TRABAJO DE GRADO
Opción Seminario-Diplomado.

Implementación de Arquitectura en AWS para la startup Digital Tech BECS

Corporación Universitaria Remington
Ingeniería de Sistemas
Seminario Amazon Web Services

Blanca Elizeth Castiblanco Silva
Juan Pablo Berrio López
Seminario de grado
2024

Tabla de Contenidos

Resumen

En este trabajo se da a conocer la implementación de una arquitectura en AWS haciendo uso de los servicios que ofrece este reconocido proveedor de computación en la nube.
Tiene como objetivo optimizar la infraestructura física tecnológica del sistema.
Los servicios más usados en este trabajo son instancias y contenedores, lo que hace que sea una plataforma de alta disponibilidad. Asegura un alto rendimiento, mayor seguridad, eficiencia y escalabilidad en función del crecimiento del trabajo.
Por ser recursos virtuales genera flexibilidad para ajustar la capacidad de almacenamiento, procesamiento y red ya que no se requieren equipos ni infraestructuras físicas.
Gracias a los recursos que ofrece se pueden rastrear costos y rendimiento.
Este trabajo se implementa una infraestructura ágil y flexible alineada con las demandas de la tecnología moderna.

Palabras clave

Seguridad, rendimiento, disponibilidad, crecimiento, sistematización.

Marco conceptual y contextual

Computación en la nube

La informática en la nube es la distribución de recursos de TI bajo demanda a través de Internet mediante un esquema de pago por uso. En lugar de comprar, poseer y mantener servidores y centros de datos físicos, puede acceder a servicios tecnológicos, como capacidad informática, almacenamiento y bases de datos, en función de sus necesidades a través de un proveedor de la nube como Amazon Web Services (AWS).

¿Qué son los servicios en la nube?

Los servicios en la nube son recursos de TI administrados por AWS que se entregan bajo demanda a través de Internet. Tradicionalmente, las organizaciones debían adquirir y configurar todo, desde el hardware de servidores y sistemas de almacenamiento hasta las tecnologías de redes y seguridad, antes de lanzar cualquier sistema digital. El aprovisionamiento y la administración de la infraestructura de TI representan un desafío costoso y complejo, además de restar tiempo a la innovación.

Los servicios en la nube permiten a cualquier persona acceder a la infraestructura de TI necesaria para crear y mantener sistemas digitales, al abstraer infraestructuras complejas de forma que todos puedan desarrollar aplicaciones sofisticadas con rapidez y a escala global. Ejecutar sus aplicaciones en un servidor en la nube es solo el comienzo. Puede utilizar los servicios en la nube para agregar inteligencia artificial y machine learning (IA/ML), en tiempo real y muchas otras capacidades a sus aplicaciones.

¿Qué son los servicios administrados en la nube?

Los servicios en la nube también se denominan servicios administrados en la nube porque AWS administra por completo la infraestructura subyacente. Todo el hardware, los sistemas operativos y otras capas de infraestructura necesarios se almacenan y administran en centros de datos de AWS distribuidos en todo el mundo, con una altamente segura. Compramos y mantenemos todo tipo de recursos de TI y los ponemos a disposición como servicios a los que puede acceder en el código de su aplicación. AWS monitorea y mantiene servidores en la nube, almacenamiento, redes, bases de datos y mucho más, lo que garantiza un rendimiento y un tiempo de actividad constantes.

Además del mantenimiento del hardware, nos ocupamos de todo tipo de tareas operativas de TI, desde el equilibrio de carga y la administración de parches hasta la recuperación ante desastres, entre otras.

Hoy en día, gracias a AWS, cualquier persona, desde estudiantes universitarios hasta equipos empresariales, puede acceder a los servicios en la nube por una fracción del costo que implica administrar la infraestructura local. Cualquier persona puede crear e implementar software sin grandes inversiones iniciales en infraestructura de TI.

Tipos de informática en la nube

Los tres tipos principales de informática en la nube son: infraestructura como servicio, plataforma como servicio y software como servicio. Cada tipo ofrece diferentes niveles de control, flexibilidad y administración para que pueda seleccionar el conjunto de servicios correcto para sus necesidades.

Infraestructura como servicio (IaaS)

La IaaS incluye los bloques de creación básicos para la TI basada en la nube. Generalmente, provee acceso a características de conexión en red, equipos (virtuales o en hardware exclusivo) y espacio de almacenamiento de datos. La IaaS le ofrece el mayor nivel de flexibilidad y control de administración en relación con sus recursos de TI. Es similar a los recursos de TI que muchos desarrolladores y departamentos de TI ya conocen.

Plataforma como servicio (PaaS)

La PaaS elimina la necesidad de administrar la infraestructura subyacente (normalmente hardware y sistemas operativos) y permite enfocarse en la implementación y administración de aplicaciones. Esto contribuye a mejorar el nivel de eficiencia, ya que no debe preocuparse por el aprovisionamiento de recursos, la planificación de la capacidad, el mantenimiento del software, la implementación de parches ni ninguna de las demás arduas tareas que conlleva la ejecución de su aplicación.

Software como servicio (SaaS)

El SaaS le proporciona un producto completo que el proveedor del servicio ejecuta y administra. En la mayoría de los casos, quienes hablan de SaaS en realidad se refieren a aplicaciones de usuario final (como el email basado en la Web). Con una solución basada en SaaS, ya no debe pensar en cómo mantener el servicio ni en cómo administrar la infraestructura subyacente. Solamente debe pensar en cómo utilizar ese software en particular.

VPC

Amazon Virtual Private Cloud (VPC) es un servicio que permite lanzar recursos de AWS en una red virtual aislada de forma lógica que usted defina. Puede controlar todos los aspectos del entorno de red virtual, incluida la selección de su propio rango de direcciones IP, la creación de subredes y la configuración de tablas de ruteo y puertas de enlace de red. Puede utilizar tanto IPv4 como IPv6 para la mayoría de los recursos de la VPC, lo que ayuda a garantizar el acceso seguro y fácil a los recursos y las aplicaciones.

Como uno de los servicios esenciales de AWS, Amazon VPC facilita la personalización de la configuración de red de la VPC. Puede crear una subred con acceso público para los servidores web que tengan acceso a Internet. También permite colocar los sistemas backend, como los servidores de aplicaciones o las bases de datos, en una subred con acceso privado sin acceso a Internet. Amazon VPC le permite utilizar varias capas de seguridad, incluidos los grupos de seguridad y las listas de control de acceso a la red, para ayudar a controlar el acceso a las instancias de Amazon Elastic Compute Cloud (Amazon EC2) en cada subred.

Instancias

Una instancia de computación en la nube es un recurso de servidor que brindan servicios en la nube de terceros. Si bien puede administrar y mantener los recursos del servidor físico en las instalaciones, hacerlo es costoso e ineficiente. Los proveedores de la nube mantienen el hardware en sus centros de datos y les dan acceso virtual a los recursos de computación en forma de instancia. Puede usar la instancia en la nube para ejecutar

cargas de trabajo con uso intensivo de cómputos, como contenedores, bases de datos, microservicios y máquinas virtuales.

Balanceador de Carga

El equilibrador de carga de aplicación funciona al nivel de las solicitudes (capa 7) y dirige el tráfico a los destinos (instancias EC2, contenedores, direcciones IP y funciones de Lambda) según el contenido de la solicitud. El equilibrador de carga de aplicación, ideal para el equilibrio de carga avanzado de tráfico HTTP y HTTPS, proporciona un direccionamiento de solicitudes avanzado destinado a la entrega de arquitecturas de aplicaciones modernas, incluidos microservicios y aplicaciones basadas en contenedores. El equilibrador de carga de aplicación simplifica y mejora el nivel de seguridad de su aplicación. Para ello, se asegura de que se utilicen en todo momento los protocolos y cifradores SSL/TLS más recientes.

Volúmenes

Los volúmenes HDD pasivos (sc1) ofrecen almacenamiento magnético de bajo costo que define el funcionamiento en términos de rendimiento en lugar de IOPS. Con un límite de rendimiento más bajo que el de los volúmenes HDD optimizados para el rendimiento (st1), los volúmenes sc1 representan una buena opción para las cargas de trabajo secuenciales, de datos pasivos y de gran tamaño. Si no necesita acceder a sus datos con frecuencia y quiere ahorrar costos, los volúmenes sc1 proporcionan almacenamiento en bloque económico. No se admiten volúmenes de arranque sc1.

Instantáneas / Snapshot

Las instantáneas de Amazon Elastic Block Store (EBS) proporcionan una solución de protección de datos sencilla y segura diseñada para proteger los datos de almacenamiento en bloques, como los volúmenes de EBS, los volúmenes de arranque y los datos de bloques en las instalaciones. Las instantáneas de EBS son una copia puntual de sus datos y se pueden utilizar para permitir la recuperación de desastres, migrar datos entre regiones y cuentas y mejorar la conformidad de las copias de seguridad.

Puede crear y administrar sus instantáneas de EBS a través de la consola de administración de AWS, la interfaz de la línea de comandos (CLI) de AWS o los SDK de AWS. Las instantáneas de Amazon EBS también se integran con Amazon Data Lifecycle Manager (DLM), que le permite definir políticas que lo ayudan a automatizar la administración del ciclo de vida de las instantáneas. Las instantáneas de EBS se almacenan de forma progresiva, lo que significa que solo se facturarán los bloques almacenados que han cambiado

Security Groups

Un grupo de seguridad controla el tráfico que puede llegar y salir de los recursos a los que está asociado. Por ejemplo, después de asociar un grupo de seguridad con una instancia de EC2, este controla el tráfico entrante y saliente de la instancia.

Cuando crea una VPC, esta viene con un grupo de seguridad predeterminado. Puede crear grupos de seguridad adicionales para una VPC, cada uno con sus propias reglas de entrada y salida. Puede especificar el origen, el rango de puertos y el protocolo para cada regla de entrada. Puede especificar el destino, el rango de puertos y el protocolo para cada regla de salida.

El siguiente diagrama muestra una VPC con una subred, una puerta de enlace de Internet y un grupo de seguridad. La subred contiene una instancia de EC2. El grupo de seguridad está asignado a la instancia. El grupo de seguridad actúa como un firewall virtual. El único tráfico que llega a la instancia es el tráfico permitido por las reglas del grupo de seguridad. Por ejemplo, si el grupo de seguridad contiene una regla que permite el tráfico ICMP a la instancia desde su red, entonces podría hacer ping a la instancia desde su computadora. Si el grupo de seguridad no contiene una regla que permita el tráfico SSH, entonces no podría conectarse a su instancia mediante SSH.

Auto Scaling Group

Puede ajustar manualmente el número de EC2 instancias de su grupo de Auto Scaling en cualquier momento. Este proceso de cambiar el recuento de instancias manualmente se

denomina *escalado manual*. El escalado manual es una alternativa al escalado automático, especialmente si desea realizar cambios de capacidad únicos.

El escalado dinámica escala la capacidad del grupo de escalado automático a medida que se producen cambios de tráfico

Amazon EC2 Auto Scaling admite los siguientes tipos de políticas de escalado dinámico:

- **Escalado de seguimiento de objetivos**: aumenta y reduce la capacidad actual del grupo en función de una CloudWatch métrica de Amazon y un valor objetivo. Funciona de forma similar a los termostatos, que mantienen la temperatura del hogar: seleccionamos una temperatura y el termostato hace el resto.
- **Step scaling** (Escalado por pasos): permite aumentar o reducir la capacidad actual del grupo en función de una serie de ajustes de escalado, denominados *ajustes por pasos*, que variarán en función del tamaño de la interrupción de alarma.
- **Simple scaling** (Escalado sencillo): permite aumentar o reducir la capacidad actual del grupo en función de un único ajuste de escalado con un periodo de recuperación entre cada actividad.

¿Qué es 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 versión ejecutable. Con Docker, puede implementar y ajustar la escala de aplicaciones rápidamente en cualquier entorno con la certeza de saber que su código se ejecutará.

- La ejecución de Docker en AWS les ofrece a desarrolladores y administradores una manera muy confiable y económica de crear, enviar y ejecutar aplicaciones distribuidas en cualquier escala.

- La frecuencia media de envío de software de los usuarios de Docker es siete veces superior a la de aquellos que no lo usan. Docker le permite entregar servicios aislados con la frecuencia necesaria.
- Las aplicaciones con contenedores facilitan la implementación, la identificación de problemas y el retorno a una fase anterior para remediarlos.
- Las aplicaciones basadas en Docker pueden transferirse a la perfección desde equipos de desarrollo locales a implementaciones de producción en AWS.
- Las aplicaciones con contenedores facilitan la implementación, la identificación de problemas y el retorno a una fase anterior para remediarlos.

Contenedores

Los servicios de contenedores de AWS facilitan la administración de la infraestructura subyacente, ya sea en las instalaciones locales o en la nube, para que usted se pueda centrar en la innovación y en las necesidades de su negocio.

Desarrollo e implementación del aprendizaje

Entrega I

Investigue y explique cuáles son los recursos de AWS que permite ejecutar instancias de AWS.

En Amazon Web Services (AWS), una instancia depende de varios recursos y componentes para operar correctamente. A continuación, se describen los principales recursos y servicios necesarios:

1. Recursos de Cómputo

Instancias EC2: Son las máquinas virtuales que proporcionan el poder de procesamiento.
Tipos de instancias: Determinan la capacidad de la máquina (CPU, memoria, almacenamiento, y optimización de red).

2. Almacenamiento

Amazon Elastic Block Store (EBS): Proporciona almacenamiento persistente para las instancias EC2. Un volumen EBS es como un disco duro que se adjunta a la instancia.
Amazon S3: Almacenamiento para objetos (archivos, backups, datos de aplicaciones).
Instance Store: Almacenamiento efímero directamente vinculado al hardware de la instancia.

3. Redes

Amazon Virtual Private Cloud (VPC): Una red virtual dedicada a tu cuenta AWS.
Subnets: Segmentos de red dentro del VPC que contienen las instancias.
Internet Gateway (IGW): Para proporcionar acceso a Internet a las instancias.
Elastic IPs: Direcciones IP estáticas que pueden asignarse a las instancias.
Security Groups: Reglas de firewall que controlan el tráfico entrante y saliente de las instancias.
Network Access Control Lists (NACLs): Reglas adicionales para controlar el tráfico en las subredes.

4. **Administración de Identidad y Seguridad**

AWS Identity and Access Management (IAM): Control de acceso a los recursos a través de políticas y roles.

Roles de IAM para instancias: Permiten que las instancias interactúen con otros servicios de AWS de manera segura.

5. Balanceo y Escalado

Elastic Load Balancer (ELB): Distribuye el tráfico entrante entre varias instancias.

Auto Scaling Groups (ASG): Escala automáticamente el número de instancias EC2 según la demanda.

6. Monitoreo y Gestión

Amazon CloudWatch: Monitorea métricas como el uso de CPU, memoria y disco, y configura alertas.

AWS Systems Manager: Gestiona las instancias, automatiza tareas administrativas y soluciona problemas.

7. DNS y Control de Tráfico

Amazon Route 53: Administra el enrutamiento de tráfico a las instancias.

8. Backup y Recuperación

Amazon Backup: Permite realizar copias de seguridad automáticas de volúmenes EBS.

Snapshots de EBS: Copias puntuales de los volúmenes para recuperación de datos.

9. Opciones de Configuración

Key Pairs: Claves SSH para acceder a las instancias.

User Data y Metadata: Scripts y configuraciones que se ejecutan al iniciar una instancia.

Implemente un servidor web en Amazon Linux

Imagen 1. Crear VPC

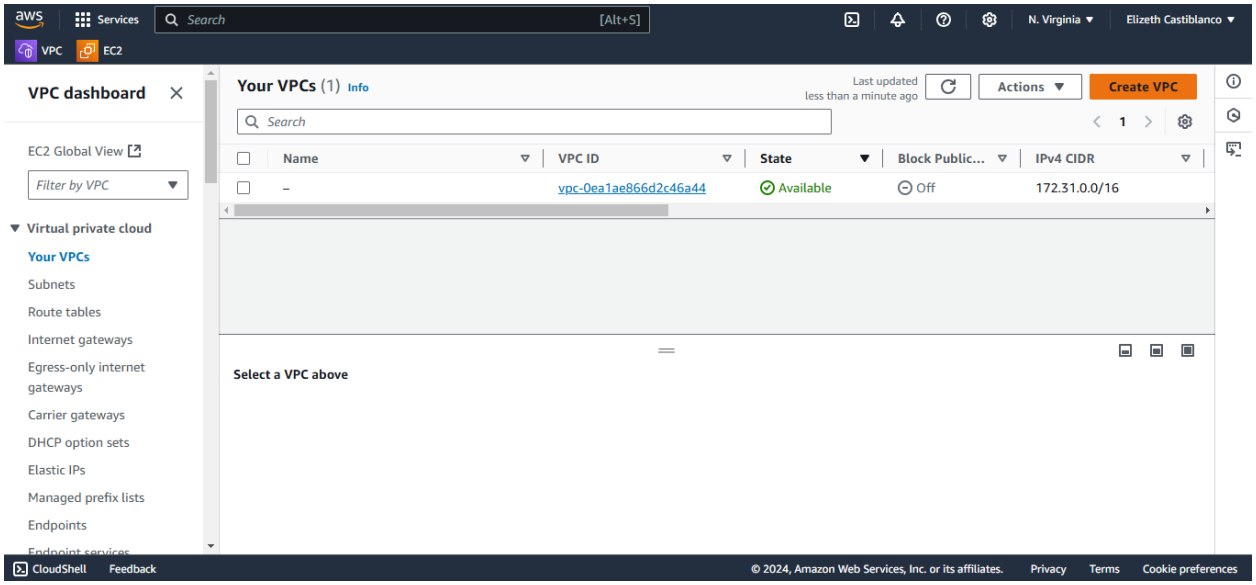


Imagen 2. Asignar nombre

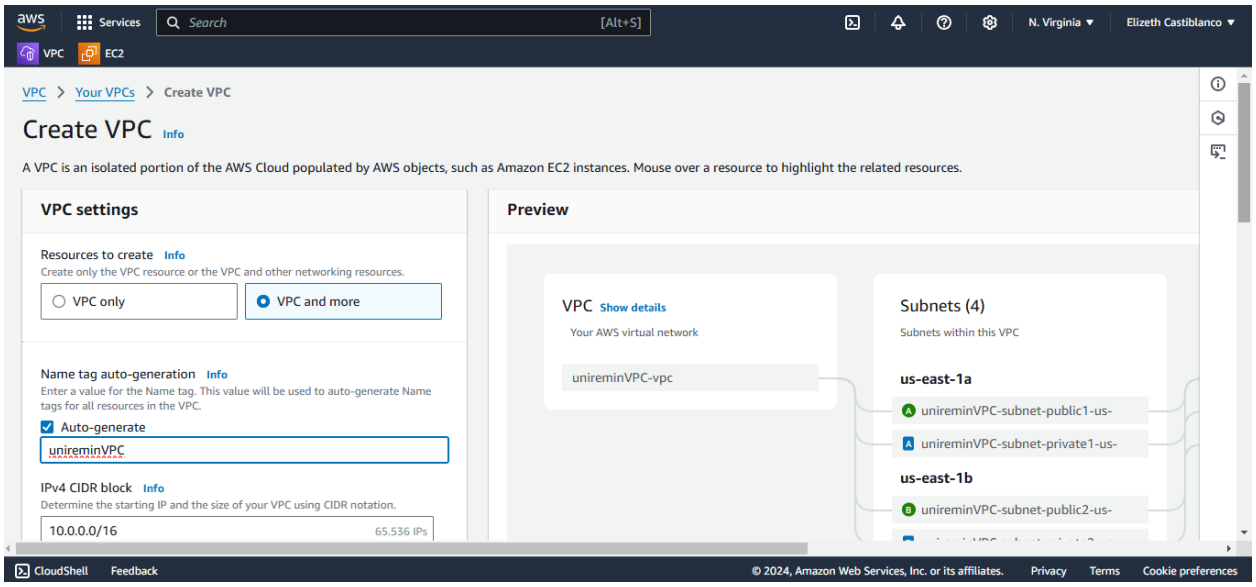


Imagen 3. Elegir recursos de la VPC

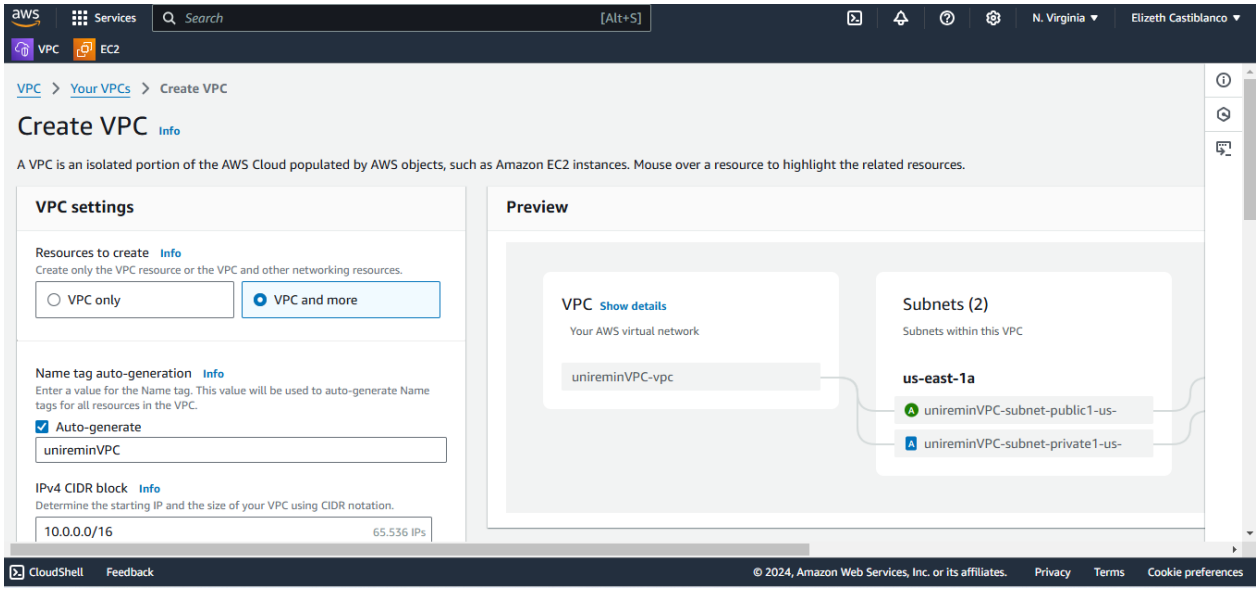


Imagen 4. Elegir número de zonas de disponibilidad, numero de subredes públicas y privadas.

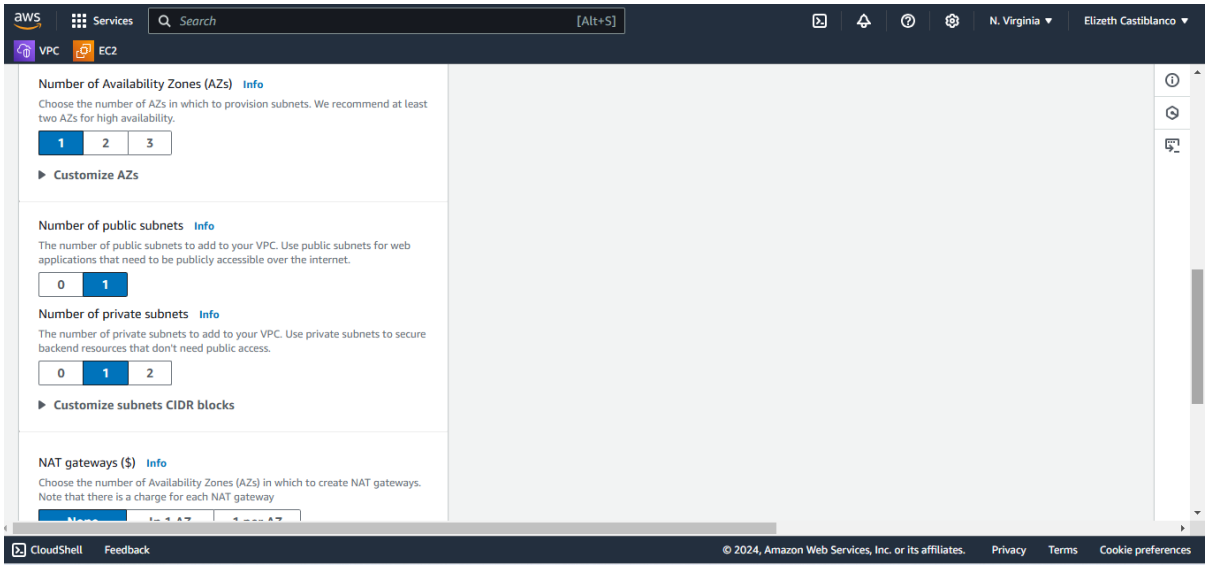


Imagen 5. Ejecución de la creación de la VPC

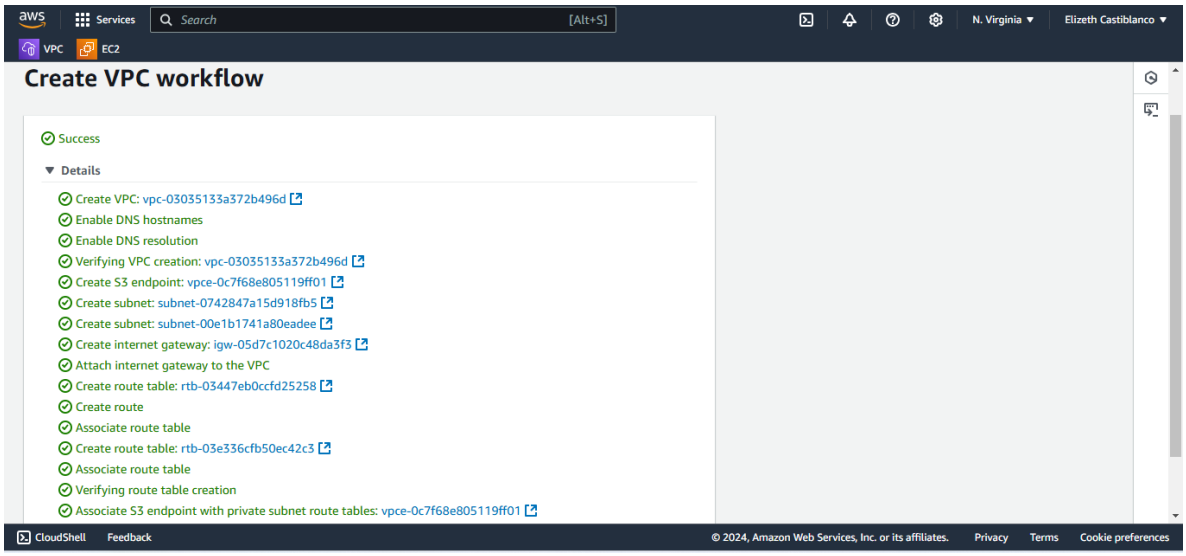


Imagen 6. VPC creada con su respectiva información

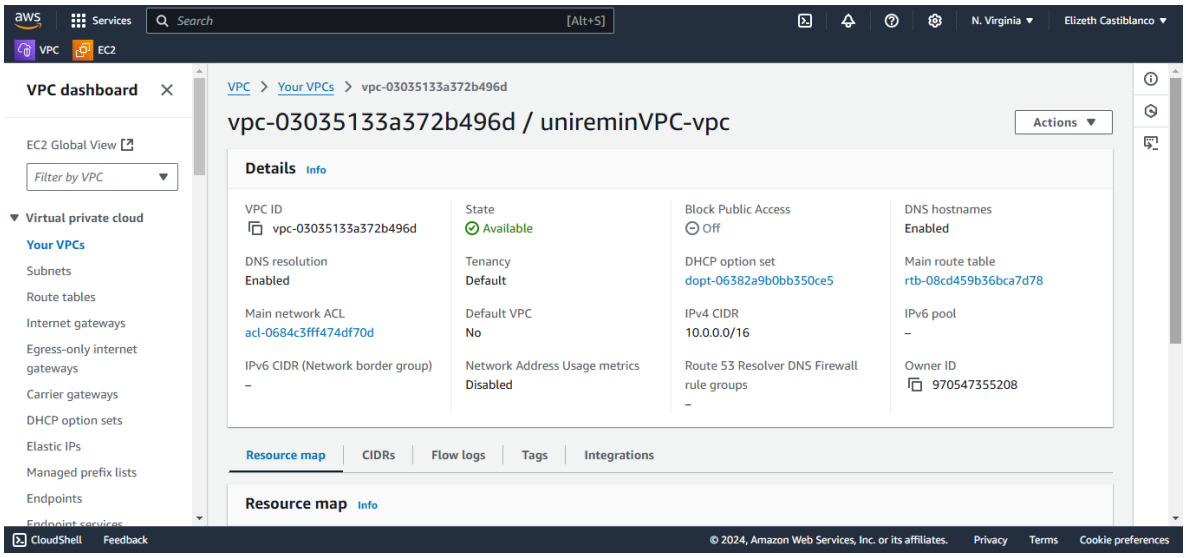


Imagen 7. Creación de Instancia

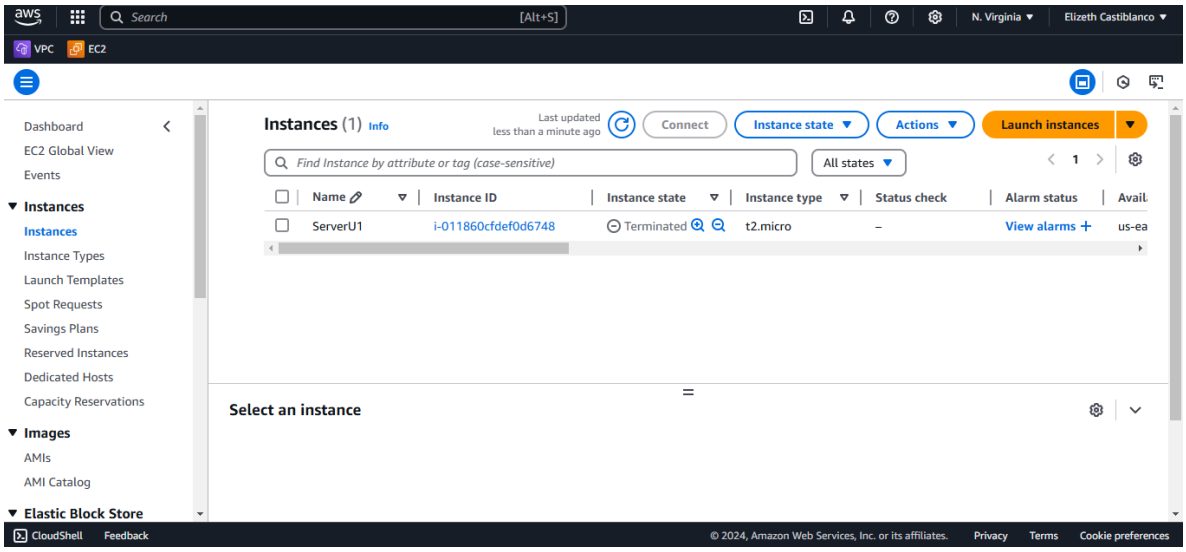


Imagen 8. Asignación de nombre

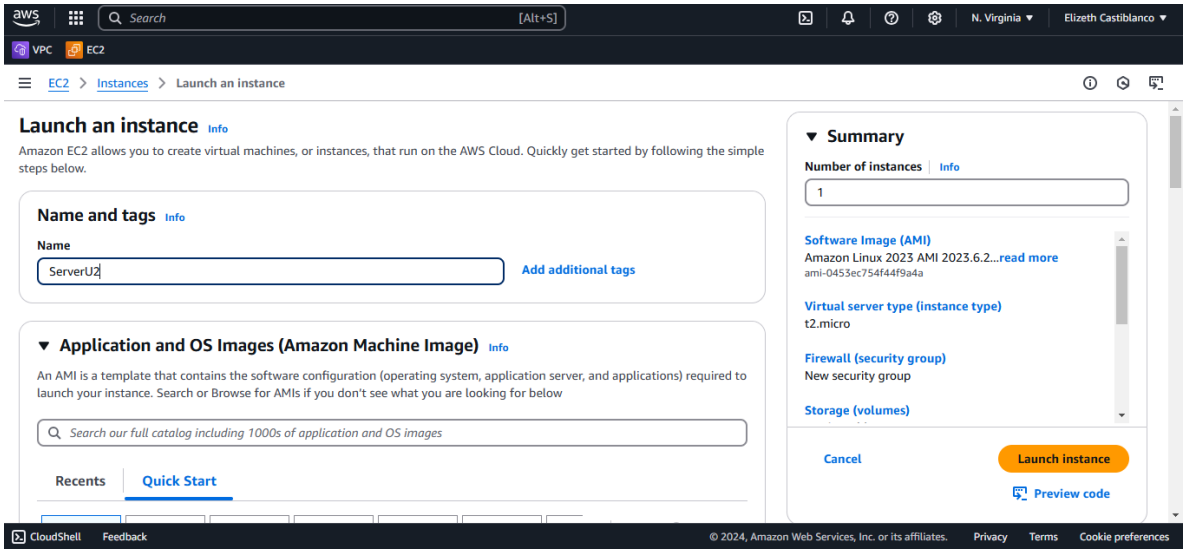


Imagen 9. Elección de Aplicación y Sistema Operativo para lanzar la creación de una máquina virtual.

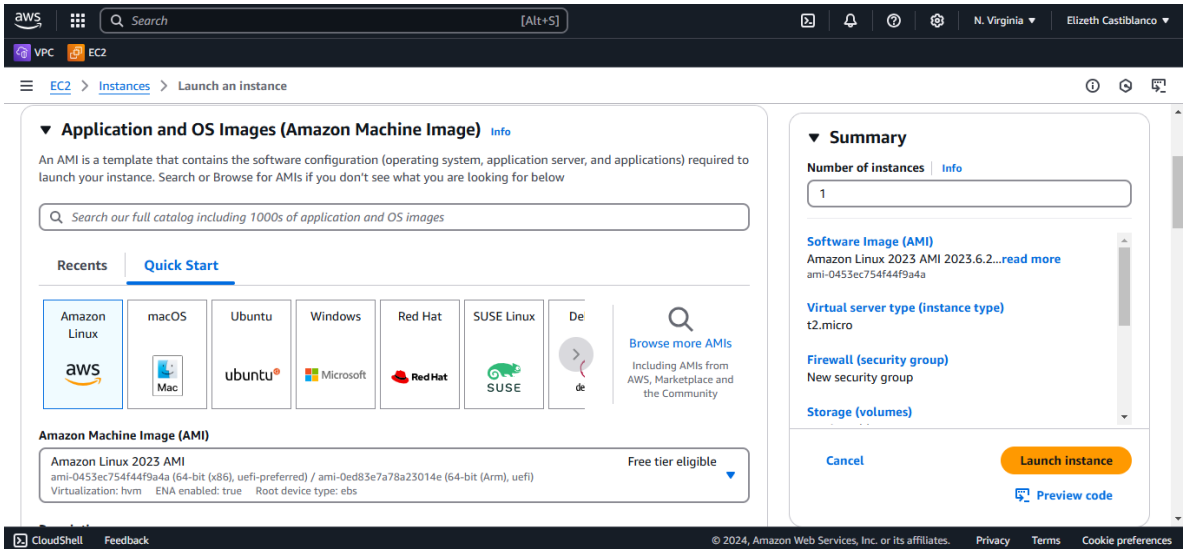


Imagen 10. Elección de versión de sistema operativo gratuito

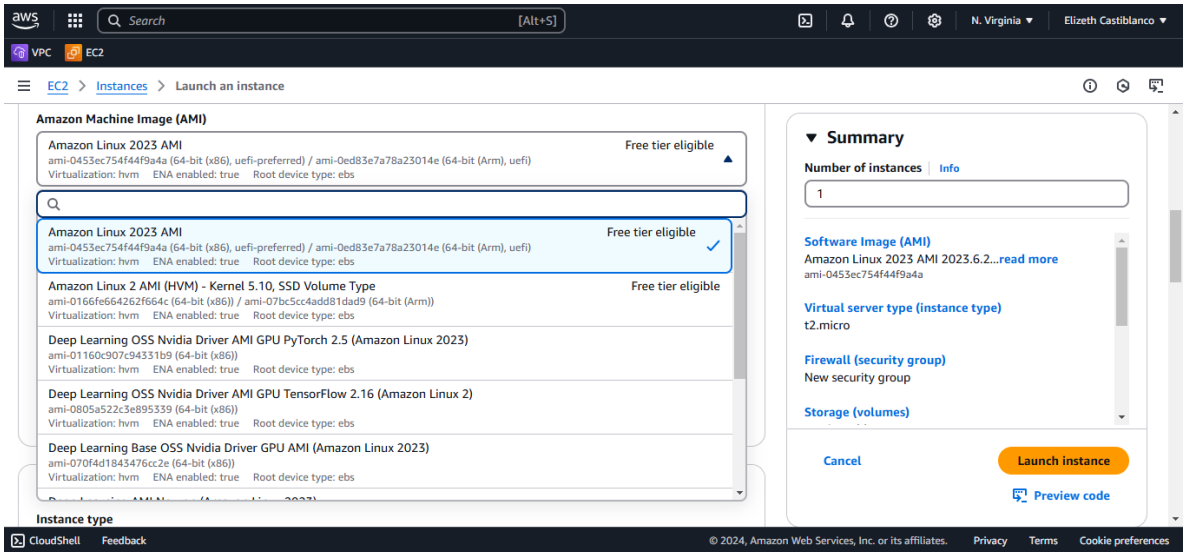


Imagen 11. Elección de tipo de instancia gratuita (recursos de hardware de nuestra máquina virtual).

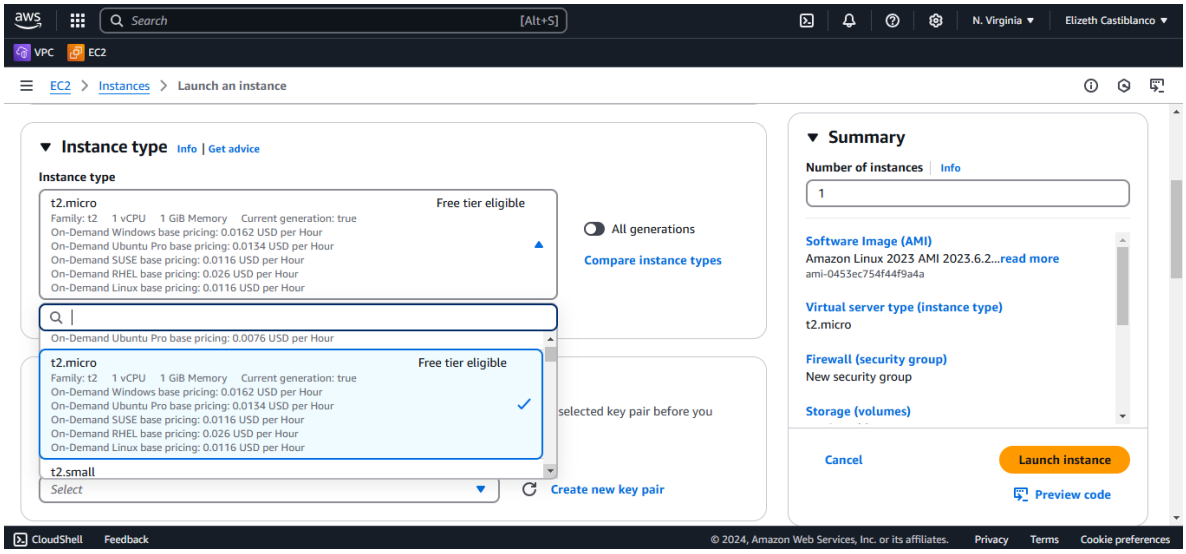


Imagen 12. Creación del Certificado de Seguridad (permite loguear luego de crear la instancia).

