

TRABAJO DE GRADO
Opción Seminario-Diplomado.

Amazon Web Services - AWS

Corporación Universitaria Remington.

Facultad Ciencias Básicas e Ingeniería

Pregrado Ingeniería de Sistemas

Cristian M. Piedrahita y Rubén D. Salas.

Nombre del Tutor del trabajo de grado Juan Pablo Berrío López

Opción de Trabajo de grado Seminario-Diplomado.

2024

Tabla de Contenidos

Desarrollo e implementación del aprendizaje.....	6
Entrega 1	6
Orígenes de la Virtualización.....	6
Esta tecnología se origina en los años sesenta, durante la era de los enormes mainframes, cuando estas gigantescas máquinas solo podían llevar a cabo una tarea a la vez. Con el paso del tiempo, los clientes exigieron que estas inversiones costosas pudieran soportar la ejecución de múltiples usuarios o procesos al mismo tiempo.	6
IBM y los Mainframes:.....	7
VMware:	8
Microsoft:.....	9
Tipos de virtualización.....	9
Virtualización del almacenamiento:	10
Virtualización de aplicaciones:.....	10
Virtualización de escritorios:	10
Virtualización de red:.....	11
Beneficios y casos de uso de la virtualización.....	11
Ahorros de costes y eficiencia:	11
Escalabilidad y flexibilidad:	11
Continuidad del negocio y recuperación ante desastres:	11
Entornos de prueba y desarrollo:	12

	3
Componentes y tecnologías de virtualización:	12
Hipervisores:	12
Máquinas virtuales:	12
Conectividad de red virtual:	13
Tecnologías de virtualización del almacenamiento:	13
Orígenes de la Computación en la Nube.....	14
Salesforce:	15
Amazon Web Services (AWS)	15
Google:	15
Microsoft Azure	16
Figuras clave en la historia de la virtualización y la computación en la nube.....	16
J.C.R. Licklider:	16
Douglas Engelbart:	16
Joseph Weizenbaum:	16
Marc Benioff:	17
Diane Greene, Mendel Rosenblum, Edouard Bugnion y Scott Devine:	17
Jeff Bezos:	17
Línea de tiempo.....	18
Nombres de Servicios AWS, AZURE Y GCP	19
Entrega 2	22
EC2 (Amazon Elastic Compute Cloud).....	22
VPC (Amazon Virtual Private Cloud)	23

	4
Security Group	24
Amazon Linux 2 AWS.....	24
SSH (Secure Shell)	25
Docker.....	27
Instalación de Docker:	27
Apache	29
Virtual host.....	29
Configuración de apache para atender las solicitudes por nombres de dominio ..	30
Páginas Web.....	31
Sitio tienda.com	32
Sitio comercio.com	32
Entrega 3	35
Amazon Simple Storage Service (Amazon S3).....	35
Actividad.....	35
Amazon Relational Database Service (Amazon RDS).....	42
Entrega 4	47
Amazon Elastic Load Balancing (ELB):	47
AWS Auto Scaling:.....	47
Amazon EC2 Auto Scaling.....	47
Actividad.....	47
Índice de Tablas	72
Índice de ilustraciones.....	73

Conclusiones 76

Referencias..... 78

Desarrollo e implementación del aprendizaje

Entrega 1

Orígenes de la Virtualización

La virtualización es una tecnología que revoluciona la gestión de recursos de TI al liberar el potencial de hardware físico. A través de un software intermedio llamado hypervisor, se fragmentan las capacidades de una máquina física en múltiples entornos virtuales, maximizando su utilización. Este hypervisor, también conocido como monitor de máquina virtual (VMM), orquesta la creación y gestión de estas entidades virtuales en un solo servidor físico. Cada una de estas máquinas virtuales, independientes entre sí, aloja un sistema operativo huésped y sus respectivas aplicaciones, asegurando su aislamiento del hardware físico. Al interceptar las solicitudes del sistema operativo huésped hacia los recursos físicos, el hypervisor optimiza el uso de los mismos, actuando como un intermediario eficiente. En resumen, el hypervisor juega un papel esencial en la virtualización, facilitando la creación, gestión y protección de estos entornos virtuales, lo que conlleva a una consolidación de servidores y una óptima utilización de recursos. (García Pombo & Fernández Romero, 2011)

Esta tecnología se origina en los años sesenta, durante la era de los enormes mainframes, cuando estas gigantescas máquinas solo podían llevar a cabo una tarea a la vez. Con el paso del tiempo, los clientes exigieron que estas inversiones costosas pudieran soportar la ejecución de múltiples usuarios o procesos al mismo tiempo.

IBM y los Mainframes:

“A finales de los años 60, IBM desarrolló el CP-40 y posteriormente el CP-67, que eran sistemas de control de programa que permitían a un mainframe simular múltiples máquinas virtuales. Este enfoque fue revolucionario porque permitía a los usuarios ejecutar diferentes sistemas operativos en un solo hardware físico” (Creasy, 1981). Un hipervisor pionero que introdujo la memoria virtual en su línea de servidores System 360.

VM/32 de IBM: Una pionera en la virtualización: En 1972, IBM presentó VM/32, el primer sistema operativo de virtualización a gran escala. Esta tecnología permitía crear múltiples máquinas virtuales dentro de un solo mainframe, aislando cada una de las demás y permitiendo una mayor flexibilidad y eficiencia en la gestión de recursos.

Sin embargo, otras alternativas surgieron para facilitar que múltiples usuarios trabajaran en un solo servidor, y la virtualización quedó relegada a un papel de tecnología de nicho durante muchas décadas. En los años siguientes, emergieron otras soluciones para abordar la creciente demanda de usuarios en una única máquina, pero la virtualización no alcanzó la misma popularidad. Un ejemplo de estas soluciones fue el tiempo compartido, que permitía aislar a los usuarios dentro de los sistemas operativos y fue precursor de sistemas como UNIX, que es la base de Linux. Mientras tanto, la virtualización se mantuvo como una tecnología especializada. (redhat, 2023)

Durante los años 90, numerosas compañías se encontraron con el desafío de manejar plataformas de tecnología de la información proveniente de un único proveedor, junto con

aplicaciones heredadas. Esta situación resaltó la importancia de optimizar el uso de los recursos de sus servidores, que frecuentemente estaban subutilizados. Al adoptar la virtualización, no solo lograron mejorar la eficiencia en la distribución de su infraestructura de servidores, sino que también pudieron ejecutar sus aplicaciones heredadas en diversos tipos y versiones de sistemas operativos. (hpe, 2024)

La mayoría de las empresas se apoyaban en servidores físicos y sistemas de tecnología de la información de un único proveedor, lo que complicaba la ejecución de aplicaciones heredadas en hardware proveniente de diferentes proveedores. Al modernizar sus entornos de TI con servidores más económicos y diversos, el hardware era utilizado de manera ineficiente, dado que cada servidor estaba destinado a una tarea específica del proveedor (redhat, 2023)

VMware:

En 1999, VMware presentó su primer producto, VMware Workstation, fue pionero en la virtualización x86, la cual facultaba a los usuarios para ejecutar varios sistemas operativos de forma eficaz y segura en una sola máquina física. Este producto representó un momento crucial en la difusión generalizada de la virtualización tanto en el ámbito personal como empresarial de la informática (zippia, 2024). Luego, en 2001 VMware desarrolló tecnologías de virtualización de servidores con su producto VMware ESX Server, esta plataforma permitió la virtualización a nivel de servidor, lo que significa que múltiples sistemas operativos podían ser ejecutados simultáneamente en un solo servidor físico. Este avance revolucionó la forma en que se

administraban los centros de datos, proporcionando mayor eficiencia y flexibilidad en la utilización de recursos (VMware, 2021)

Microsoft:

Microsoft también contribuyó significativamente al desarrollo de la virtualización con su tecnología de virtualización Hyper-V, que fue lanzada inicialmente en 2008 como parte de Windows Server.

