



TRABAJO DE GRADO
Opción Seminario-Diplomado.

CREACIÓN DE SERVICIOS AWS PARA
EMPRESA FICTICIA VIRTUALGOURMET

Corporación Universitaria Remington.
Ingeniería de sistemas
SEMINARIO AMAZON AWS

Diego Fernando Peña Salinas
Seminario de grado
2024.

Tabla de Contenidos

Resumen.....	3
Marco conceptual y contextual	4
Desarrollo e implementación del aprendizaje.....	4
Entrega 1	
Figuras y tablas	7
Entrega	
Final.....	20
Figuras y Graficos.....	20
Conclusiones	226
Referencias.....	30

Resumen

En el presente proyecto se maneja la creación de instancias, que son las máquinas virtuales que podemos crear y manejar dentro de nuestro servicio de AWS, las instancias son como computadores servidores que tenemos en la nube, y donde se puede hacer instalación de programas para poder manejar bases de datos o páginas web.

Las instancias tienen un nombre específico, se denominan AMI, Imagen de Máquina de Amazon, y en estas AMI podemos usar el software que necesitamos para nuestro negocio o proyecto.

Una instancia en AWS EC2, es la máquina virtual de Amazon por sus siglas Elastic Compute Cloud, que da al usuario los recursos de un equipo físico, de forma virtual.

También vemos la creación de Balanceadores de Carga, que son aquellas reglas de tráfico, que simplemente redirigen las peticiones de clientes hacia nuestras instancias, en caso de que alguna de ellas falle.

Marco conceptual y contextual

La tecnología ha permitido crear cada vez mas productos de los cuales la gran mayoría ya se manejan de manera virtual o en la nube, los servicios de Amazon Web, es uno de ellos, y el siguiente proyecto vamos a hablar de ello.

¿Qué es Amazon Web Services? AWS provee servicios en la nube que permite tener un almacenamiento virtual de recursos de computadores, aplicaciones para móviles y bases de datos. Todo esto basado en la arquitectura en la Nube.

Sus principales ventajas esta en la seguridad para procesar los datos de manera certificada y sin costo (una parte de sus servicios, lo otros tienen un valor relativo).

El manejo de bases de datos para los negocios, se pueden tener en un rápido acceso y confiable, para su manejo y almacenamiento.

Desarrollo e implementación del aprendizaje

Entrega 1

1. Recursos de AWS disponibles para la ejecución de instancias.

Amazon Web Services (AWS) pone a nuestra disposición diversos recursos que nos permiten llevar a cabo la ejecución de máquinas virtuales, conocidas como instancias. Estas instancias son empleadas para diversas actividades, tales como hospedar aplicaciones, procesar información o llevar a cabo simulaciones.

2. Almacenamiento para las instancias.

Una máquina requiere de almacenamiento para conservar la información.

AWS ofrece diversas alternativas. EBS, también conocido como Elastic Block Store, es un tipo de disco duro que se puede conectar fácilmente a tu instancia en la nube.

Lo positivo es que al apagar la máquina, los datos permanecen guardados. El almacenamiento de instancia es una opción rápida que está incluida en la máquina, pero ten en cuenta que los datos se eliminarán si la apagas.

S3 (Simple Storage Service): Se utiliza para almacenar grandes volúmenes de información, como archivos o copias de respaldo, sin embargo, no se trata de un disco duro como EBS.

3. Conexiones y redes.

Para que las máquinas puedan comunicarse entre sí o acceder a internet, es necesario establecer una red.

VPC (Virtual Private Cloud): Se asemeja a una red privada personalizada para tus dispositivos.

Los Grupos de Seguridad son similares a las reglas de un cortafuegos que definen las conexiones permitidas, como quién tiene acceso dentro o fuera de la máquina.

Dirección IP estática: En caso de que desees que tu dispositivo cuente con una dirección IP fija para acceder desde cualquier ubicación.

4. Amazon EC2, también conocido como Elastic Compute Cloud.

Este servicio principal nos permite lanzar y gestionar instancias. En otras palabras, EC2 sería como arrendar una computadora en la nube.

Existen diversas variedades de instancias en AWS que se adaptan a tus necesidades específicas, ya sea que requieras mayor velocidad, mayor capacidad de memoria o costos más accesibles.

A modo de ilustración:

El t2.micro es una opción sencilla y económica.

m5.Large: Una máquina con mayor potencia.

AMIs (Amazon Machine Images): Representan una especie de réplica o modelo que abarca tanto el sistema operativo como el software preciso para tu instancia, Windows o Linux.

Figuras y tablas.

Figura 1. Creamos la VPC para la entrega 1, la llamamos **Entrega1-DiegoPena-VPC**.

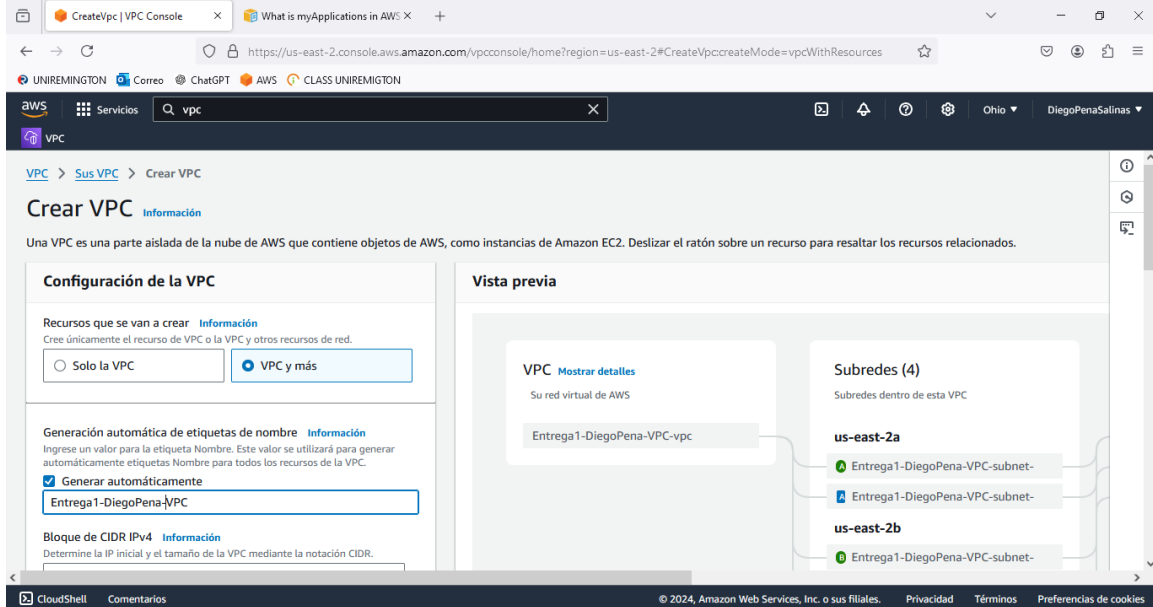


Figura2. Aplicamos la configuración para la VPC de subredes privadas y públicas, en este caso creamos una subred publica y una privada.

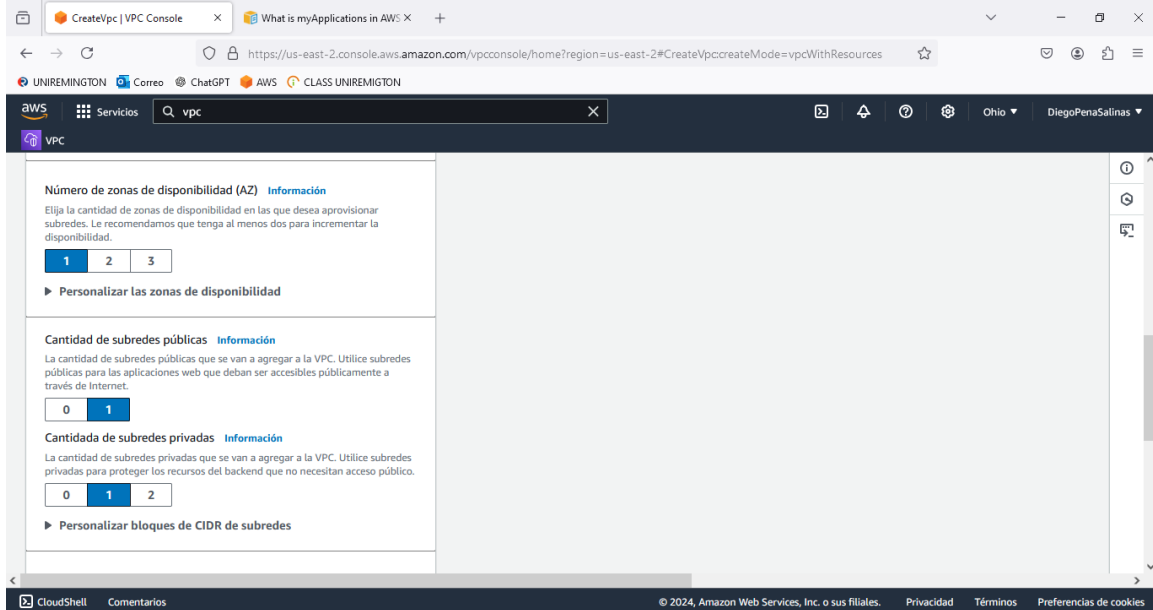


Figura 3. Los demás valores los dejamos por defecto y creamos la instancia.