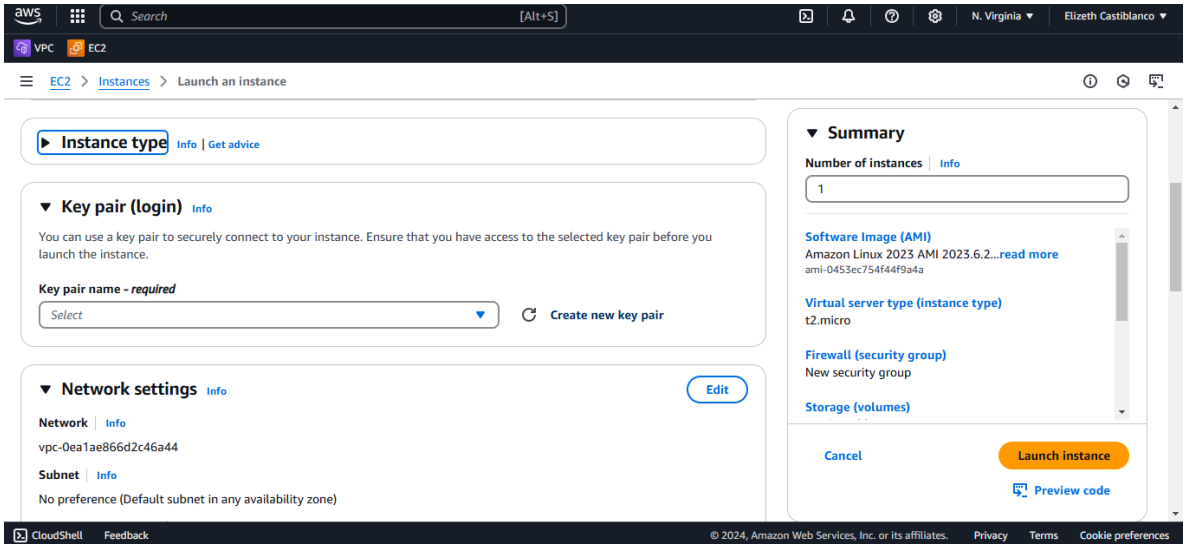


Imagen 13. Selección de tipo de certificado (algoritmo descifrado que se va a utilizar) y formato (PuTTY aplicación ayuda).

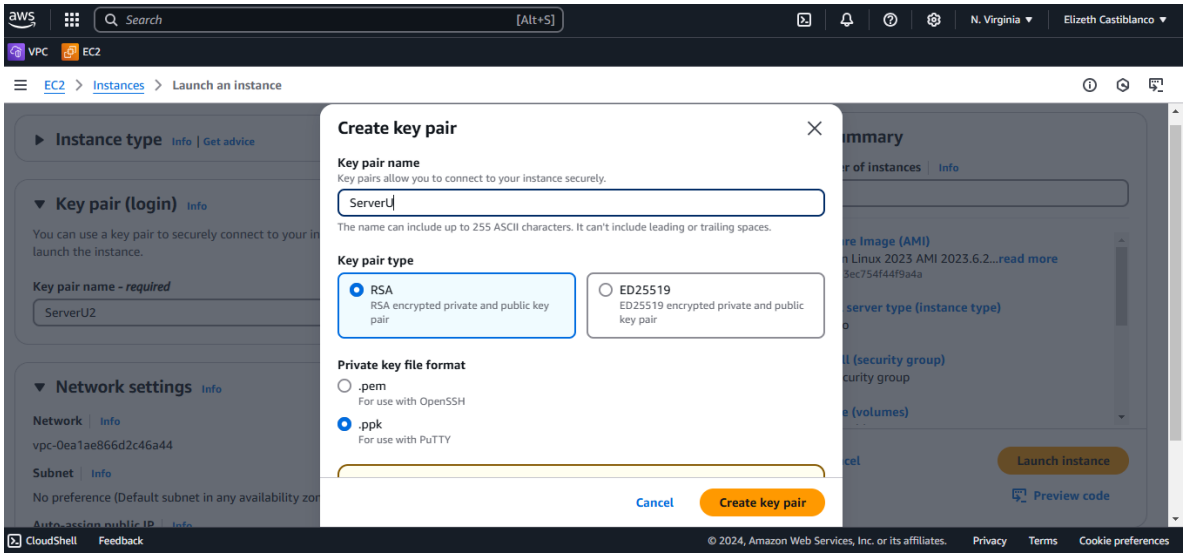


Imagen 14. Configuración de red, elección de VPC creada.

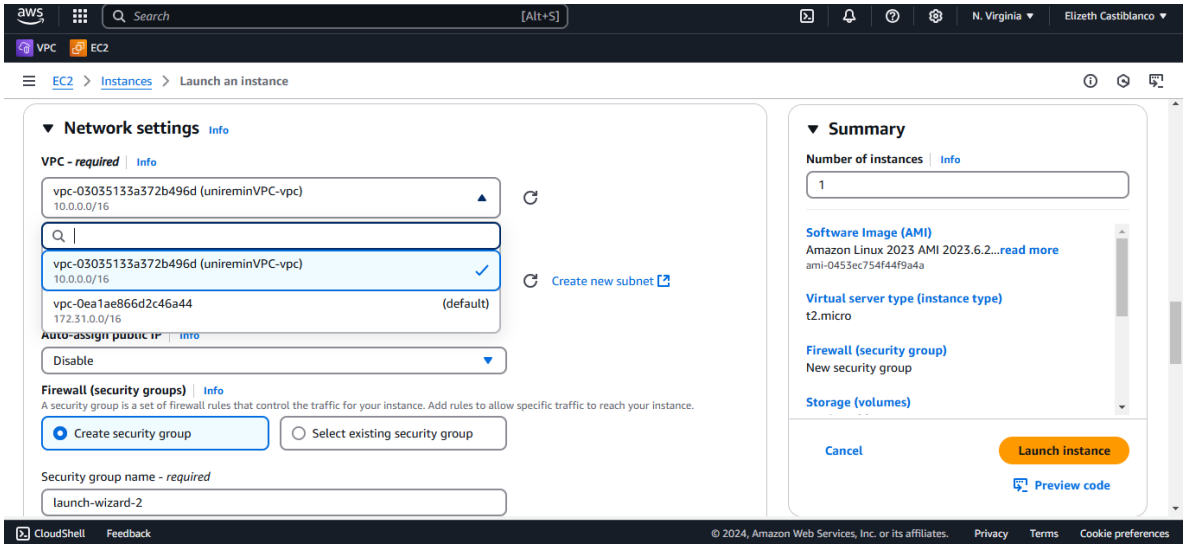


Imagen 15. Elección de Subred (Publica ya que es un servidor web).

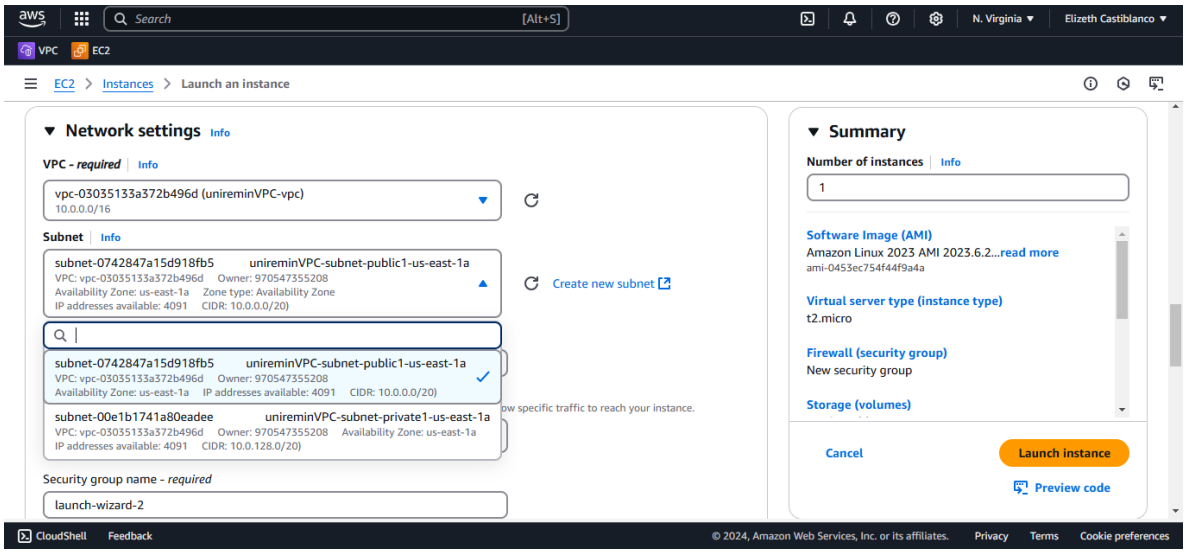


Imagen 16. Habilitar la opción de asignación automática de IP Publica (para que se vea en internet).

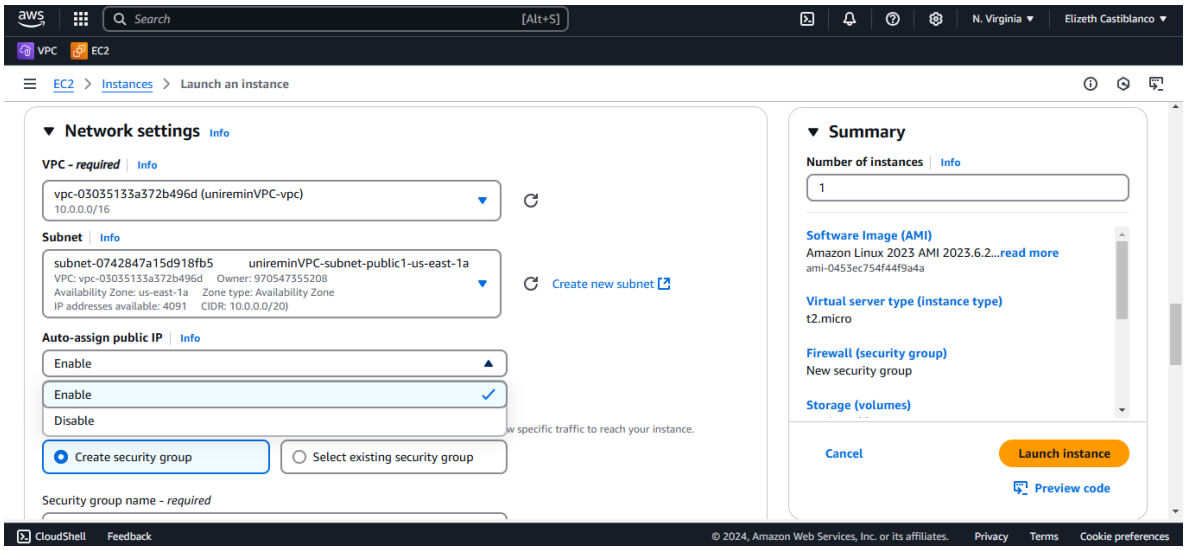


Imagen 17. Asignación de nombre al grupo de seguridad (para mantener las reglas siempre dentro de ese grupo) y las reglas de entrada se dejan por defecto (tipo ssh, puerto 22).

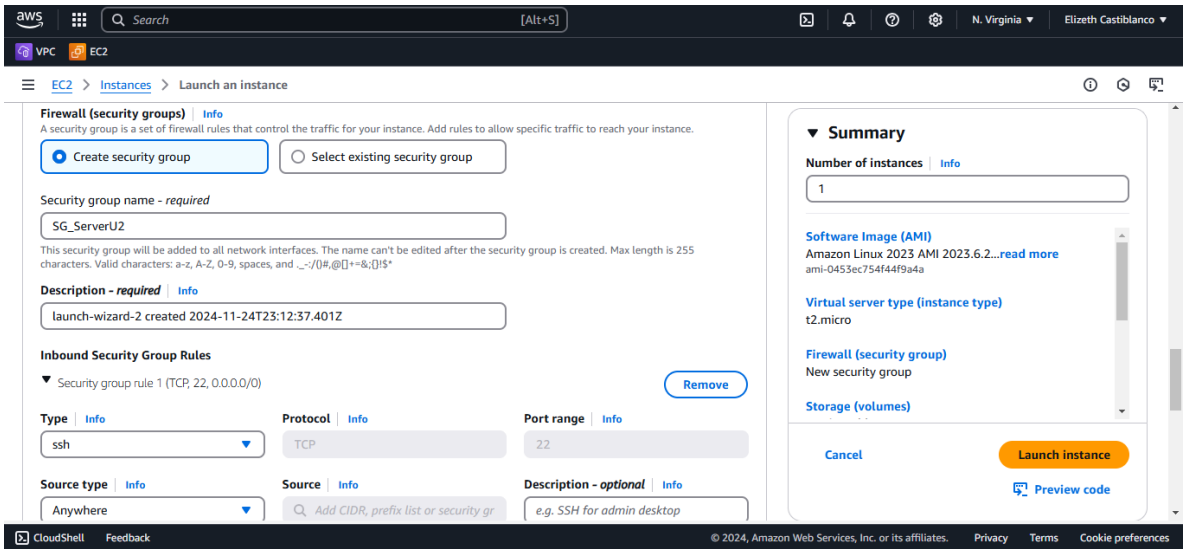


Imagen 18. Características de Almacenamiento se dejan por defecto.

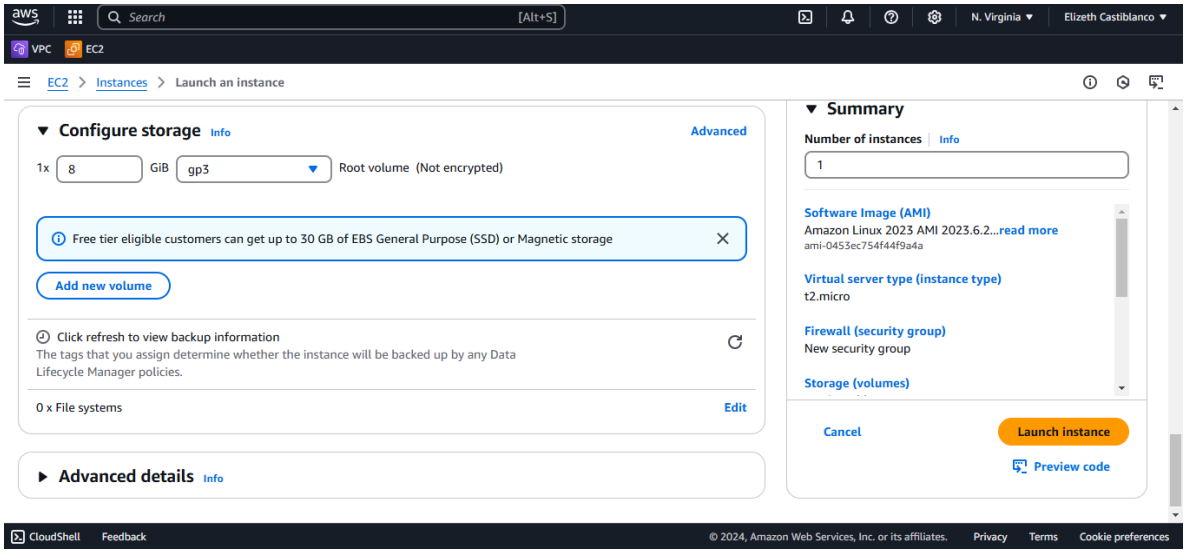


Imagen 19. Lanzamiento de Instancia exitoso.

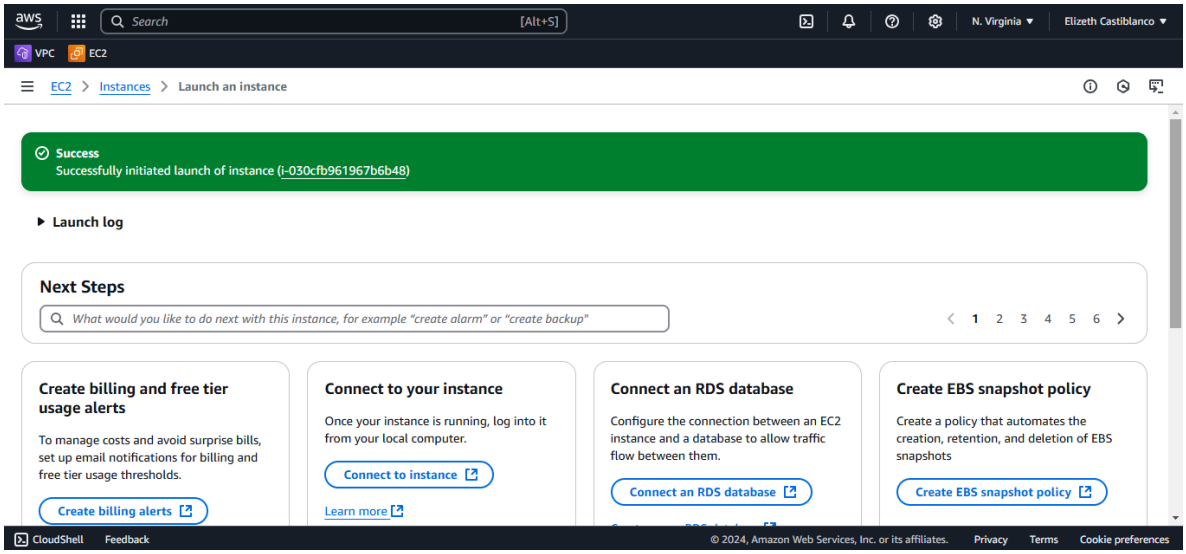


Imagen 20. Estado de la Instancia Inicializando.

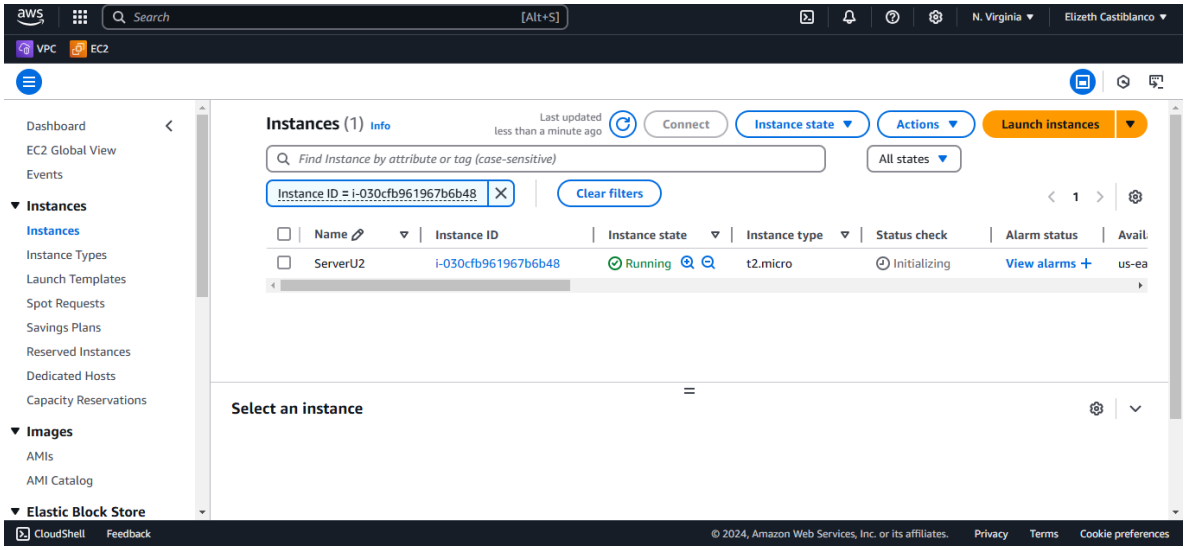


Imagen 21. Información de la Instancia

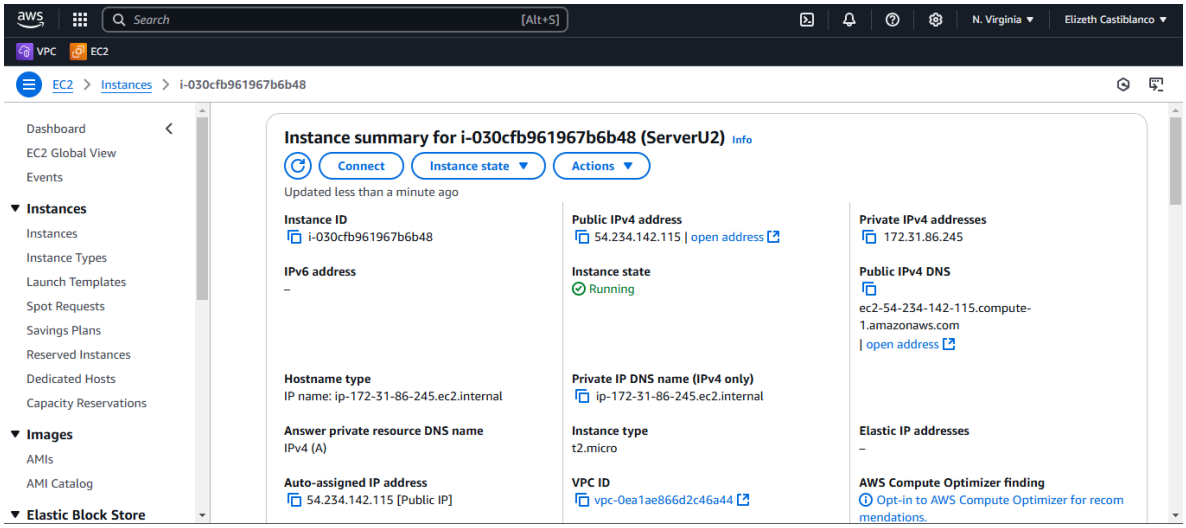


Imagen 22. Instancia Inicializada 2/2 (corriendo)

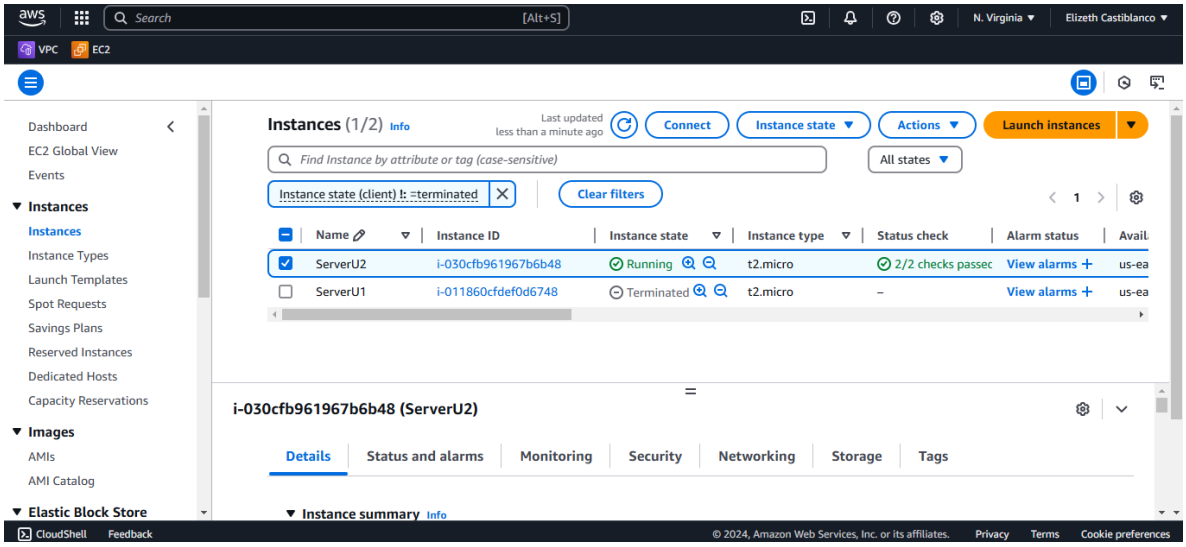


Imagen 23. Conectar Instancia

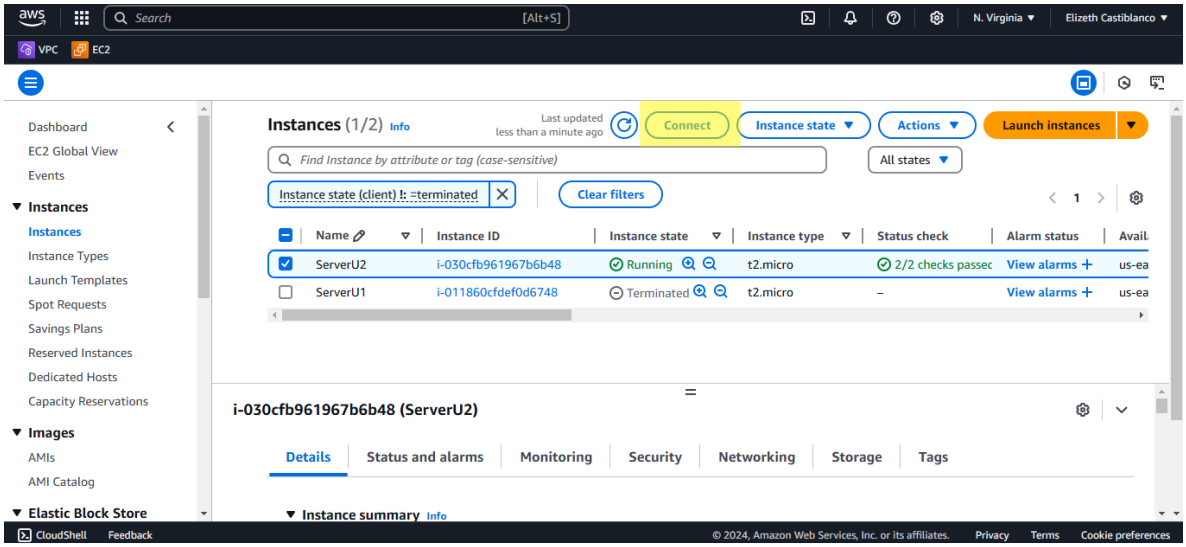


Imagen 24. Copiar el nombre de la máquina para conectarse desde Windows.

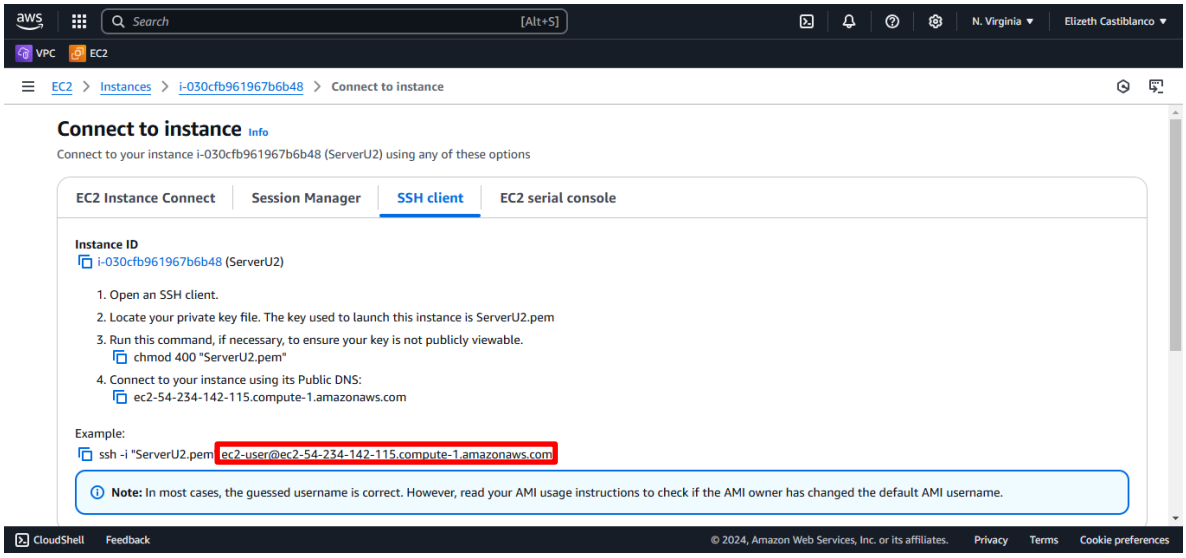


Imagen 25. Usando la aplicación PuTTY copio el nombre de la maquina en Host Name y se deja el puerto 22.

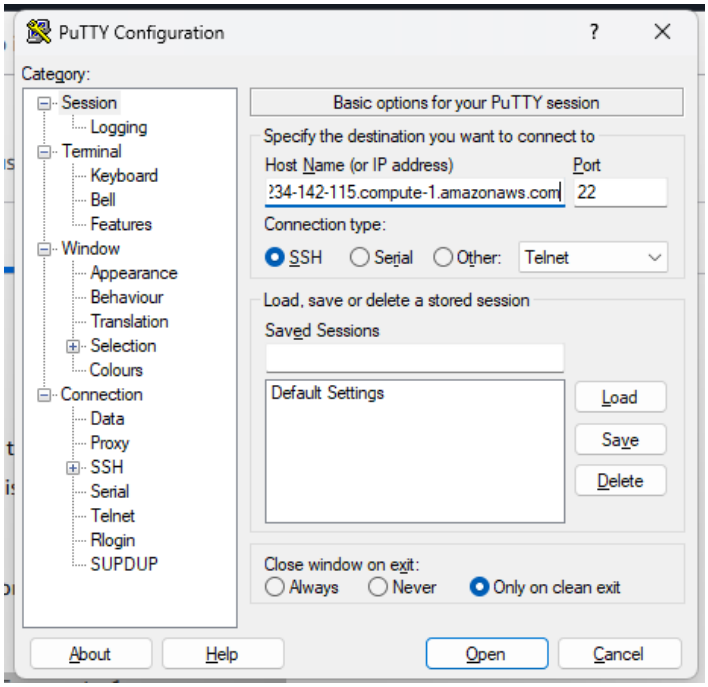


Imagen 26. Opcion SSH, Auth, Credentials. Opcion Private key se carga el certificado descargado anteriormente.

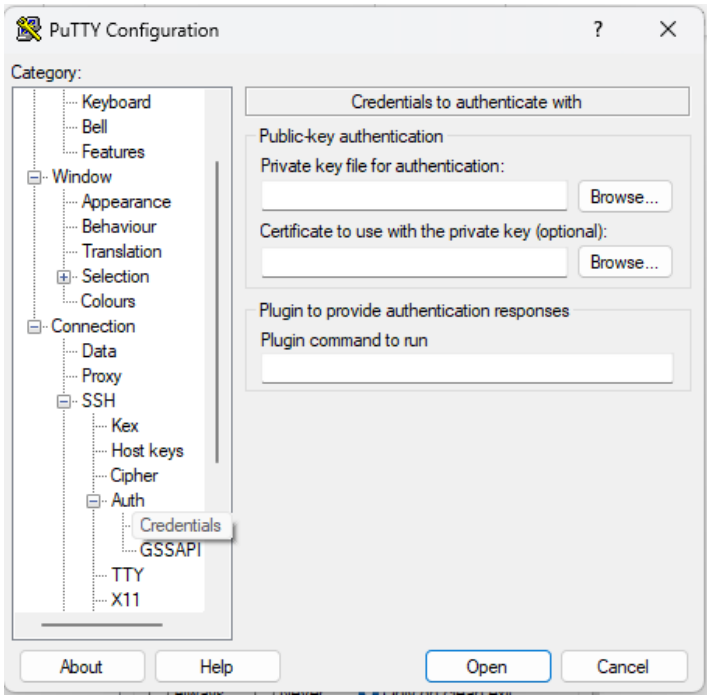


Imagen 27. Cargar el certificado de seguridad descargado.

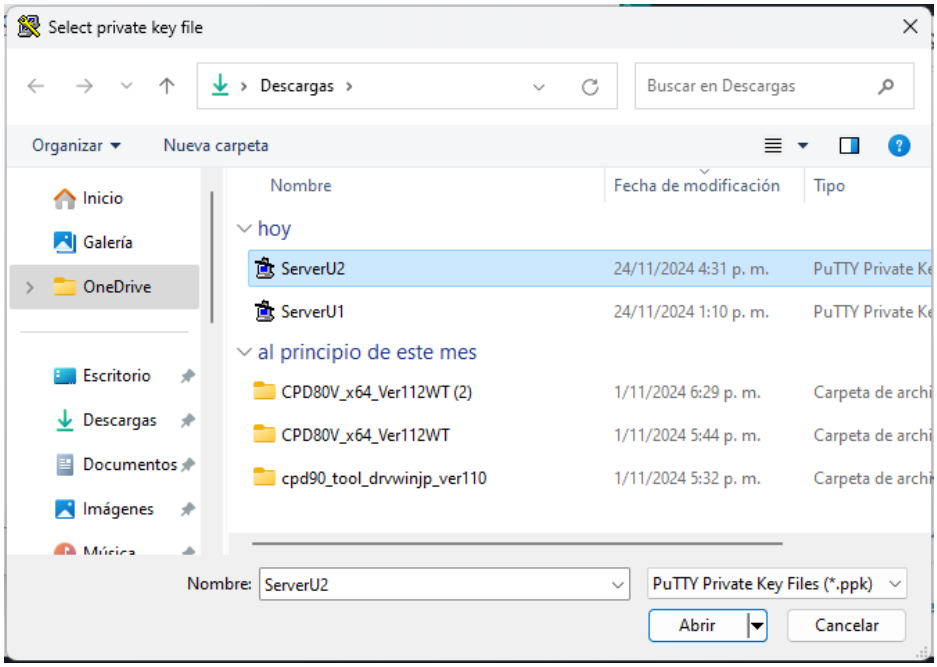


Imagen 28. Aceptar alerta de seguridad.

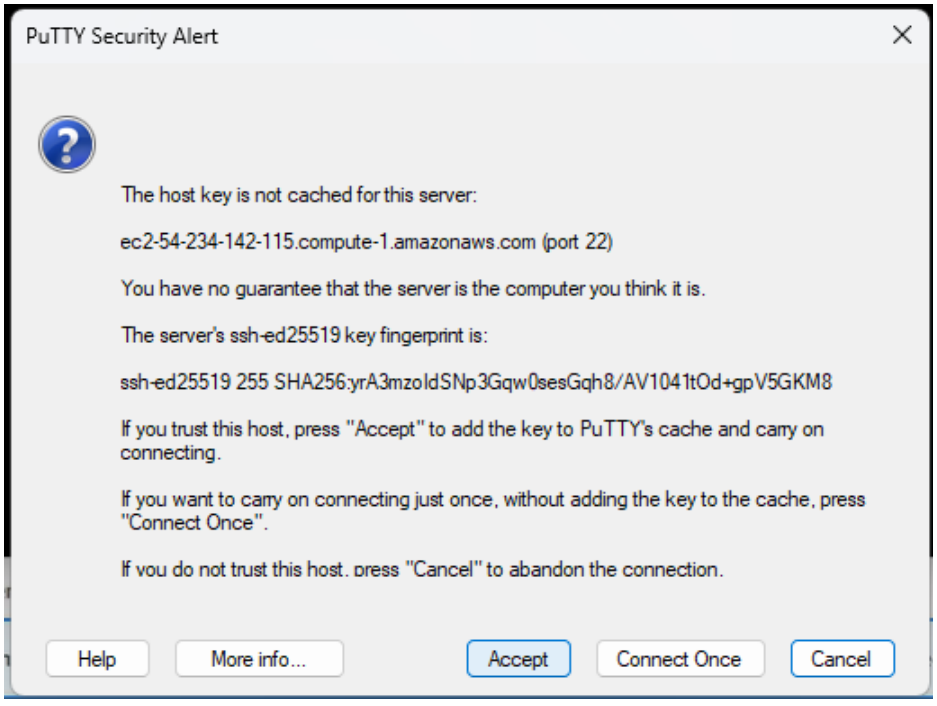


Imagen 29. Acceso a Linux a través de la consola.

Imagen 30. Permisos de Administrador (Para loguear como administrador).

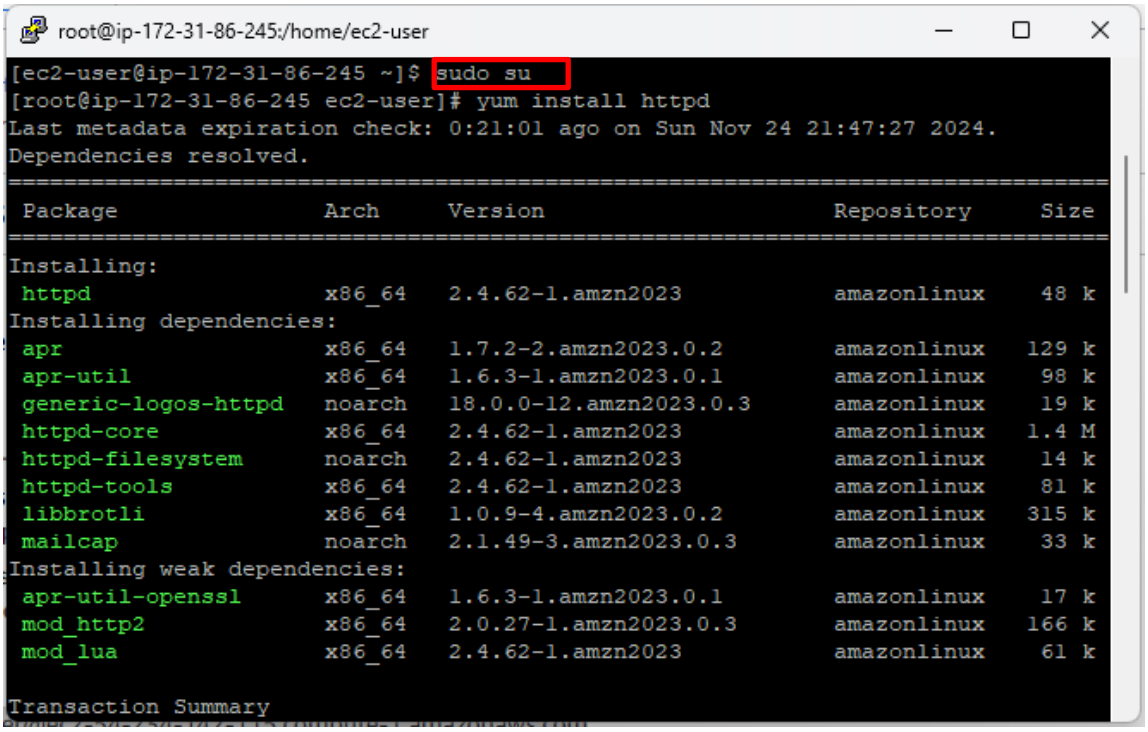


Imagen 31. Instalación de aplicación Apache (yum: gestor de paquetes de Amazon Linux, install: comando de la orden, httpd: apache.