Aunque la virtualización se originó en los años sesenta, su adopción masiva comenzó a principios del 2000. Las tecnologías que permitieron la virtualización, como los hipervisores, se crearon hace muchas décadas para permitir que varios usuarios accedieran simultáneamente a computadoras que realizaban procesamiento por lotes. Este tipo de procesamiento, popular en el ámbito empresarial, ejecutaba tareas repetitivas rápidamente, como las nóminas. (redhat, 2023)

En consecuencia, la virtualización emergió como una solución fundamental para estas dificultades. Facilitó a las empresas la división de servidores y la ejecución de aplicaciones heredadas en múltiples sistemas operativos distintos, lo que mejoró la eficacia en la utilización del hardware y redujo los costos asociados con la adquisición, instalación, refrigeración y mantenimiento (redhat, 2023)

Tipos de virtualización

Virtualización de servidores:

Es la forma más extendida de aplicar tecnología de virtualización en el mercado actual. Dado que los servidores están optimizados para procesar una gran cantidad de tareas, la partición

de estas tareas para que los componentes puedan ser utilizados de manera más eficaz para múltiples funciones en todo el sistema puede resultar muy beneficioso para muchas organizaciones.

Virtualización del almacenamiento:

Consiste en un conjunto de servidores administrados por un sistema de almacenamiento virtual. Este sistema administra el almacenamiento procedente de diversas fuentes y lo trata como un conjunto único de almacenamiento, incluso si existen discrepancias de hardware entre los sistemas anfitriones. Esta virtualización simplifica las tareas de copia de seguridad, almacenamiento y recuperación.

Virtualización de aplicaciones:

Separa la aplicación del sistema operativo y del hardware en el que opera. En general, el usuario final accede a aplicaciones virtualizadas mediante un cliente liviano, mientras que la aplicación misma se ejecuta en un servidor del centro de datos conectado a través de Internet. Esto puede simplificar la ejecución de aplicaciones que necesitan versiones anteriores del sistema operativo, pero también puede representar un riesgo para otros recursos del sistema.

Virtualización de escritorios:

llamada infraestructura de escritorios virtuales (VDI), representa un entorno de escritorio del usuario en un sistema basado en software al que se puede acceder remotamente a través de Internet. Todos los componentes del entorno de trabajo físico se encuentran almacenados en el servidor, lo que proporciona a los usuarios finales una experiencia uniforme, independientemente del dispositivo que utilicen. Toda la información y los programas del usuario residen en el servidor principal, no en el dispositivo del usuario final.

Virtualización de red:

Desacopla las redes virtuales de la infraestructura física subyacente. Los switches virtuales gestionan todos los aspectos de la red. Esta virtualización de red simplifica la tarea de los administradores para asignar y distribuir recursos, lo que conlleva a un rendimiento de red más amplio y estable. (hpe, 2024)

Beneficios y casos de uso de la virtualización***Ahorros de costes y eficiencia:***

- La virtualización ahorra recursos y costos al consolidar la infraestructura y disminuir los gastos relacionados con el hardware.
- Incrementa la utilización de recursos, la eficiencia energética y optimiza la administración de la infraestructura para reducir costos

Escalabilidad y flexibilidad:

- La virtualización posibilita la rápida provisión de recursos virtuales, lo que agiliza la capacidad de escalabilidad para adaptarse a los cambios en la demanda.
- Ofrece una asignación de recursos adaptable y rentable al incrementar o disminuir los recursos de forma eficiente según los requisitos de la carga de trabajo.

Continuidad del negocio y recuperación ante desastres:

- La virtualización asegura la seguridad de los datos y una disponibilidad óptima mediante funcionalidades como la replicación y la migración en tiempo real, lo que asegura la continuidad de las operaciones comerciales.

- Facilita la implementación de estrategias de recuperación ante desastres, como la conmutación por error de sitios, reduciendo al mínimo el tiempo de inactividad y asegurando la integridad de los datos.

Entornos de prueba y desarrollo:

- La virtualización simplifica la configuración de entornos de prueba independientes, disminuyendo las posibles discrepancias y asegurando resultados exactos.
- Agiliza los procesos de prueba y desarrollo mediante la rápida provisión y replicación de entornos virtuales, lo que mejora la eficacia de las pruebas y el desarrollo de software. (hpe, 2024)

Componentes y tecnologías de virtualización:

Hipervisores:

- Los hipervisores de tipo 1 (bare metal) se ejecutan directamente en el hardware del servidor, mientras que los hipervisores de tipo 2 (hosted) se ejecutan sobre un sistema operativo ya existente.
- Los hipervisores ofrecen una variedad de funciones, capacidades de gestión y opciones de proveedores, lo que proporciona flexibilidad y control sobre los entornos virtualizados.

Máquinas virtuales:

- Las máquinas virtuales son representaciones de software de computadoras físicas que pueden ejecutar sistemas operativos y aplicaciones.

- La creación, gestión y migración de máquinas virtuales se lleva a cabo mediante software de virtualización, lo que proporciona flexibilidad, portabilidad y una asignación eficiente de recursos.

Conectividad de red virtual:

- Los switches virtuales, las redes LAN virtuales (VLAN) y los enrutadores virtuales permiten la segmentación y la conexión de redes en entornos virtualizados.
- La configuración y la conectividad de la red en entornos virtualizados se controlan mediante componentes de conectividad de red virtuales, brindando flexibilidad y control.

Tecnologías de virtualización del almacenamiento:

- Las infraestructuras de almacenamiento, como las redes de área de almacenamiento (SAN) y el almacenamiento conectado a la red (NAS), suministran recursos de almacenamiento para entornos virtualizados.
- Las plataformas de virtualización de almacenamiento y las soluciones de almacenamiento definido por software se encargan de abstraer y administrar los recursos de almacenamiento, facilitando un almacenamiento y una gestión de datos eficientes. (hpe, 2024).

La amplia adopción de la virtualización no solo redujo la dependencia de un único proveedor, sino que también estableció los fundamentos del cloud computing. En la actualidad, su

uso se ha generalizado tanto que muchas empresas requieren software especializado para administrar la virtualización. (redhat, 2023).

Orígenes de la Computación en la Nube

Computación en la nube es una expresión general que abarca la provisión de recursos de computación mediante Internet. Los usuarios pueden acceder a una considerable capacidad de procesamiento según su necesidad, adquiriéndola por lapsos cortos de tiempo y utilizando únicamente la cantidad requerida para sus tareas informáticas (Amazon, 2024)

La computación en la nube consiste en la disponibilidad casi instantánea de recursos de procesamiento a través de servicios en línea. Este modelo elimina la obligación de que las empresas adquieran, configuren o gestionen sus propios recursos, permitiéndoles pagar únicamente por el uso que hagan de ellos (Google Cloud, 2024).

La computación en la nube, como la conocemos hoy en día, tiene sus raíces en una combinación de avances tecnológicos, sus orígenes en conceptos como el tiempo compartido y ha evolucionado gracias modelos de negocios innovadores y necesidades de la industria. Si bien no hay un consenso absoluto sobre su origen exacto, podemos rastrear sus antecedentes a varios hitos importantes en la historia de la tecnología de la información.

Uno de los primeros conceptos que sentaron las bases para la computación en la nube fue la idea de tiempo compartido, que surgió en la década de 1960. Este concepto permitía a múltiples usuarios acceder a una única computadora central a través de terminales remotas, lo que optimizaba los recursos de computación y mejoraba la eficiencia del procesamiento de datos (Cerf, 1982).

El término "computación en la nube" se popularizó a principios del siglo XXI, aunque sus fundamentos se remontan décadas atrás. Una de las primeras instancias documentadas del término apareció en un documento interno de Google en 2006, donde se describía el concepto de "computación en nube" como una forma de proporcionar servicios basados en Internet (Armbrust, y otros, 2010).

Salesforce:

Fue uno de los primeros en ofrecer software como servicio (SaaS) con su plataforma Salesforce.com en 1999. Este enfoque ayudó a popularizar el concepto de software basado en la nube.