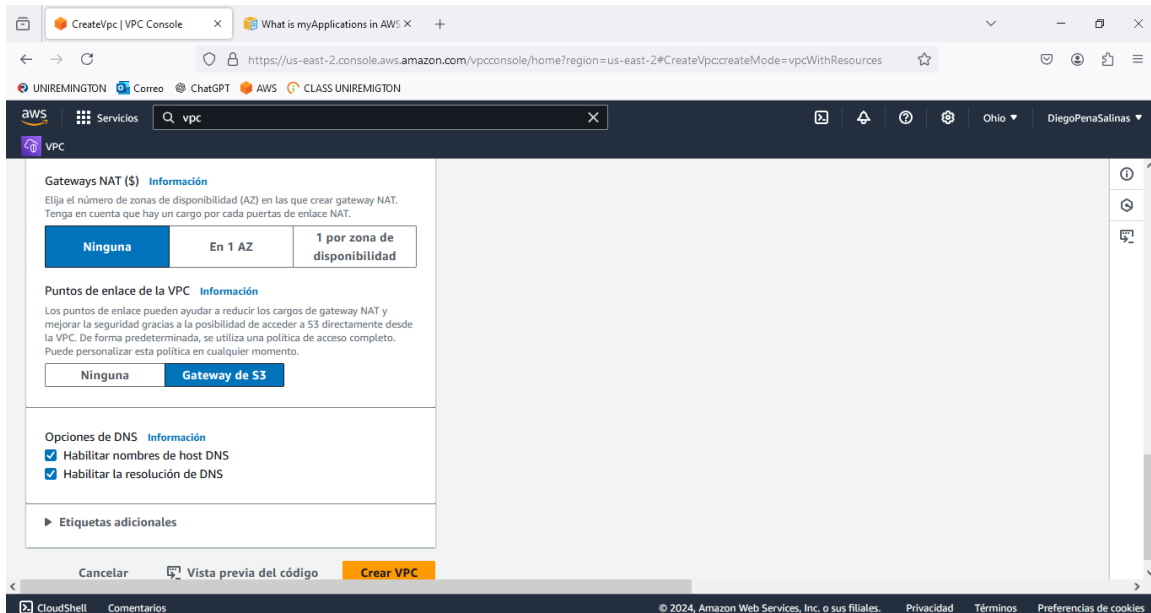


Figura 4. Una vez creada la instancia VPC, nos arroja un resumen de la acción, con todos los valores que le dimos en la configuración de la VPC.

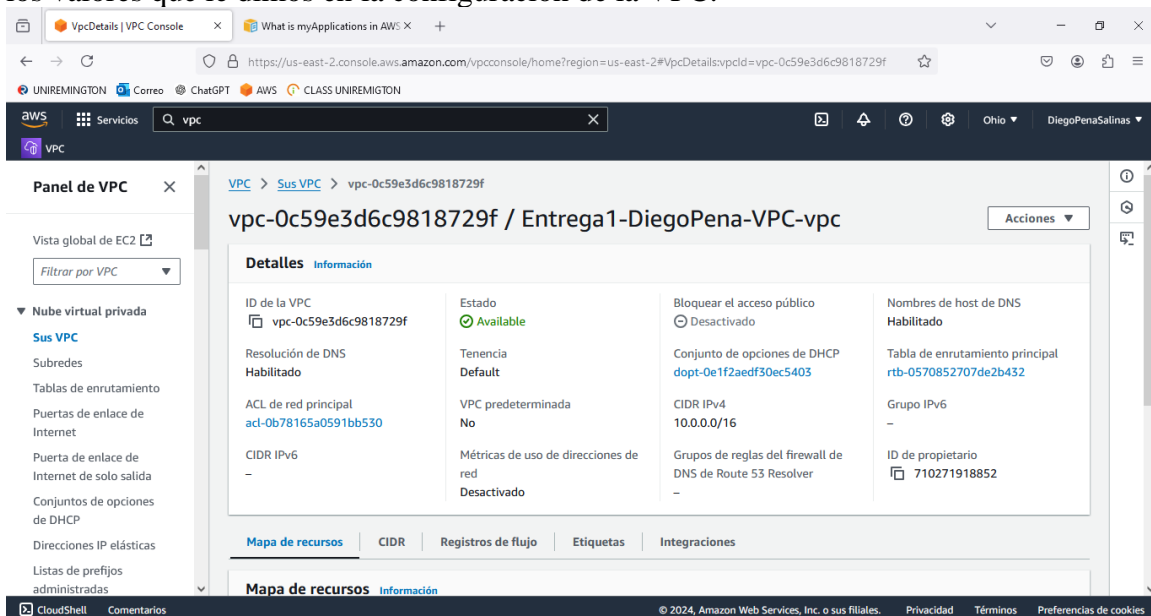


Figura 5. Creamos la instancia o la máquina virtual con Linux.

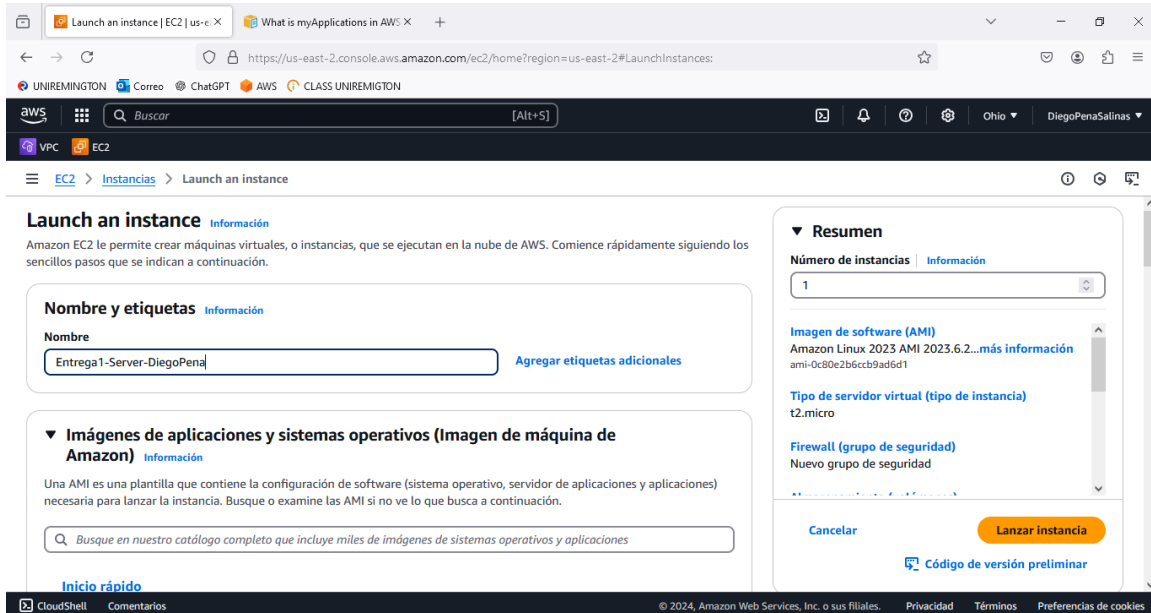


Figura 6. Seleccionamos el sistema Operativo que vamos a utilizar en nuestro servidor, seleccionamos Linux para AWS.