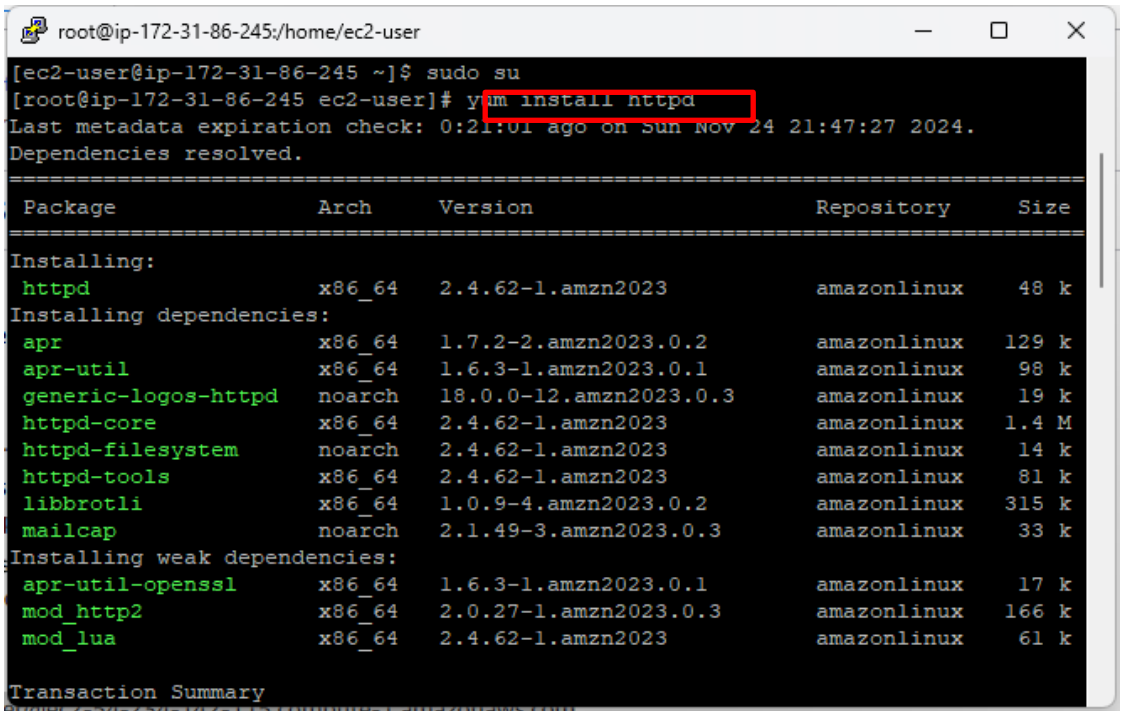


Imagen 32. Instalar paquete de 2.3 M

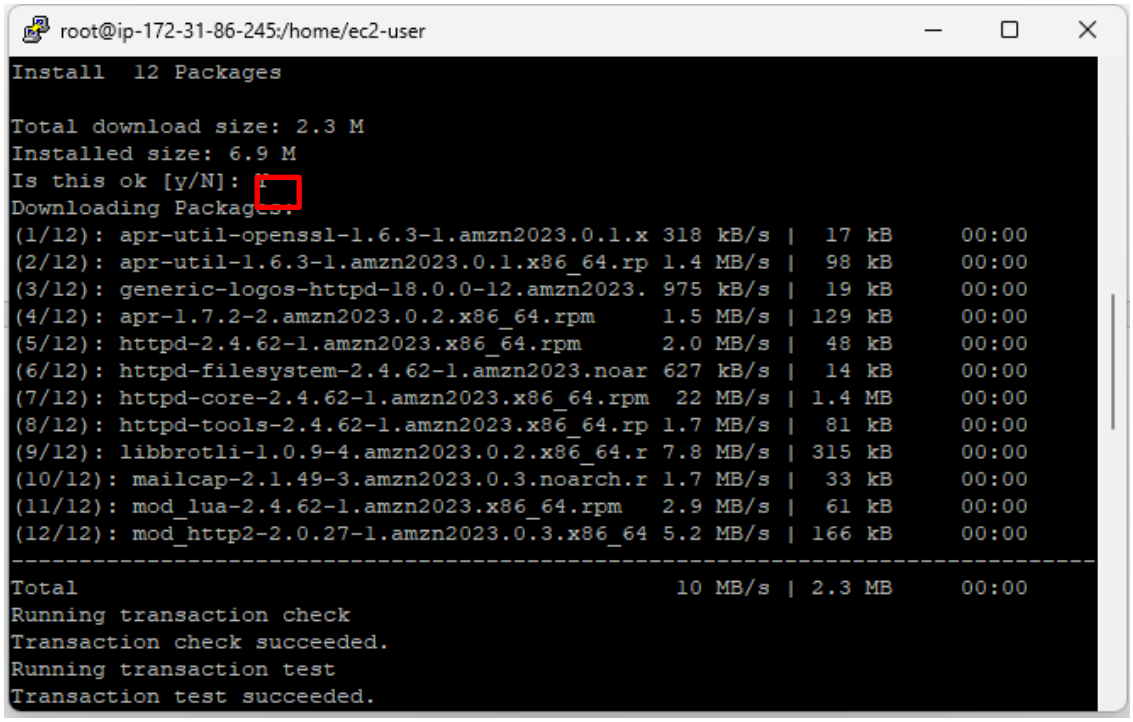


Imagen 33. Consultar el estado del servicio.

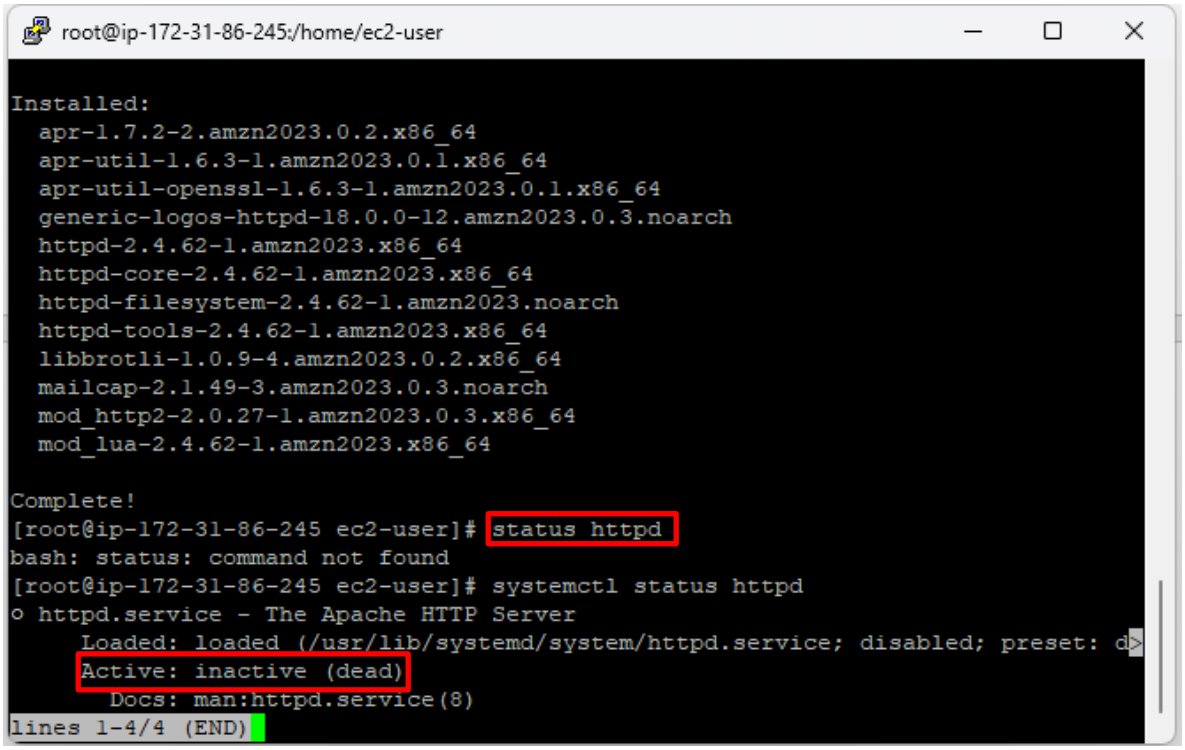


Imagen 34. Activar el estado del servicio y consultarlo nuevamente (debe queda corriendo).

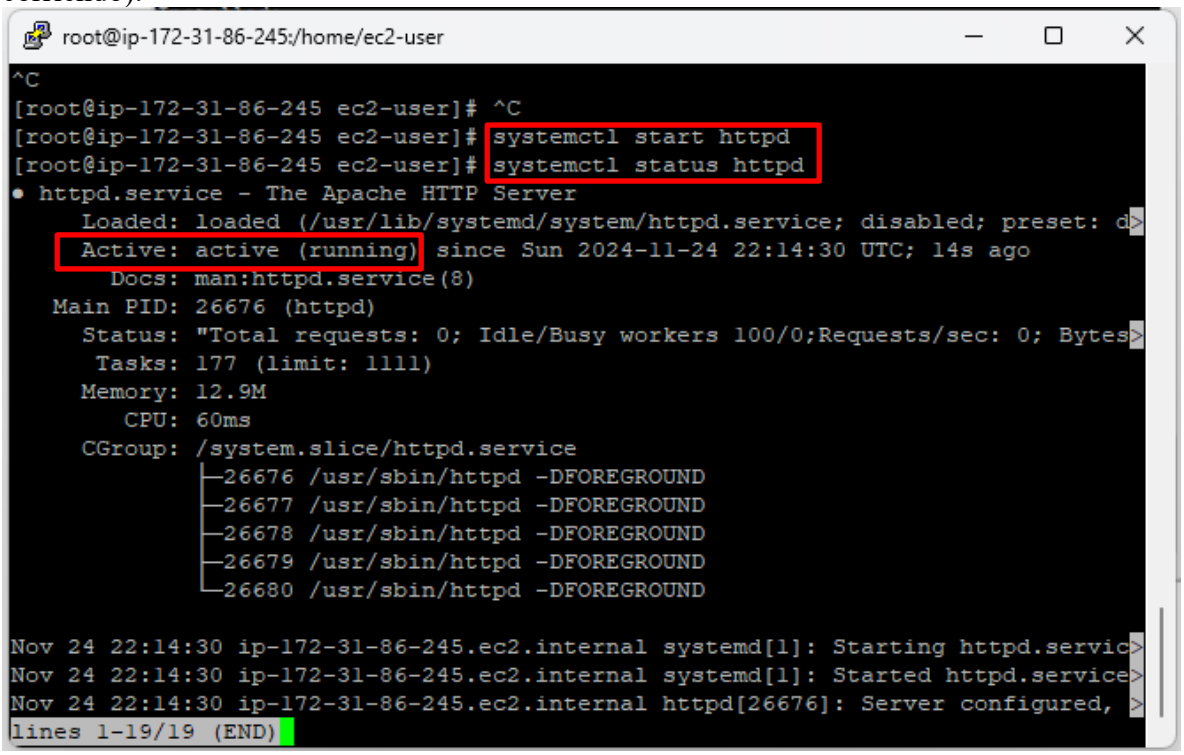


Imagen 35. Editar y agregar regla de seguridad en la configuración de la Instancia

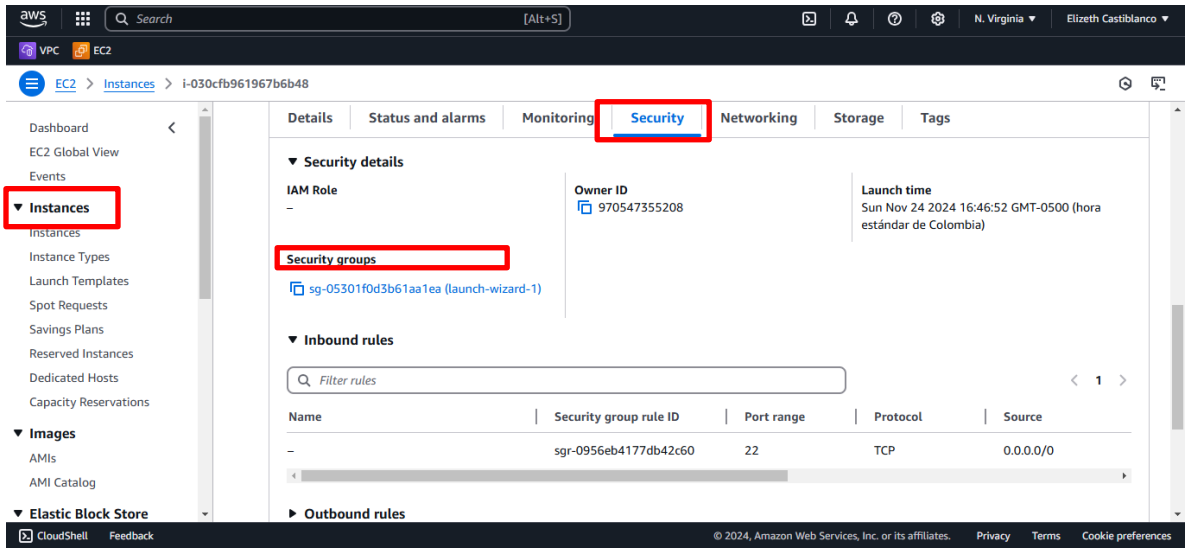


Imagen 36. Agregar regla, colocar número de puerto y desde donde quiero aceptar que se conecten al servicio web (en este caso Todos (0.0.0.0/0)).

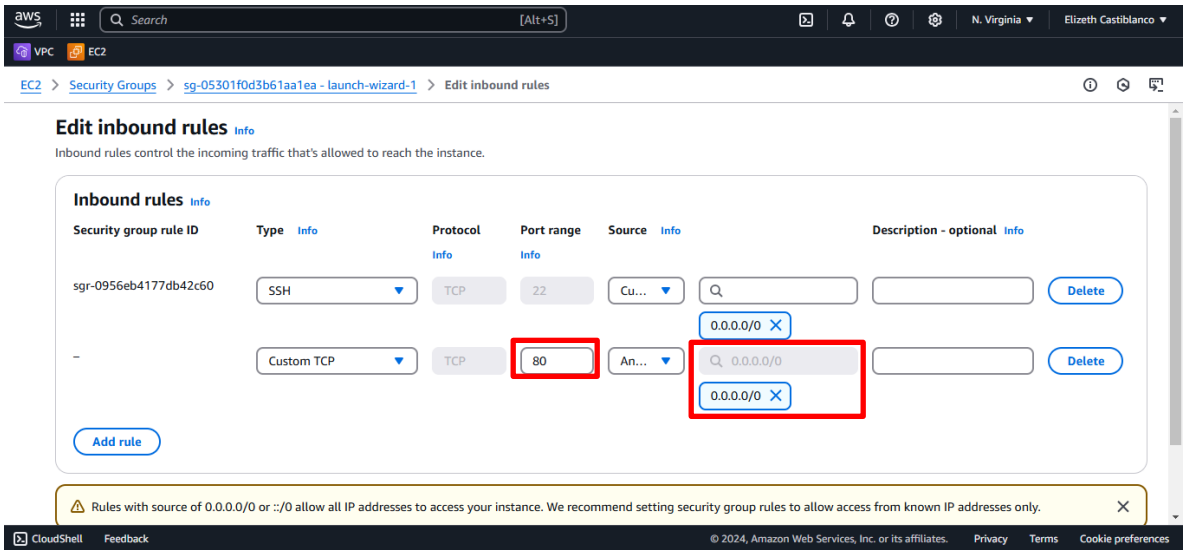


Imagen 37. Reglas de seguridad actualizadas.

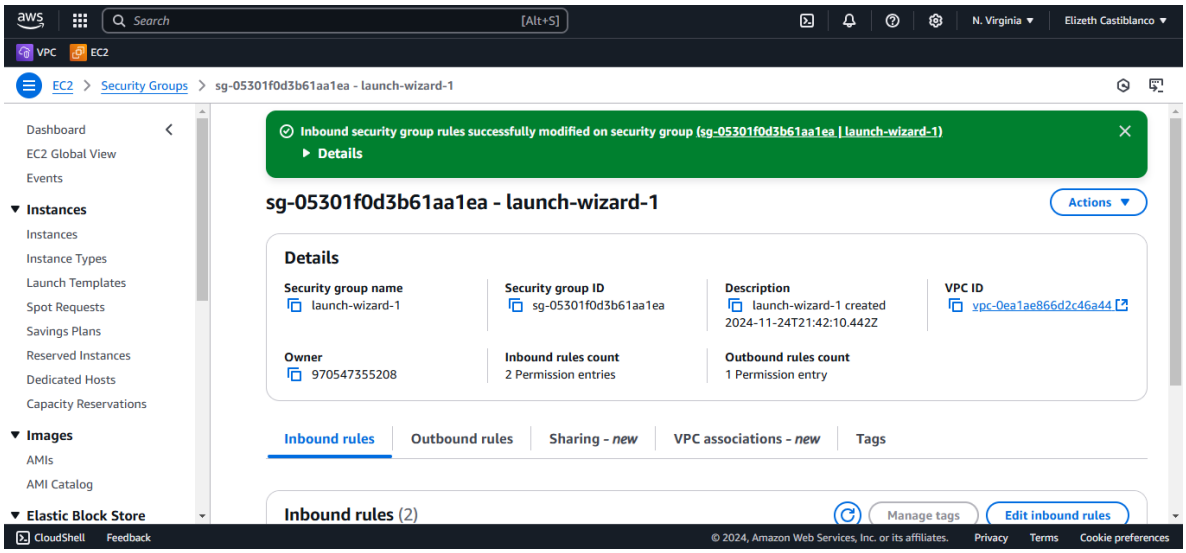


Imagen 38. Copiar la IPv4 asignada en la información de la instancia.

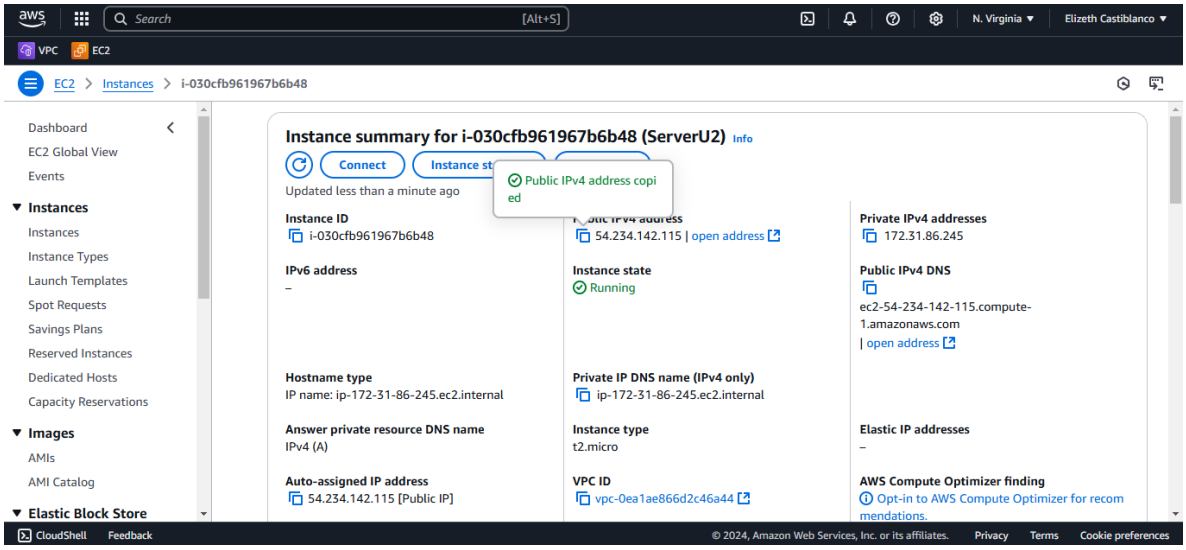
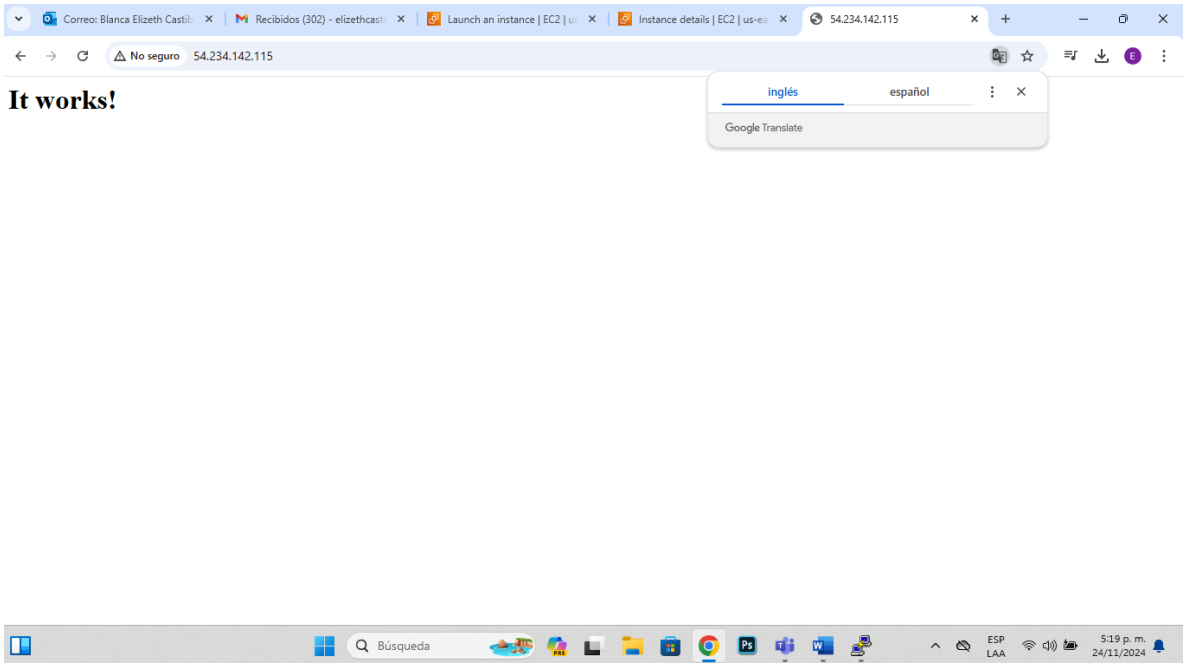


Imagen 39. Pegar la IP en el navegador, ejecutar y ya estaría corriendo correctamente el servidor web.



Video de explicación Servidor Web de Amazon Linux en AWS

https://youtu.be/56m_94_09QM

Entrega II

Imagen 1. Crear Snapshot a partir del volumen de la instancia creada (es una copia del disco duro).

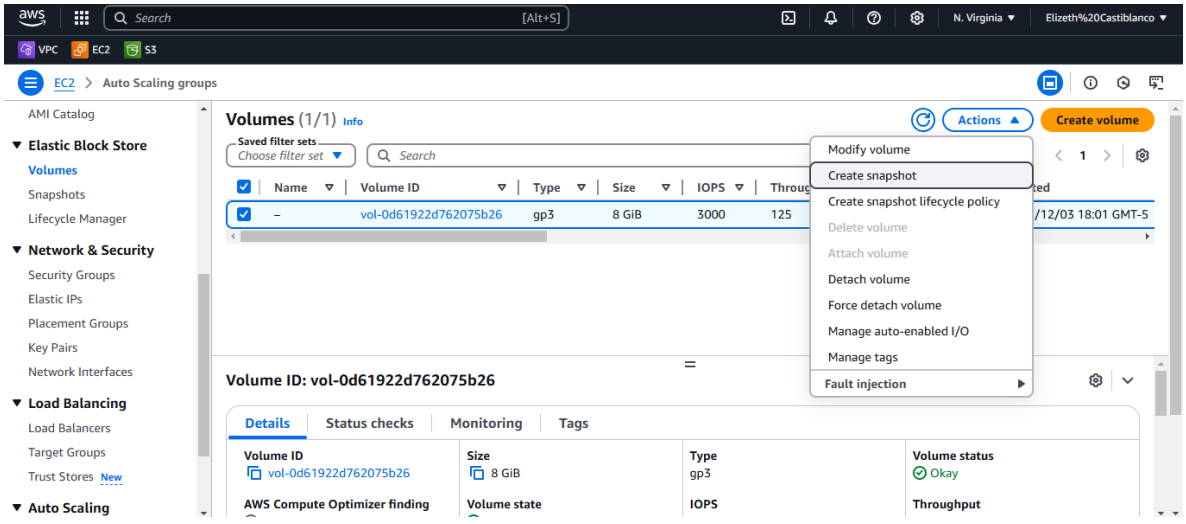


Imagen 2. Asignación de nombre y la configuración que viene por defecto se deja

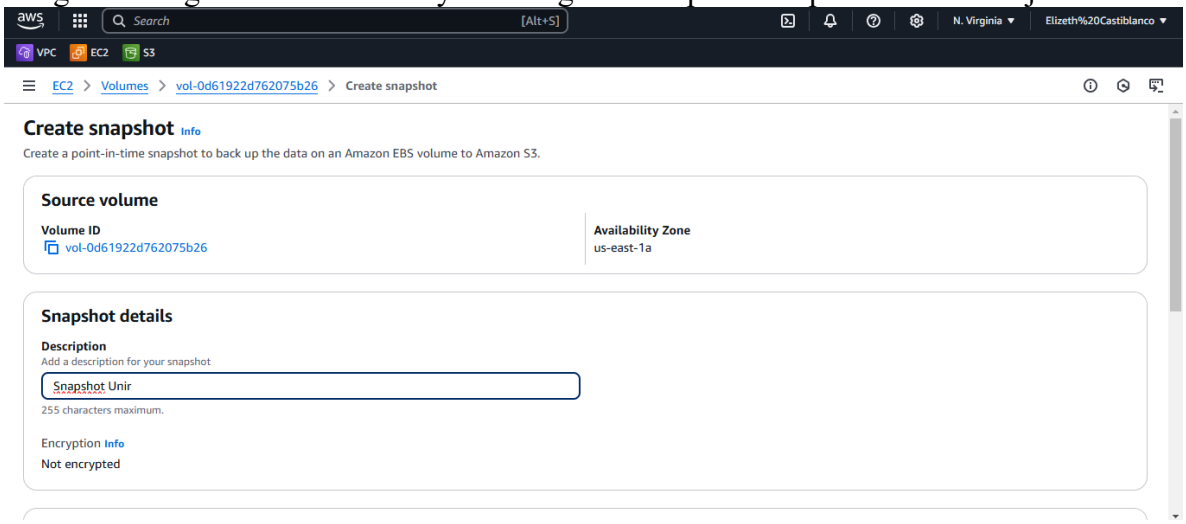


Imagen 3. Snapshot creado

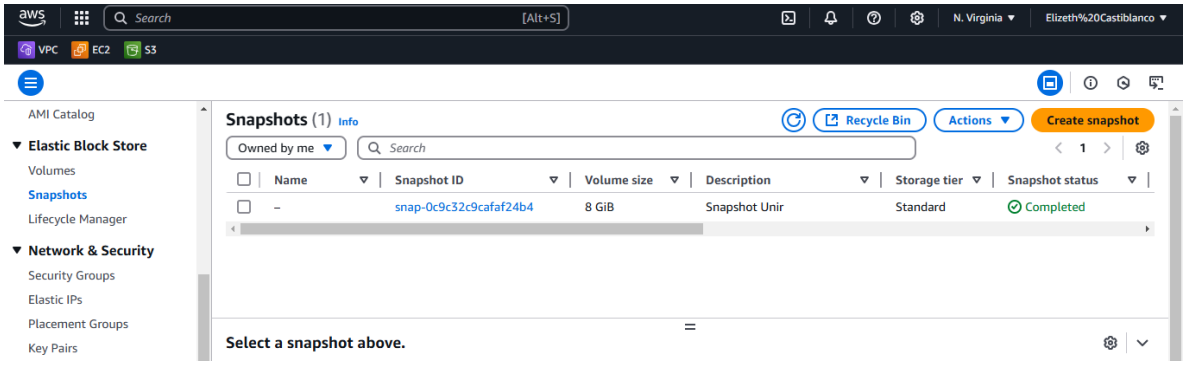


Imagen 4. Creación de etiquetas en volúmenes (Ayuda a filtrar rápidamente información).

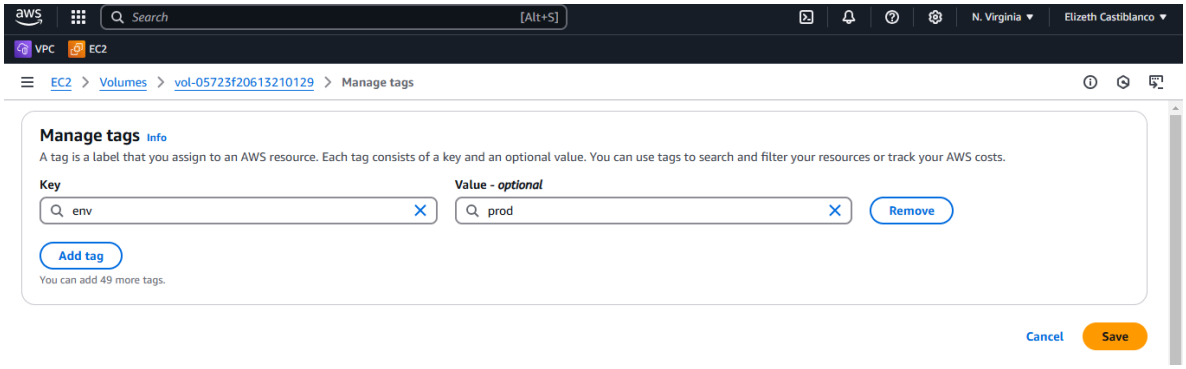


Imagen 5. Creación de AMI (imagen personalizada) a partir del snapshot creado anteriormente (Cuando se crean nuevas instancias, se puede seleccionar esa imagen para ahorrarse la configuración web del servidor).

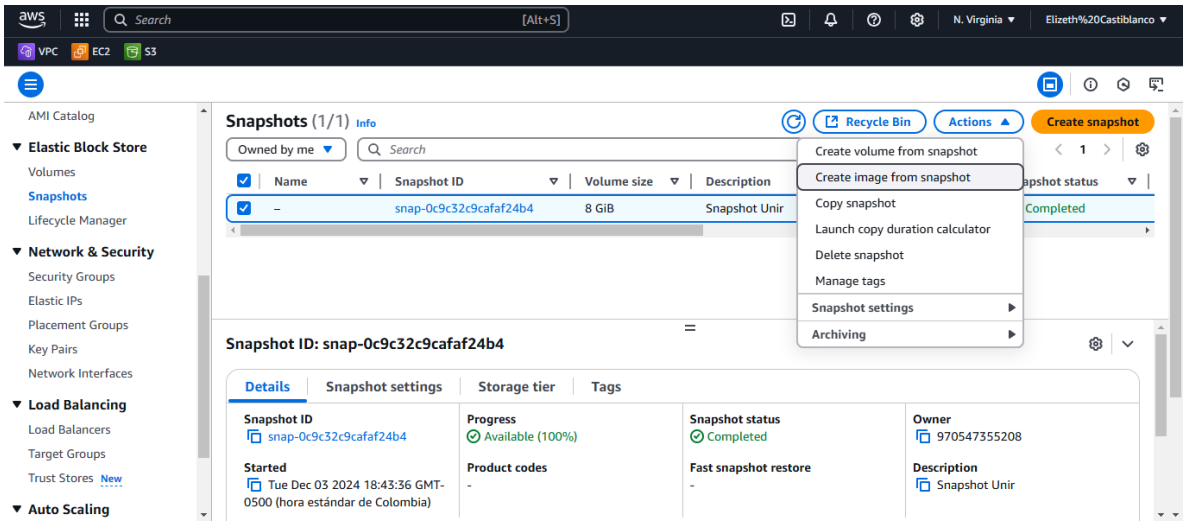


Imagen 6. Asignación de nombre y la configuración que viene por defecto se deja

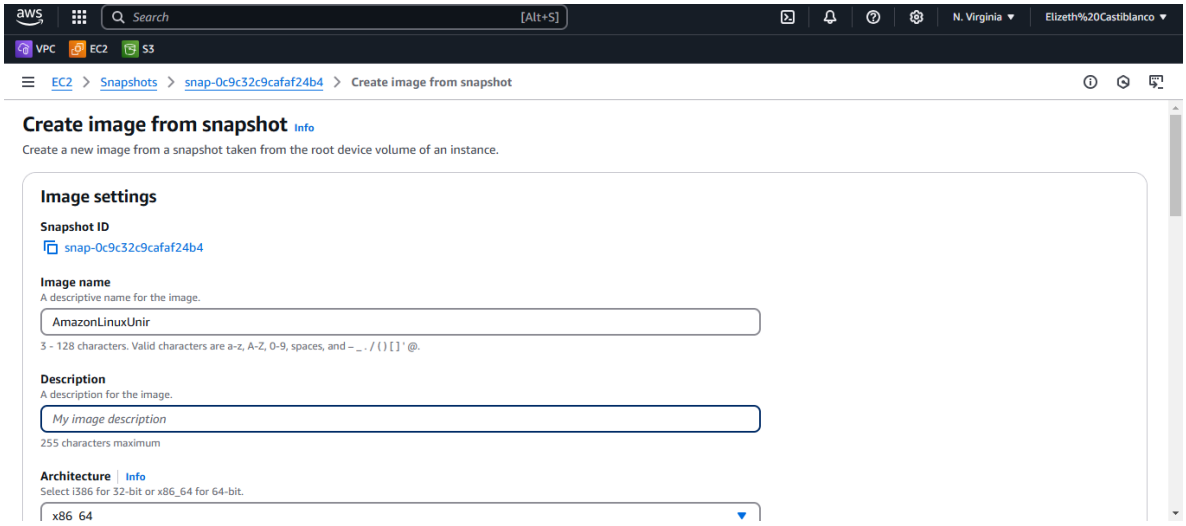


Imagen 7. AMI creada

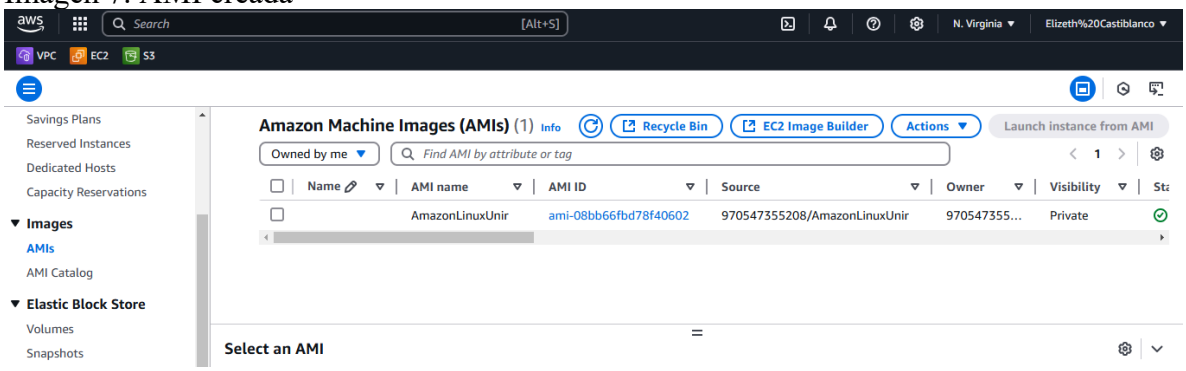


Imagen 8. Crear la primera instancia usando la AMI creada

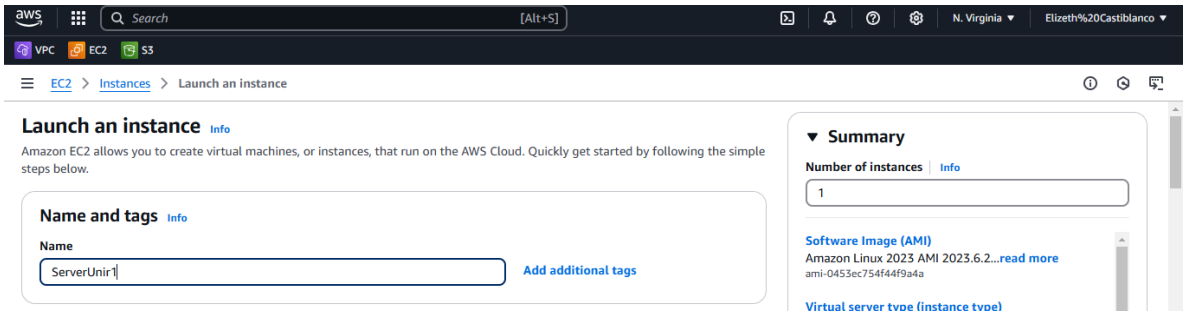


Imagen 9. Se elige la AMI creada

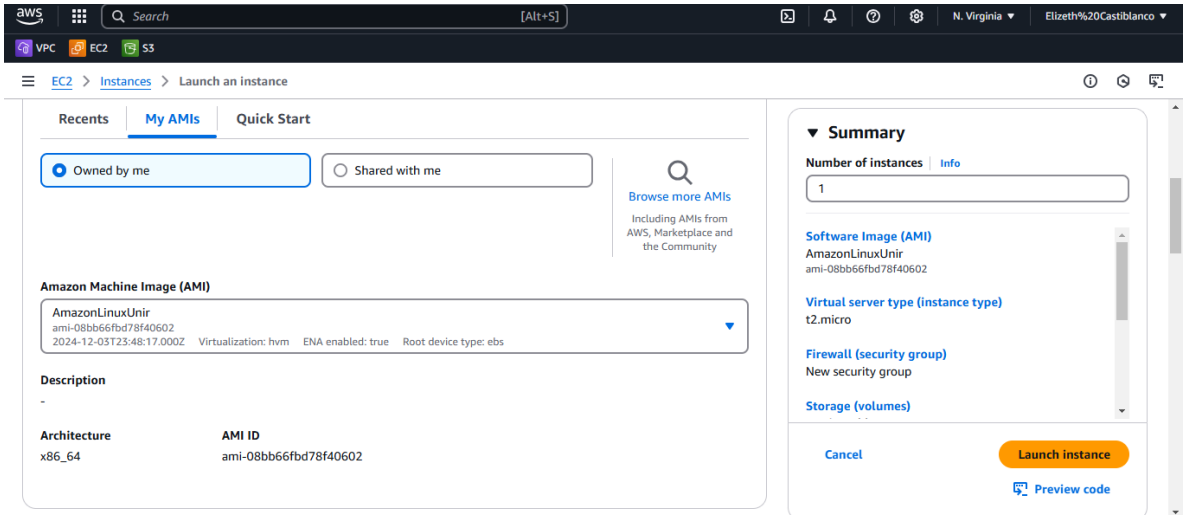


Imagen 10. Se elige el almacenamiento y certificado de seguridad

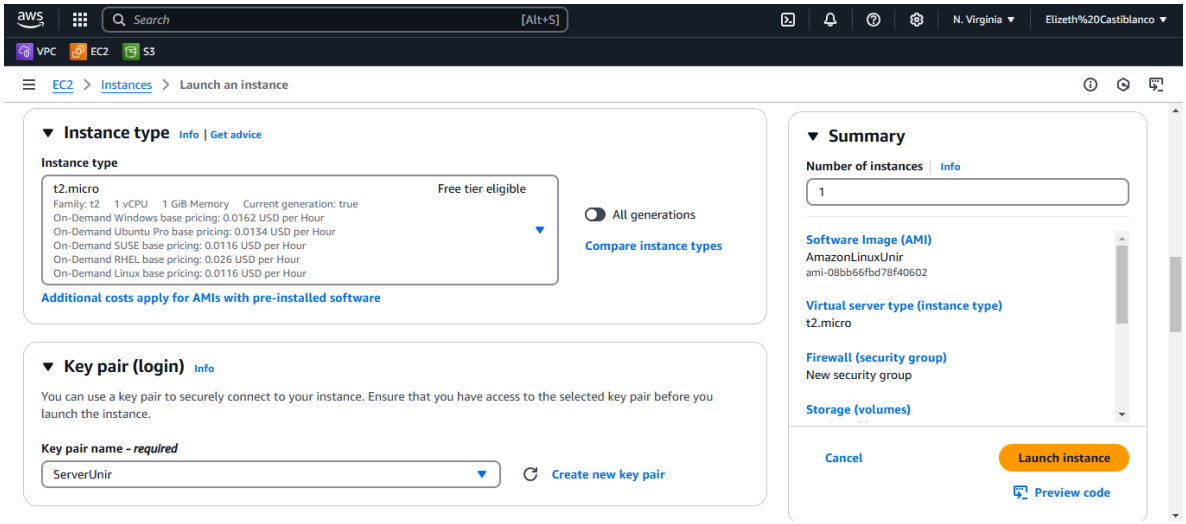


Imagen 11. Se elige la misma VPC, la subred publica y se activa la asignación de IP pública.