Amazon Web Services (AWS)

En 2006 también marcó un hito importante en la historia de la computación en la nube. AWS ofrecía una variedad de servicios de infraestructura, como almacenamiento, bases de datos y capacidad informática, a través de Internet, lo que permitía a las empresas externalizar sus necesidades de infraestructura de TI de manera flexible y escalable (Amazon, s.f.). Ampliamente reconocido como uno de los pioneros de cloud computing.

Google:

En 2006 presenta Google Apps, una suite de aplicaciones de oficina basadas en la nube (ahora conocida como Google Workspace) (Google Workspace, 2023). Jugó un papel importante en la evolución de la computación en la nube con el lanzamiento de Google App Engine en 2008, que permitía a los desarrolladores crear y ejecutar aplicaciones en la infraestructura de Google; en este mismo año se lanza Google Cloud Platform (GCP) se une a la competencia, ofreciendo una

alternativa a AWS con enfoque en la innovación y el análisis de datos (Google Cloud Platform, 2023).

Microsoft Azure

Ingresa al mercado, brindando a las empresas una opción familiar y segura para la computación en la nube (Microsoft Azure, 2023).

Desde entonces, la computación en la nube ha evolucionado rápidamente, con la introducción de nuevos servicios, modelos de implementación y tecnologías emergentes. Hoy en día, es una parte integral de la infraestructura de TI para muchas organizaciones en todo el mundo, ofreciendo beneficios como la agilidad, la escalabilidad y la eficiencia operativa (Gartner, 2021).

Figuras clave en la historia de la virtualización y la computación en la nube

J.C.R. Licklider:

"Padre del concepto de la nube" con su visión de una red de computadoras interconectadas (Subrahmanyam & Buyya, 2010)

Douglas Engelbart:

Pionero de la interacción hombre-computadora y creador del concepto de "escritorio virtual" (Barroso & Cidon, 2009).

Joseph Weizenbaum:

Creador de ELIZA, uno de los primeros programas de inteligencia artificial que simulaba la conversación humana (BBC, 2018)

Marc Benioff:

Fundador de Salesforce, considerado uno de los "padres fundadores" de la computación en la nube (Subrahmanyam, R., & Buyya, R., 2010).

Diane Greene, Mendel Rosenblum, Edouard Bugnion y Scott Devine:

Fundadores de VMware, cuyo trabajo pionero en virtualización fue fundamental para el desarrollo de la virtualización moderna.

Jeff Bezos:

Fundador de Amazon, cuya visión ayudó a establecer AWS como un líder en el mercado de la computación en la nube.

Línea de tiempo

Grandes hitos de la computación en la nube

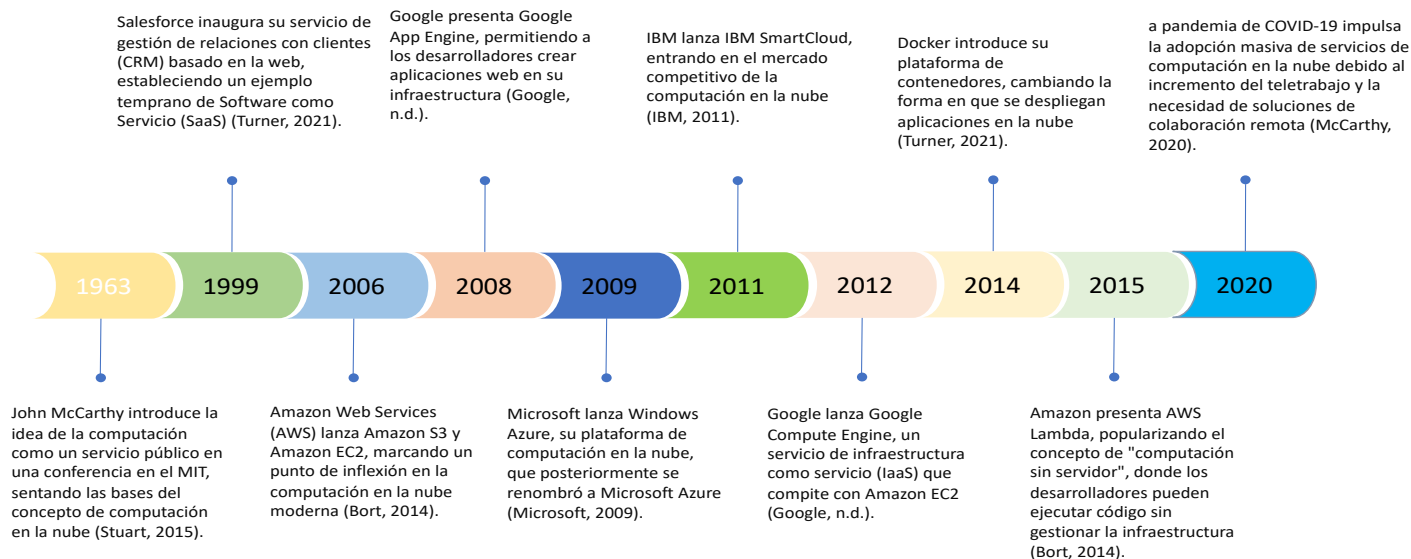


Ilustración 1 Línea de tiempo Hitos Computación en la nube

Nombres de Servicios AWS, AZURE Y GCP

Tabla 1 Comparativa Nombre de Servicios Cloud

Servicio	AWS (AWS, 2024)	Azure (AZURE, 2024)	GCP (google cloud, 2024)
Computación			
Máquinas Virtuales	EC2 (Elastic Compute Cloud)	Virtual Machines	Compute Engine
Almacenamiento			
Almacenamiento de Objetos	S3 (Simple Storage Service)	Blob Storage	Cloud Storage
Almacenamiento en Bloques	EBS (Elastic Block Store)	Managed Disks	Persistent Disks
Bases de Datos			
Base de Datos Relacional	RDS (Relational Database Service)	SQL Database	Cloud SQL
Base de Datos NoSQL	DynamoDB	Cosmos DB	Firestore / Datastore
Redes y Entrega de Contenidos			
Red Privada Virtual	VPC (Virtual Private Cloud)	Virtual Network	Virtual Private Cloud (VPC)
Entrega de Contenidos	CloudFront	CDN (Content Delivery Network)	Cloud CDN

Contenedores y Orquestación			
Servicio de Kubernetes	EKS (Elastic Kubernetes Service)	AKS (Azure Kubernetes Service)	GKE (Google Kubernetes Engine)
IA y Machine Learning			
Plataforma de Machine Learning	SageMaker	Azure Machine Learning	AI Platform
Monitorización y Gestión			
Monitorización y Logging	CloudWatch	Azure Monitor	Cloud Monitoring
Herramientas de Desarrollo			
CI/CD	CodePipeline	Azure DevOps	Cloud Build
Seguridad y Gestión de Identidades			
Gestión de Identidades	IAM (Identity and Access Management)	Azure Active Directory	IAM (Identity and Access Management)
Servicios de Mensajería y Colas			

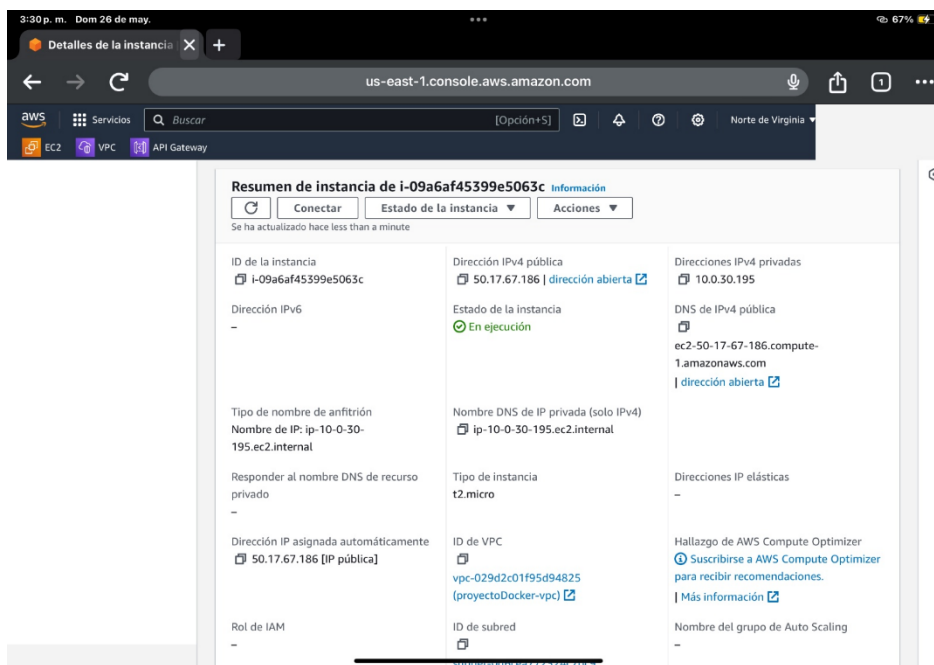
Colas de Mensajería	SQS (Simple Queue Service)	Queue Storage	Pub/Sub
Big Data y Análisis			
Plataforma de Big Data	EMR (Elastic MapReduce)	HDInsight	Dataproc

Entrega 2

EC2 (Amazon Elastic Compute Cloud)

Permite a los usuarios alquilar computadores virtuales en los cuales pueden ejecutar sus propias aplicaciones. (Amazon AWS, s.f.)