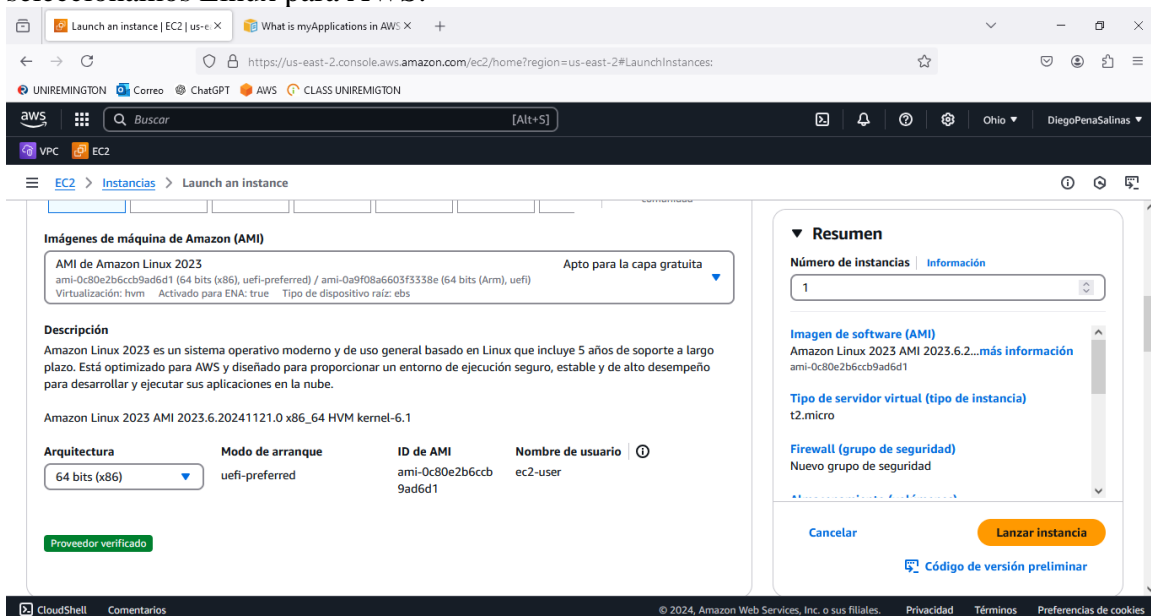


Figura 7. Seleccionamos las características de Hardware que debe tener nuestra máquina como Procesadores, Disco y Memoria RAM.

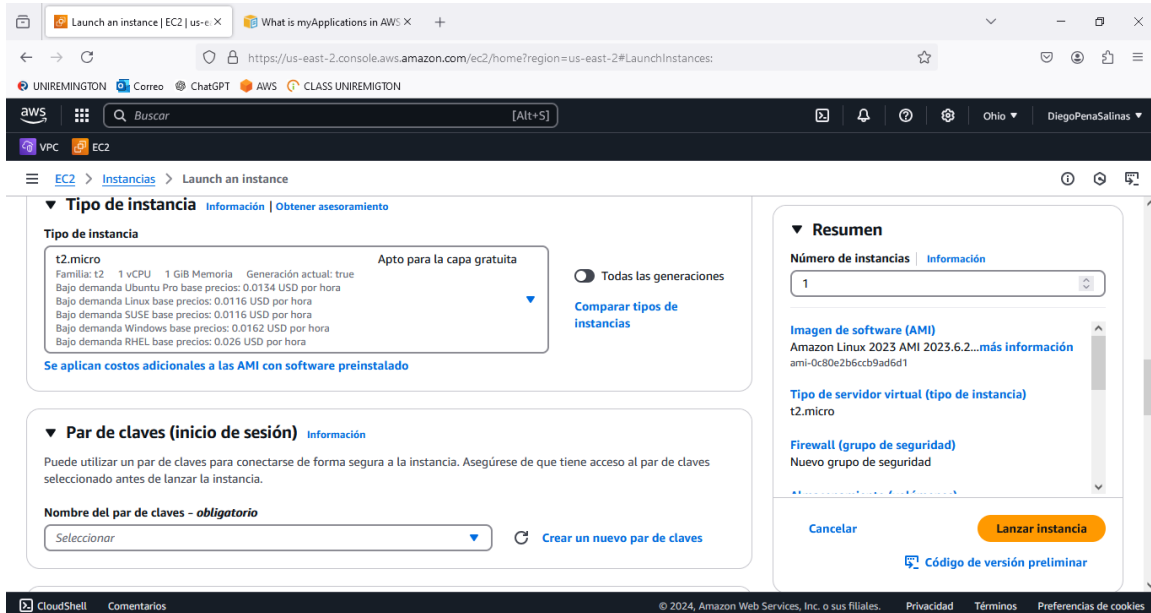


Figura 8. Creamos el Pair Key, que son las claves para el acceso a nuestro servidor de manera remota.

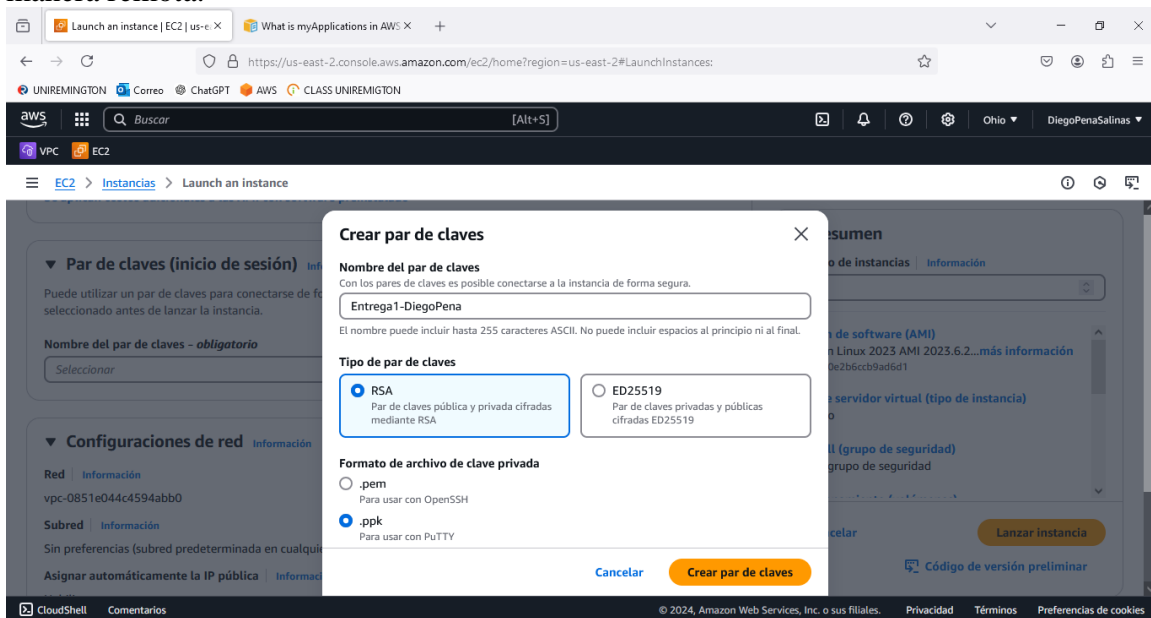


Figura 9. Creamos nuestras Reglas de entrada y salida para las conexiones desde y hacia nuestro servidor. Configuramos puerto 22 de salida y 80 de entrada. También configuramos nuestra dirección IP Pública.

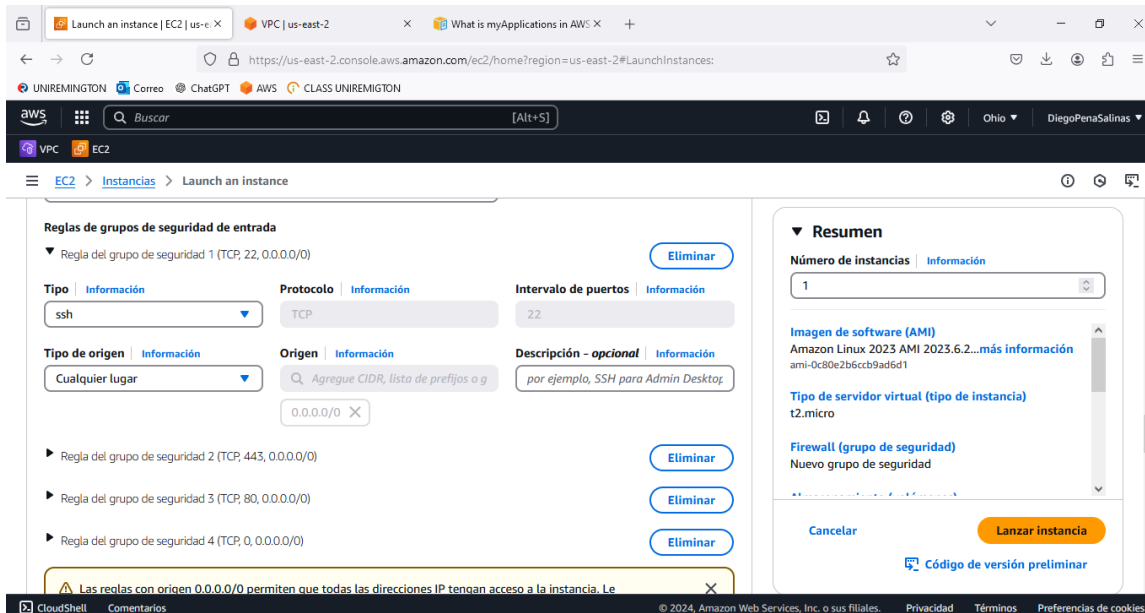


Figura 10. Creamos nuestra computadora en la nube.