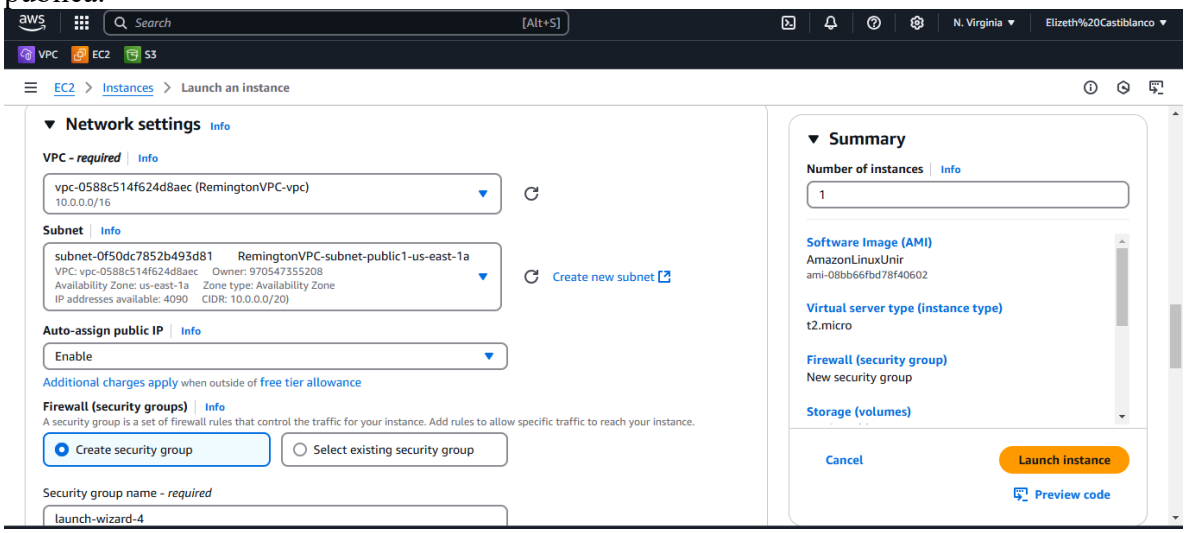


Imagen 12. Se crea una regla de seguridad con el puerto 80 y las demás configuraciones que vienen por defecto se dejan

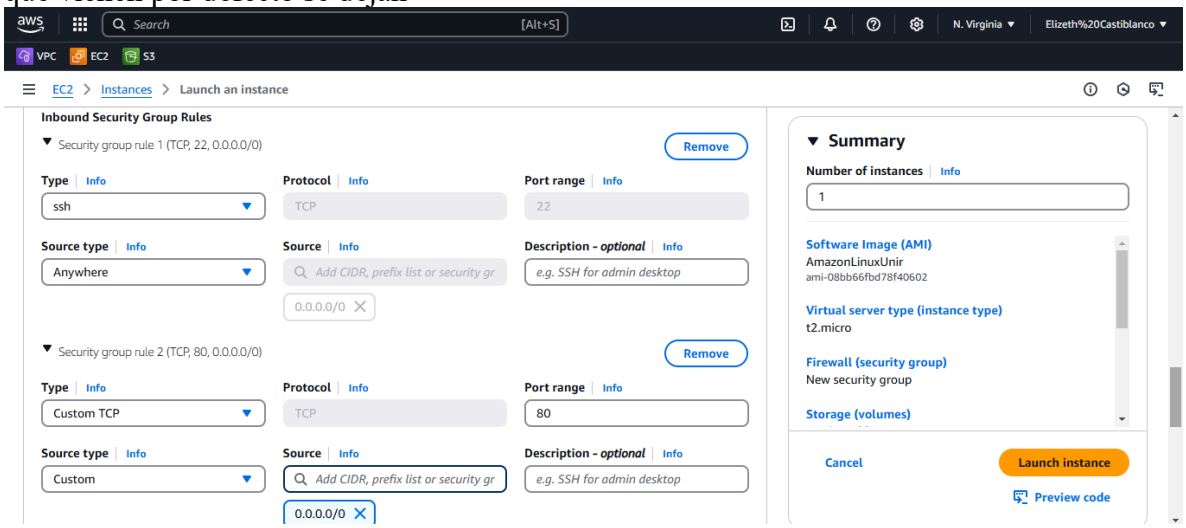


Imagen 13. Instancia inicializando

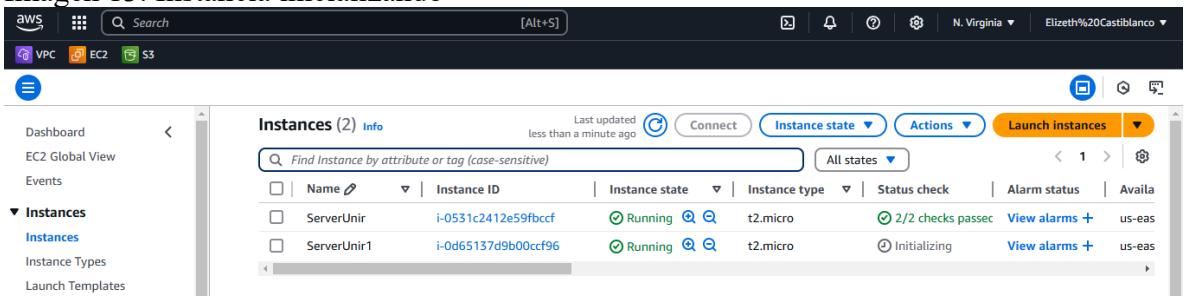


Imagen 14. Instancia en estado 2/2

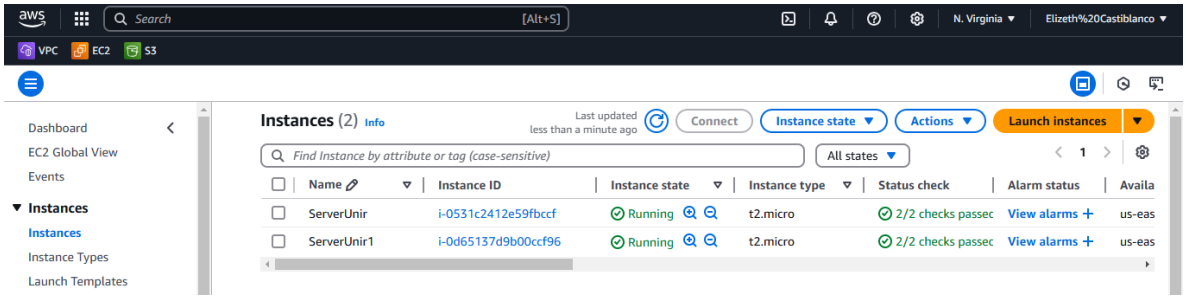


Imagen 15. Ejecución del servidor con éxito

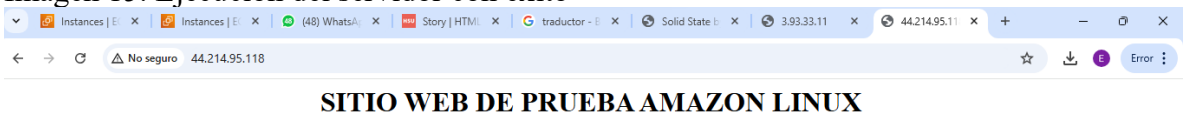


Imagen 16. Creación de segunda instancia con la misma configuración de la anterior

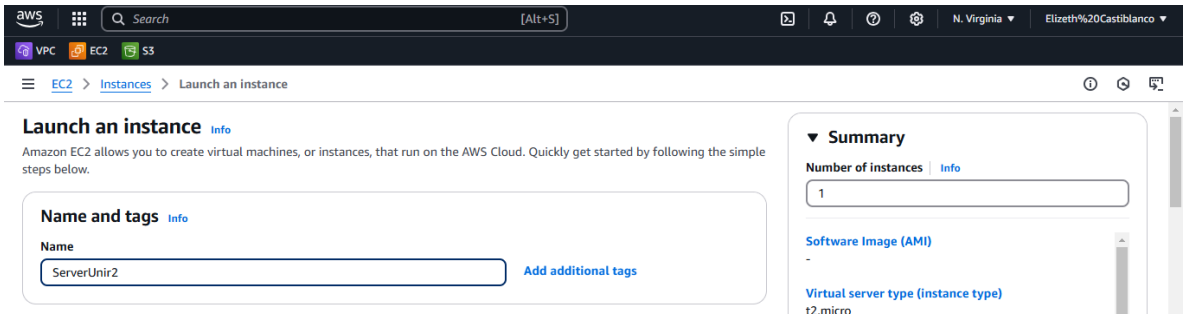


Imagen 17. Instancia inicializando

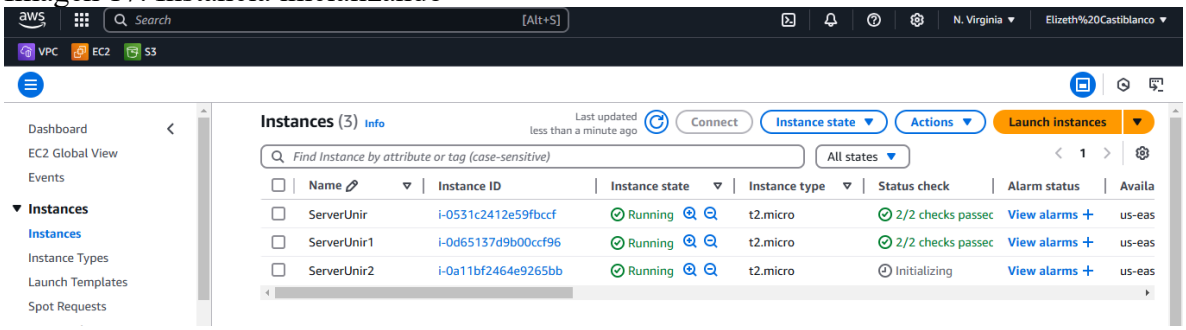


Imagen 18. Instancia en estado 2/2

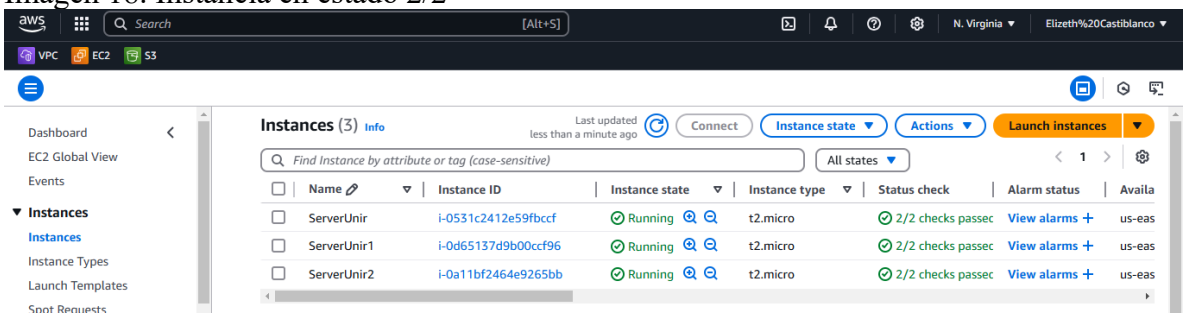


Imagen 19. Ejecución del servidor con éxito

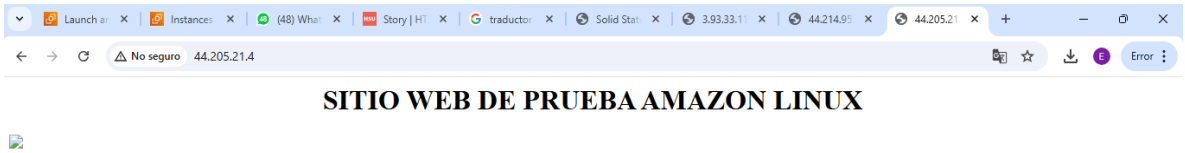


Imagen 20. Crear Balanceador de carga

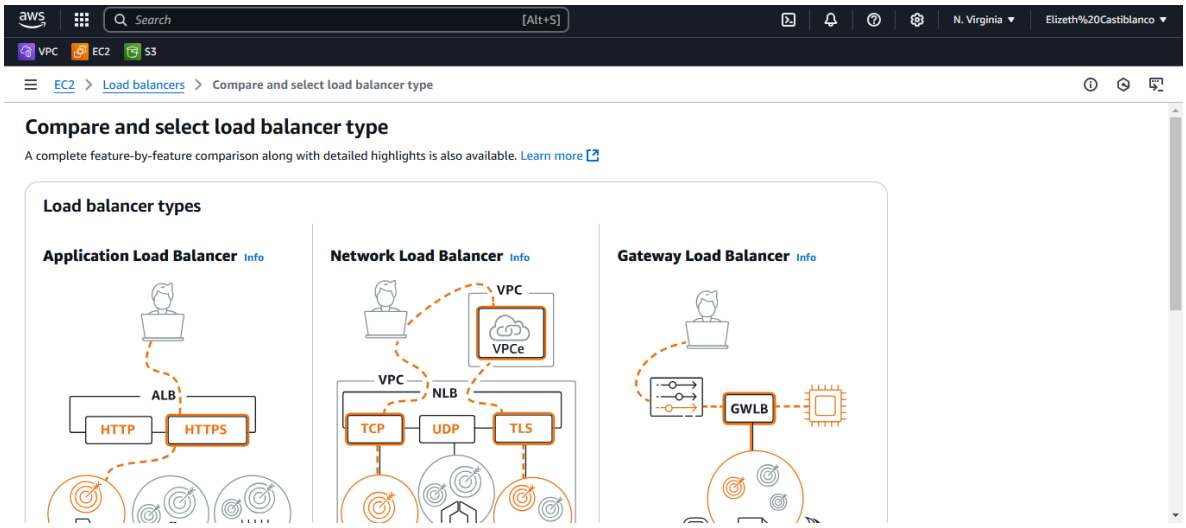


Imagen 21. Asignación de nombre

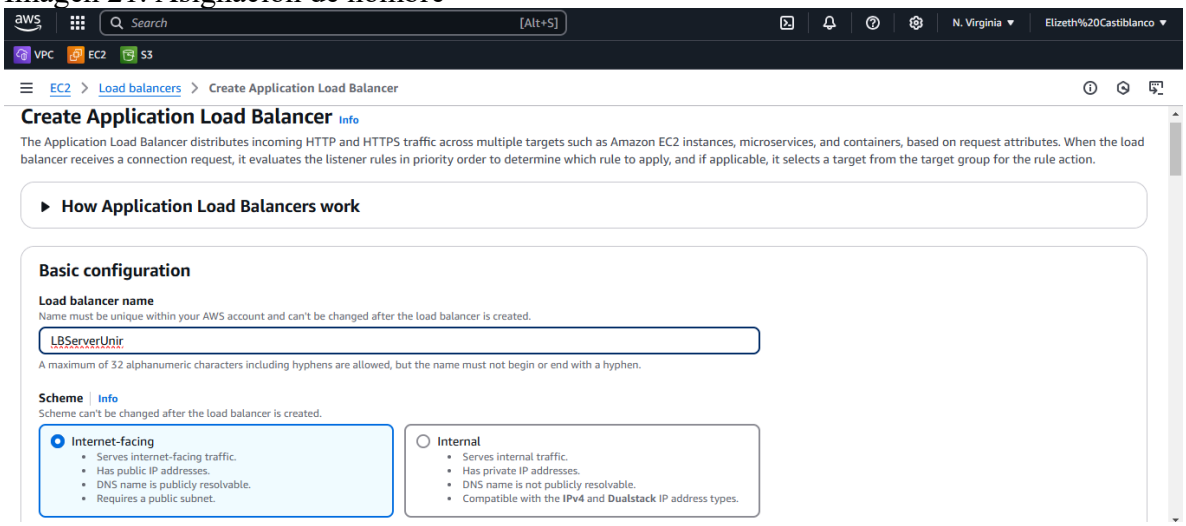


Imagen 22. Configuración de IP y VPC

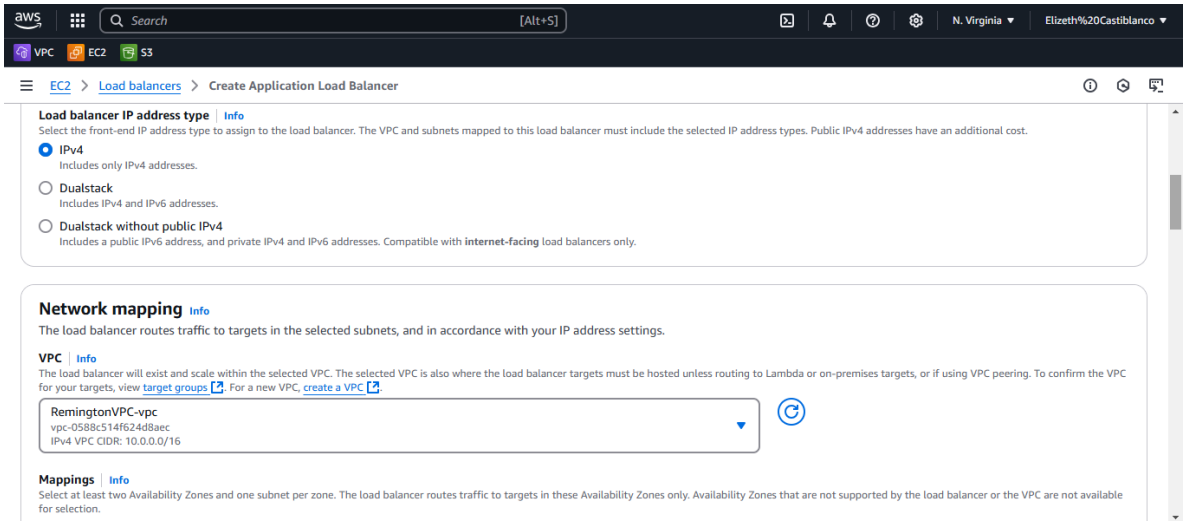


Imagen 23. Elección de zonas de disponibilidad y redes públicas (mínimo dos)

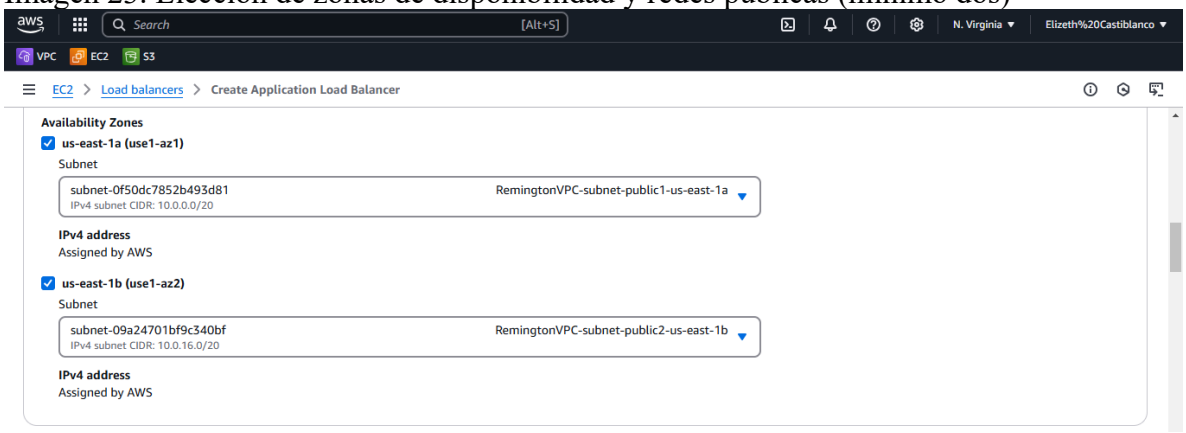


Imagen 24. Crear grupo de seguridad independiente para el balanceador de carga

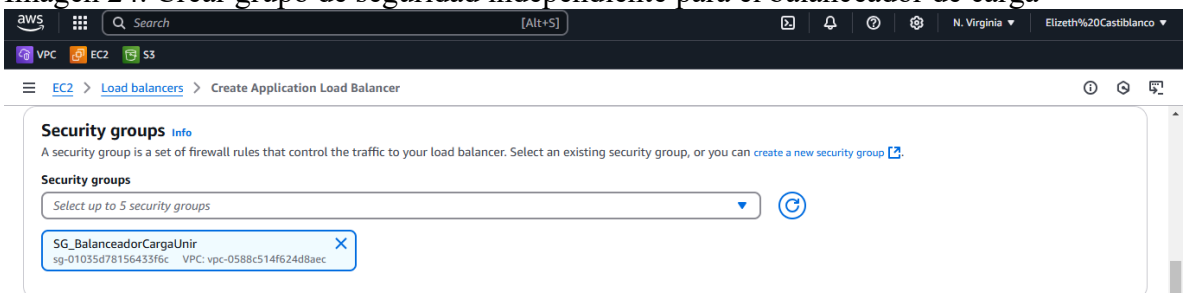


Imagen 25. Asignación de reglas (puertos desde donde se va a escuchar y para donde salen las instrucciones)

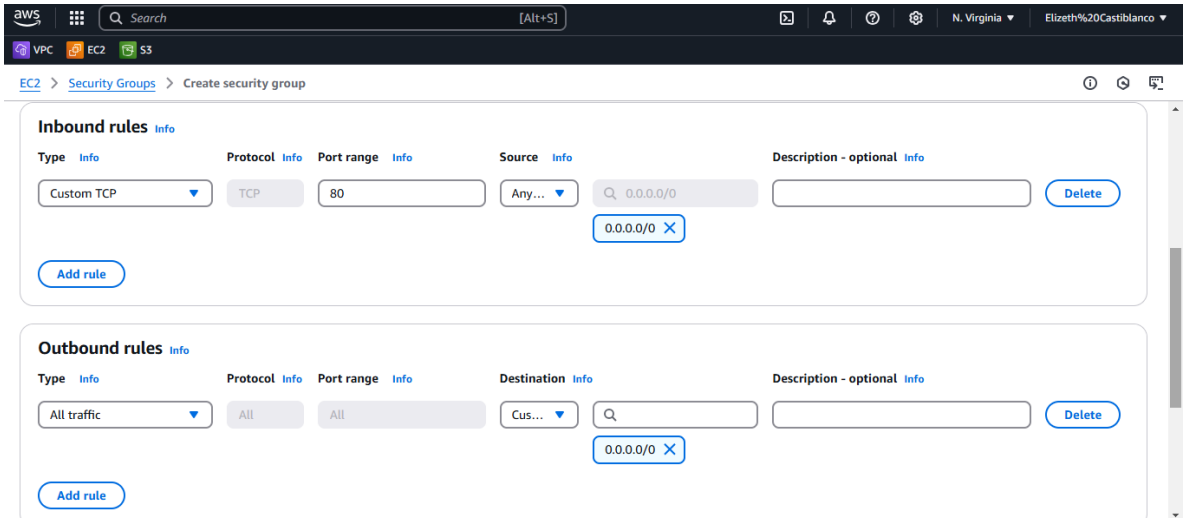


Imagen 26. Creación de grupo de objetivos-Target Group (cuando ingresan las peticiones se envían a los objetivos o máquinas de destino). Se elige la misma VPC y se deja la demás configuración que viene por defecto.

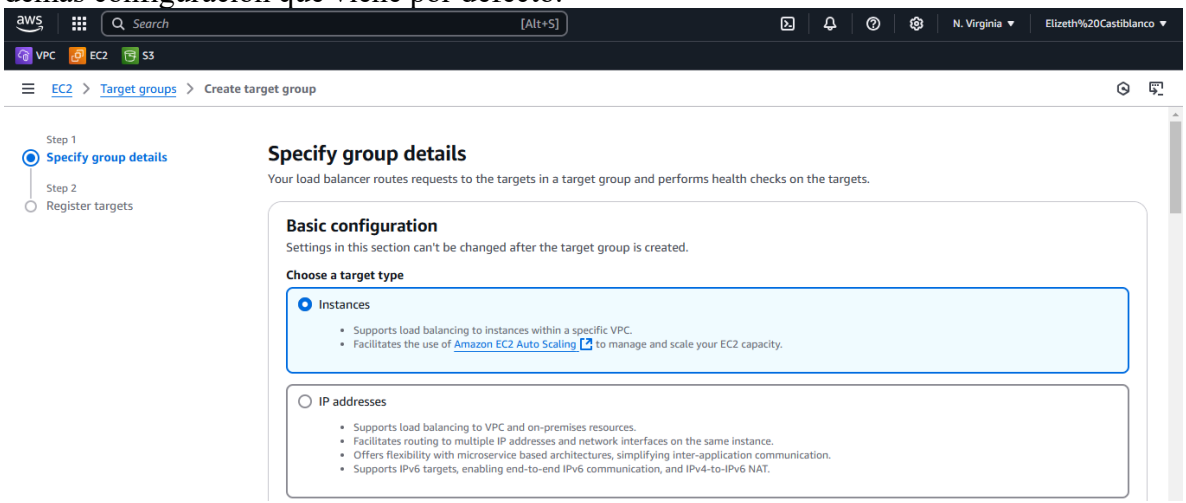


Imagen 27. Configuración del grupo de objetivos-Target Group

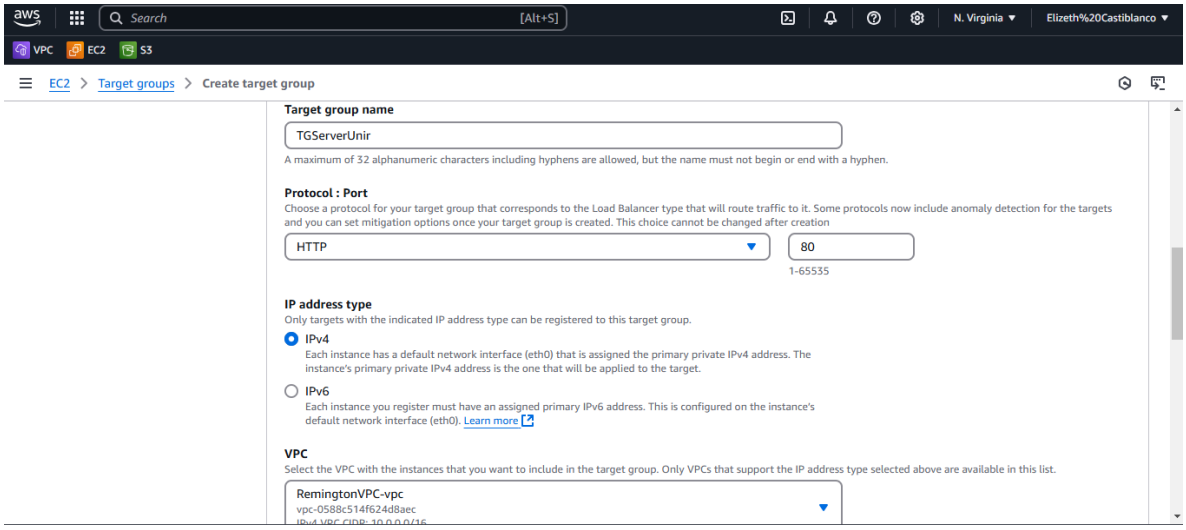


Imagen 28. Configuración del grupo de objetivos-Target Group

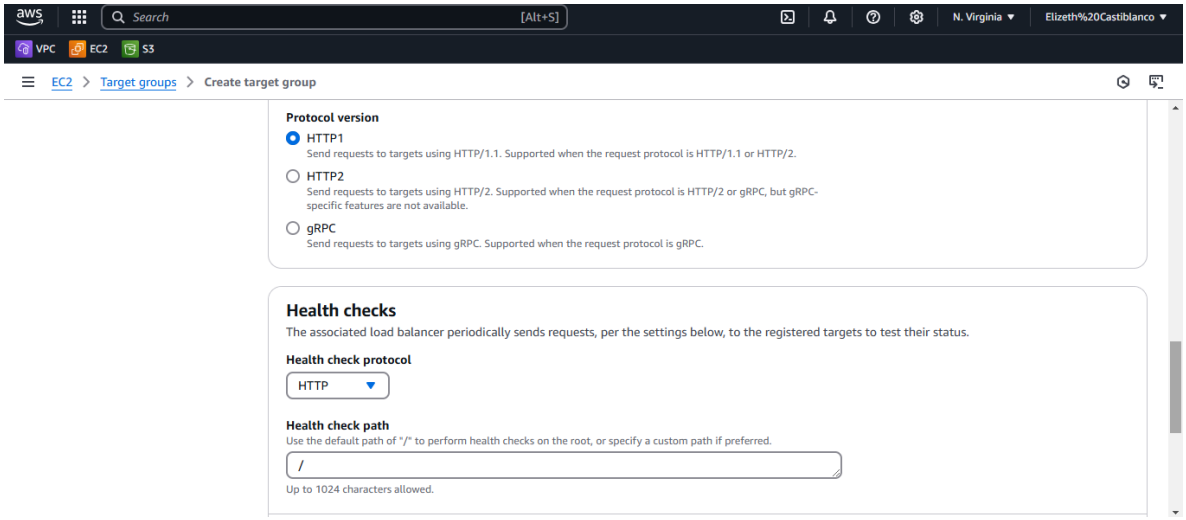


Imagen 29. Se agregan las instancias al grupo de objetivos

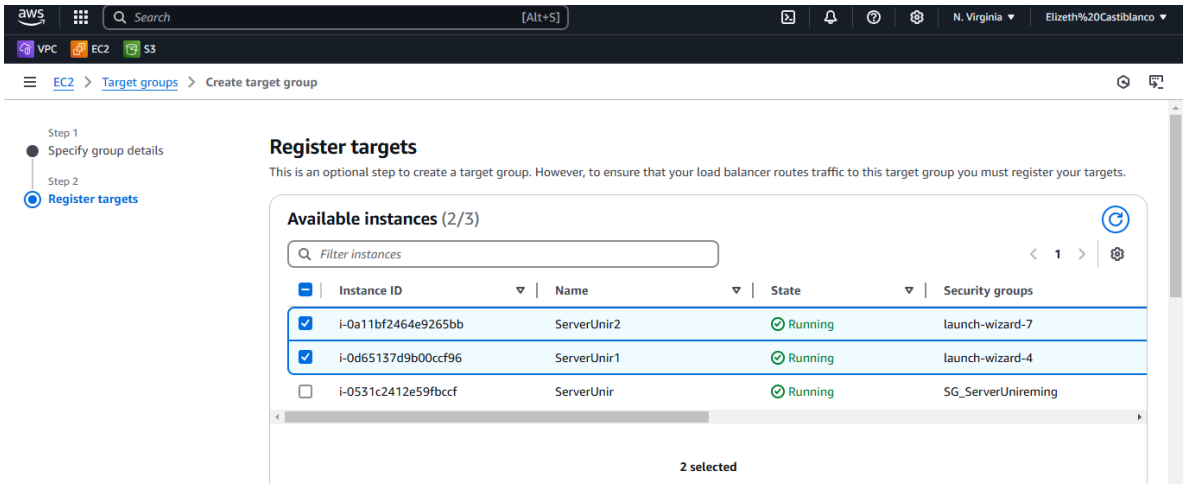


Imagen 30. Las instancias aparecen como objetivos

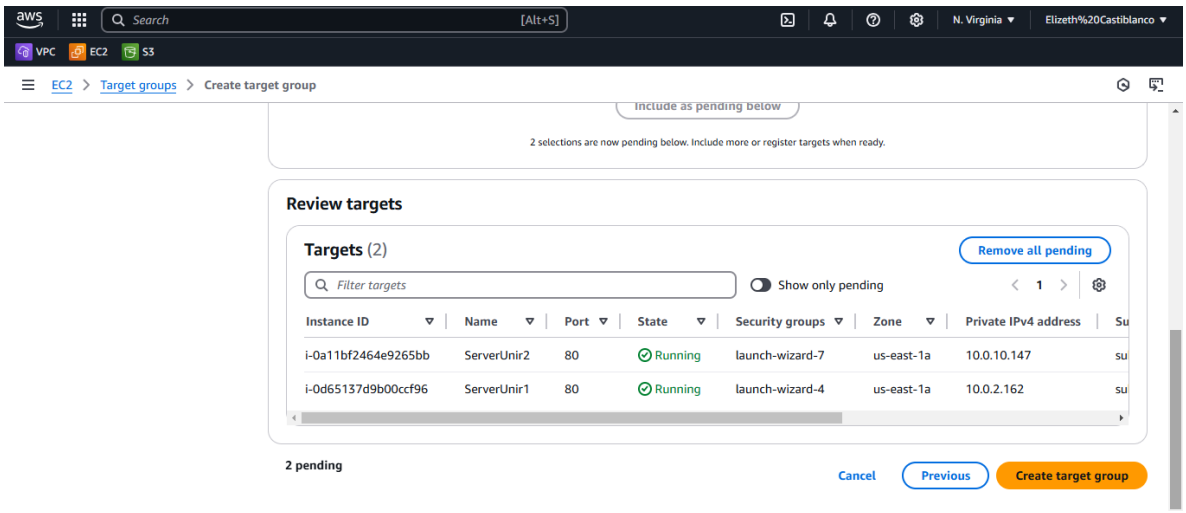


Imagen 31. Queda creado el Target Group y empieza a validar en la ruta que le indicamos (/) cada uno de los servidores e ira indicando si responden o no aumentando el número en el detalle Saludable y/o No Saludable.

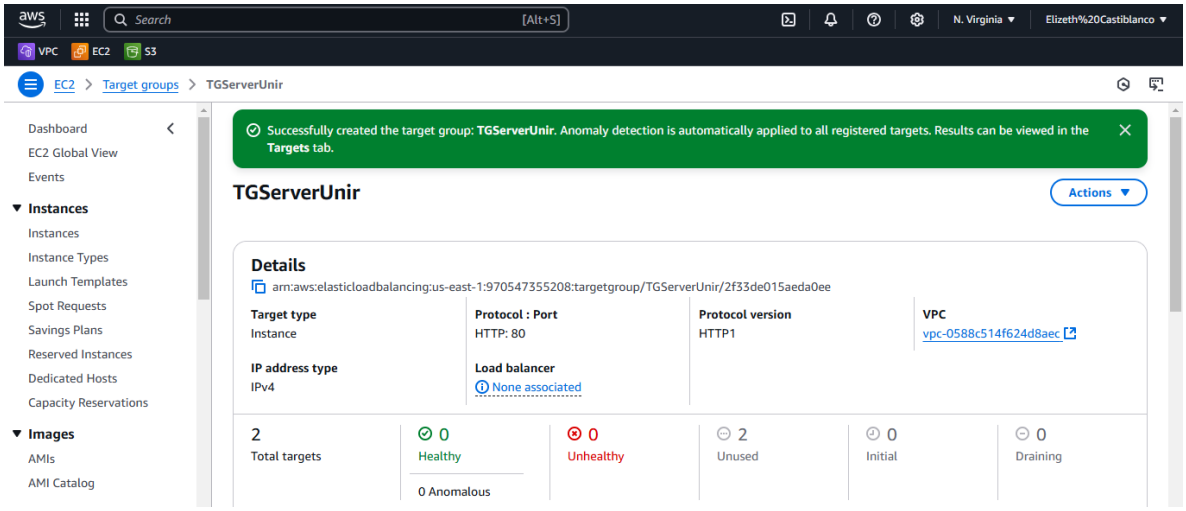


Imagen 32. Se elige el grupo de objetivos creado anteriormente y el resto de configuración se deja por defecto

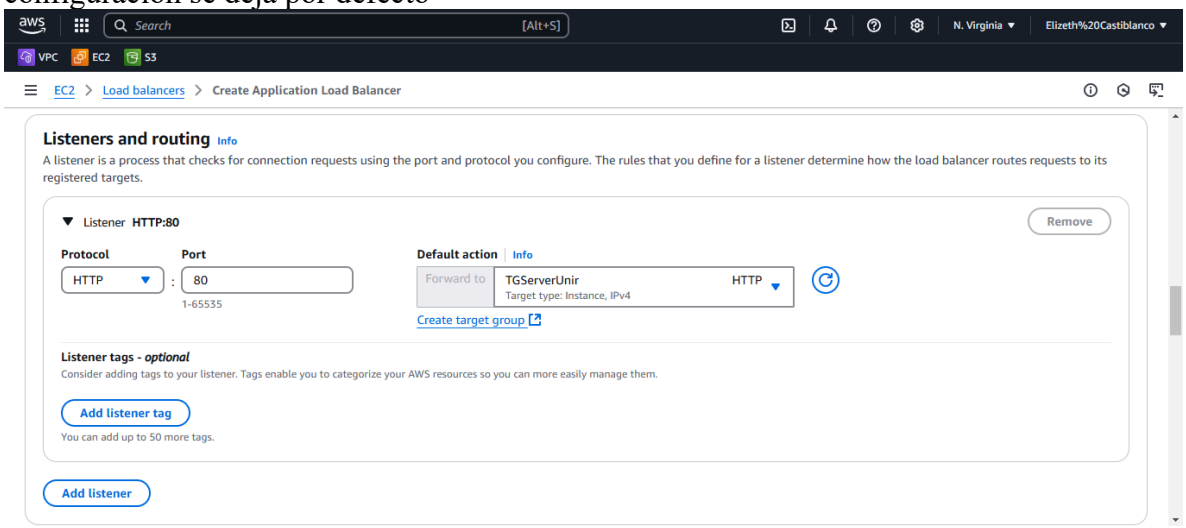


Imagen 33. Balanceador de carga inicializando

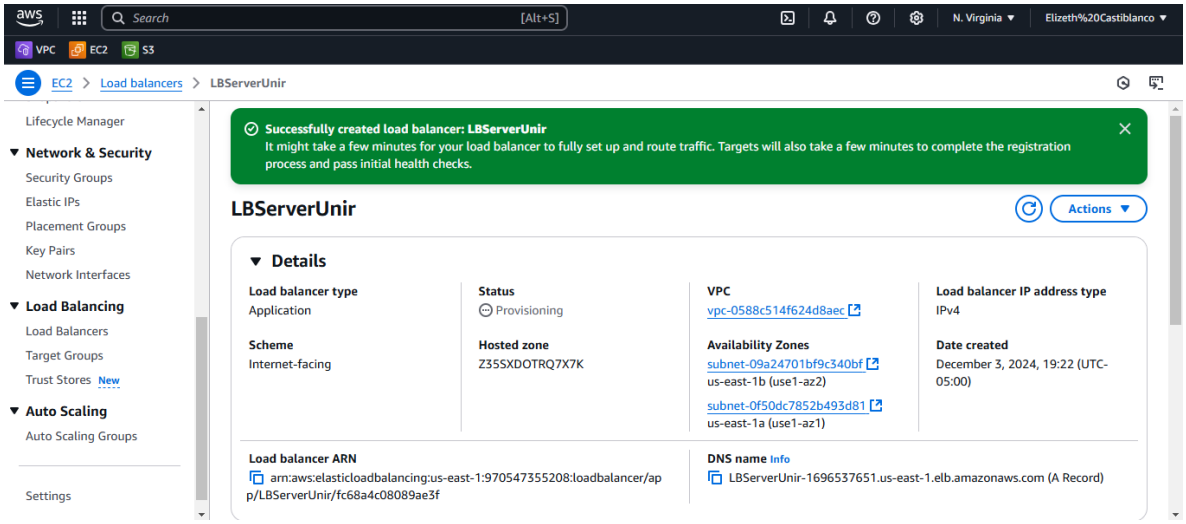


Imagen 34. Balanceador de carga activo

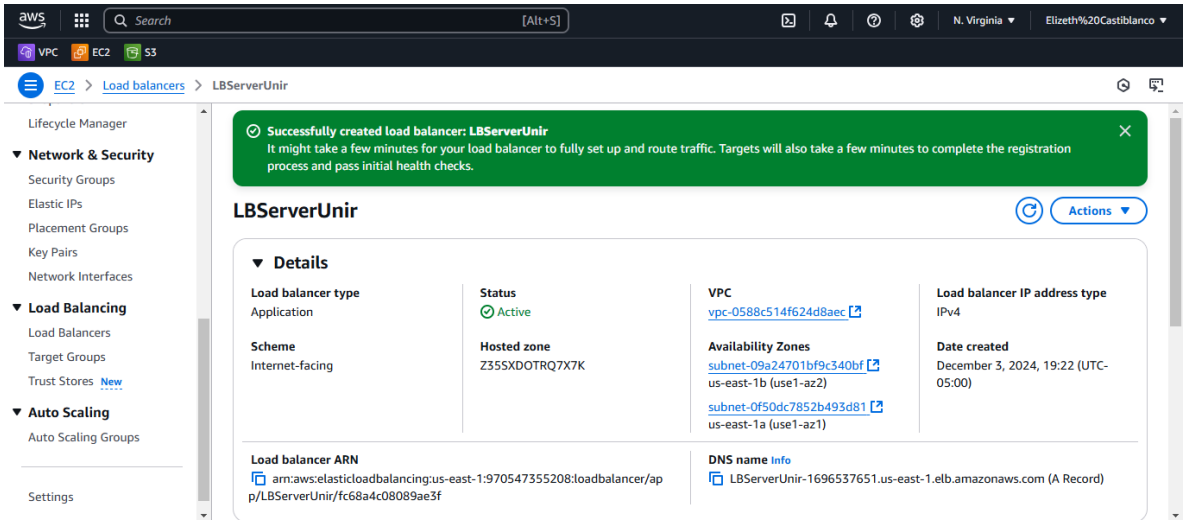


Imagen 35. Una vez el Balanceador de carga se activa el grupo de objetivos inicia a cargar el estado de las instancias, en este caso las dos están en estado Saludable.