De acuerdo con las indicaciones compartidas en las tutorías se procedió a la creación de una instancia EC2, previa creación de una cuenta en AWS fue seleccionada una maquina tipo T2 Micro que ofrece la capa gratuita, se desplego una AMI con Amazon Linux 2023, se le asigno una IP publica para poder acceder desde el exterior, así como una dirección IP privada de la VPC



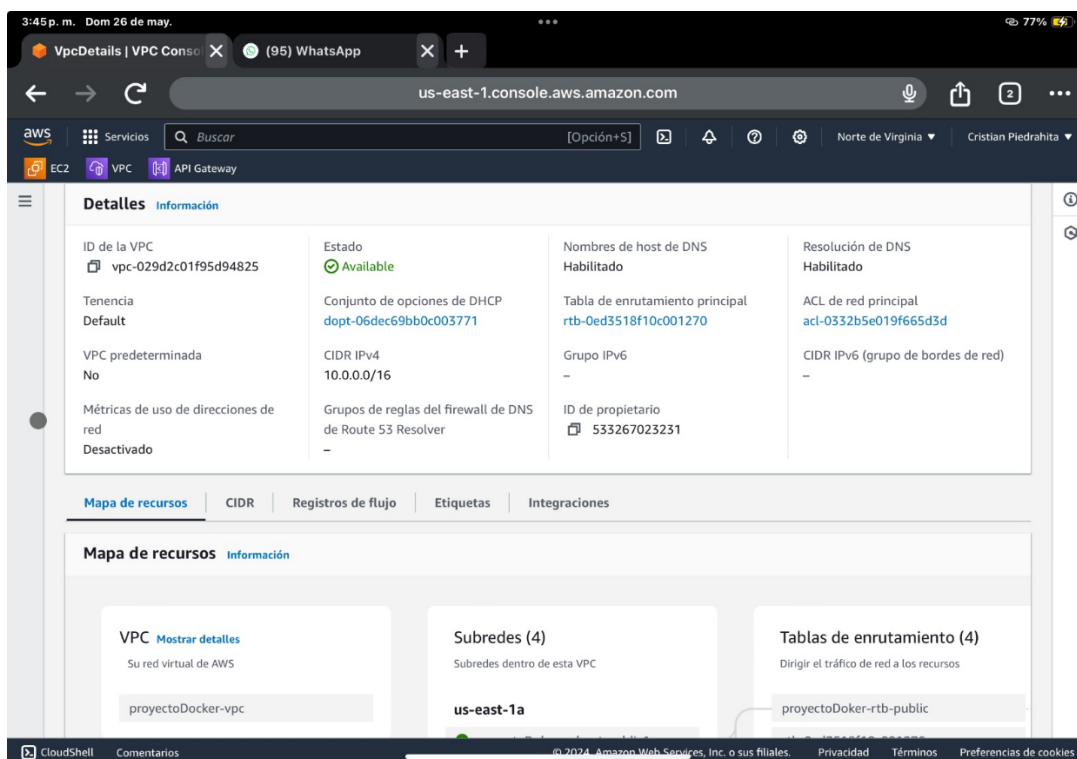
The screenshot displays the AWS Management Console interface for an EC2 instance. The browser address bar shows 'us-east-1.console.aws.amazon.com'. The page title is 'Resumen de instancia de i-09a6af45399e5063c'. The instance is in the 'En ejecución' (Running) state. Key details include:

Property	Value
ID de la instancia	i-09a6af45399e5063c
Dirección IPv4 pública	50.17.67.186 dirección abierta
Direcciones IPv4 privadas	10.0.30.195
Dirección IPv6	-
Estado de la instancia	En ejecución
DNS de IPv4 pública	ec2-50-17-67-186.compute-1.amazonaws.com dirección abierta
Tipo de nombre de anfitrión	Nombre de IP: ip-10-0-30-195.ec2.internal
Nombre DNS de IP privada (solo IPv4)	ip-10-0-30-195.ec2.internal
Responder al nombre DNS de recurso privado	-
Tipo de instancia	t2.micro
Direcciones IP elásticas	-
Dirección IP asignada automáticamente	50.17.67.186 [IP pública]
ID de VPC	vpc-029d2c01f95d94825 (proyectoDocker-vpc)
Hallazgo de AWS Compute Optimizer	Suscribirse a AWS Compute Optimizer para recibir recomendaciones. Más información
Rol de IAM	-
ID de subred	-
Nombre del grupo de Auto Scaling	-

Ilustración 2 Resumen de instancia Creada

VPC (Amazon Virtual Private Cloud)

Se estableció el segmento de red que va a permitir la conexión a la red al interior de la nube de AWS y las respectivas subredes. (Amazon AWS, s.f.)



The screenshot displays the AWS Management Console interface for a VPC. The browser address bar shows 'us-east-1.console.aws.amazon.com'. The page title is 'VpcDetails | VPC Console'. The main content area is titled 'Detalles Información' and contains a table of VPC details:

ID de la VPC vpc-029d2c01f95d94825	Estado Available	Nombres de host de DNS Habilitado	Resolución de DNS Habilitado
Tenencia Default	Conjunto de opciones de DHCP dopt-06dec69bb0c003771	Tabla de enrutamiento principal rtb-0ed3518f10c001270	ACL de red principal acl-0332b5e019f665d3d
VPC predeterminada No	CIDR IPv4 10.0.0.0/16	Grupo IPv6 -	CIDR IPv6 (grupo de bordes de red) -
Métricas de uso de direcciones de red Desactivado	Grupos de reglas del firewall de DNS de Route 53 Resolver -	ID de propietario 533267023231	

Below the details table, there are tabs for 'Mapa de recursos', 'CIDR', 'Registros de flujo', 'Etiquetas', and 'Integraciones'. The 'Mapa de recursos' tab is active, showing a resource map with three main components:

- VPC**: Su red virtual de AWS. Includes a sub-resource 'proyectoDocker-vpc'.
- Subredes (4)**: Subredes dentro de esta VPC. Includes a sub-resource 'us-east-1a'.
- Tablas de enrutamiento (4)**: Dirigir el tráfico de red a los recursos. Includes a sub-resource 'proyectoDocker-rtb-public'.

The footer of the console shows '© 2021 Amazon Web Services, Inc. o sus filiales.' and links for 'Privacidad', 'Términos', and 'Preferencias de cookies'.

Ilustración 3 VPC creada

Security Group

Se establecieron reglas básicas en el security group (funciona como un Firewall virtual para permitir o negar las conexiones entrantes y salientes) (Amazon AWS, s.f.) para permitir el acceso desde el exterior a los servicios SSH (puerto TCP 11), HTTP (puerto TCP 80), puertos TCP 8080, 8081 que serán usados para las pruebas de funcionamiento de las páginas web que estarán en los contenedores.