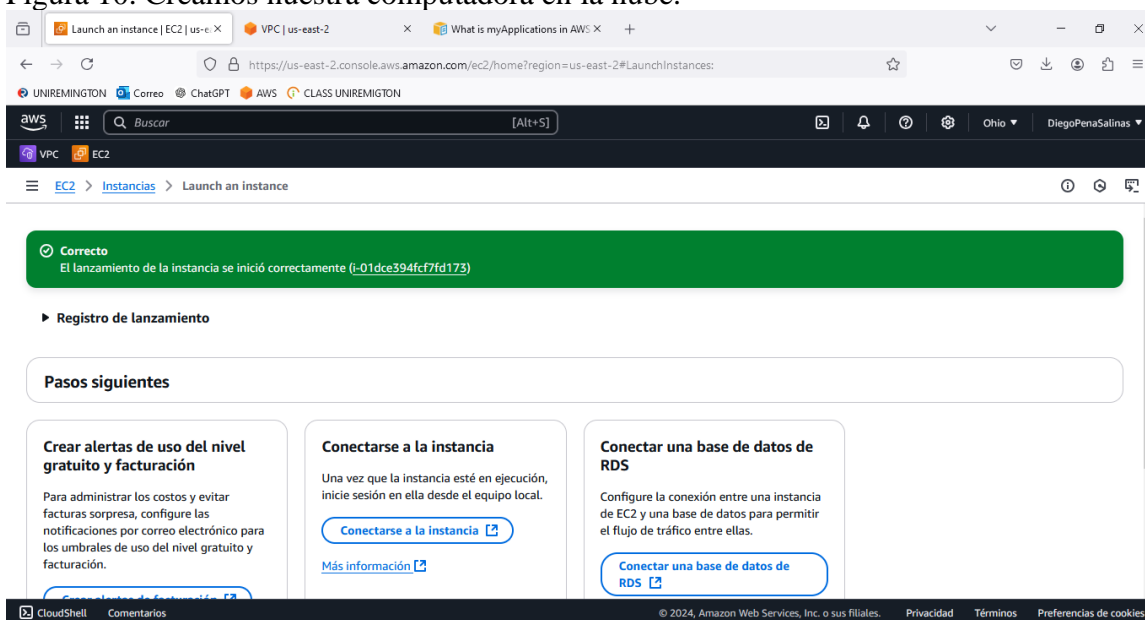


Figura 11. Nos muestra un resumen de nuestra computadora, configuración establecida y hardware dedicado; también el estado de funcionamiento, en este caso nuestra máquina se encuentra funcionando.

The screenshot displays the AWS Management Console interface for an EC2 instance. The left sidebar shows navigation options like 'Panel', 'Eventos', 'Instancias', and 'Imágenes'. The main content area is divided into several sections:

- ID de la instancia:** i-01dce394fcf7fd173
- Dirección IPv6:** -
- Tipo de nombre de anfitrión:** Nombre de IP: ip-10-0-4-141.us-east-2.compute.internal
- Responder al nombre DNS de recurso privado:** -
- Dirección IP asignada automáticamente:** 3.144.12.180 [IP pública]
- Rol de IAM:** -
- Dirección IPv4 pública:** 3.144.12.180 | dirección abierta
- Estado de la instancia:** En ejecución
- Nombre DNS de IP privada (solo IPv4):** ip-10-0-4-141.us-east-2.compute.internal
- Tipo de instancia:** t2.micro
- ID de VPC:** vpc-0c59e3d6c9818729f (Entrega1-DiegoPena-VPC-vpc)
- ID de subred:** -
- Direcciones IPv4 privadas:** 10.0.4.141
- DNS de IPv4 pública:** ec2-3-144-12-180.us-east-2.compute.amazonaws.com | dirección abierta
- Direcciones IP elásticas:** -
- Hallazgo de AWS Compute Optimizer:** Suscribirse a AWS Compute Optimizer para recibir recomendaciones. | Más información
- Nombre del grupo de Auto Scaling:** -

Figura12. Iniciamos nuestro programa de conexión a nuestro servidor. En este caso la herramienta para conexión será el programa PUTTY. Introducimos los datos de conexión que nos brinda el sistema AWS EC2, en el momento de crear nuestro computador en la nube.

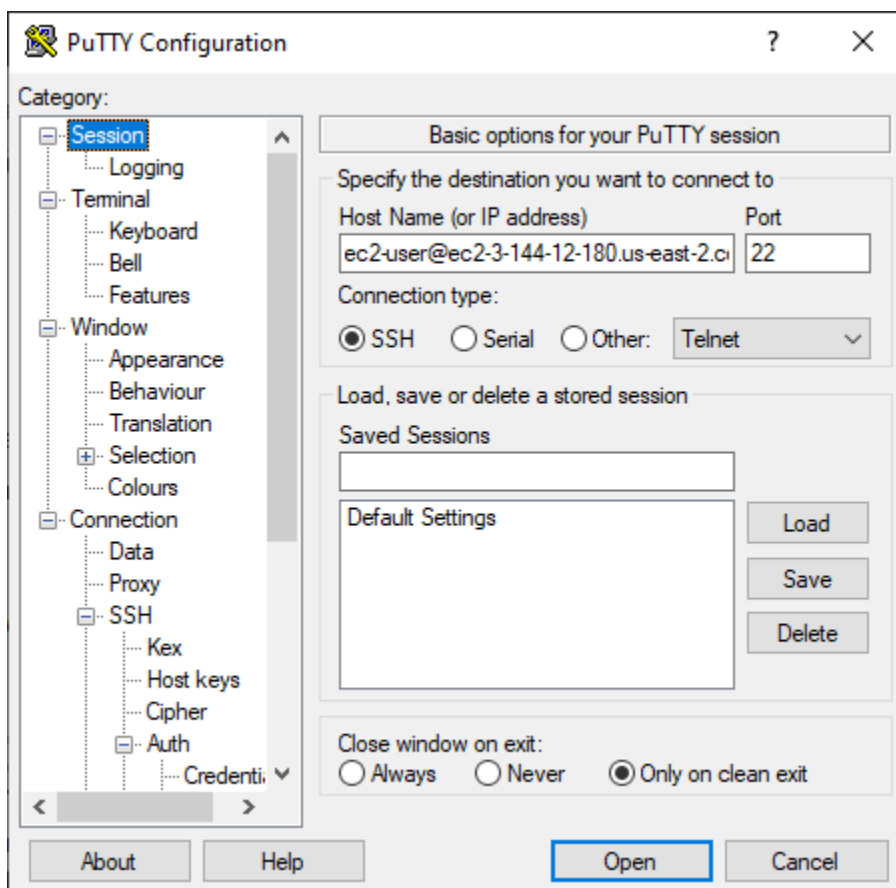


Figura 13. Nos autenticamos con el certificado de seguridad que nos generó nuestro EC2 en el momento de la configuración.

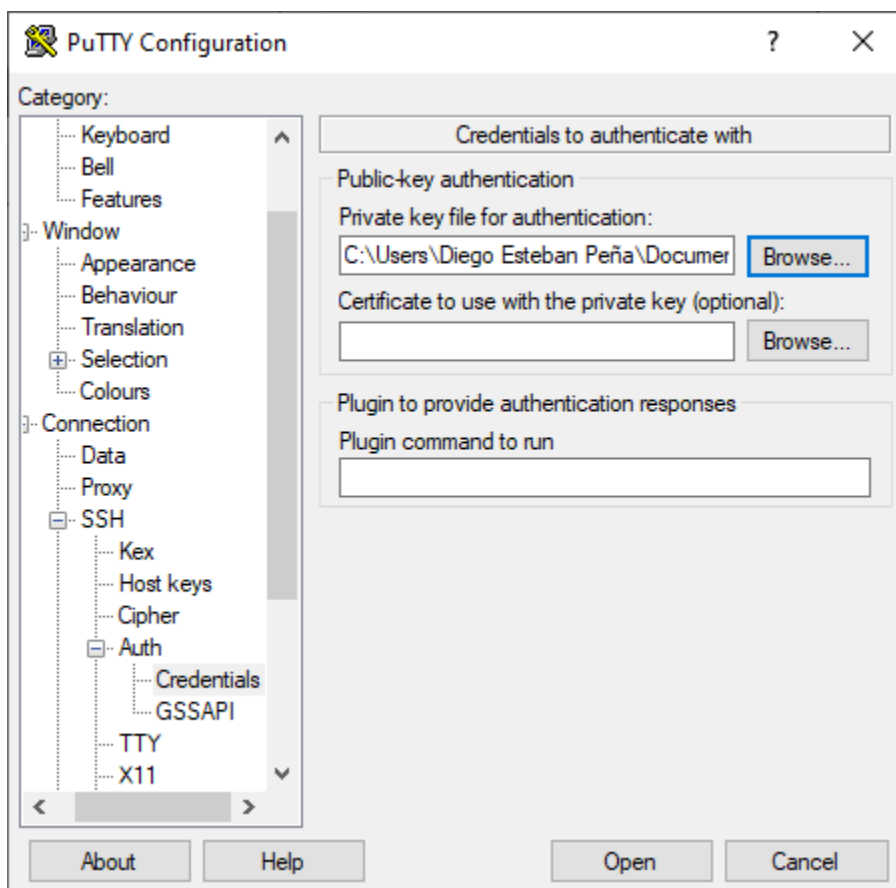


Figura 14. Podemos entrar a la interfaz de comandos de nuestra computadora en la nube. Esto nos muestra que la máquina se encuentra en normal funcionamiento.

```

ec2-user@ip-10-0-4-141:~
Using username "ec2-user".
Authenticating with public key "Entregal-DiegoPena"

#
~\  ###          Amazon Linux 2023
~~ \  #####\
~~  \  ###|
~~   \  \#/      https://aws.amazon.com/linux/amazon-linux-2023
~~    V~'  '->
~~~~
~~~~
~/m/'
[ec2-user@ip-10-0-4-141 ~]$

