explorar y manipular las múltiples opciones de configuración que nos brinda esta plataforma.

Referencias

Amazon (2024). AWS Identity and Access Management. https://docs.aws.amazon.com/es_es/IAM/latest/UserGuide/introduction.html

Amazon (2024). Amazon Simple Storage Service. https://docs.aws.amazon.com/es_es/AmazonS3/latest/userguide/Welcome.html

Amazon (2024). Amazon EBS. https://docs.aws.amazon.com/es_es/ebs/latest/userguide/what-is-ebs.html

Amazon (2024). Amazon Relational Database Service. https://docs.aws.amazon.com/es_es/AmazonRDS/latest/UserGuide/Welcome.html

Amazon (2024). Amazon Elastic Compute Cloud. https://docs.aws.amazon.com/es_es/AWSEC2/latest/WindowsGuide/concepts.html

Amazon (2023). ¿Qué es Docker?. https://aws.amazon.com/es/docker/#:~:text=Docker%20es%20una%20plataforma%20de,probar%20e%20implementar%20aplicaciones%20r%C3%A1pidamente&text=a%20usar%20Docker-,Docker%20es%20una%20plataforma%20de%20software%20que%20le%20permite,probar%20e%20implementar%20aplicaciones%20r%C3%A1pidamente.

Amazon (2024). Amazon Elastic Compute Cloud. https://docs.aws.amazon.com/es_es/AWSEC2/latest/WindowsGuide/concepts.html#:~:text=Amazon%20Elastic%20Compute%20Cloud%20(Amazon,implementar%20aplicaciones%20con%20mayor%20rapidez.

Amazon (2024). Amazon Virtual Private Cloud. https://docs.aws.amazon.com/es_es/vpc/latest/userguide/what-is-amazon-vpc.html

Ardila, Laura (2019). ¿Cómo ha evolucionado la computación en la nube a través del tiempo?. https://www.pragma.co/es/blog/como-ha-evolucionado-la-computacion-en-la-nube-a-traves-del-tiempo#:~:text=Hist%C3%B3ricamente%2C%20el%20concepto%20de%20computaci%C3%B3n,la%20informaci%C3%B3n%20desde%20cualquier%20lugar.

Deloitte (2024). Introducción a Amazon Web Services. https://www2.deloitte.com/es/es/pages/technology/articles/introduccion-a-amazon-web-services.html

Fiz, Juan María (2019). AWS vs Azure vs GCP: todos los servicios cloud frente a frente. https://www.paradigmadigital.com/dev/comparativa-servicios-cloud-aws-azure-gcp/

hpe (2024). Virtualización. https://www.hpe.com/lamerica/es/what-is/virtualization.html

Microsoft (2023). ¿Qué es el Subsistema de Windows para Linux?. https://learn.microsoft.com/es-es/windows/wsl/about

Red Hat (2023). Las máquinas virtuales. https://www.redhat.com/es/topics/virtualization/what-is-a-virtual-machine