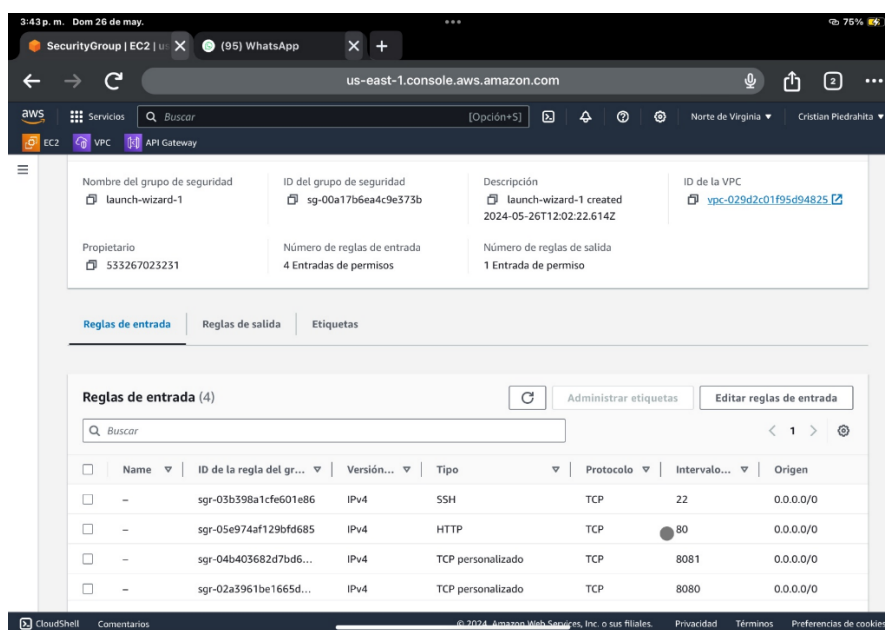


Ilustración 4 Grupo de Seguridad

Amazon Linux 2 AWS

Es un sistema operativo Linux de Amazon Web Services (AWS), optimizado para funcionar en esta plataforma, enfocado en brindar seguridad, estabilidad, alto rendimiento, para la ejecución de aplicaciones en la nube de Amazon, no tiene costos de licenciamiento asociados de

forma adicional, con la ventaja que AWS proporciona de forma periódica las actualizaciones de seguridad y mantenimiento. (Amazon AWS, s.f.)

SSH (Secure Shell)

Es un protocolo para la conexión remota a un servidor, por medio de un canal seguro en el cual la información va cifrada, usando criptografía para autenticar y encriptar las conexiones entre dispositivos. (Cloudfrare, s.f.)

Con el archivo .PEM que se genero en el proceso de la creación de la instancia, se descargo y guardo de forma segura, para garantizar el acceso a la instancia y poder administrarla.

Se procedió a realizar la conexión a la instancia, usando el nombre DNS de la instancia, la credencial y la llave previamente cargada.

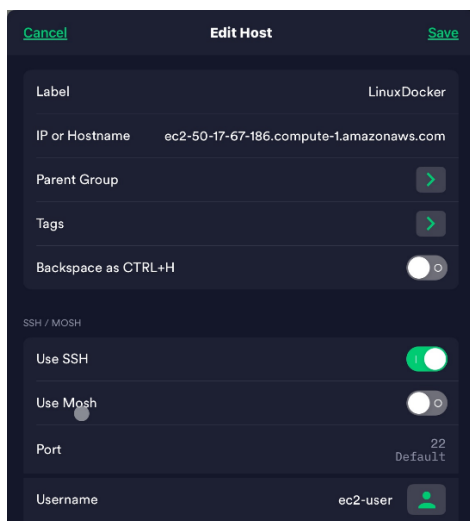


Ilustración 5 Configuración SSH -1

Docker

Proyecto de código abierto que automatiza y acelera el despliegue de aplicaciones dentro de contenedores (containers) de software, proporcionando una capa adicional de aislamiento (abstracción) y automatización para la virtualización de aplicaciones distintos sistemas operativos.

(Docker, s.f.)

Instalación de Docker:

Elevamos los permisos como super usuario (root) a través de:

```
sudo su
```

instalamos Docker:

```
yum install -y docker
```

En este caso usaremos la imagen de docker para apache, la buscamos en docker hub que un repositorio que comparte diferentes imágenes para docker, la cual usaremos en los contenedores más adelante

```
docker pull httpd
```

Verificar estado de docker

```
systemctl status docker
```

Procedemos a iniciar docker

```
systemctl start docker
```

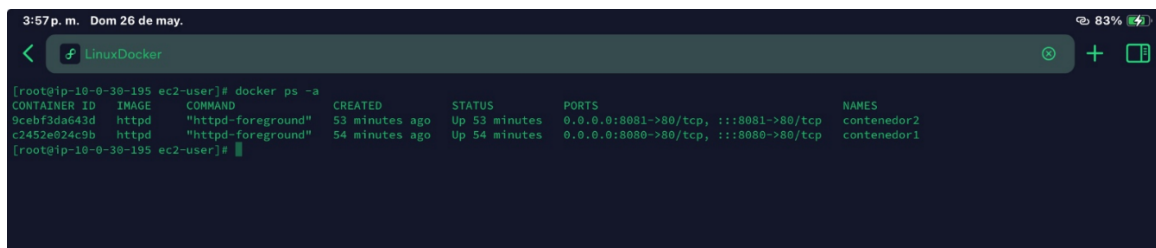
Si fuera necesario detener docker

```
systemctl stop docker
```

Creamos los contenedores

- `docker run -dit --name contenedor1 -p 8080:80 -v /var/www/contenedor1:/usr/local/apache2/htdocs/ httpd`
- `docker run -dit --name contenedor2 -p 8081:80 -v /var/www/contenedor2:/usr/local/apache2/htdocs/ httpd`

Se crearon las carpetas en las cuales estarán alojados los archivos de las paginas en este caso contendrán sus respectivos index.html, se definieron los puertos a los cuales se accederán de forma externa e interna del contenedor.



```
3:57 p. m. Dom 26 de may. 83%
LinuxDocker
[root@ip-10-0-30-195 ec2-user]# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
9ceb3da643d   httpd    "httpd-foreground"      53 minutes ago Up 53 minutes 0.0.0.0:8081->80/tcp, :::8081->80/tcp contenedor2
c2452e024c9b   httpd    "httpd-foreground"      54 minutes ago Up 54 minutes 0.0.0.0:8080->80/tcp, :::8080->80/tcp contenedor1
[root@ip-10-0-30-195 ec2-user]#
```

Ilustración 8 Ejecucion del comando `docker ps -a`

Listamos los contenedores en ejecución

`docker ps`

También se podría usar `docker ps -a` para listar todos los contenedores activos e inactivos

Para detener la ejecución de un contenedor se usa número de instancia (ID) que se puede obtener al listar los contenedores con `docker ps`.

`docker stop` (se escribe número de instancia)

Para iniciar un contenedor el comando va acompañado del número de la instancia

`docker start` (se escribe número de instancia)

Para conocer las estadísticas en tiempo real del contenedor se usa

`docker stats`

Apache

Apache es un servidor web de código abierto multiplataforma, ampliamente usado (The Apache Software Foundation, s.f.).

Lo usaremos para la creación de un proxy reverso, instalado en la instancia EC2 que estamos ejecutando, esto con el fin de canalizar correctamente las conexiones entrantes a nuestros contenedores que contienen sitios con diferentes dominios a través de una conexión http a la instancia, el proxy reverso recibirá las conexiones en el puerto http (puerto TCP 80) y de acuerdo con el nombre de dominio solicitado se encarga de redireccionar y proporcionar la página que corresponda según el dominio.