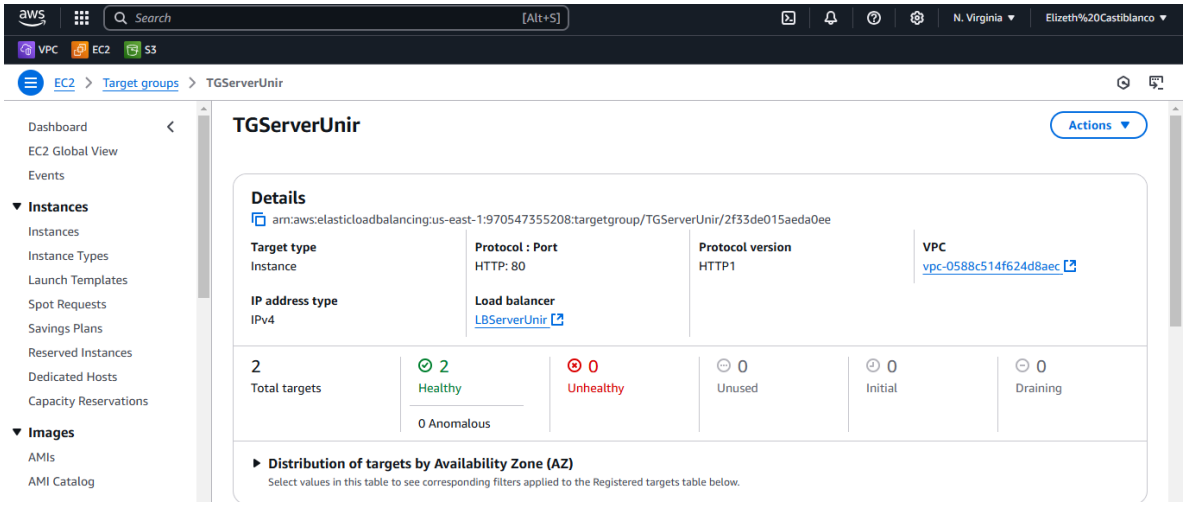


Imagen 36. Se ingresa ahora con el DNS del Balanceador de Carga (ya no con las IP de las instancias) al navegador y se evidencia que el servidor está funcionando correctamente.

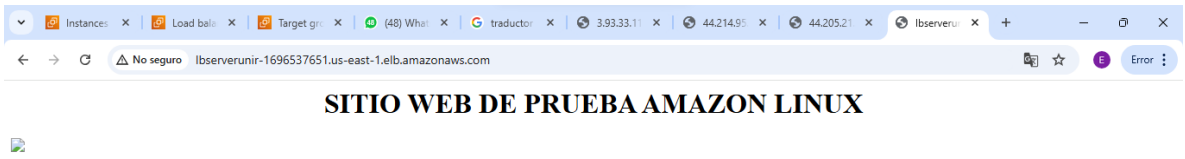


Imagen 37. Crear un Auto Scaling Group (cuando alguna de las instancias deje de funcionar por alguna razón se crean nuevas instancias automáticamente con la misma configuración de la AMI para que el servidor web no deje de responder en ningún momento).

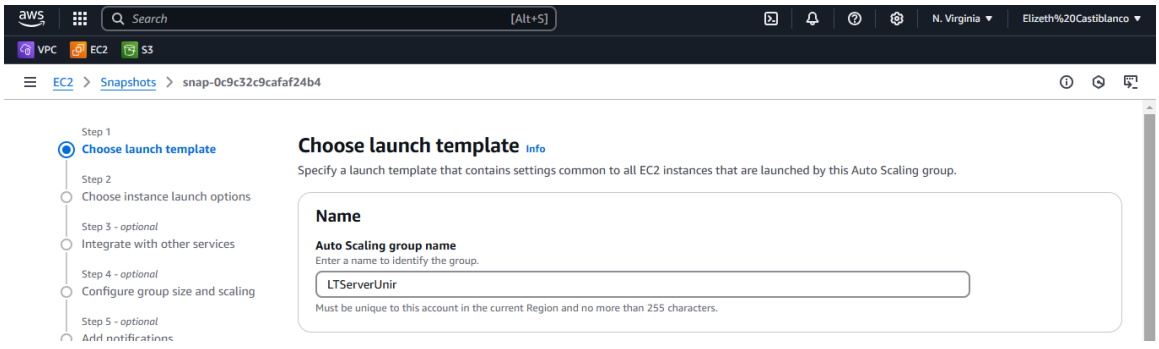


Imagen 38. Configuración del grupo de escalado automatico

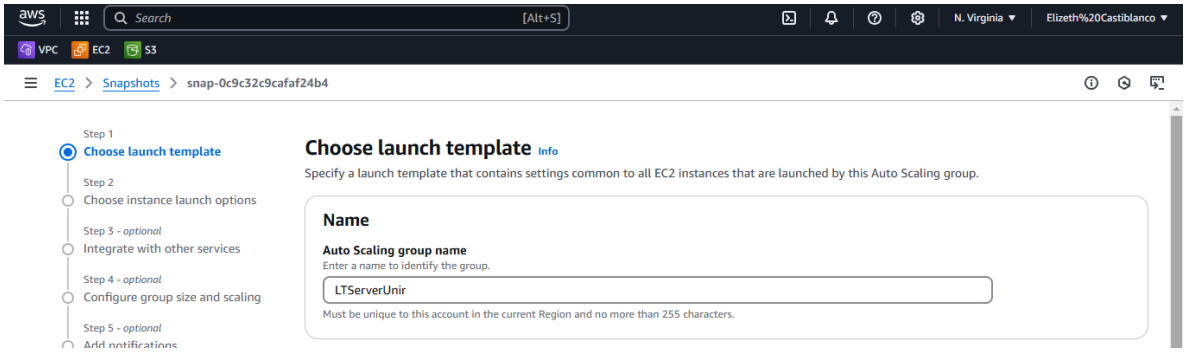


Imagen 39. Se elige la AMI creada al inicio

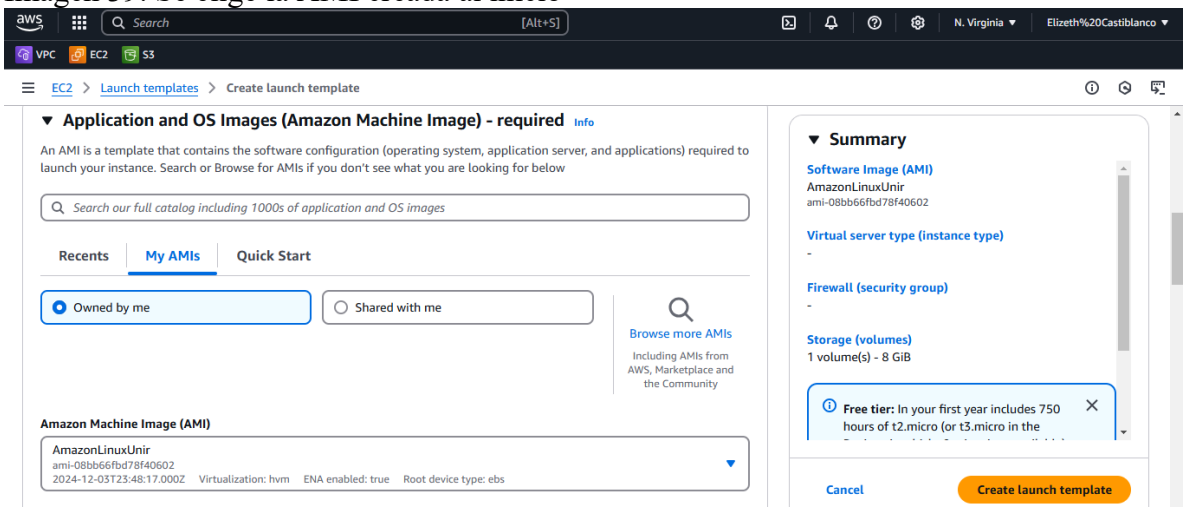


Imagen 40. Se configura el almacenamiento y se elige el mismo certificado de seguridad con el que se ha trabajado

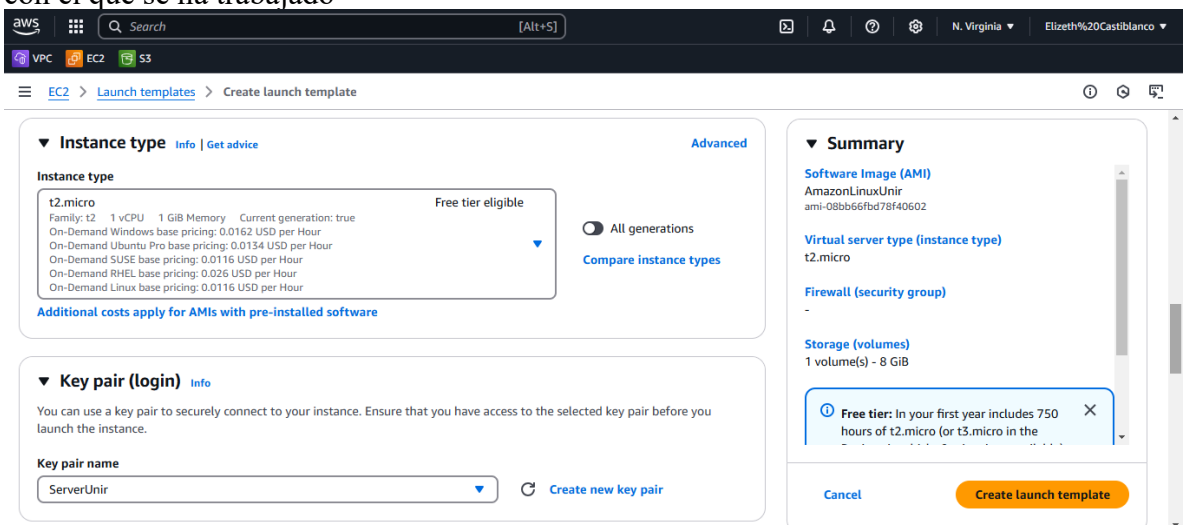


Imagen 41. Se elige el grupo de seguridad creado para el Balanceador de carga y se activan las IP publicas

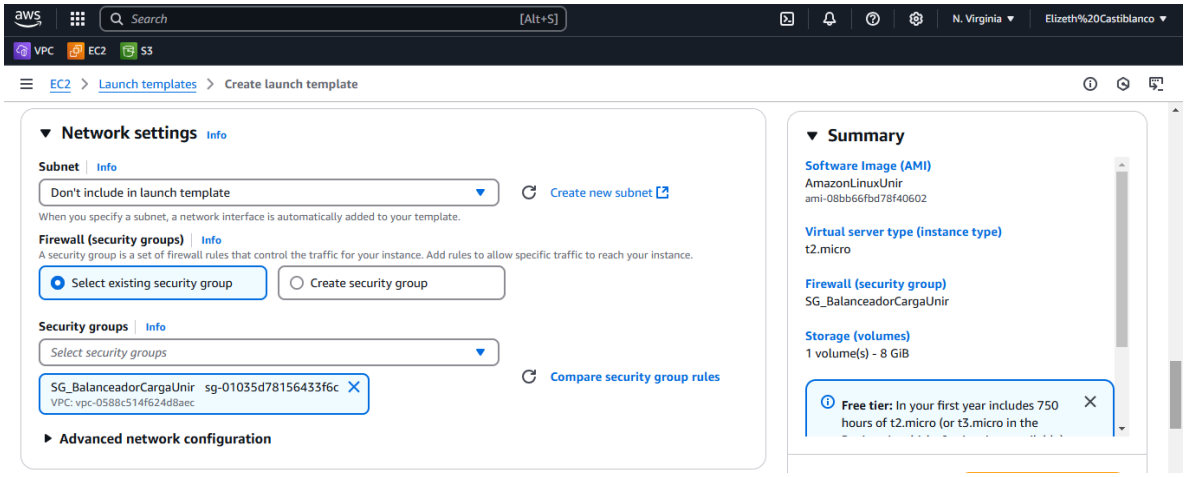


Imagen 42. Crear plantilla de lanzamiento

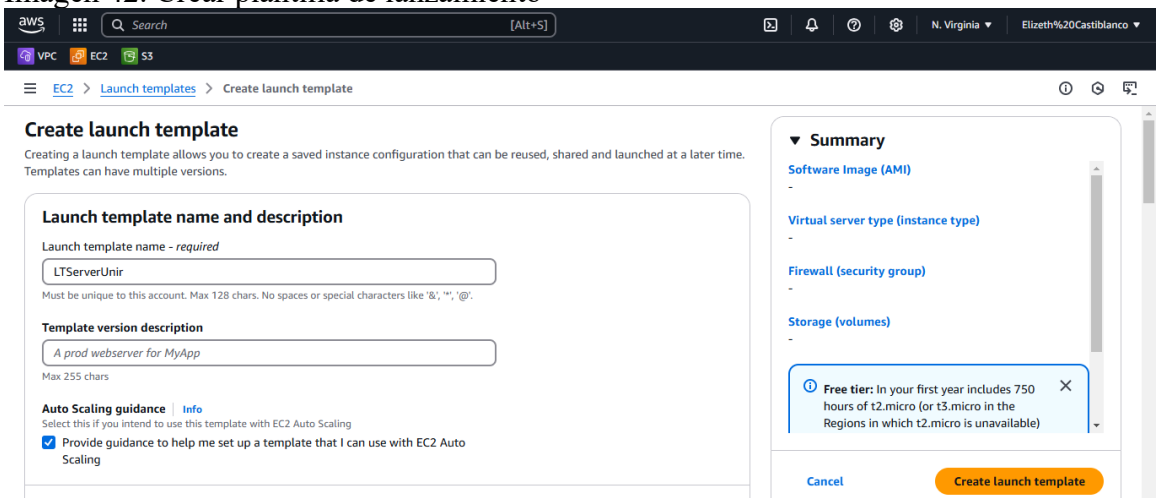


Imagen 43. Configuración plantilla de lanzamiento

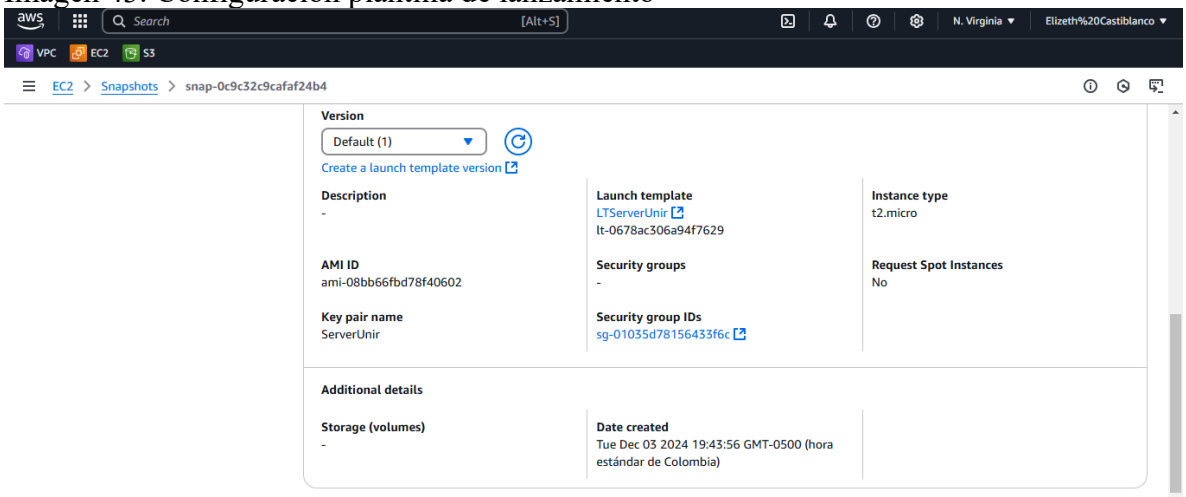


Imagen 44. Se elige la plantilla de lanzamiento creada para seguir configurando el grupo de escalado automático

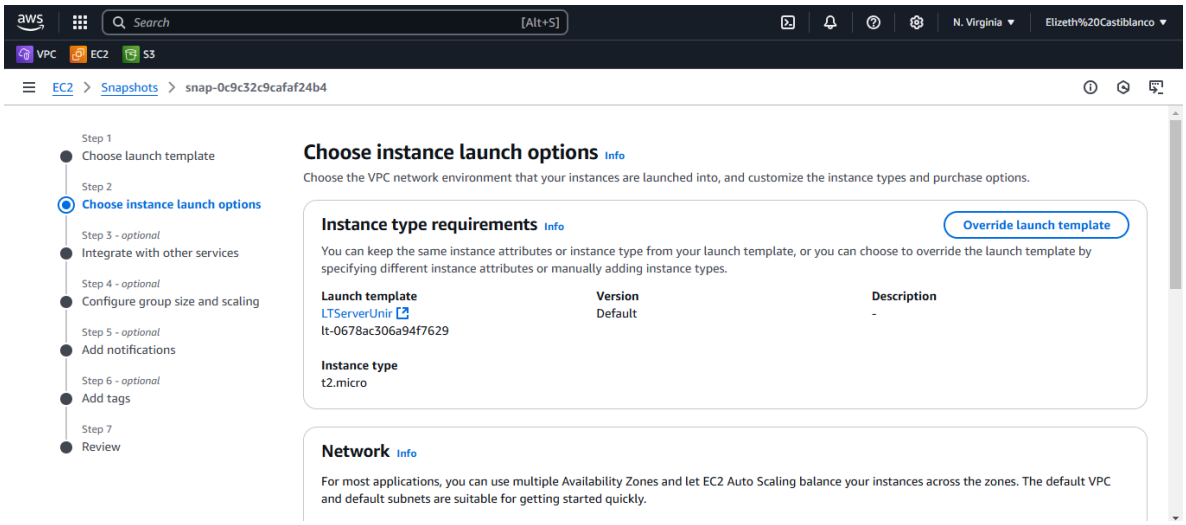


Imagen 45. Se configuran las dos zonas de disponibilidad de la VPC

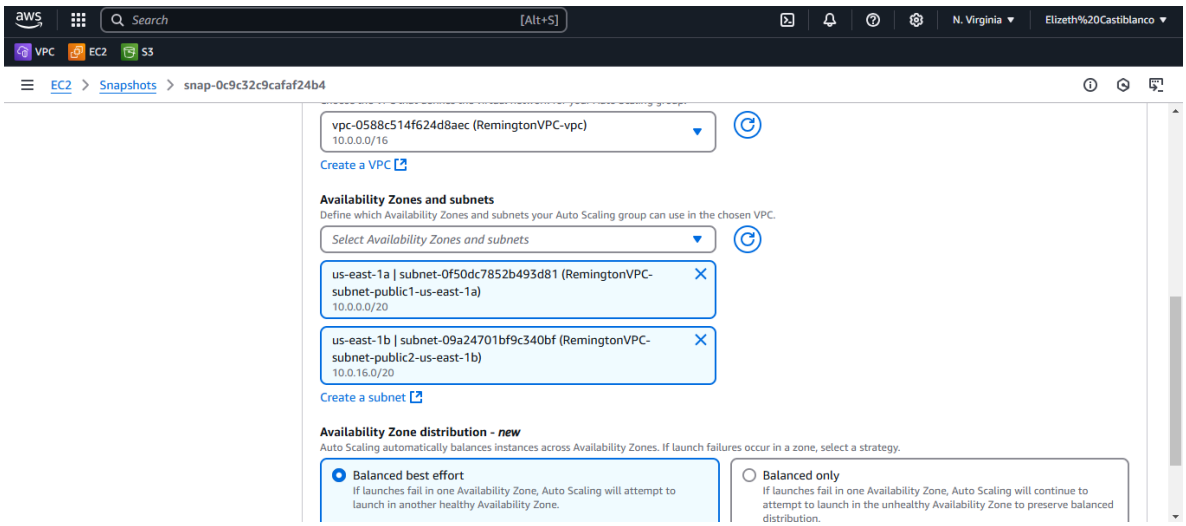


Imagen 46. Se continua con la configuración por defecto

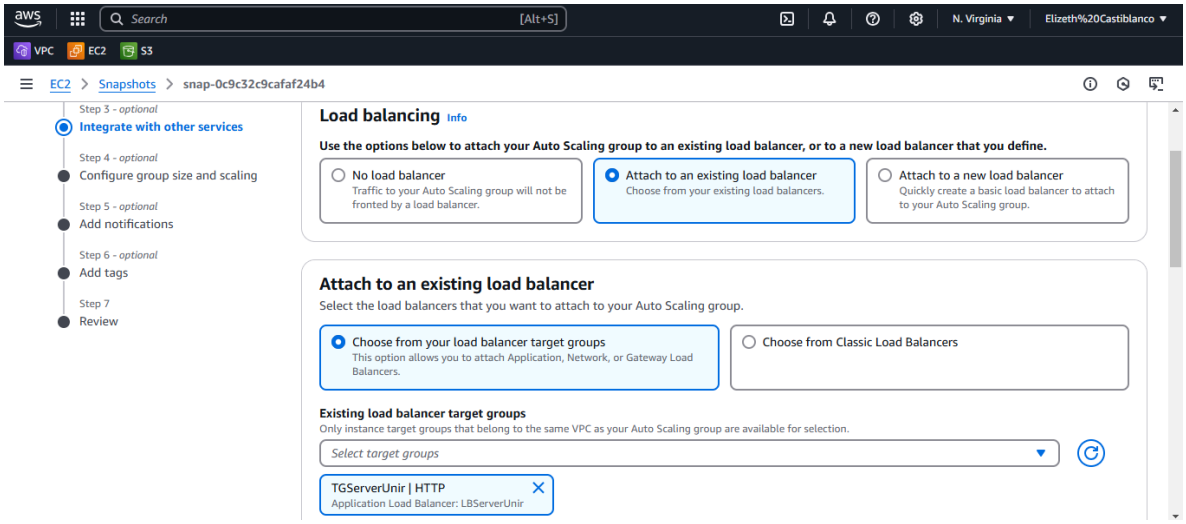


Imagen 47. Se configura la cantidad de instancias que se desea se creen cuando alguna deja de funcionar, en este caso será de mínimo 2 máximo 5 y deseadas 3

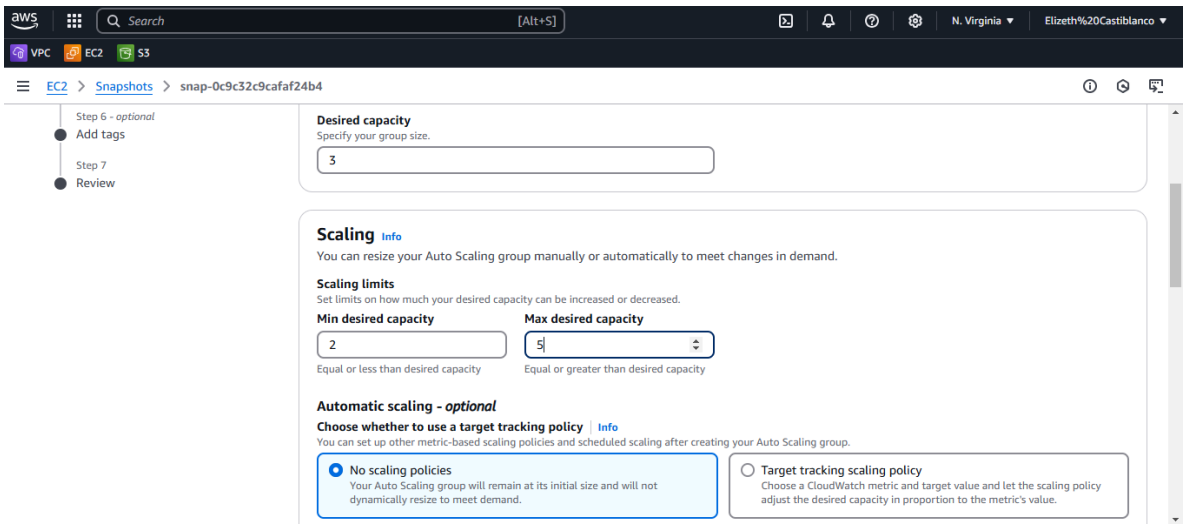


Imagen 48. Se continua con la configuración por defecto

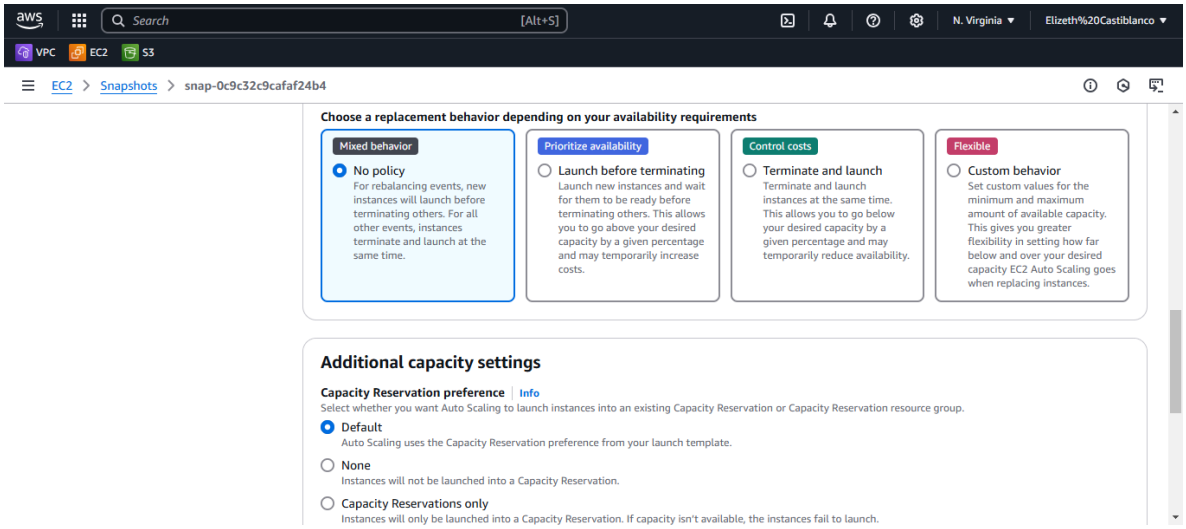


Imagen 49. Se continua con la configuración por defecto

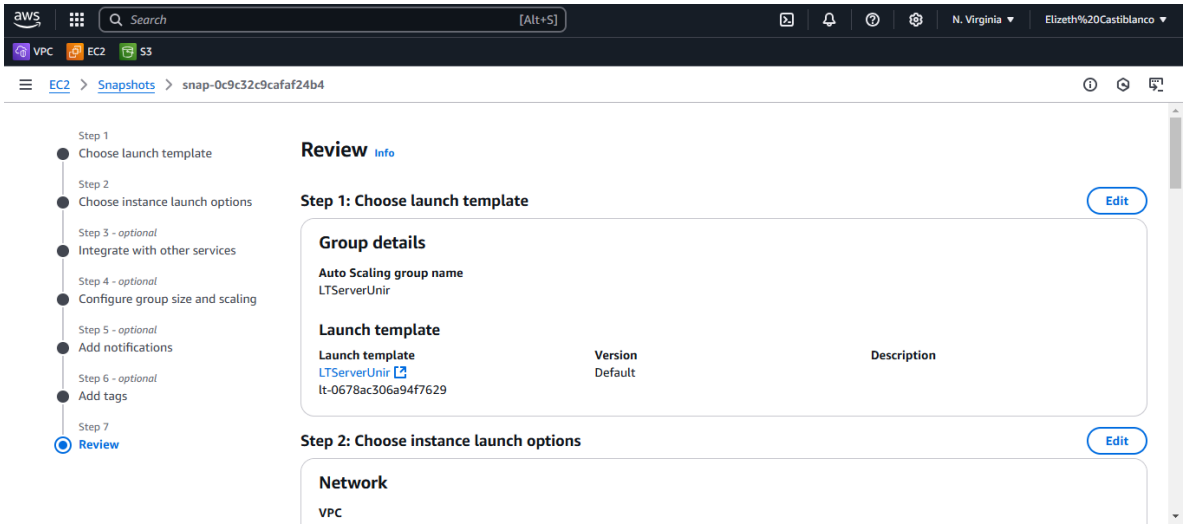


Imagen 50. Se crea el grupo de escalado automático con éxito

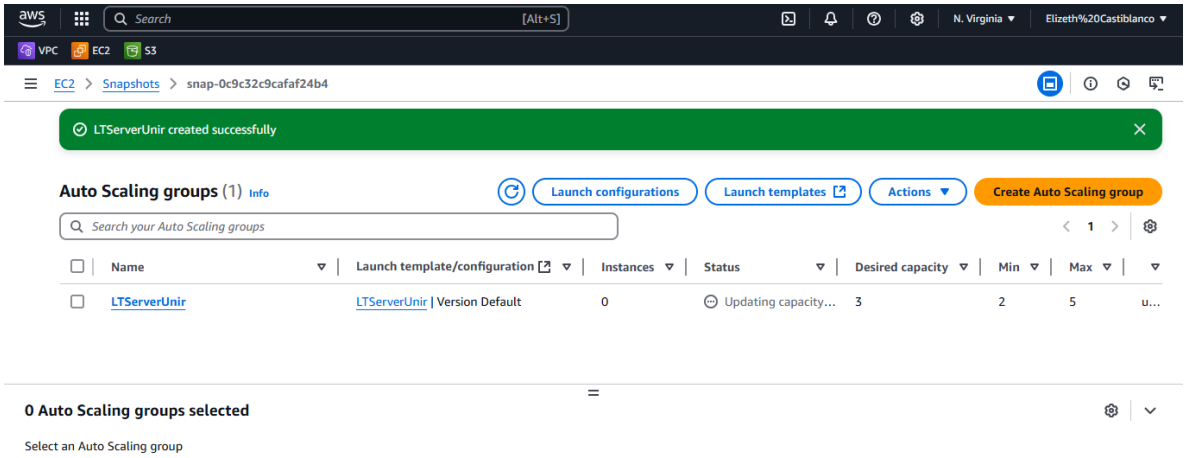


Imagen 51. Una vez creado el Auto Scaling Group, se crean automaticamente el numero de instancias que se halla configurado (en este caso fueron tres)

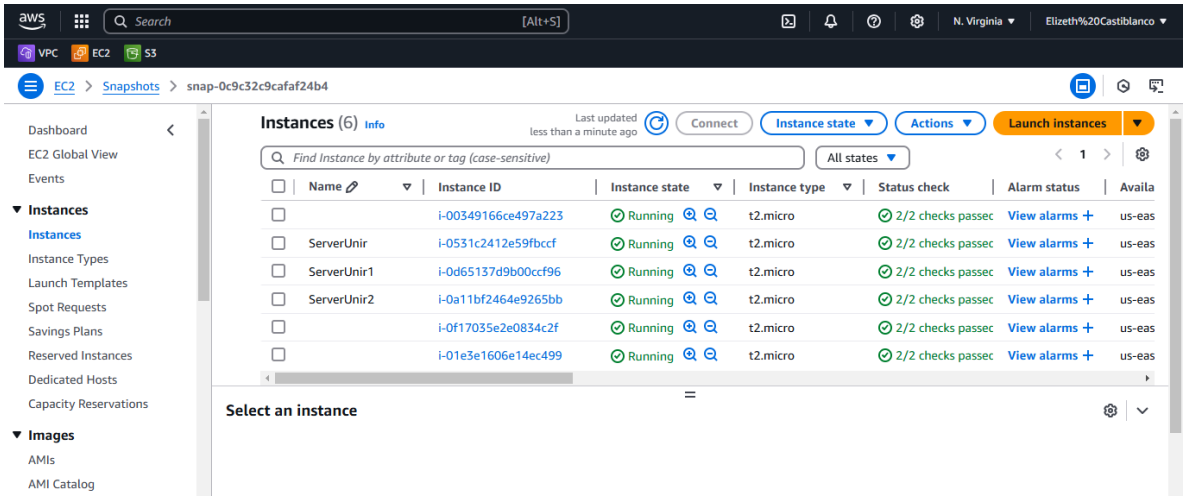


Imagen 52. El grupo de objetivos marca las 5 instancias en estado Saludable.