```

Figura 15. Instalamos nuestro servicio HTTP en nuestra máquina, crearemos el servicio Apache para acceso desde internet a nuestra computadora en la nube. Una vez instalado el servicio, procedemos a ponerlo a funcionar inicializándolo.

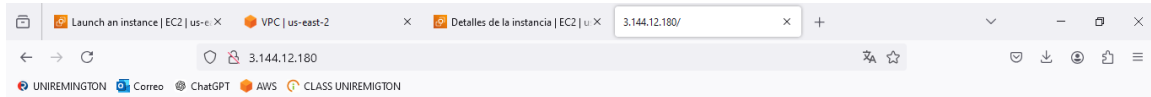
```

root@ip-10-0-4-141:/home/ec2-user
lines 1-4/4 (END)
^C
[root@ip-10-0-4-141 ec2-user]# systemctl start httpd
[root@ip-10-0-4-141 ec2-user]# systemctl status httpd
• httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: d
   Active: active (running) since Tue 2024-11-26 03:40:03 UTC; 5s ago
     Docs: man:httpd.service(8)
  Main PID: 26491 (httpd)
    Status: "Started, listening on: port 80"
    Tasks: 177 (limit: 1111)
   Memory: 12.9M
      CPU: 55ms
   CGroup: /system.slice/httpd.service
           └─26491 /usr/sbin/httpd -DFOREGROUND
             └─26492 /usr/sbin/httpd -DFOREGROUND
               └─26493 /usr/sbin/httpd -DFOREGROUND
                 └─26494 /usr/sbin/httpd -DFOREGROUND
                   └─26495 /usr/sbin/httpd -DFOREGROUND

Nov 26 03:40:03 ip-10-0-4-141.us-east-2.compute.internal systemd[1]: Starting h
Nov 26 03:40:03 ip-10-0-4-141.us-east-2.compute.internal systemd[1]: Started ht
Nov 26 03:40:03 ip-10-0-4-141.us-east-2.compute.internal httpd[26491]: Server c
lines 1-19/19 (END)


```

Figura 16. Verificamos funcionamiento desde nuestro navegador a nuestro servidor creado, mediante la dirección IP Pública, que el sistema AWS nos asignó en el momento de crear nuestra Computadora en la nube.



It works!

Figura 17. Modificamos nuestro archivo de internet Index.html, mediante herramientas de edición de texto de la consola de comandos de nuestra máquina virtual creada con AWS. Así, al momento de entrar desde un computador nos muestra información personalizada de nuestro servidor.



```
root@ip-10-0-4-141:/var/www/html
GNU nano 5.8 index.html
SEMINARIO DE AWS

HOLA A TODOS

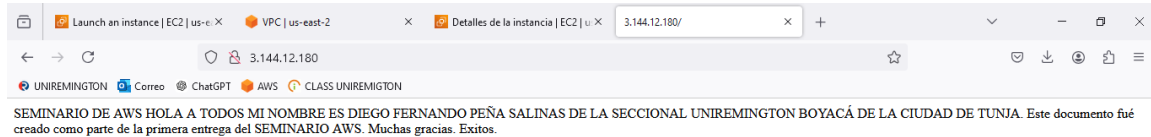
MI NOMBRE ES DIEGO FERNANDO PEÑA SALINAS DE LA SECCIONAL UNIREMINGTON BOYACÁ
DE LA CIUDAD DE TUNJA.
Este documento fué creado como parte de la primera entrega del SEMINARIO AWS.

Muchas gracias.

Exitos.

[ Read 11 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

Figura 18. Verificamos conexión desde nuestro navegador, a nuestra pagina recién modificada.