Instalar Apache:

```
sudo yum install httpd
```

Habilitar Apache al inicio del sistema:

```
sudo systemctl enable httpd
```

Iniciar el servicio de Apache

```
sudo systemctl start httpd
```

Virtual host

Es una forma para compartir el alojamiento web, que permite hospedar más de una página en un mismo servidor, optimizando los recursos.

```

4:14 p. m. Dom 26 de may.
LinuxDocker
[root@ip-10-0-30-195 conf.d]# pwd
/etc/httpd/conf.d
[root@ip-10-0-30-195 conf.d]# ls
README autoindex.conf contenedor1.conf contenedor2.conf userdir.conf welcome.conf
[root@ip-10-0-30-195 conf.d]# cat contenedor1.conf contenedor2.conf
<VirtualHost *:80>
    ServerName tienda.com
    ProxyPass / http://localhost:8080/
    ProxyPassReverse / http://localhost:8080/
</VirtualHost>
<VirtualHost *:80>
    ServerName comercio.com
    ProxyPass / http://localhost:8081/
    ProxyPassReverse / http://localhost:8081/
</VirtualHost>
[root@ip-10-0-30-195 conf.d]#

```

Ilustración 9 Contenido de los archivos de configuración de apache

Configuración de apache para atender las solicitudes por nombres de dominio

Accederemos a la ruta `/etc/httpd/conf.d/` crearemos archivos para cada contenedor que usara el proxy inverso (The Apache Software Foundation, s.f.), especificares las redirecciones hacia los puertos correctos en sus respectivos contenedores.

Accedemos a la ruta

```
cd /etc/httpd/conf.d/
```

creamos los archivos

```
sudo nano contenedor1.conf
```

agregamos el siguiente contenido:

```
<VirtualHost *:80>
```

```
    ServerName tienda.com
```

```
    ProxyPass / http://localhost:8080/
```

```
    ProxyPassReverse / http://localhost:8080/
```

```
</VirtualHost>
```

Guardamos el archivo

Procedemos a crear el siguiente

```
sudo nano contenedor2.conf
```

```
<VirtualHost *:80>  
  
    ServerName comercio.com  
  
    ProxyPass / http://localhost:8081/  
  
    ProxyPassReverse / http://localhost:8081/  
  
</VirtualHost>
```

Guardamos el archivo

En estos archivos, ProxyPass indica la dirección del contenedor y el puerto al que Apache debe redirigir el tráfico entrante para ese nombre de dominio, mientras que ProxyPassReverse ajusta los encabezados HTTP para que las respuestas del servidor en el contenedor apunten al nombre de dominio correcto.

Después de actualizar los archivos de configuración, reiniciamos Apache para que los cambios surtan efecto:

```
sudo systemctl restart httpd
```

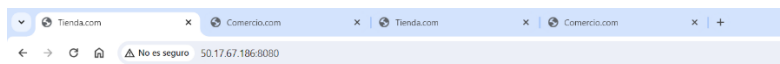
Después de esta configuración el servidor web Apache funcionará como proxy reverso, estará escuchando las solicitudes en el puerto HTTP (puerto 80 TCP) y dirigirá el tráfico entrante a los puertos 8080 y 8081 en función del nombre de dominio solicitado.

Páginas Web

A continuación, realizaremos las pruebas para validar que las páginas se estén sirviendo correctamente de acuerdo con la solicitud del nombre de dominio que realicen los clientes

Sitio tienda.com

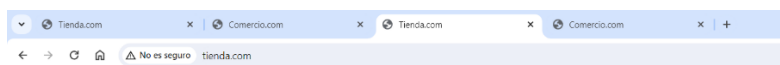
Acceso por IP y puerto al contenedor1



Bienvenidos a Tienda.com

Ilustración 10 Acceso por IP y puerto al contenedor1

Acceso por nombre de dominio al contenedor1

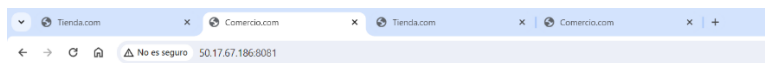


Bienvenidos a Tienda.com

Ilustración 11 Acceso por nombre de dominio al contenedor1

Sitio comercio.com

Acceso por IP y puerto al contenedor2



Bienvenidos a Comercio.com

Ilustración 12 Acceso por IP y puerto al contenedor2

Acceso por nombre de dominio al contenedor2

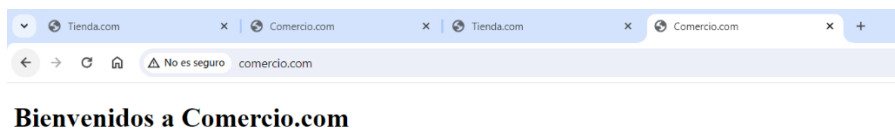


Ilustración 13 Acceso por nombre de dominio al contenedor2

Para poder usar estos dominios ficticios y realizar las pruebas sin comprar los dominios modificamos el archivo del sistema host, que se encuentra en la ruta:

C:\Windows\System32\drivers\etc\hosts

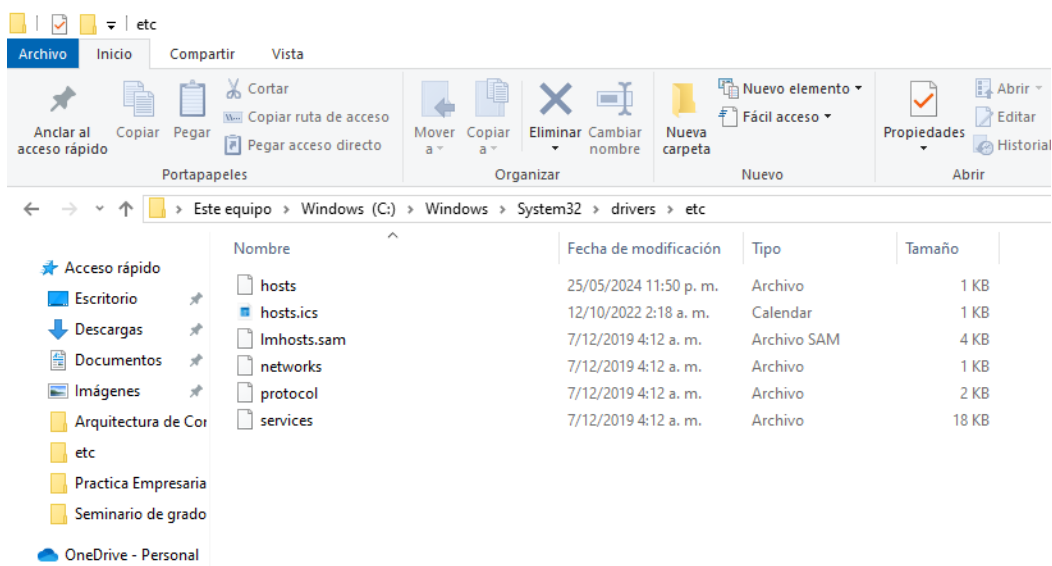
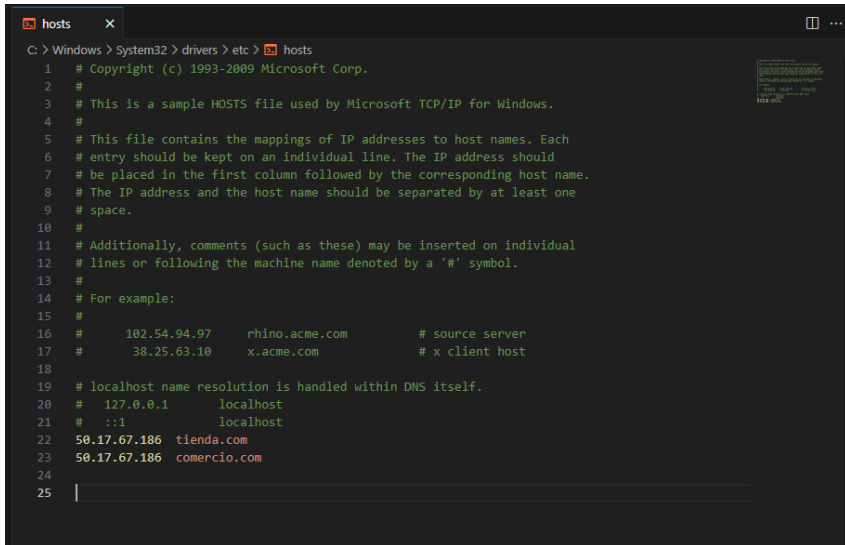