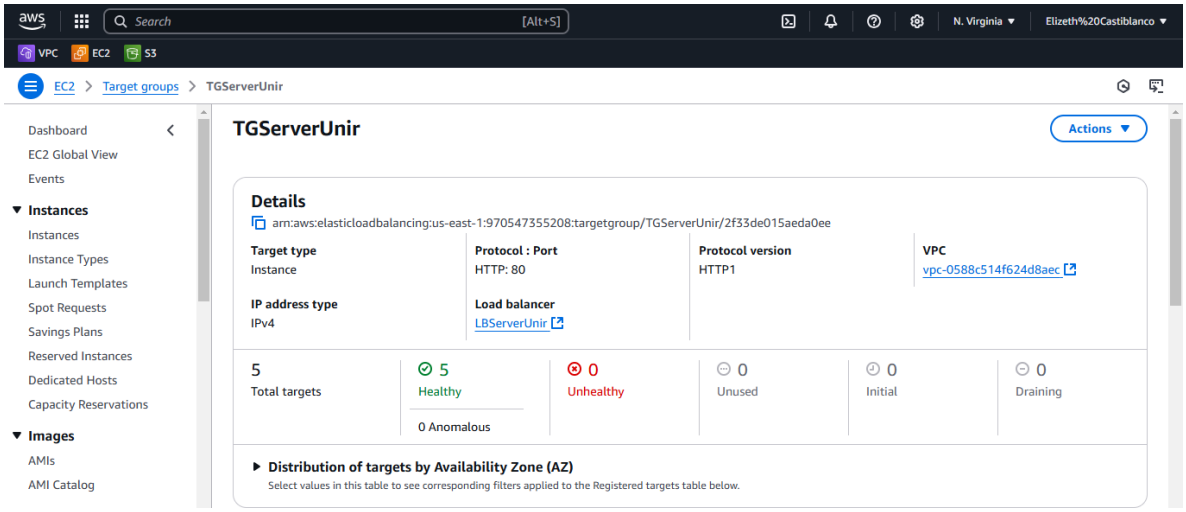


Imagen 53. Se hace la comprobación del Auto Scaling Groups para verificar que trabaje correctamente, por lo tanto, eliminamos una de las 5 instancias manualmente

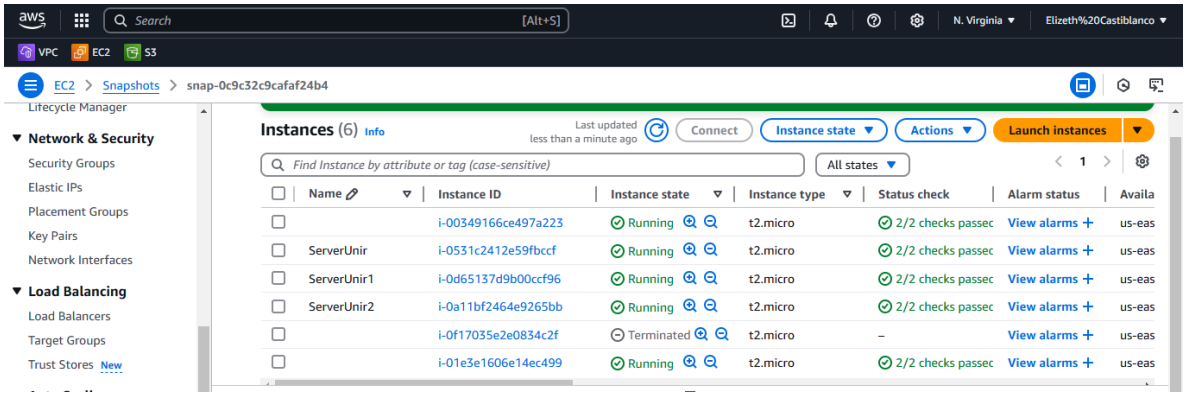


Imagen 54. En la actividad del Auto Scaling ya aparece la instancia eliminada

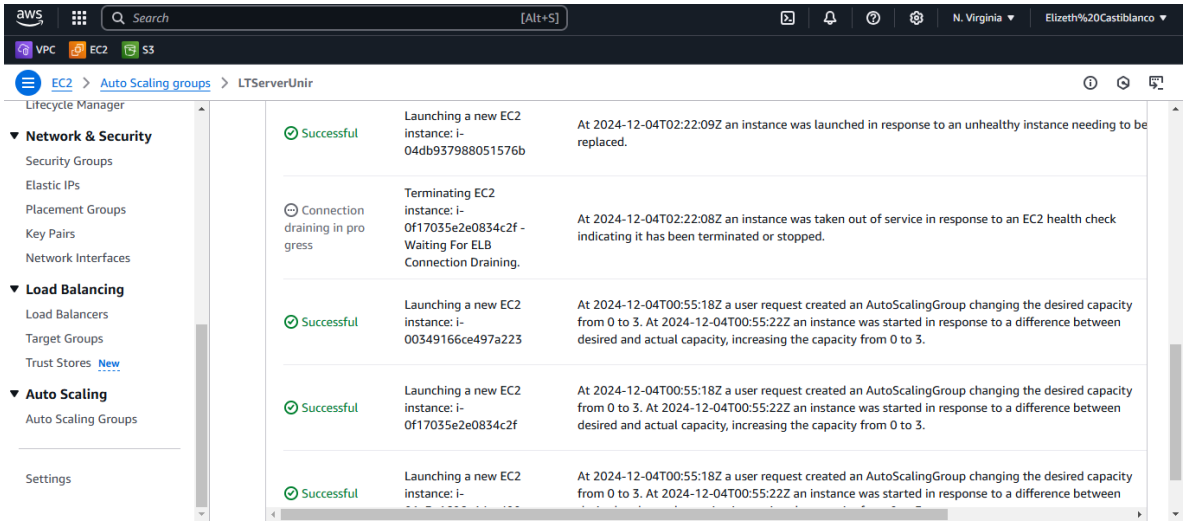
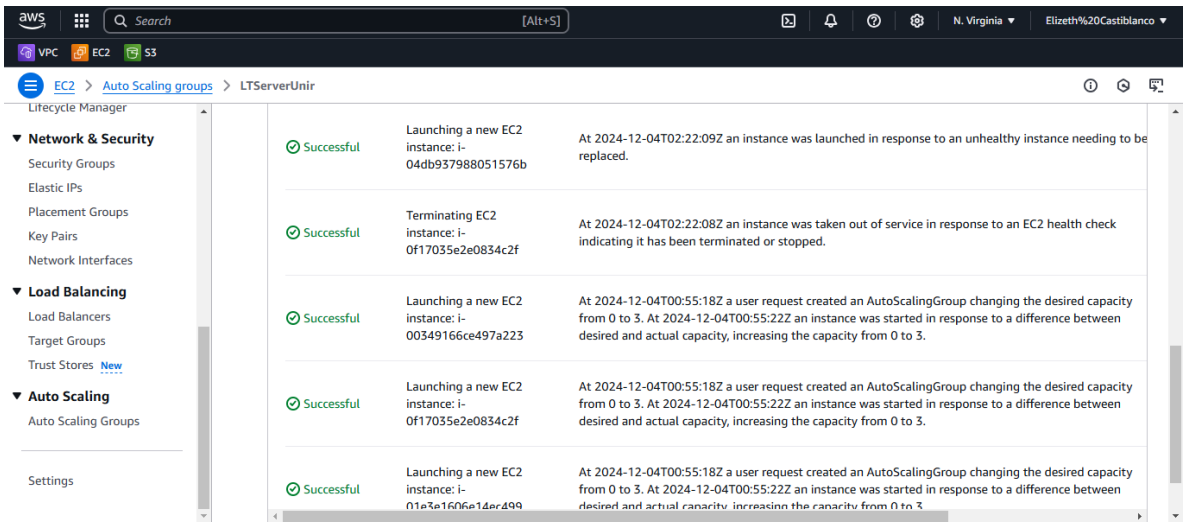


Imagen 55. Finalmente se crea una nueva instancia automáticamente completando siempre las 3 instancias deseadas



Video de explicación implementación de Balanceador de Carga en AWS

https://youtu.be/bddliaoiG_o

Implementación de S3

Imagen 1. S3 Almacenamiento instantáneo que me brinda objetos

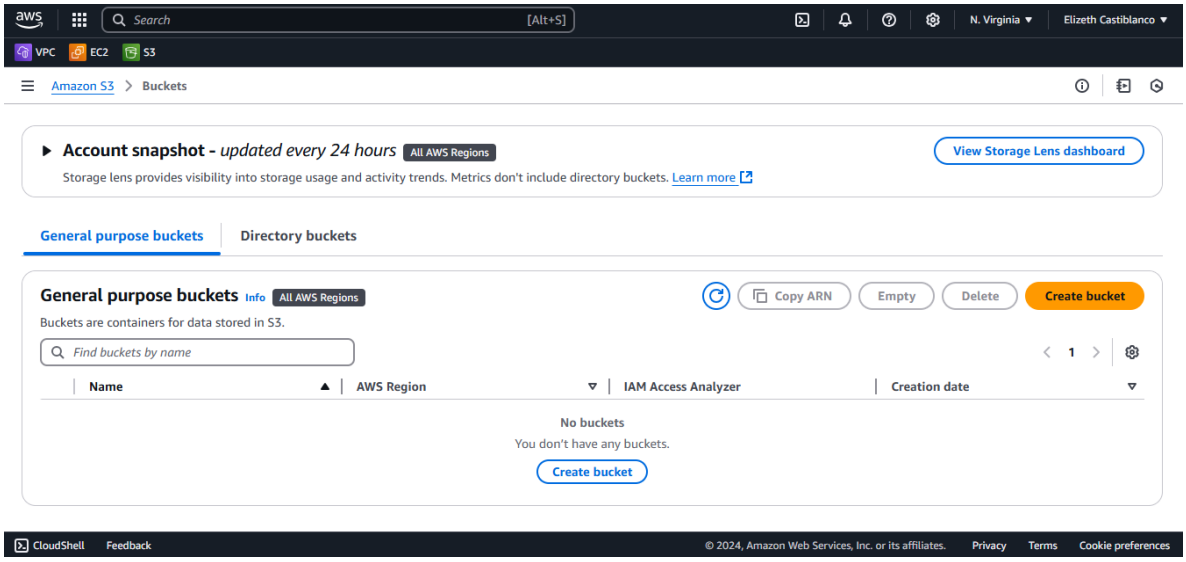


Imagen 2. Asignación de nombre

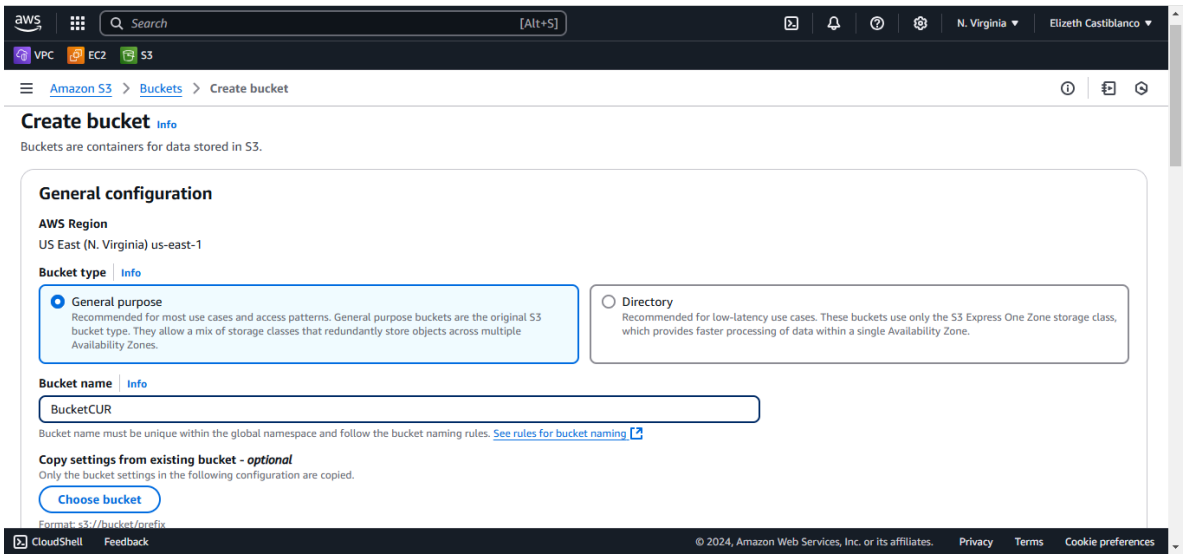


Imagen 3. Manejar los permisos y definir si los objetos que se están manejando son públicos o privados

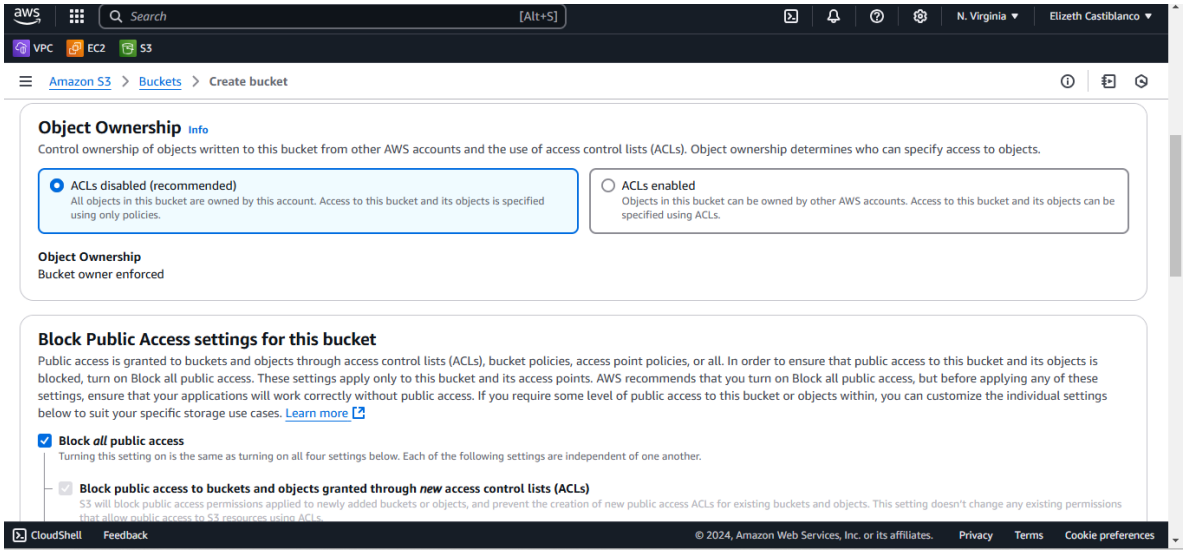


Imagen 4. Bloquear acceso publico

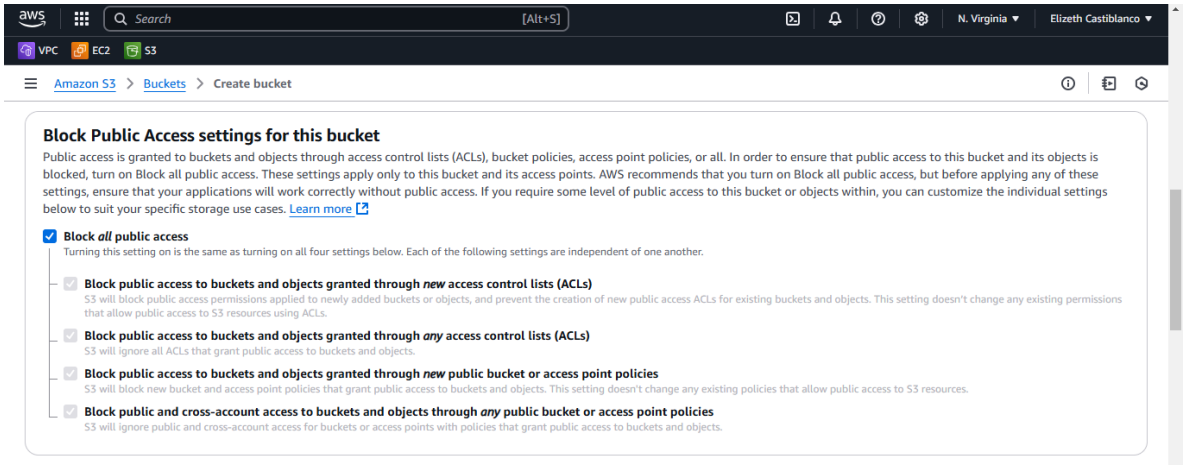


Imagen 5. Crear bucket

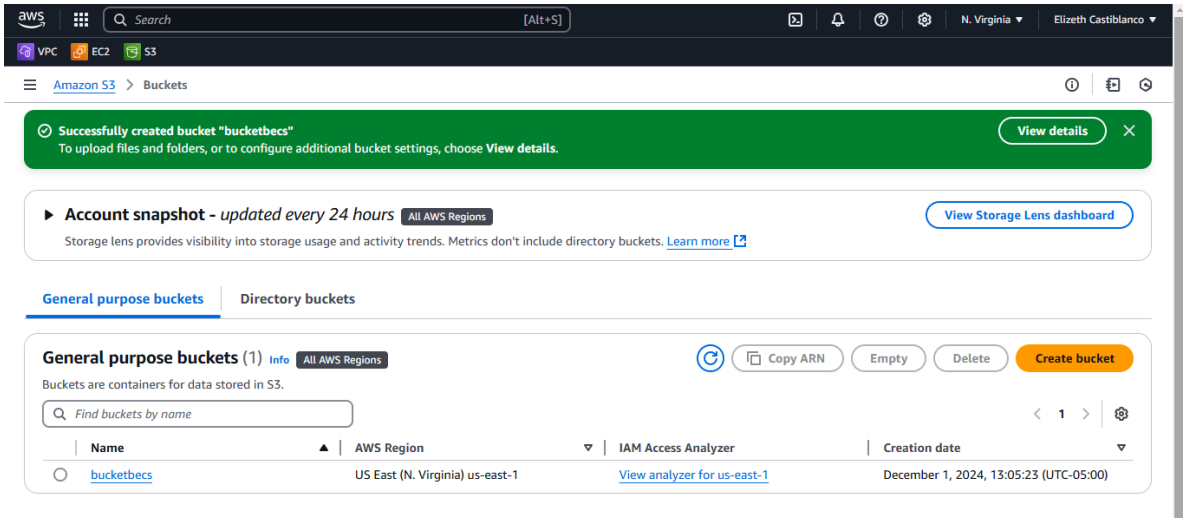


Imagen 6. Crear objetos

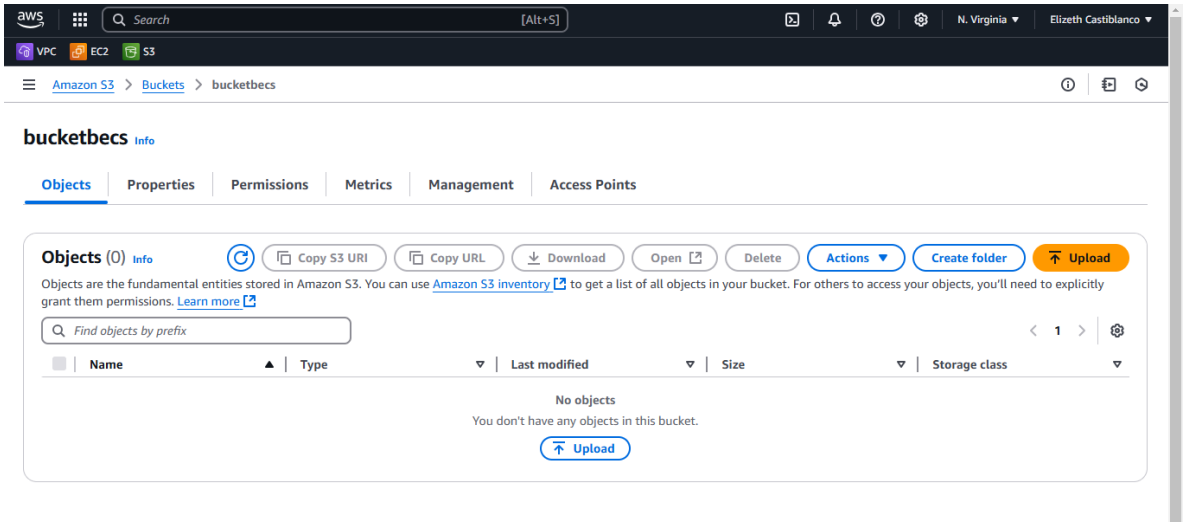


Imagen 7. Subir archivos como objetos

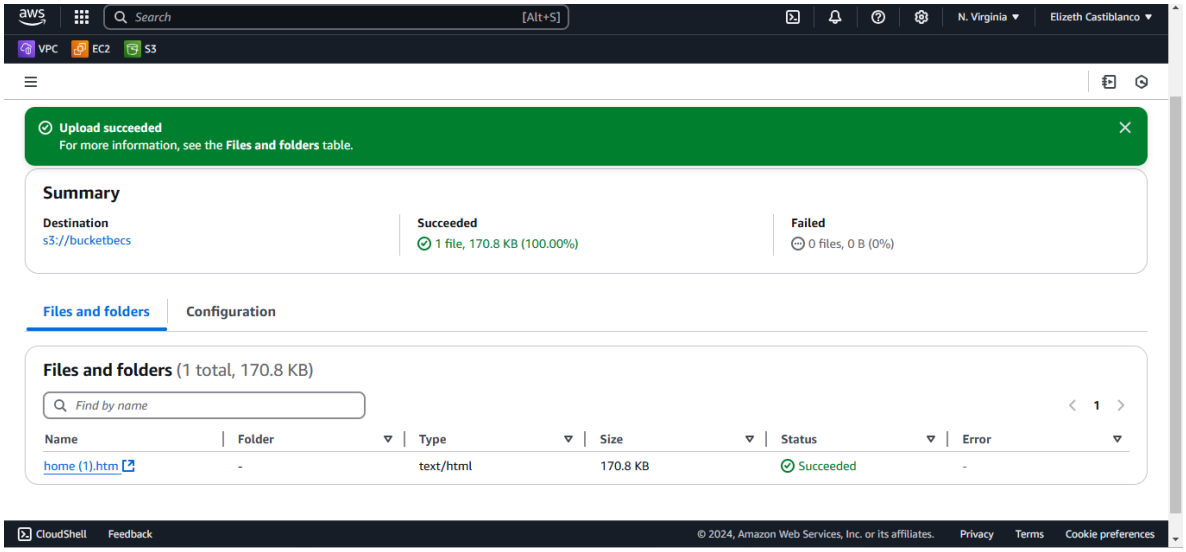


Imagen 8. Dentro de las características del objeto ejecutar la URL

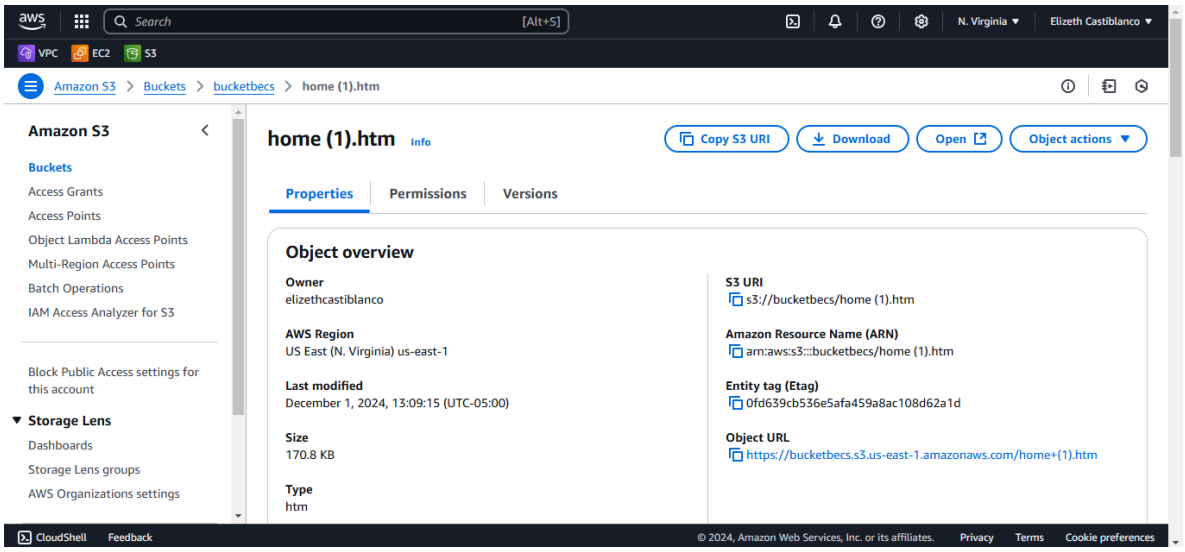


Imagen 9. Teniendo en cuenta que el objeto no tiene permiso de acceso público no se verá.

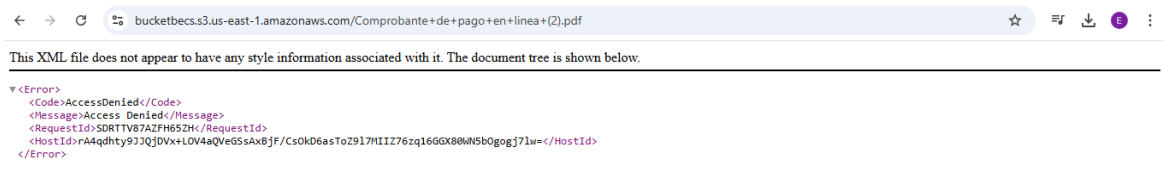


Imagen 10. Desbloquear el acceso público en el bucket

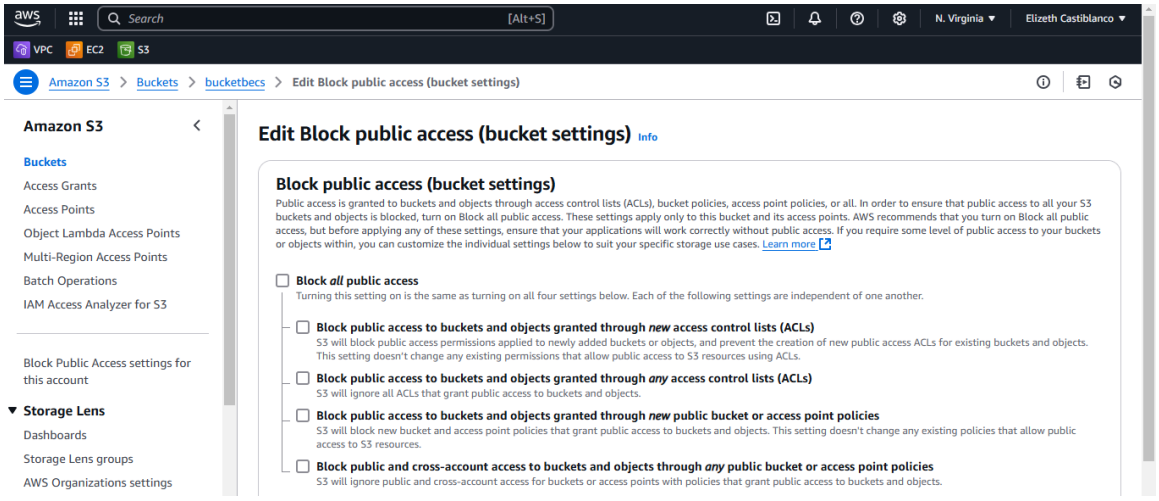


Imagen 11. Dar permisos de acceso público al objeto y ejecutar

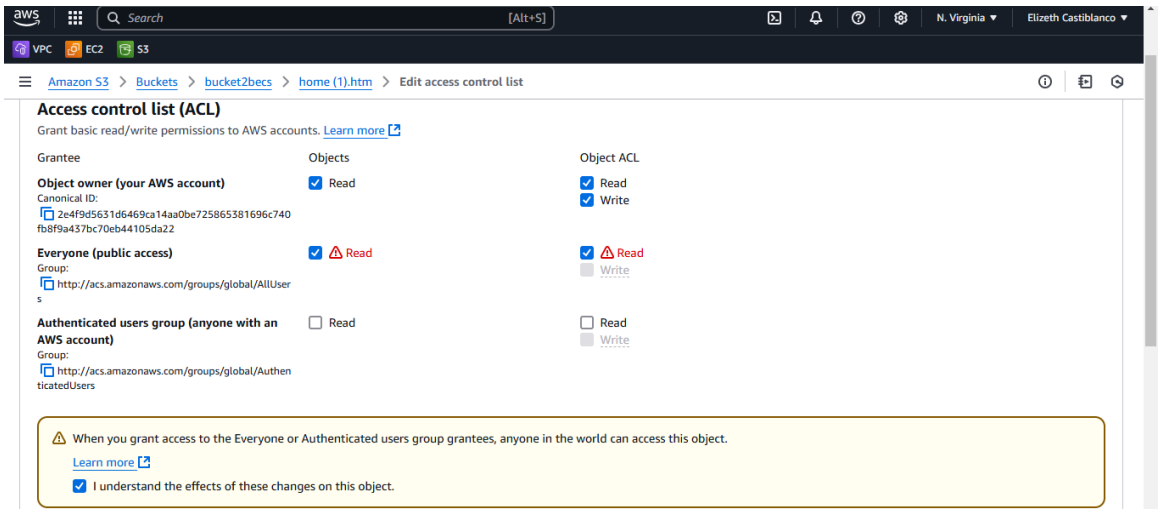


Imagen 12. Verificar la correcta ejecución

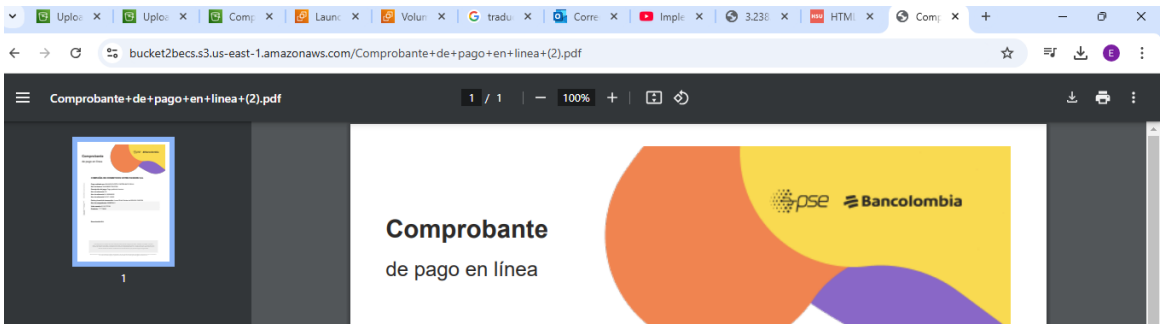


Imagen 13. Usar plantilla para crear como objetos

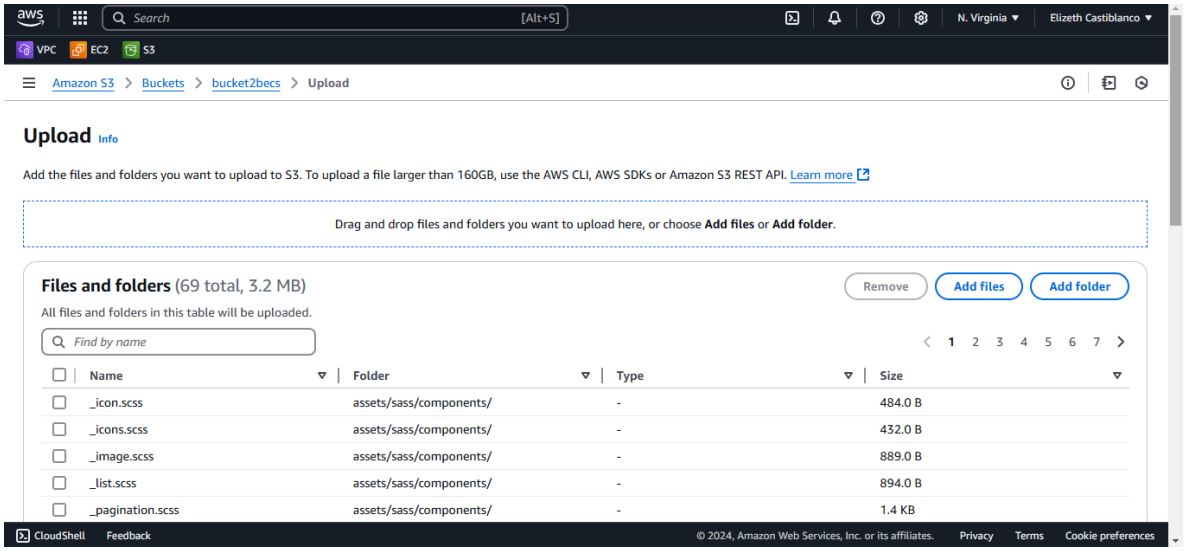


Imagen 14. Garantizar acceso público de lectura y subir

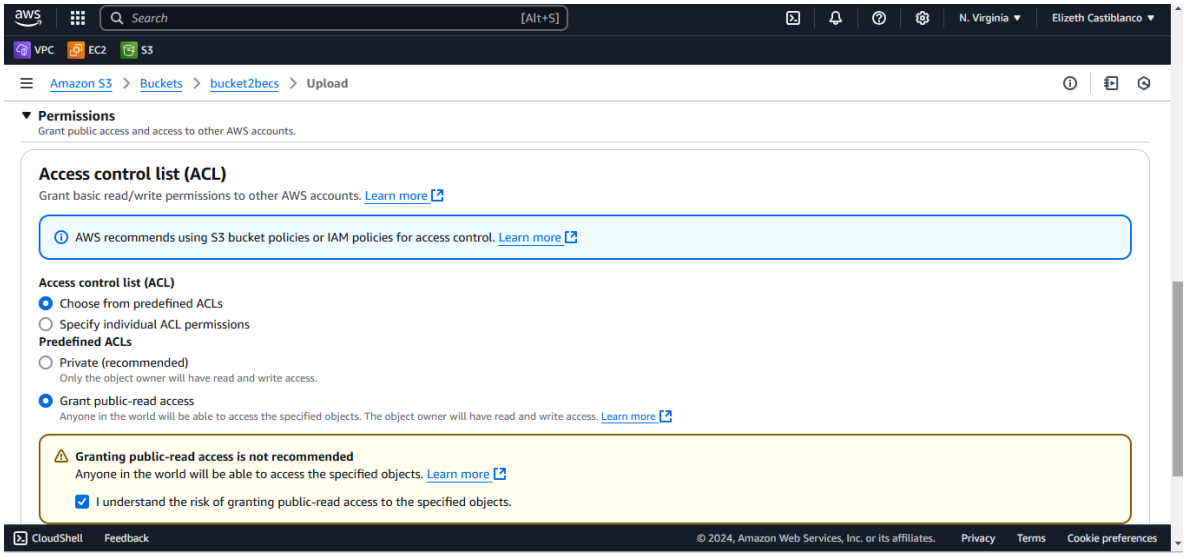


Imagen 15. Plantilla cargada correctamente

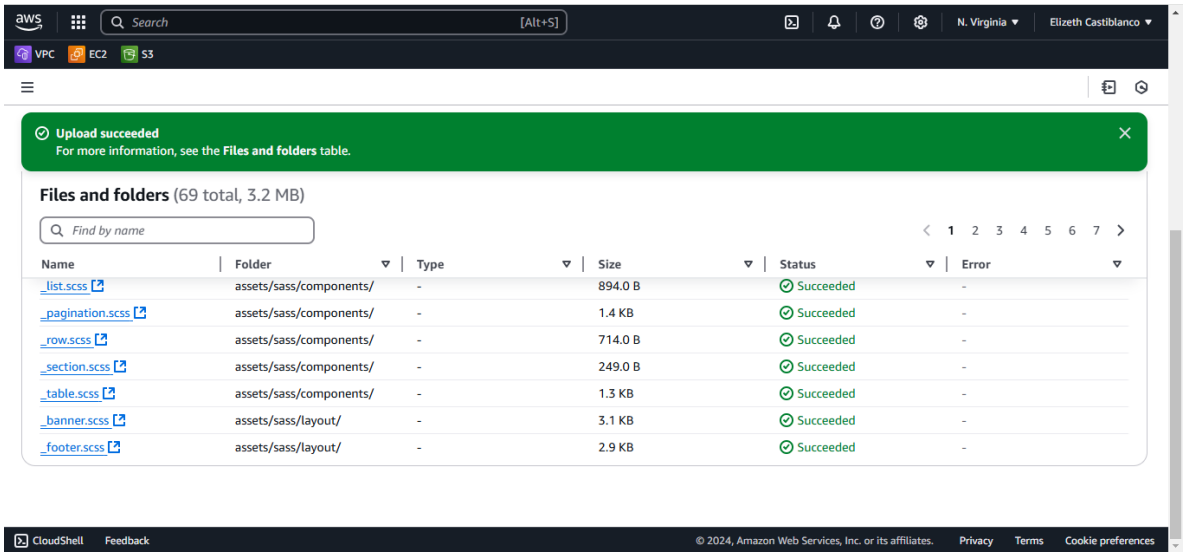
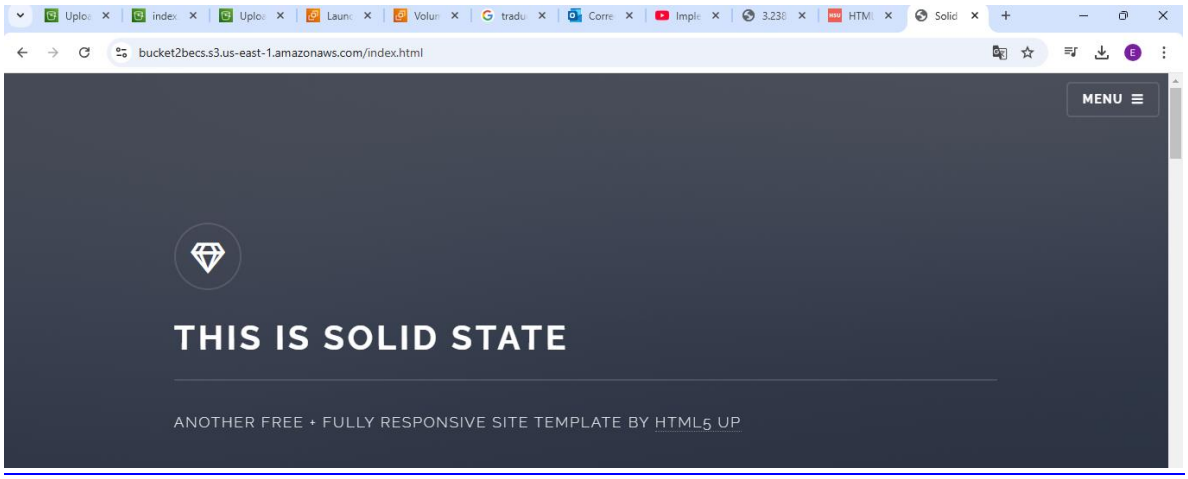


Imagen 16. Ejecutar URL y verificar



Implementación de Auto Scaling Group y configuración de políticas

Imagen 1. Ahora que se tienen únicamente las instancias creadas automáticamente por el Auto Scaling Group (ya no tienen IP publica) se crea una nueva instancia con la misma configuración de las anteriores sin usar la AMI creada, para conectarse por ssh y administrar las demás.

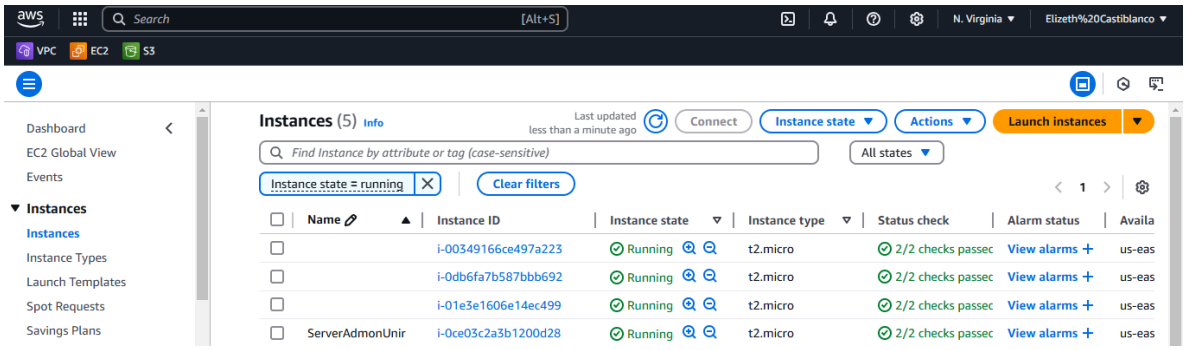


Imagen 2. Se crea una politica dinamica de escalado dentro del Auto Scaling Group para que cuando el uso de CPU sea mas o menos del 25% escale.

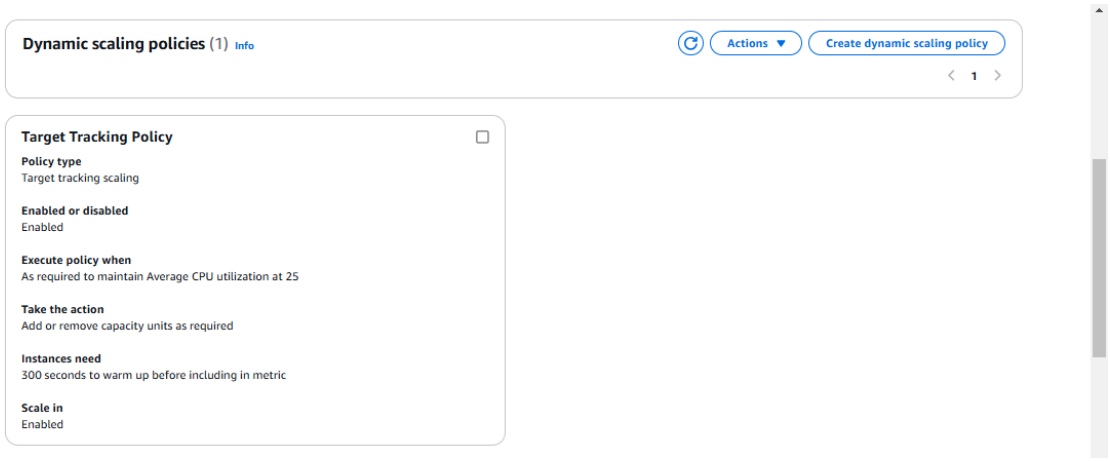


Imagen 3. Se conecta desde la instancia y se instala un administrador de recursos.