```
Launch an instance | EC2 | us-e- X  VPC | us-east-2  Detalles de la instancia | EC2 | u X  3.144.12.180/
3.144.12.180
UNIREMINGTON Correo ChatGPT AWS CLASS UNIREMIGTON
SEMINARIO DE AWS HOLA A TODOS MI NOMBRE ES DIEGO FERNANDO PEÑA SALINAS DE LA SECCIONAL UNIREMINGTON BOYACÁ DE LA CIUDAD DE TUNJA. Este documento fué creado como parte de la primera entrega del SEMINARIO AWS. Muchas gracias. Exitos.
```

ENTREGA FINAL

Implementar una arquitectura en AWS que cumpla con los siguientes requisitos:

1. **Balanceador de Carga:** Configure un **Application Load Balancer (ALB)** para distribuir el tráfico entrante a múltiples instancias EC2.

Figura 19. Creamos el equilibrador de cargas llamado **ALB_VIRTUALGOURMET**, que es el nombre que damos a nuestra empresa virtual.

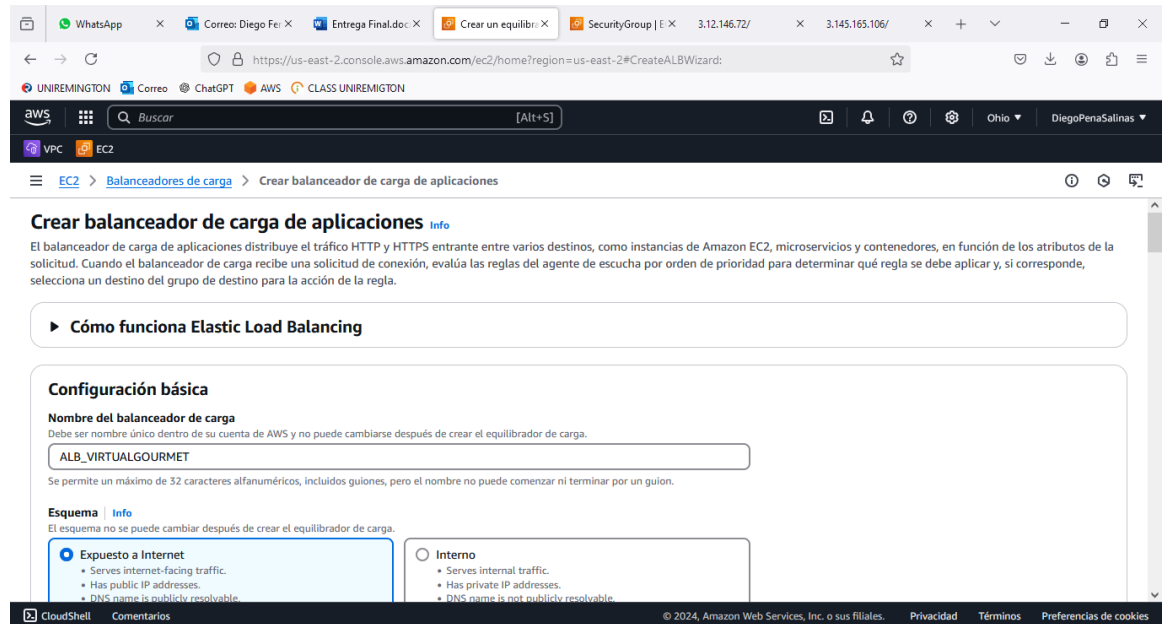


Figura 20. Procedemos a crear el grupo de seguridad, donde va a trabajar nuestro balanceador de cargas, en este caso se llamará **SG-VIRTUALGOURMET**.

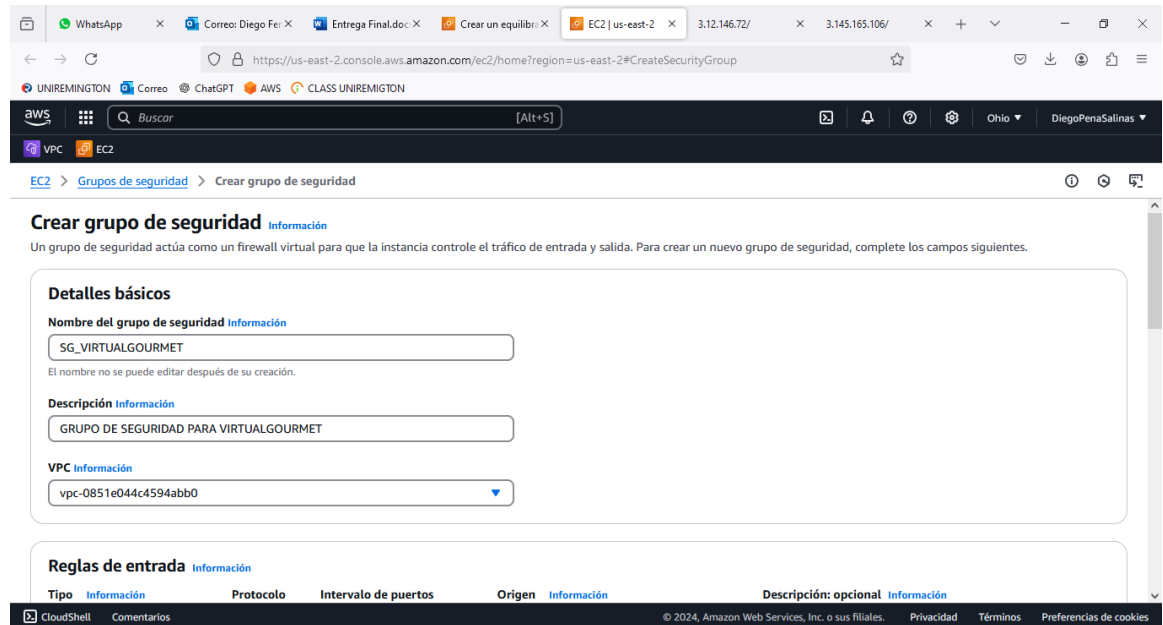


FIGURA 21. En el grupo de seguridad, verificamos que el puerto de entrada 80, esté abierto a cualquier entrada externa por ip4.

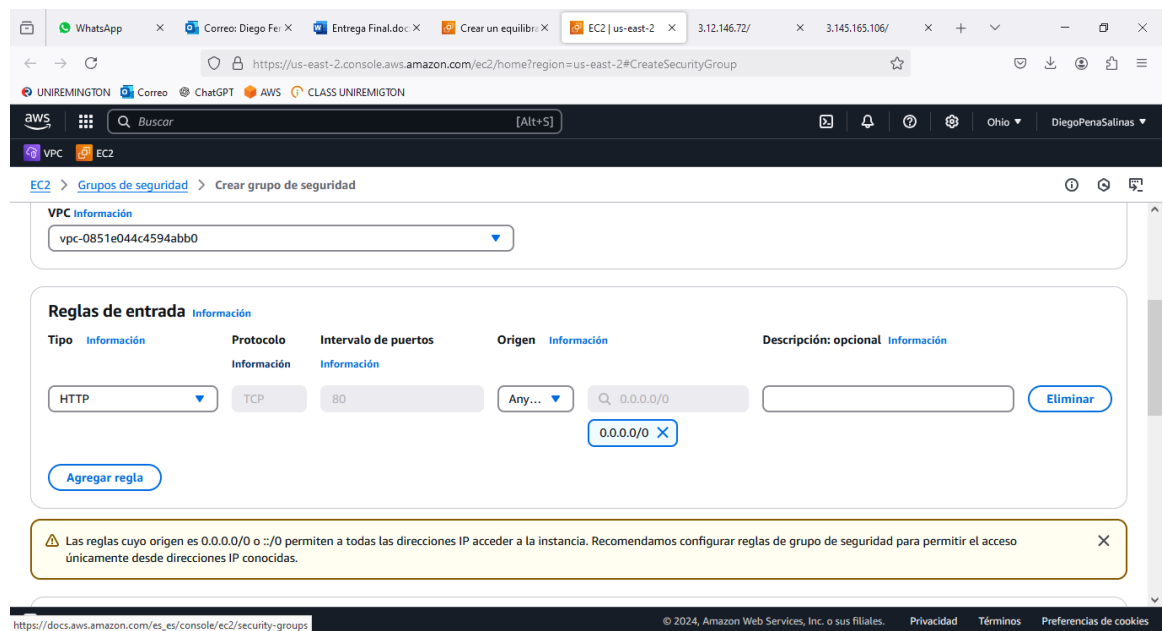


Figura 22. Creamos el grupo de seguridad y verificamos sus configuraciones.

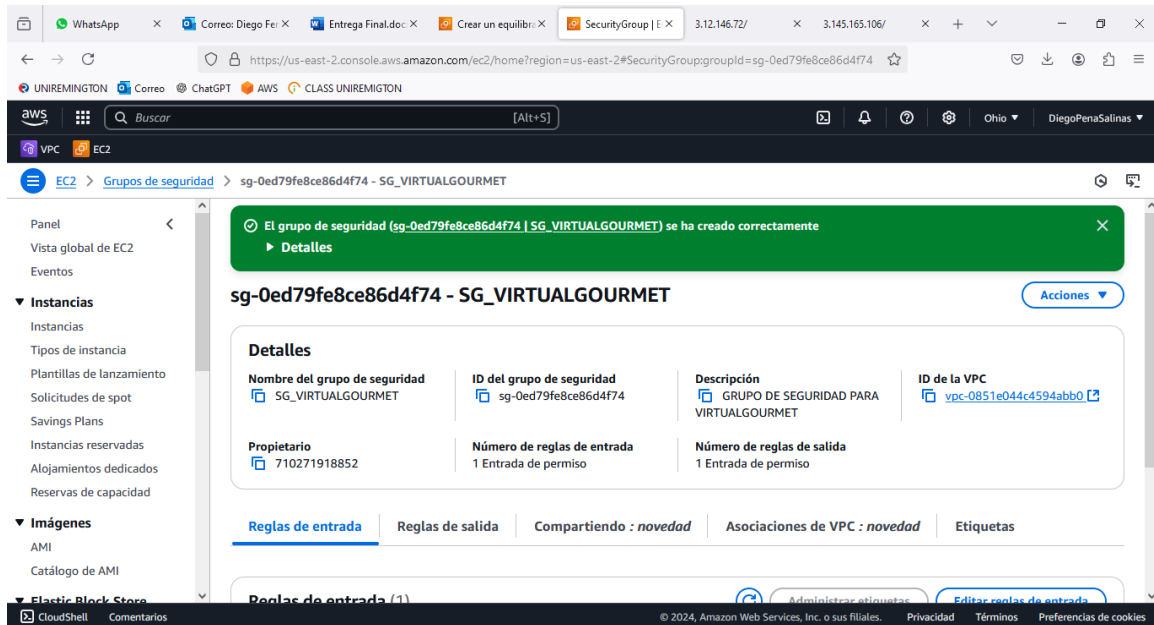


Figura 23. Seguimos con el procedimiento de creación del balanceador de carga, y en esta instancia vamos a adicionar las instancias para trafico, cargamos el grupo de seguridad previamente creado.

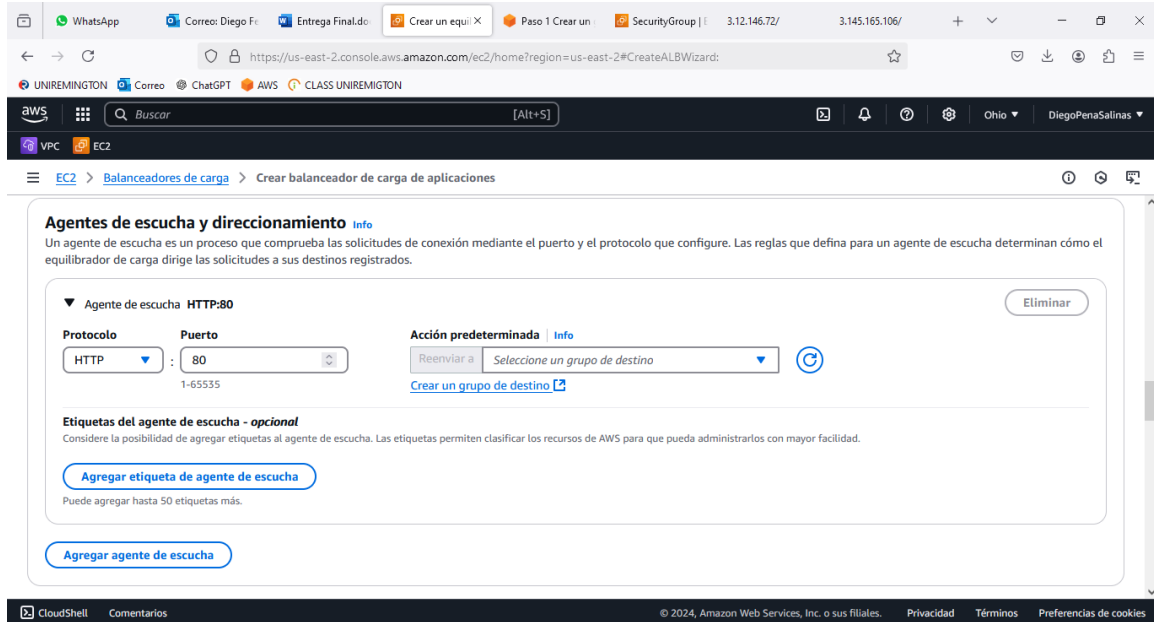


Figura 24. Cargamos los detalles del grupo.

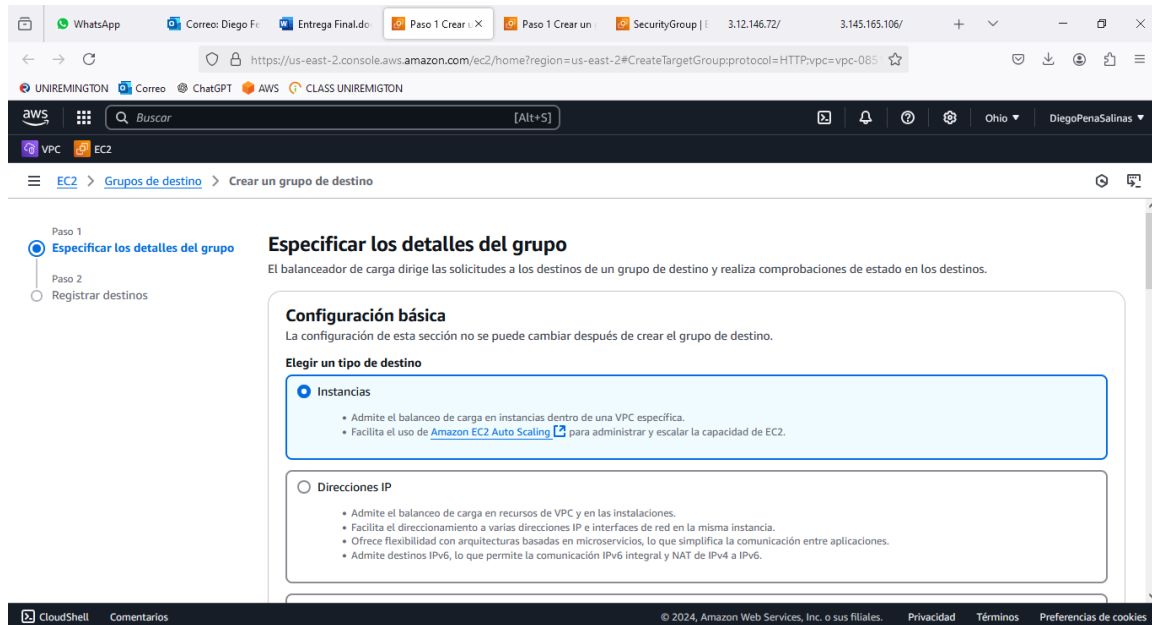


Figura 25. Cargamos las dos instancias previamente creadas para el desarrollo del proyecto, que son **STARTUP-VIRTUALGOURMET-DP** y **STARTUP-VIRTUALGOURMET-DP2**. Y modificamos sus grupos de seguridad al grupo creado **SG-VIRTUALGOURMET**.

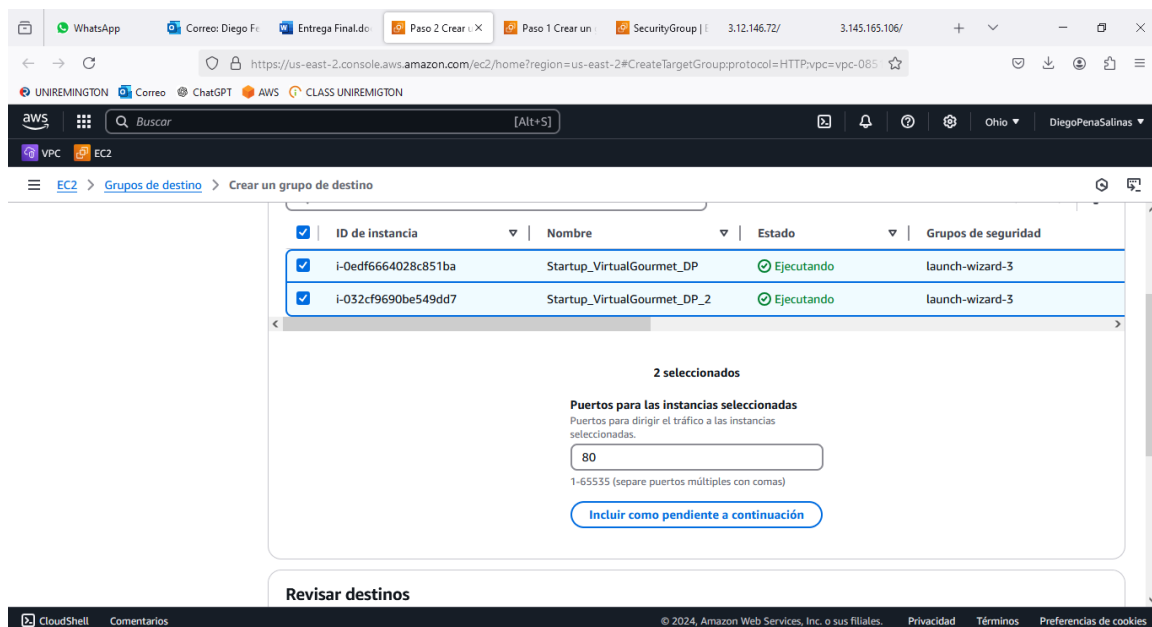


Figura 26. Ya tenemos el grupo creado.

The screenshot shows the AWS Management Console interface for an Amazon Elastic Load Balancing (ALB) target group. The breadcrumb navigation indicates the path: EC2 > Grupos de destino > GD-VirtualGOURMET. The main content area displays the 'Detalles' (Details) for the target group 'GD-VIRTUALGOURMET' with ID 'am:aws:elasticloadbalancing:us-east-2:710271918852:targetgroup/GD-VirtualGOURMET/4d9679b7619e6802'. The details are organized into a table with the following information:

Tipo de destino	Protocolo : Puerto	Versión del protocolo	VPC
Instancia	HTTP: 80	HTTP1	vpc-0851e044c4594abb0
Tipo de dirección IP	Balancedador de carga		
IPv4	Ninguno asociado		
2	0	2	0
Destinos totales	En buen estado	En mal estado	Sin utilizar
	0 Anómalo	0	Inicial
			Vaciado

Below the table, there is a section for 'Distribución de destinos por zona de disponibilidad (AZ)' with a note: 'Seleccione los valores de esta tabla para ver los filtros correspondientes aplicados a la tabla Destinos registrados que aparece a continuación.'

Figura 27. Continuamos con la creación del Balanceador de Carga, y verificamos que tenga cargado el grupo creado anteriormente.

The screenshot shows the 'Resumen' (Summary) page for creating an Amazon Elastic Load Balancing (ALB) application load balancer. The breadcrumb navigation indicates the path: EC2 > Balanceadores de carga > Crear balanceador de carga de aplicaciones. The summary is divided into several sections:

- Configuración básica:** LB-VIRTUALGOURMET, Expuesto a Internet, IPv4.
- Grupos de seguridad:** SG_VIRTUALGOURMET (sg-0ed79fe8ce86d4f74).
- Mapeo de red:** VPC vpc-0851e044c4594abb0, subnets: us-east-2a (subnet-0ae14f98cc6171c24), us-east-2b (subnet-0930e73cb1c3a6112), us-east-2c (subnet-0596643ea089e4a66).
- Agentes de escucha y direccionamiento:** HTTP:80 (valor predeterminado para GD-VirtualGOURMET).
- Integraciones de servicios:** Amazon CloudFront + AWS Web Application Firewall (WAF): Ninguno, AWS WAF: Ninguno, AWS Global Accelerator: Ninguno.
- Etiquetas:** Ninguno.

Figura 28. Confirmación de la creación del grupo.

Se creó correctamente el equilibrador de carga: LB-VIRTUALGOURMET
Pueden transcurrir unos minutos hasta que el equilibrador de carga esté totalmente configurado y listo para dirigir el tráfico. Los destinos también tardarán unos minutos en completar el proceso de registro y superar las comprobaciones de estado iniciales.

LB-VIRTUALGOURMET

Detalles

Tipo de equilibrador de carga Aplicación	Estado Aprovisionándose	VPC vpc-0851e044c4594abb0	Tipo de dirección IP del equilibrador de carga IPv4
Esquema Internet-facing	Zona hospedada Z3AADJG6KTTL2	Zonas de disponibilidad subnet-0ae14f98cc6171c24 us-east-2a (use2-az1) subnet-0930e73cb1c3a6112 us-east-2b (use2-az2) subnet-0596643ea089e4a66 us-east-2c (use2-az3)	Fecha creada 11 de diciembre de 2024, 00:22 (UTC-05:00)
ARN del equilibrador de carga	Nombre de DNS info		

Figura 29. Tenemos confirmación de la creación del balanceador de carga con sus dos instancias y su estado Healthy.

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

Destinos registrados (2) info Mitigación de anomalías: No aplicable Anular el registro Registrar destinos

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

Filtrar destinos

<input type="checkbox"/>	ID de instancia	Nombre	Puerto	Zona	Estado	Detalles del
<input type="checkbox"/>	i-0edf6664028c851ba	Startup_VirtualGourmet_DP	80	us-east-2a (us...)	Healthy	-
<input type="checkbox"/>	i-032cf9690be549dd7	Startup_VirtualGourmet_DP_2	80	us-east-2a (us...)	Healthy	-

Figura 30. Verificamos la creación de nuestras instancias con la conexión remota por PUTTY.

Detalles del grupo de Auto S...

Crear plantilla de lanzamien... WhatsApp 3.12.146.72/ Nueva pestaña

https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#AutoScalingGroupDetails:sid=AS-VIRTUALGOURM...

UNIREMINGTON Correo ChatGPT AWS CLASS UNIREMINGTON

Buscar [Alt+S]

VPC EC2

EC2 > Grupos de Auto Scaling > AS-VIRTUALGOURMET-DP

Política de escalado dinámico creada o editada correctamente.

Acción programada creada o editada correctamente

Las políticas de escalado predictivo utilizan datos históricos para escalar de forma horizontal su grupo antes de la carga por hora prevista.

Crear una política de escalado predictivo

Acciones programadas (1) Info

Acciones Crear acción programada

Filtrar las acciones programadas

<input type="checkbox"/>	Nombre	Hora de inicio	Hora de fin...	Recurrencia	Zona horaria	Capacidad ...	Mín.	Máx.
<input type="checkbox"/>	carga virtualg...	2024 Decemb...		21****	America/Bogota	25	10	25

CloudShell Comentarios

© 2024, Amazon Web Services, Inc. o sus filiales. Privacidad Términos Preferencias de cookies

Figura 33. Verificamos el funcionamiento de nuestra regla automática en nuestra conexión PUTTY.