Ilustración 14 Ruta archivo hosts

A screenshot of a Windows command prompt window titled 'hosts'. The window shows the contents of the hosts file, which includes copyright information, instructions on how to use the file, and several entries mapping IP addresses to hostnames. The entries include 'rhino.acme.com' pointing to '102.54.94.97', 'x.acme.com' pointing to '38.25.63.10', and two entries for 'localhost' pointing to '127.0.0.1' and ':::1'. At the bottom, two manually added entries are visible: '50.17.67.186 tienda.com' and '50.17.67.186 comercio.com'.

```
C:\Windows\System32\drivers\etc > hosts
1 # Copyright (c) 1993-2009 Microsoft Corp.
2 #
3 # This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
4 #
5 # This file contains the mappings of IP addresses to host names. Each
6 # entry should be kept on an individual line. The IP address should
7 # be placed in the first column followed by the corresponding host name.
8 # The IP address and the host name should be separated by at least one
9 # space.
10 #
11 # Additionally, comments (such as these) may be inserted on individual
12 # lines or following the machine name denoted by a '#' symbol.
13 #
14 # For example:
15 #
16 #       102.54.94.97     rhino.acme.com         # source server
17 #       38.25.63.10    x.acme.com             # x client host
18
19 # localhost name resolution is handled within DNS itself.
20 #   127.0.0.1          localhost
21 #   :::1              localhost
22 50.17.67.186 tienda.com
23 50.17.67.186 comercio.com
24
25 |
```

Ilustración 15 Contenido archivo hosts

Se agregaron las entradas manualmente al archivo de hosts del sistema operativo, para que resuelvan los dominios ficticios con los que se están trabajando apuntado a la IP publica de la instancia EC2. Para que pueda funcionar correctamente, en este caso seria agregar las siguientes líneas al final del archivo host.

50.17.67.186 tienda.com

50.17.67.186 comercio.com

Entrega 3

Amazon Simple Storage Service (Amazon S3)

Es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento, ofrece diferentes opciones de almacenamiento, los objetos se almacenan dentro de buckets (directorios, carpetas), permite la asignación de permisos para controlar el acceso. (Amazon AWS, s.f.)

Actividad

De acuerdo con las indicaciones dadas en el seminario, se habilita la funcionalidad IAM Identity Center, esto con el fin de permitir que las cuentas de miembros de una organización habiliten aplicaciones de AWS que son independientes de la instancia.