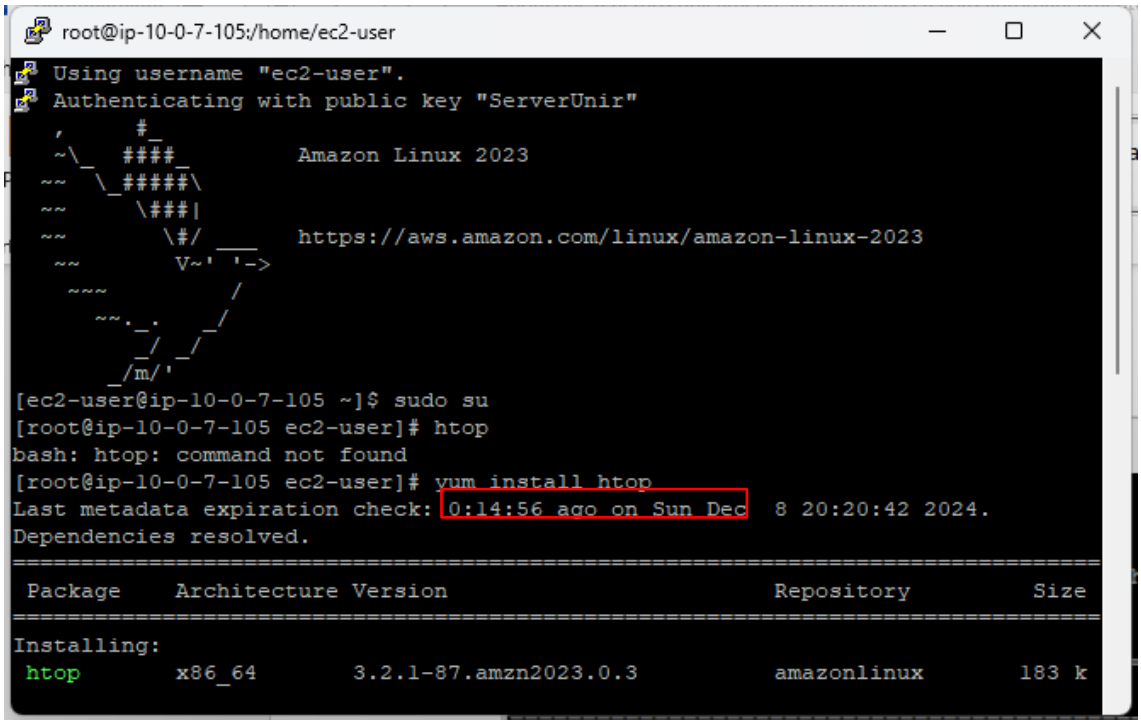


Imagen 4. Se conecta a una de las instancias que no tienen IP publica

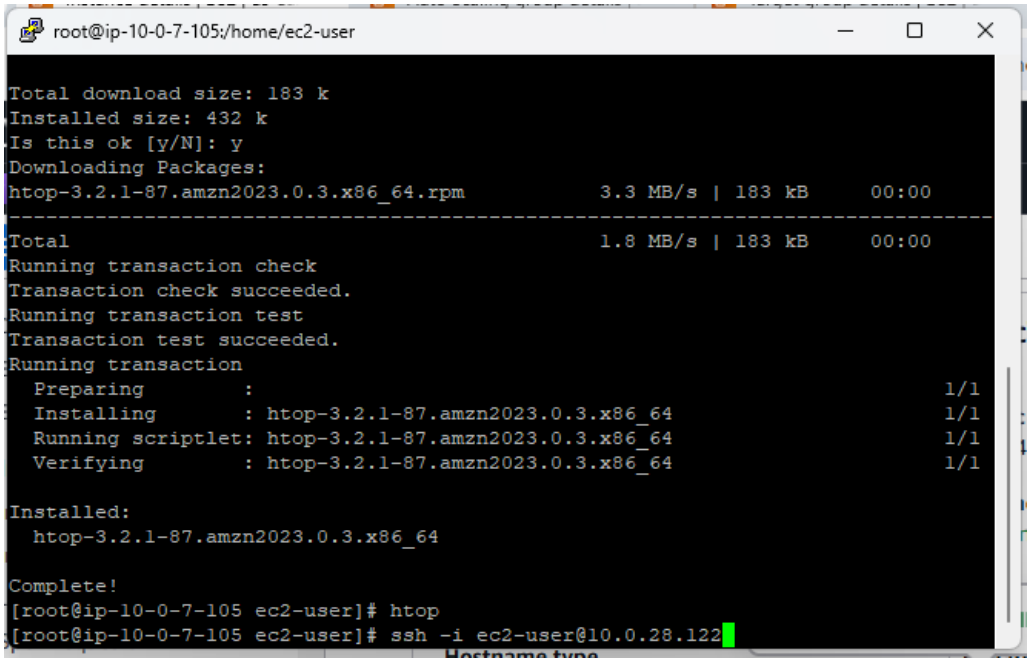


Imagen 5. Se sube la clave de seguridad. PEM

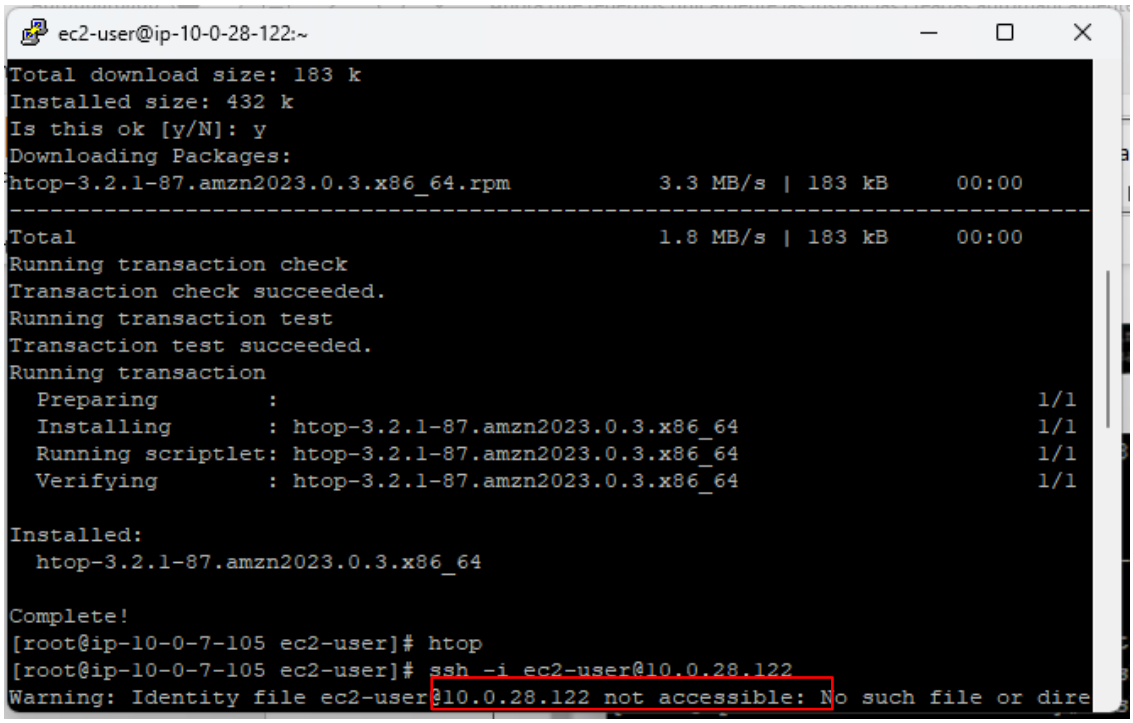


Imagen 6. Se dan los permisos al certificado

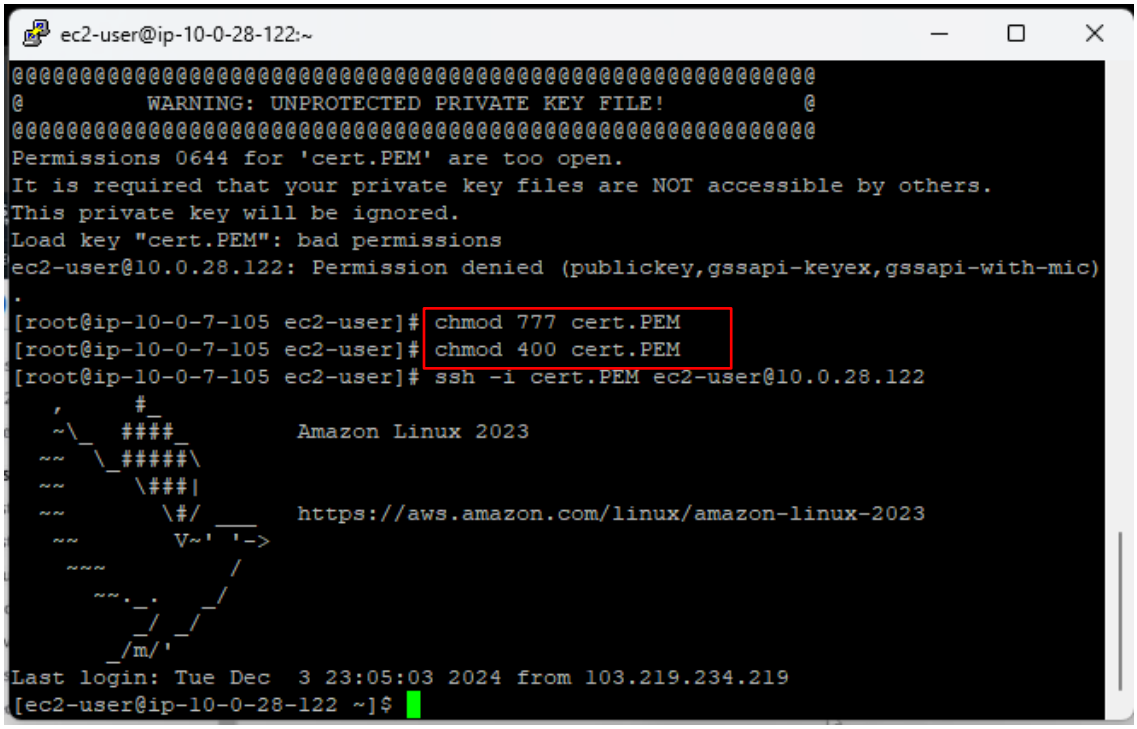


Imagen 7. Se instala una aplicación para subir el uso de CPU de una instancia y hacer la prueba de estrés

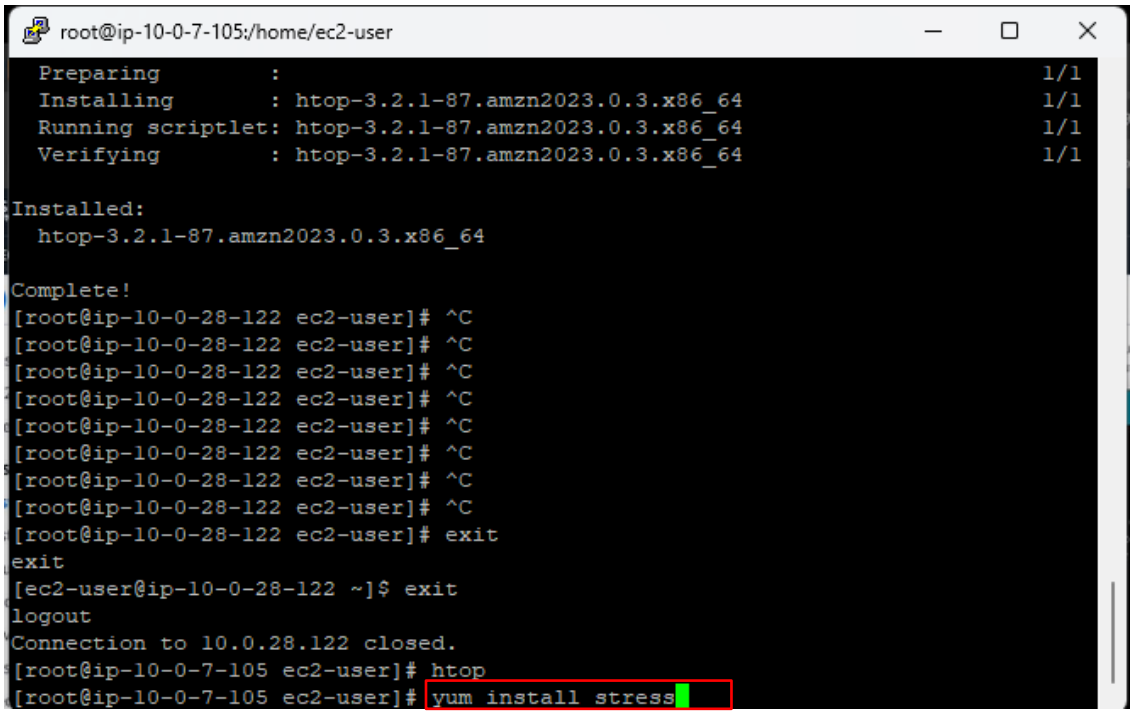


Imagen 8. Se ejecuta el comando para aumentar el rendimiento de CPU

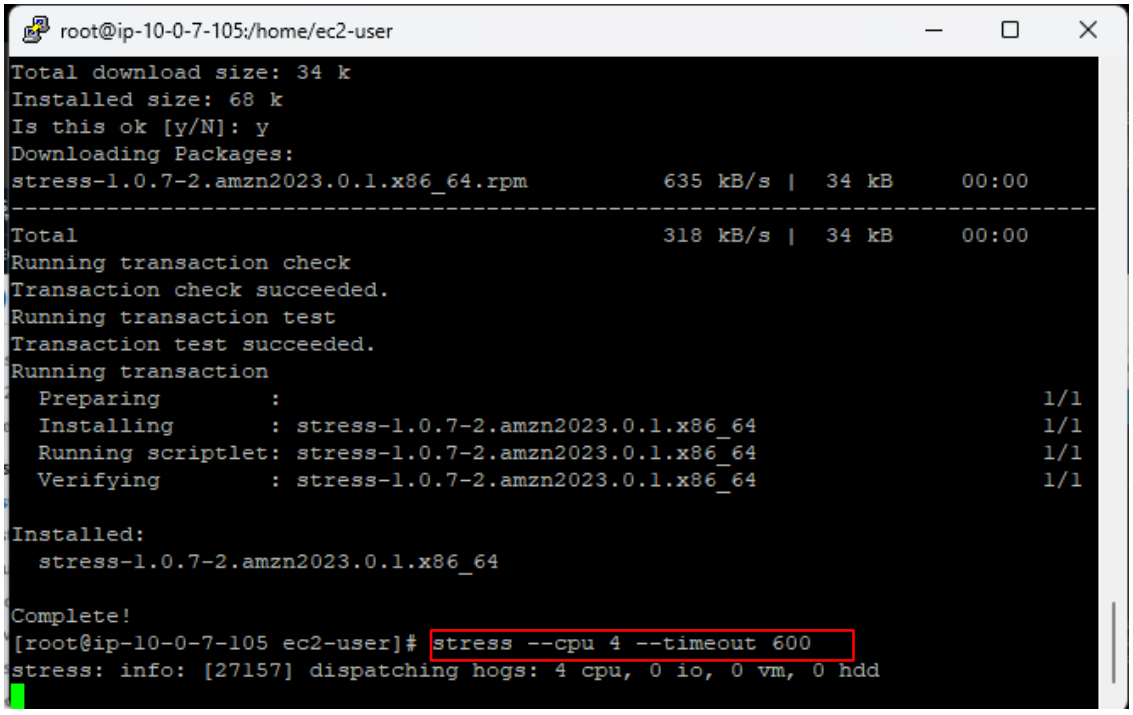


Imagen 9. Se verifica como el uso de CPU de la instancia aumento al 99.3%

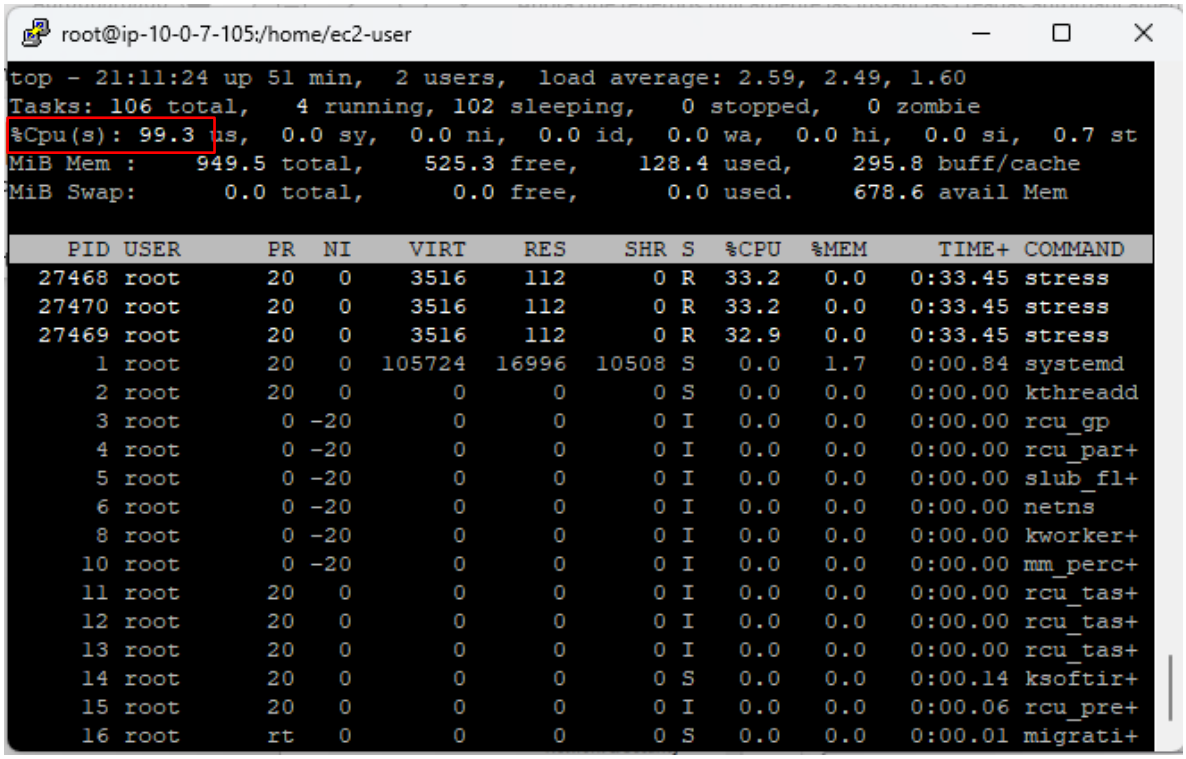


Imagen 10. Se verifica en la actividad del Auto Scaling Group que el proceso de detención de la instancia inicia

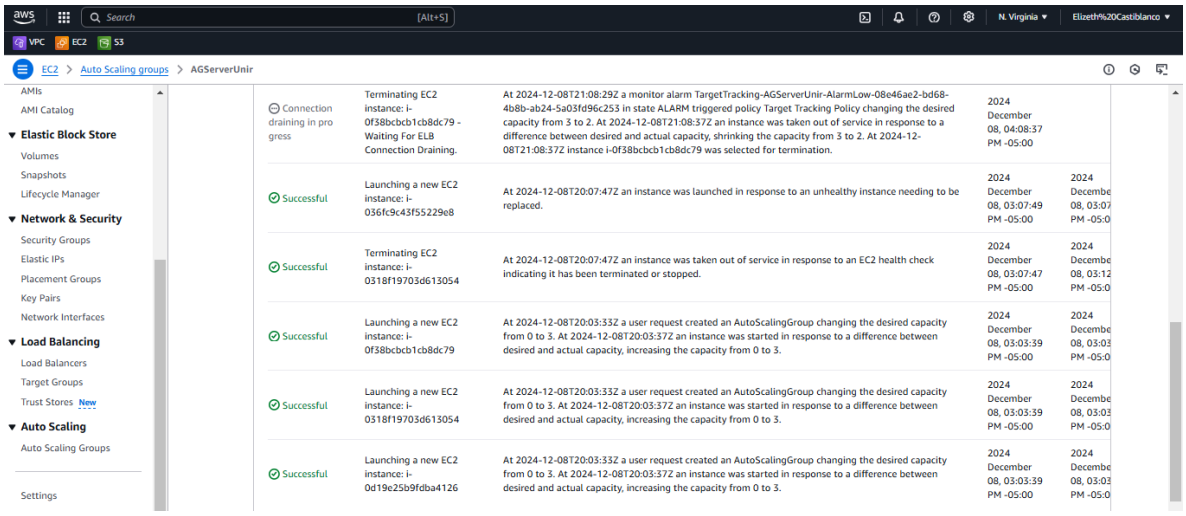


Imagen 11. Se verifica en el Target Group que ya dejo de funcionar una instancia

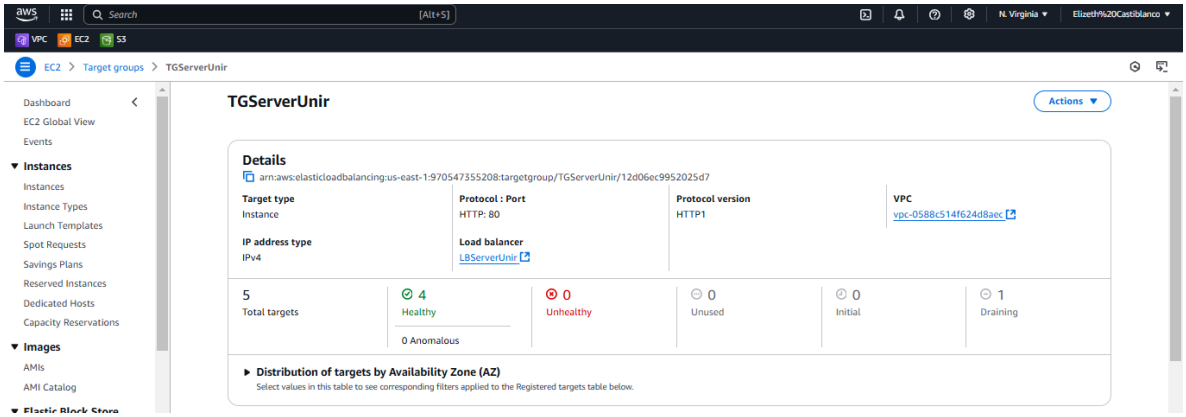


Imagen 12. Se verifica en el rendimiento de la instancia como el uso de CPU aumentó

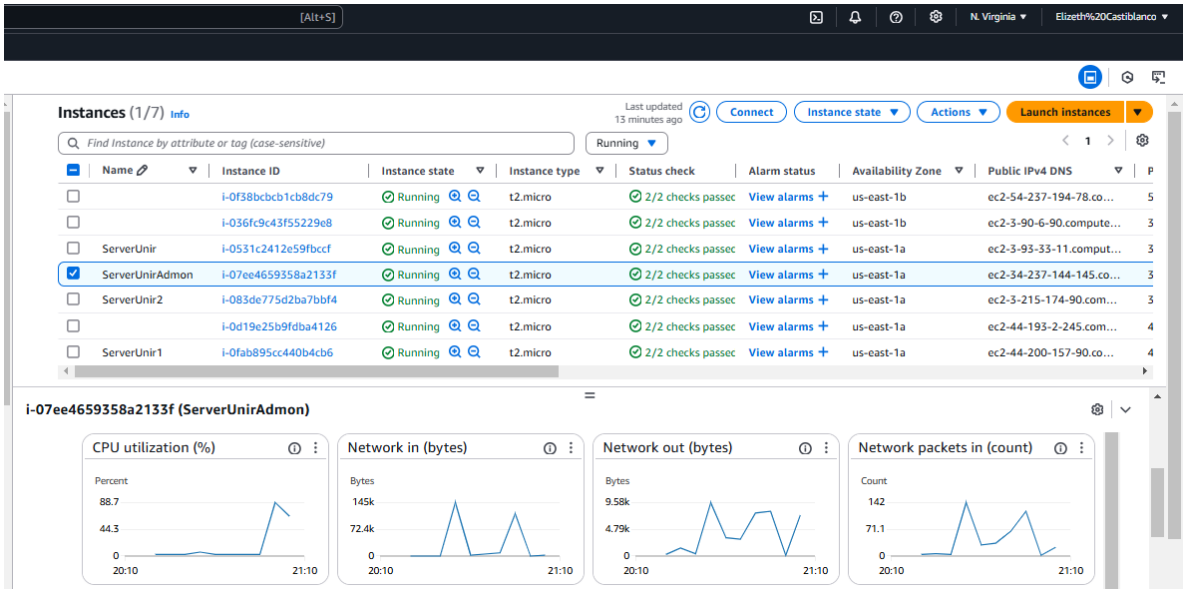


Imagen 13. Nuevamente en la actividad de la instancia se verifica que el proceso de finalización de la instancia terminó

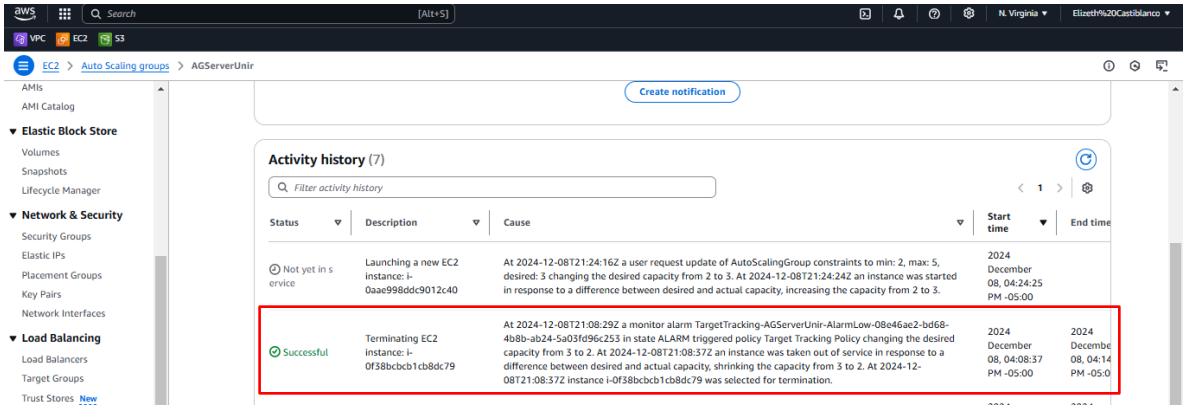


Imagen 14. Inicia automáticamente una nueva instancia en remplazo de la que dejo de funcionar por uso de CPU mayor al 25%

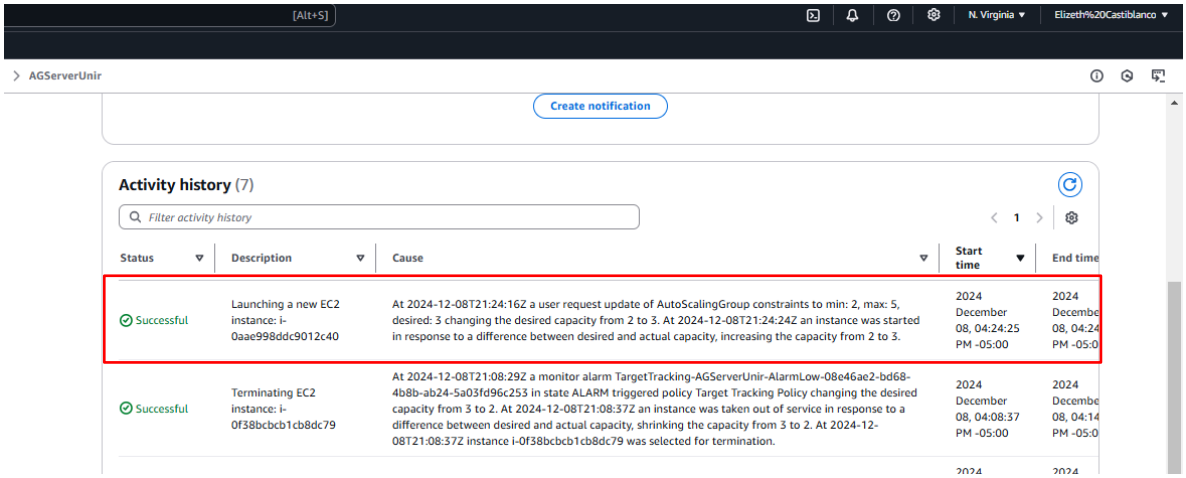
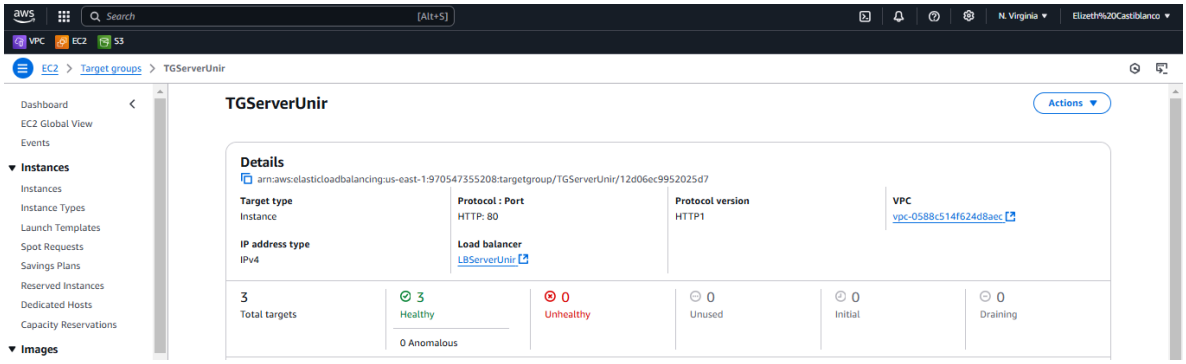


Imagen 15. Finalmente quedan las tres instancias que deben existir por defecto (según la configuración que se hizo) en estado saludable



Entrega III

Docker / Contenedores

Imagen 1. Se revisa el estado del servicio httpd

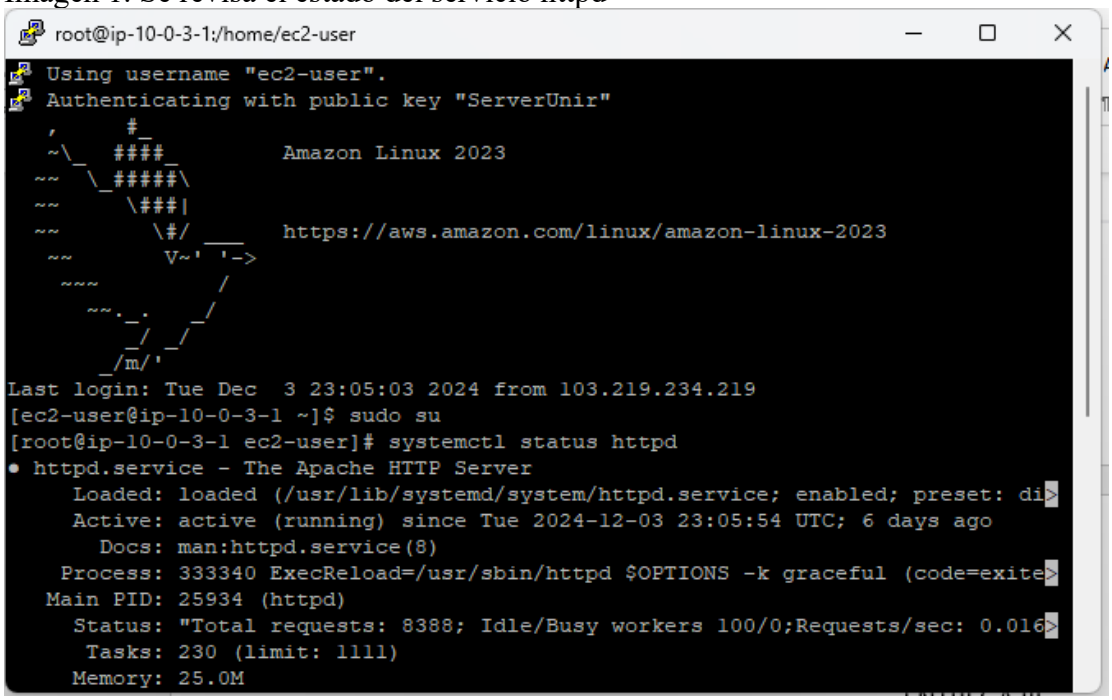


Imagen 2. Se detiene el servicio httpd

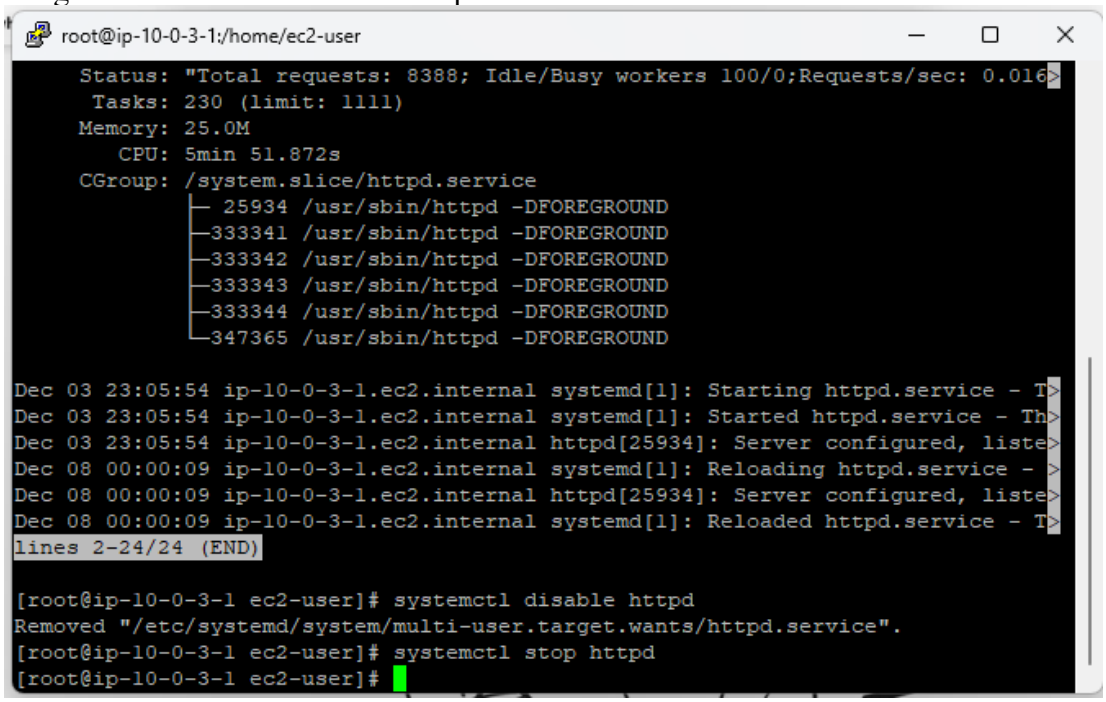


Imagen 3. Se instala Nginx

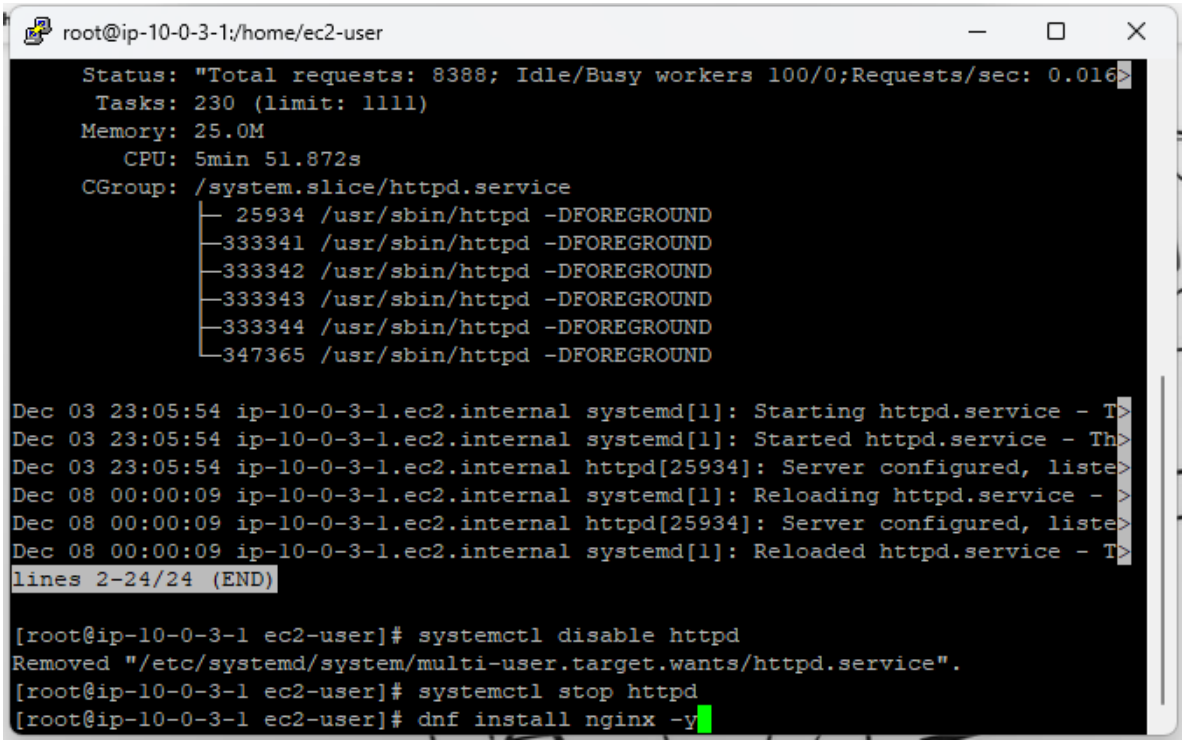


Imagen 4. Se verifica el estado del servicio nginx (en este caso esta inactivo)

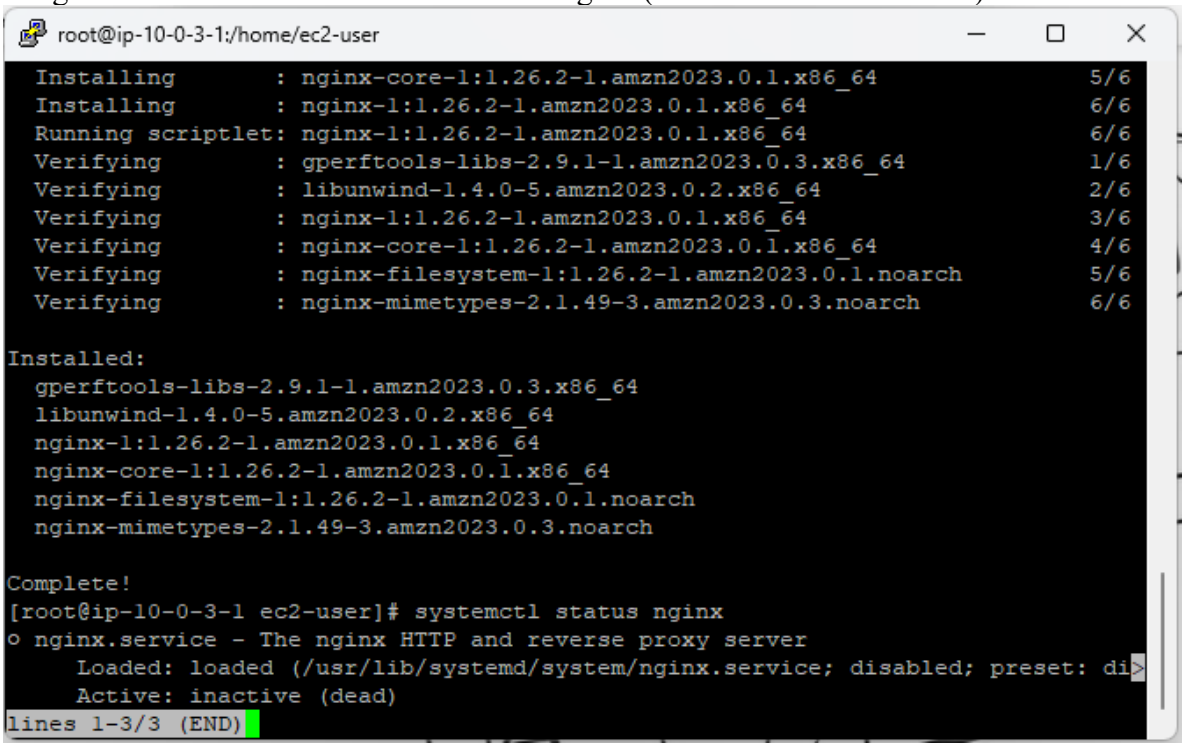


Imagen 5. Se arranca el servicio nginx y se ejecuta el comando enable para que las próximas veces arranque automáticamente