```

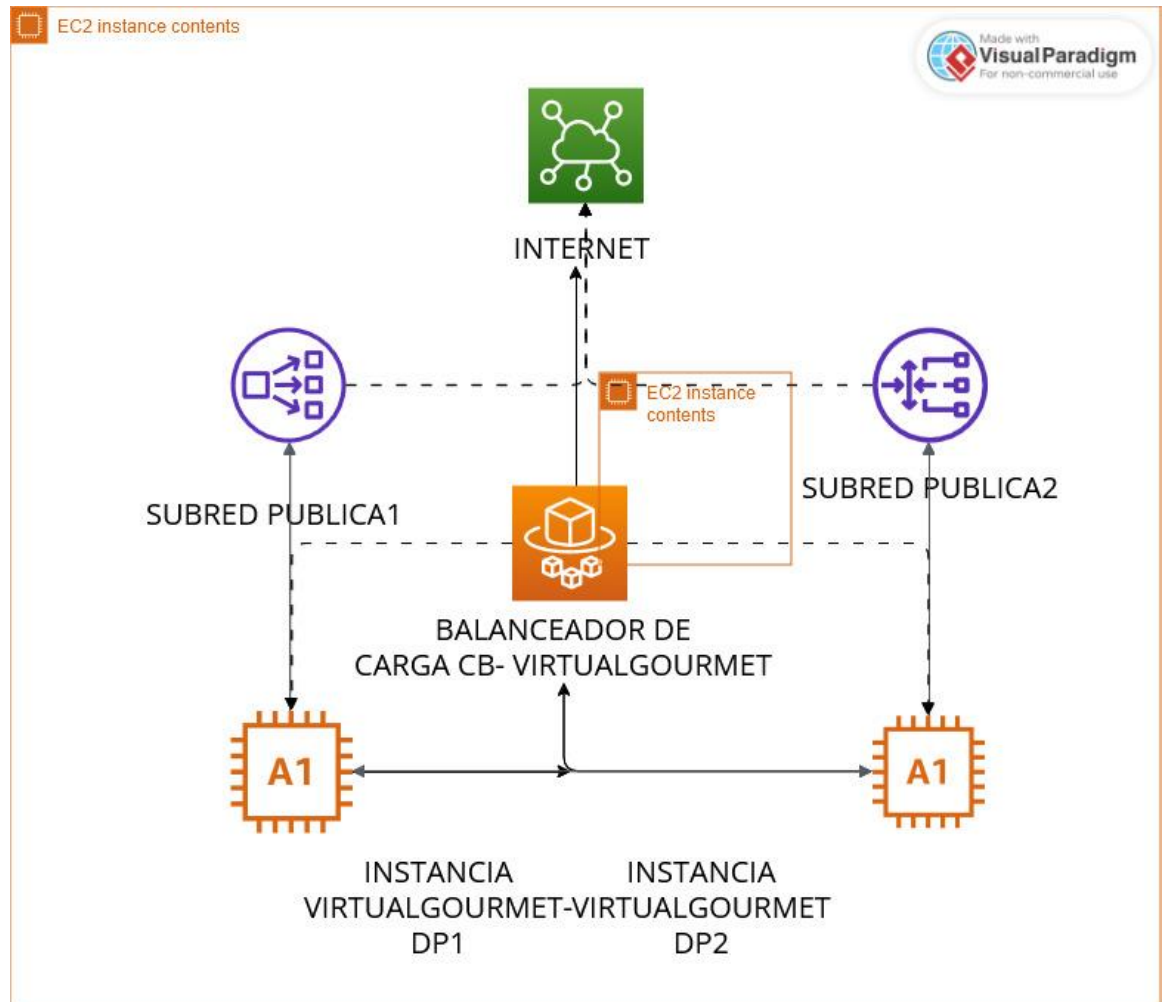
root@ip-172-31-6-74:/home/ec2-user

[root@ip-172-31-6-74 ec2-user]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor prese
   Active: active (running) since Wed 2024-12-11 04:46:10 UTC; 2h 30min ago
     Docs: man:httpd.service(8)
   Main PID: 3352 (httpd)
  Status: "Total requests: 465; Idle/Busy workers 100/0;Requests/sec: 0.0515; B
   CGroup: /system.slice/httpd.service
           └─3352 /usr/sbin/httpd -DFOREGROUND
             └─3353 /usr/sbin/httpd -DFOREGROUND
               └─3354 /usr/sbin/httpd -DFOREGROUND
                 └─3355 /usr/sbin/httpd -DFOREGROUND
                   └─3356 /usr/sbin/httpd -DFOREGROUND
                     └─3357 /usr/sbin/httpd -DFOREGROUND
                       └─3428 /usr/sbin/httpd -DFOREGROUND
                         └─3434 /usr/sbin/httpd -DFOREGROUND
                           └─3435 /usr/sbin/httpd -DFOREGROUND

Dec 11 04:46:10 ip-172-31-6-74.us-east-2.compute.internal systemd[1]: Starting T
Dec 11 04:46:10 ip-172-31-6-74.us-east-2.compute.internal systemd[1]: Started Th
[root@ip-172-31-6-74 ec2-user]# systemctl nginx
Unknown operation 'nginx'.
[root@ip-172-31-6-74 ec2-user]# sudo apt update

```

Grafico 1. Grafico de conexiones de los servicios de nuestro proyecto VIRTUALGOURMET.



Conclusiones

Al finalizar este ejercicio, reforzamos conocimientos en el tema de AWS, y nos damos cuenta que hay mucho tema por socializar, ya que esta plataforma tiene muchos servicios para aplicar a nuestra vida diaria.

Los balanceadores de carga pueden distribuir el tráfico de forma eficiente entre las instancias, con el fin de optimizar el uso de los recursos de la máquina y poder tener un rendimiento favorable para el usuario. La creación de instancias en AWS se realiza de forma fácil y ágil, en servicios virtuales con adaptaciones específicas a las necesidades de las aplicaciones.

Con el auto escalado nos aseguramos que la estructura de nuestros servicios, se ajusten a la demanda, aumentando o disminuyendo los recursos según su uso. Para así mejorar la eficiencia en las operaciones.

Todos estos conjuntos de AWS, facilitan la creación de las operaciones de fácil acceso en la nube

Referencias

diciembre de 2024, *Bienvenido a la documentación de AWS*,

https://docs.aws.amazon.com/es_es/