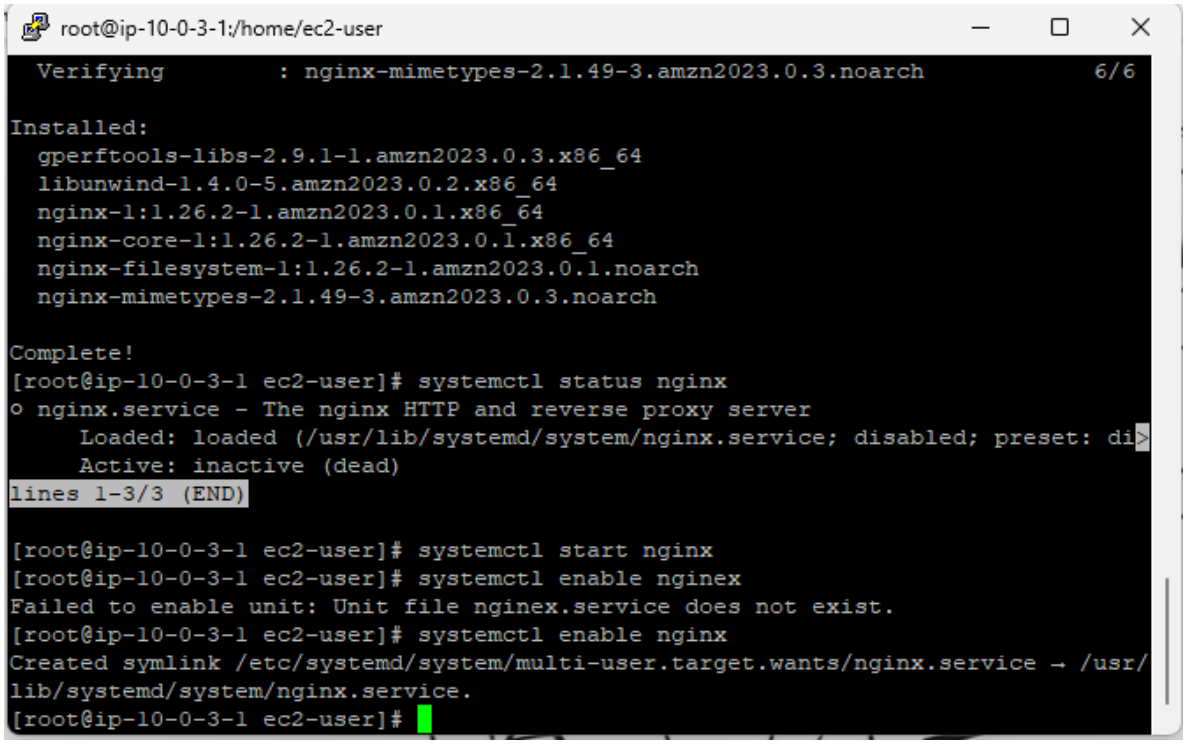


Imagen 6. Se confirma que el servidor funciona correctamente

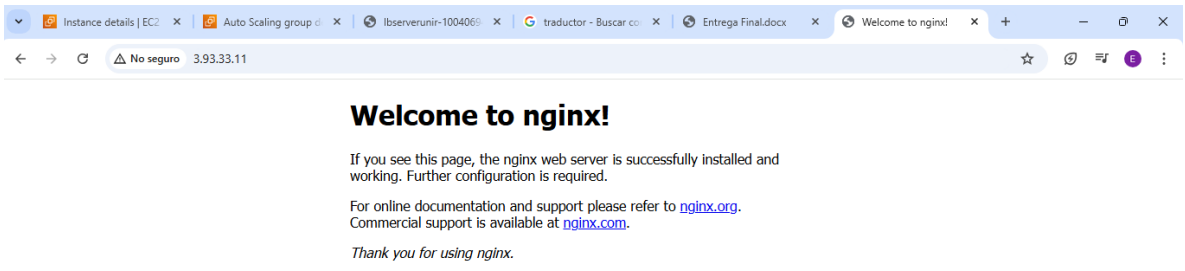


Imagen 7. Se instala Docker

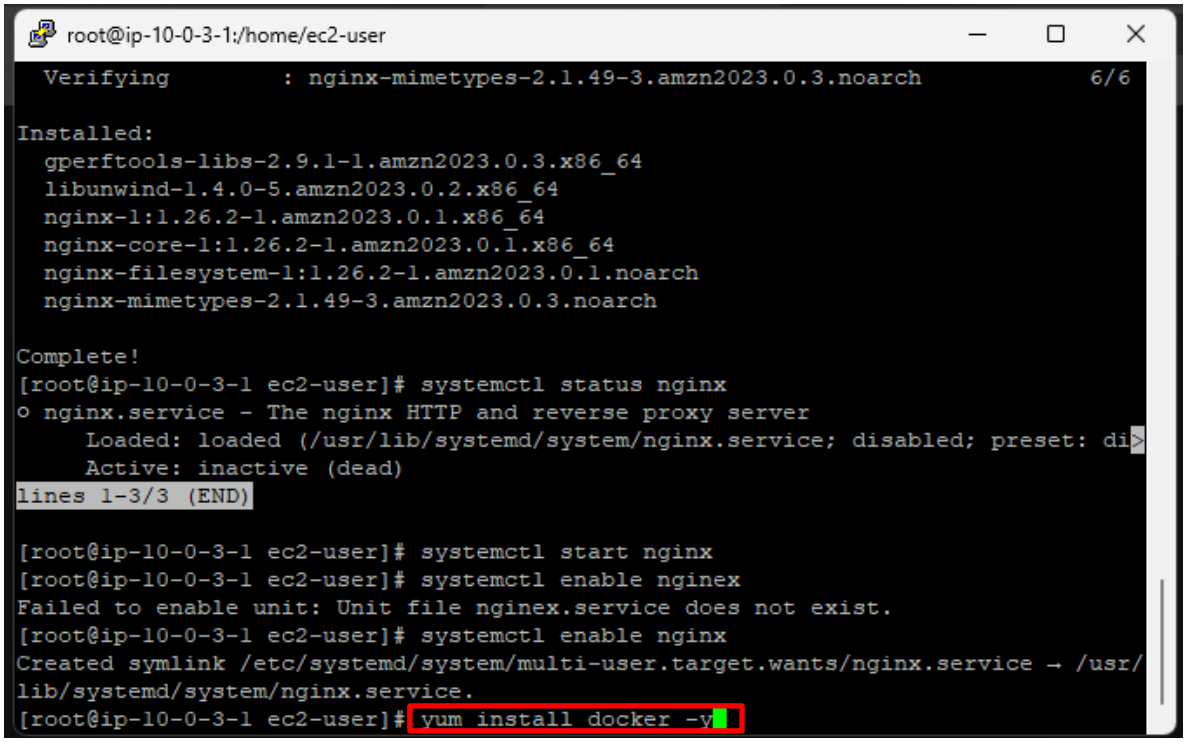


Imagen 8. Se activa el servicio y se ejecuta el comando enable para que arranque automáticamente las próximas veces

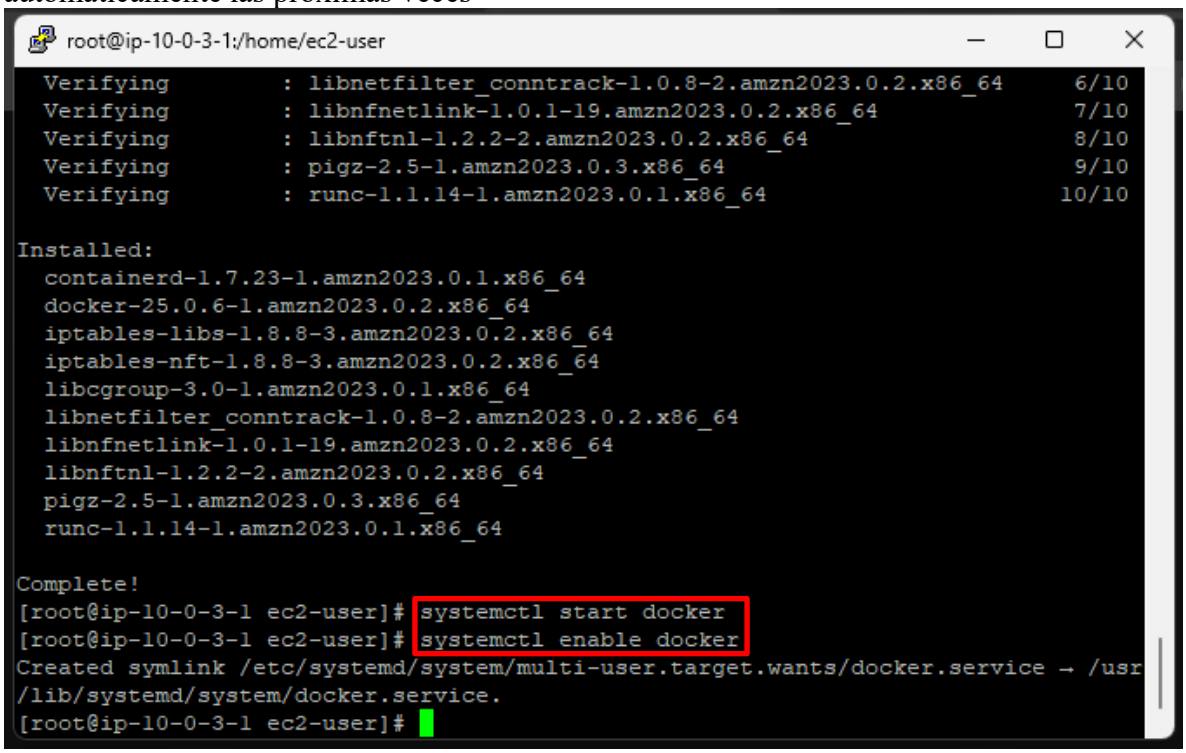


Imagen 9. Una vez Docker se ejecuta, se debe instalar un contenedor. Para ello se necesita primero montar una imagen que ejecuta un apache

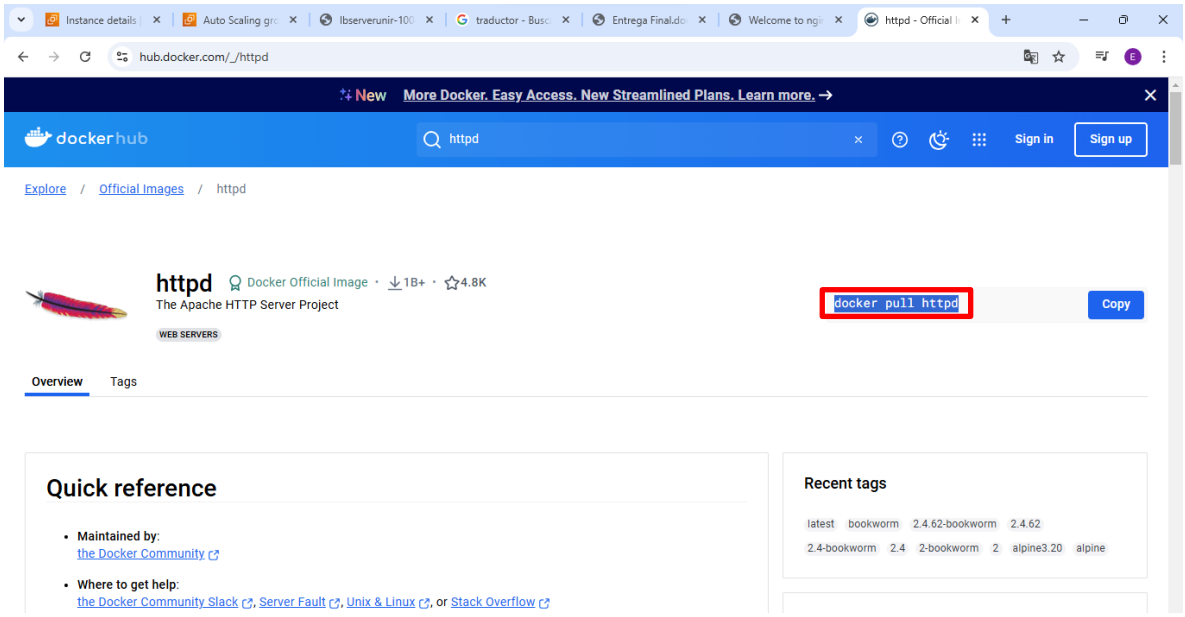


Imagen 10. Se ejecuta el código para subirla

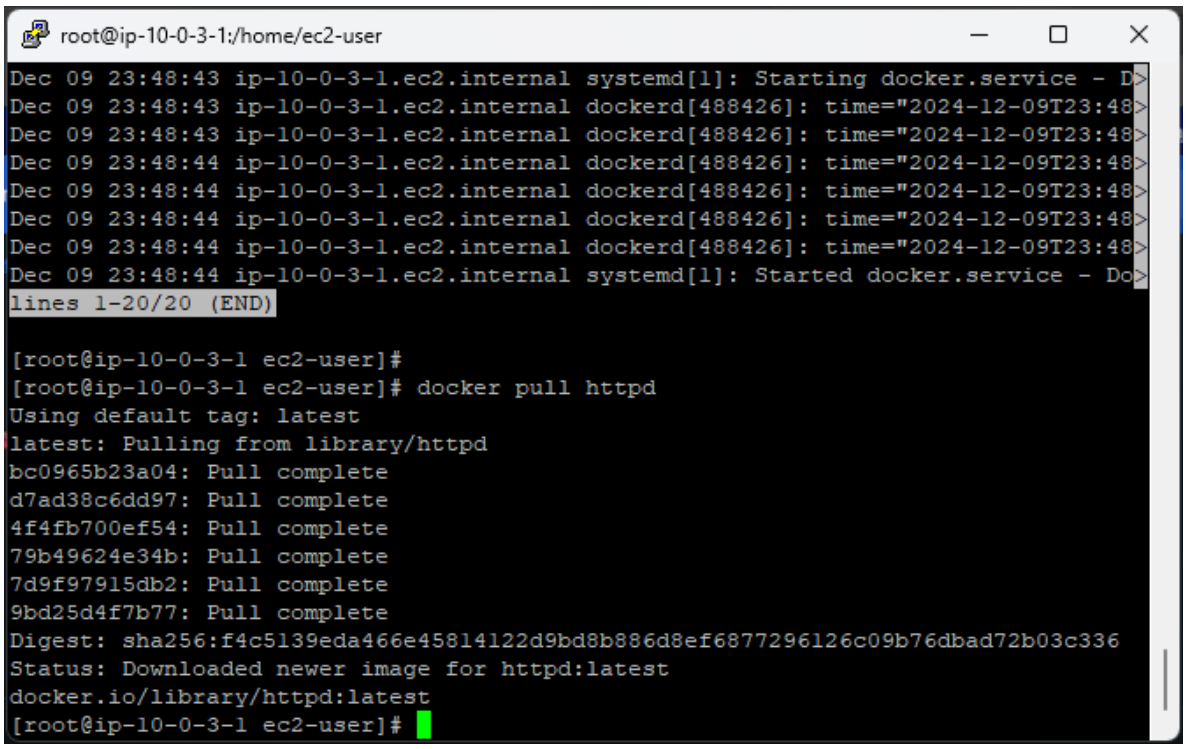


Imagen 11. Se verifican las imágenes que se tienen guardadas de forma local

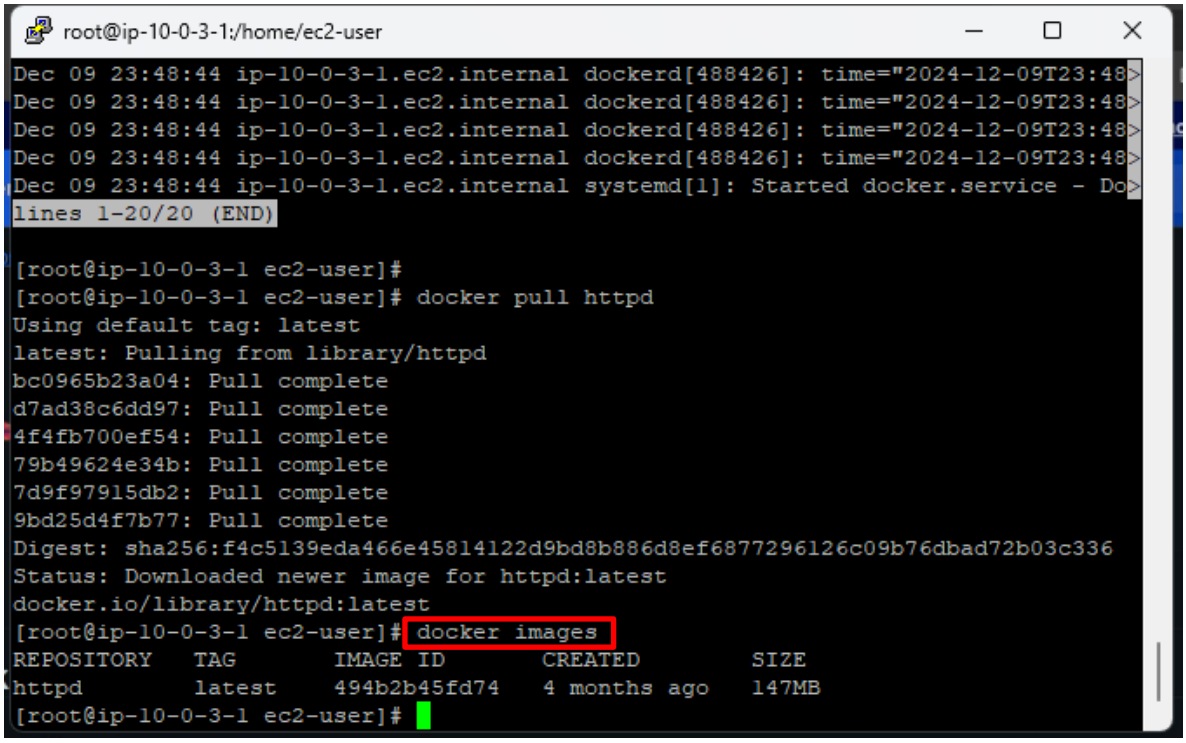


Imagen 12. Ahora se instala un contenedor:

-v: ruta (Disco duro) donde quiero que el servidor vaya y busque los archivos de mi aplicación, existe dentro del contenedor (la ruta se copia de la documentación de la misma imagen descargada anteriormente).

httpd: nombre de la imagen descargada

--restart always: hacer que el contenedor cuando se reinicie arranque de una vez y también arranque los contenedores existentes

- p 8080: puerto de ejecución del contenedor

- p 80: puerto donde la máquina virtual escucha

Finalmente debe aparecer el código de verificación

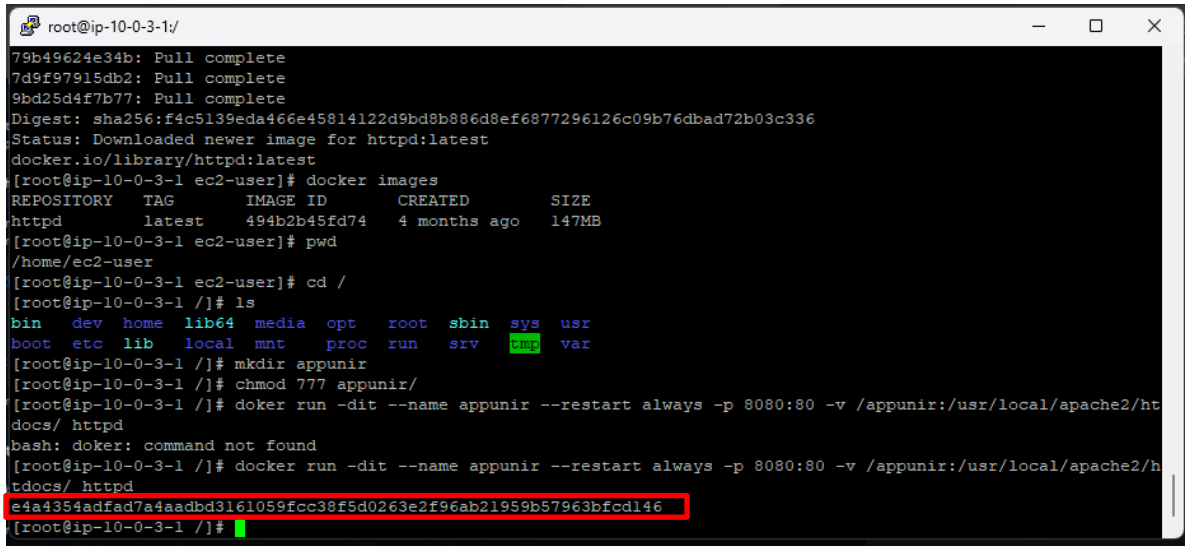


Imagen 13. Se debe detener el servicio de nginx y el contenedor. Se vuelve a iniciar el contenedor y con el comando docker ps se verifica que el contenedor se está ejecutando.

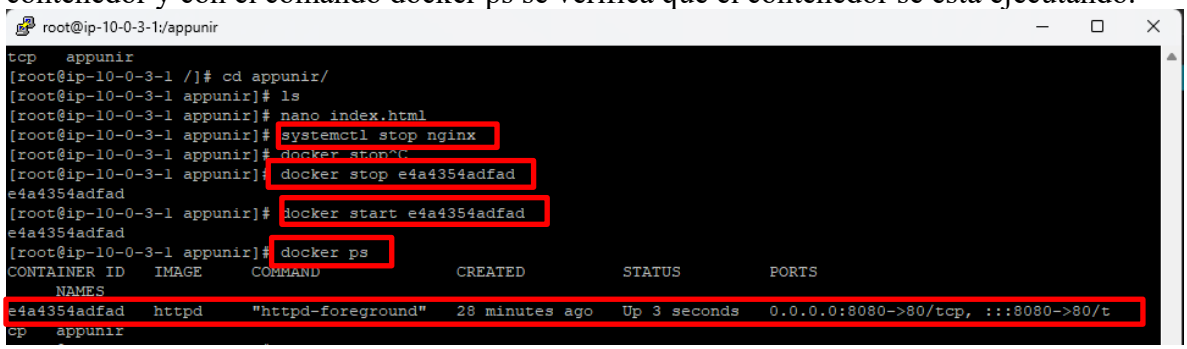


Imagen 14. Se verifican los permisos del archivo de la aplicación

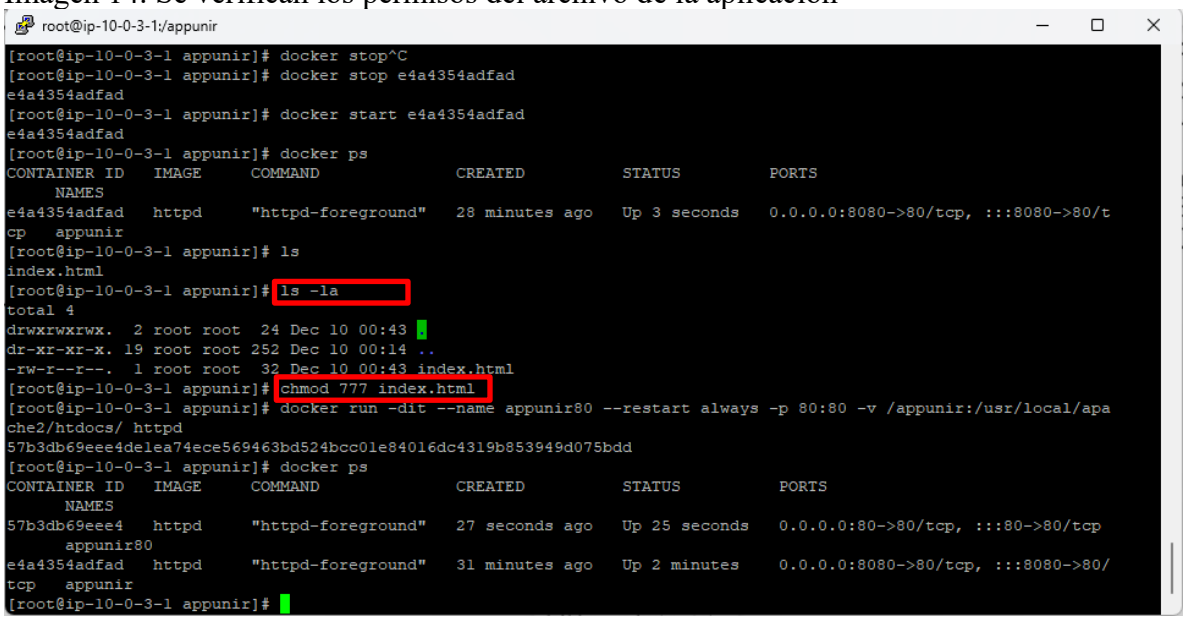


Imagen 15. Se crea otro contenedor que se ejecute en el puerto 80 y se crean mas sitios web para hacer la prueba

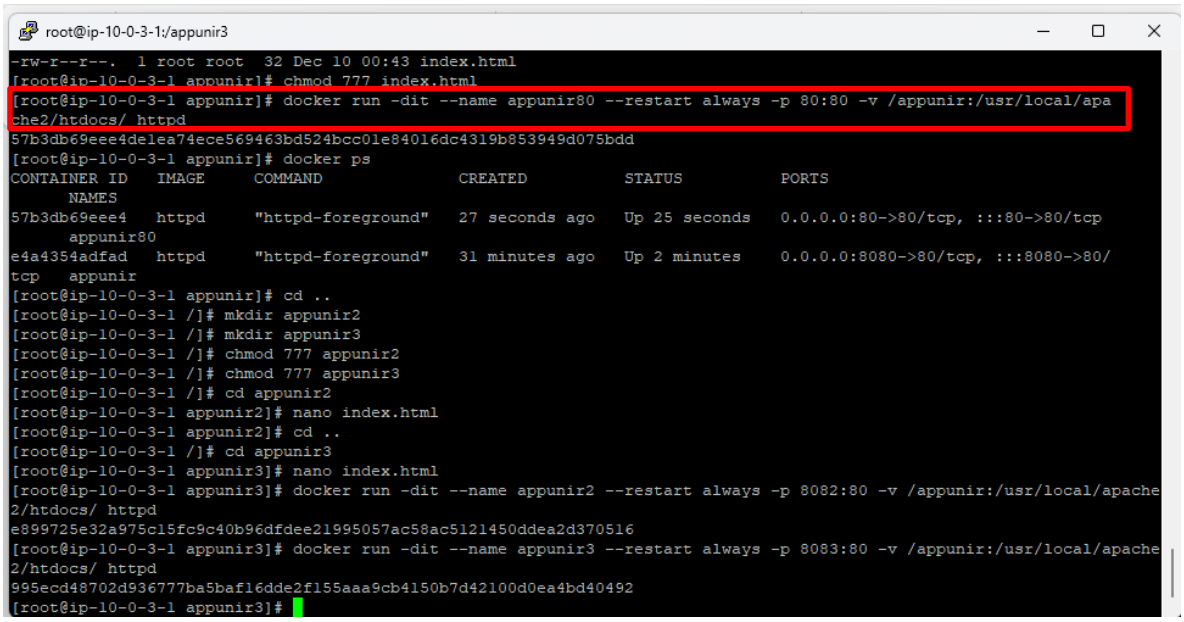


Imagen 16. Con los tres contenedores se realiza la prueba (los otros dos contenedores se crearon mal por lo tanto no se usan)

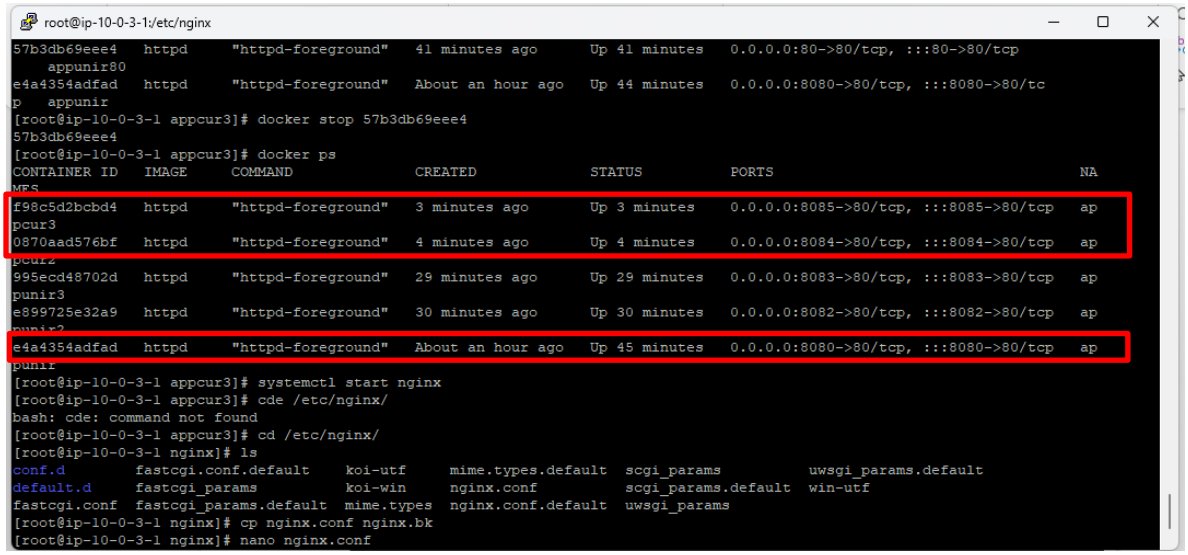


Imagen 17. Se configura el nginx que es el que se va a encargar de manejar/repartir las peticiones a los contenedores creados, lo que genera una alta disponibilidad y se hace antes un backup

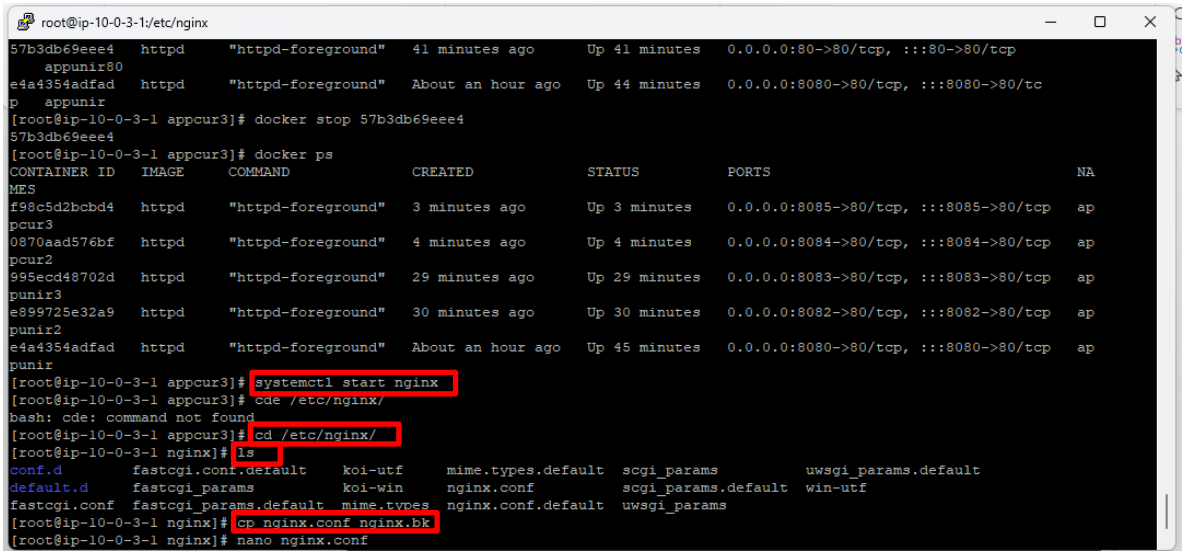


Imagen 18. Se elimina la configuración que trae por defecto nginx y se ingresa una nueva configuración básica que se requiere para un Proxy Inverso (depende de cada servidor)

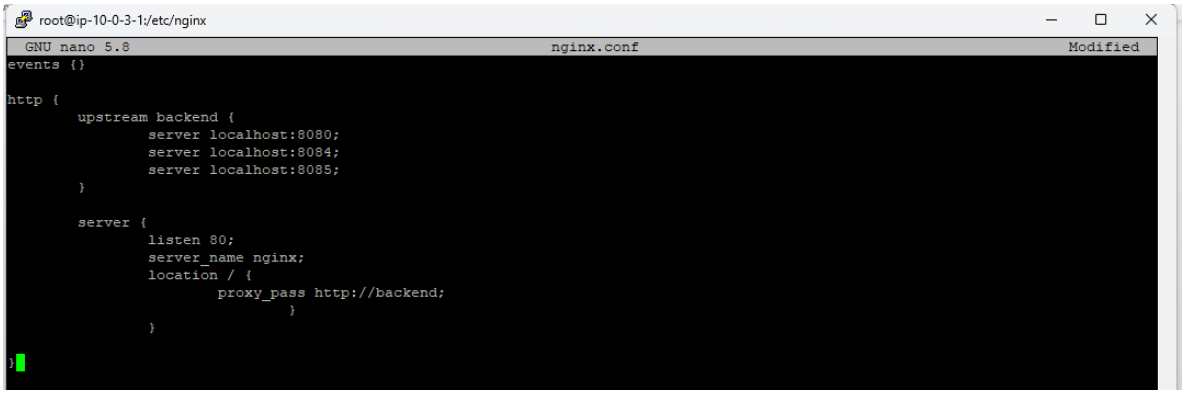


Imagen 19. Se reinicia el servicio de nginx

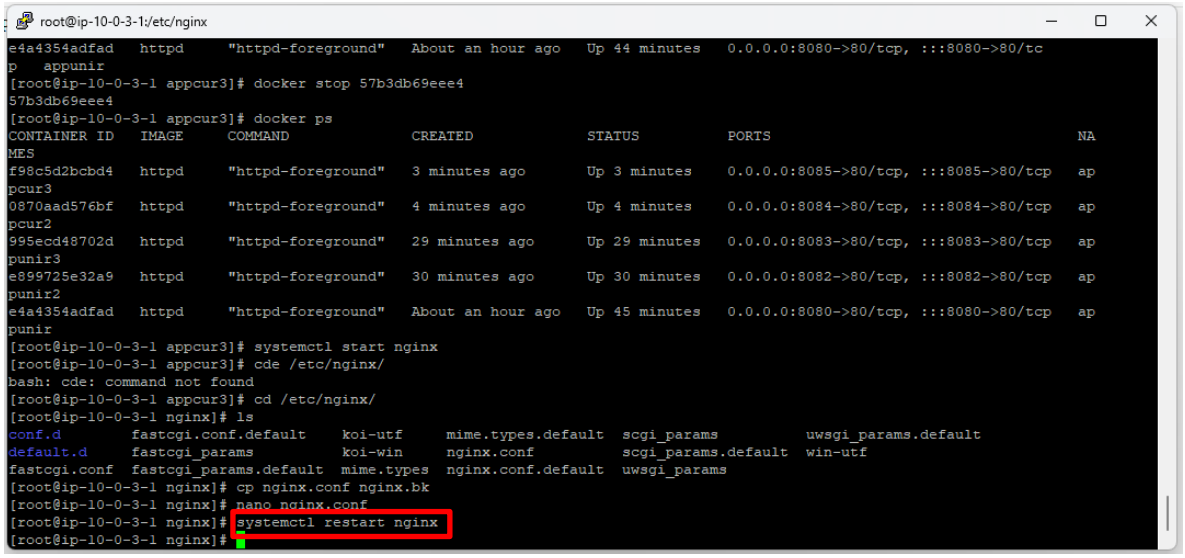
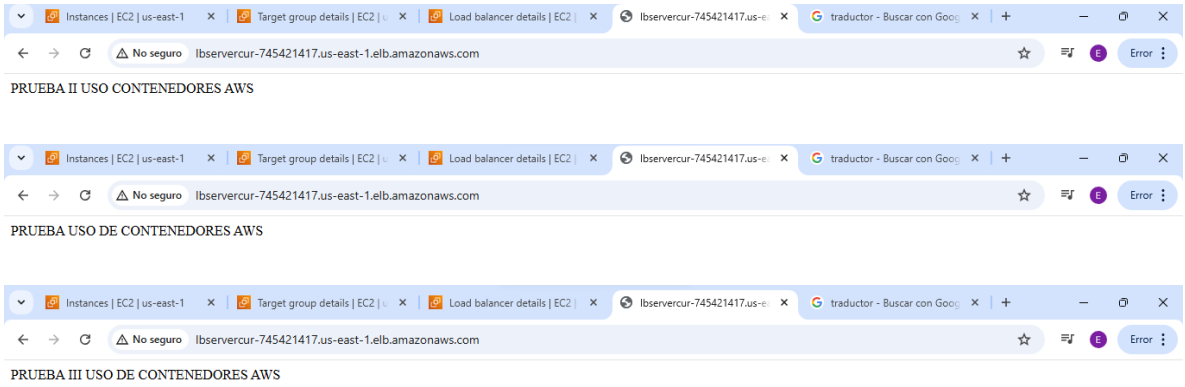


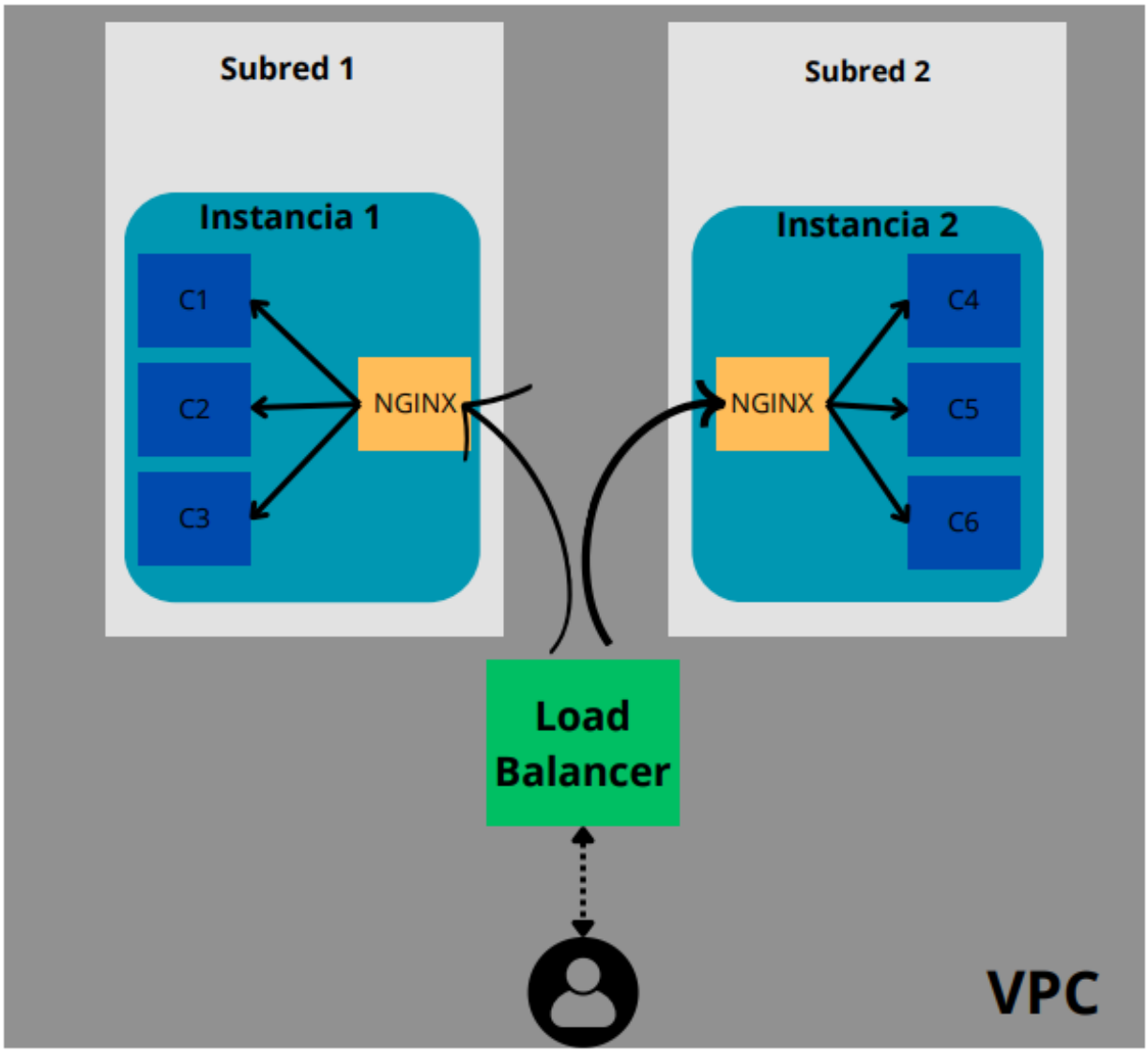
Imagen 20-21-22. Y finalmente debe iniciar a repartir las peticiones a cada uno de los contenedores creados



Video implementación de Docker en AWS

https://youtu.be/77fKx1tnRKQ

Incluya un diagrama de los recursos usados y como se comunican entre ellos



Link diagrama: https://www.canva.com/design/DAGY8OUfapQ/JrpjtXGrpwtqPMq9KF-Z3A/view?utm_content=DAGY8OUfapQ&utm_campaign=designshare&utm_medium=link2&utm_source=uniquelinks&utlId=hcf72428ca8

Conclusiones

La implementación de esta arquitectura ofrece servicios de alta disponibilidad y permite trabajar de manera mucho más eficiente, escalable, automatizada y monitorizada.

El aprendizaje y adaptación son necesarios, por lo que se requiere tiempo y dedicación pata poder empezar a hacer uso de estos servicios y sacar el mayor provecho.

El control de costos puede terminar siendo favorable ya que si se administran correctamente los gastos pueden ser más bajos.

Referencias

(Amazon Web Services, 2024)

(Amazon Web Services, 2024)