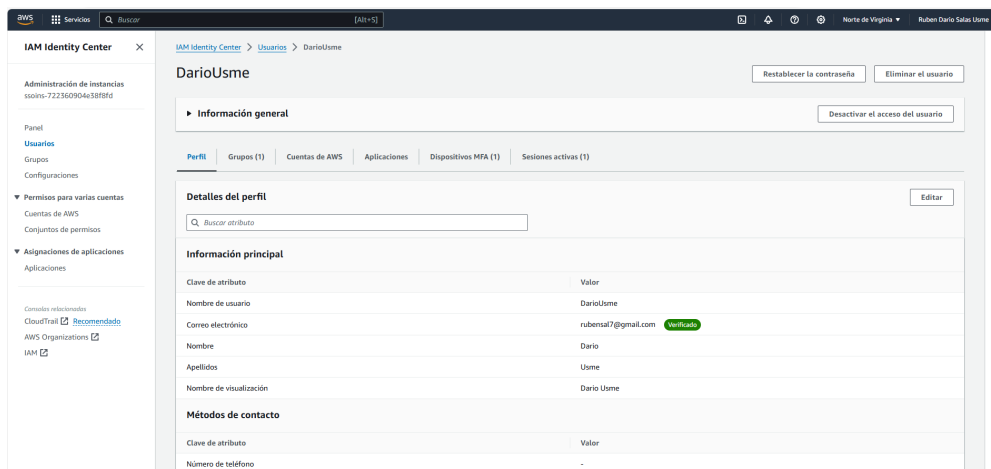


Ilustración 16 Usuario creado IAM Identity Center

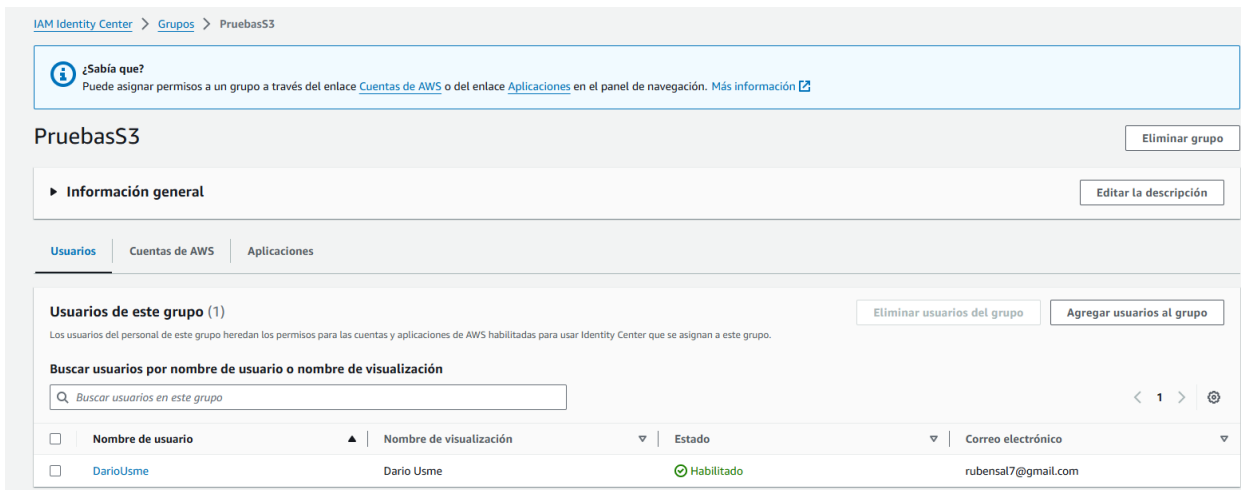


Ilustración 17 Grupo creado

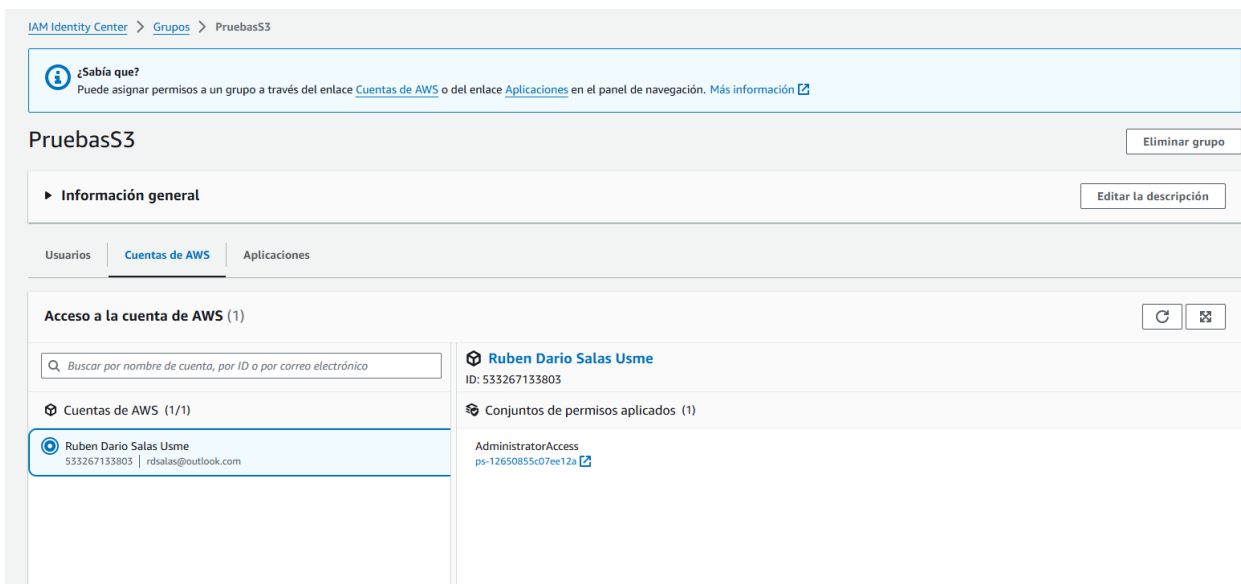


Ilustración 18 Asociación de cuenta AWS al grupo

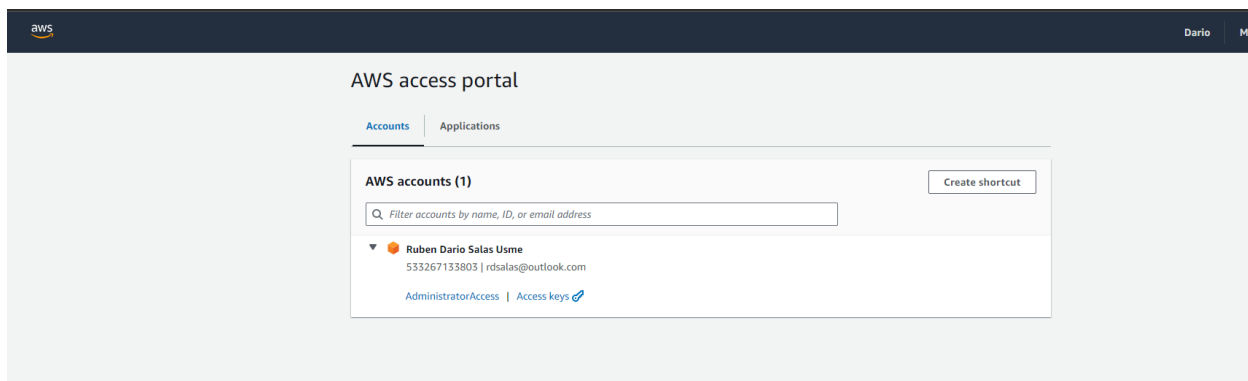


Ilustración 19 Ingreso a la cuenta creada

Una vez creada la cuenta procedemos a crear el Bucket de clase general en S3 de acuerdo con las indicaciones dadas en la clase del seminario, el bucket será el directorio donde guardaremos los objetos.

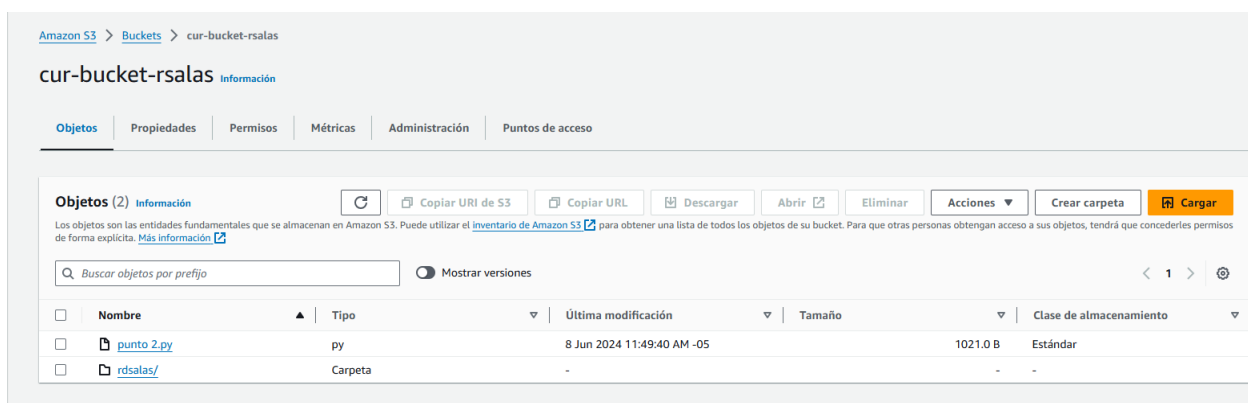


Ilustración 20 Bucket creado

Preparamos una instancia EC2 de acuerdo con las indicaciones que se han dado durante el seminario para realizar las pruebas

Se usaron los siguientes comandos:

Descarga la herramienta:

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
```

Descomprime AWS CLI

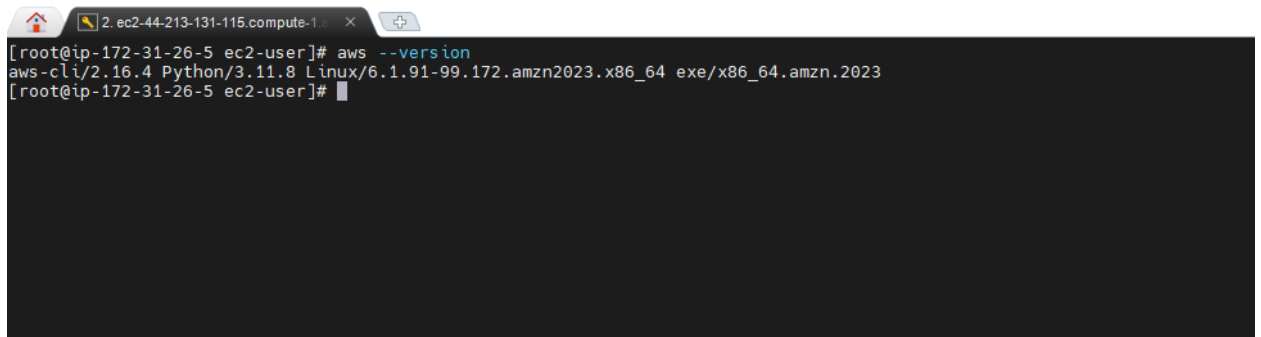
```
unzip awscliv2.zip
```

Ejecutamos la instalación:

```
sudo ./aws/install
```

Confirmación de instalación de AWS CLI

```
aws --version
```



```
2. ec2-44-213-131-115.compute-1... x +  
[root@ip-172-31-26-5 ec2-user]# aws --version  
aws-cli/2.16.4 Python/3.11.8 Linux/6.1.91-99.172.amzn2023.x86_64 exe/x86_64.amzn.2023  
[root@ip-172-31-26-5 ec2-user]#
```

Ilustración 23 Confirmación de instalación AWS CLI

En la cuenta creada previamente extraemos las Access Key

Get credentials for AdministratorAccess ✕

Create access for the account **Ruben Dario Salas Usme (533267133803)** with **AdministratorAccess**.
Use any of the following options to access AWS resources programmatically or from the AWS CLI. You can retrieve credentials as often as needed.

macOS and Linux
Windows
PowerShell

▼ **AWS IAM Identity Center credentials (Recommended)**

To extend the duration of your credentials, we recommend you configure the AWS CLI to retrieve them automatically using the `aws configure sso` command. [Learn more](#)

SSO start URL	<code>https://d-9067e73807.awsapps.com/start/#</code>	
SSO Region	<code>us-east-1</code>	

▼ **Option 1: Set AWS environment variables**

Run the following commands in your terminal to set the AWS environment variables. [Learn more](#)

```
export AWS_ACCESS_KEY_ID="ASIAXYKJSwVVZ4JV3GKH"
export AWS_SECRET_ACCESS_KEY="CgVaiKpFgIN3LpK9JdaCfM1ZvIGj0jshmkU)
export AWS_SESSION_TOKEN="IQoJb3JpZ21uX2VjEiH////////wEaCXVzLWVl
```

Copy

Ilustración 24 Access Keys

Copiamos y pegamos las variables para la conexión desde SSH, probamos que podamos listar el Bucket

Usando el comando:

```
Aws s3 ls s3://cur-bucket-rsalas/
```


Amazon Relational Database Service (Amazon RDS)

Es un servicio de bases de datos relacionales con facilidades de administración, optimizado, puede escalar según la demanda. Amazon RDS automatiza las tareas administración de bases de datos, como el aprovisionamiento, la configuración, las copias de seguridad y la aplicación de revisiones, ofrece flexibilidad para personalizar las bases de datos, tiene diversos motores disponibles para elegir entre ellos MySQL, MariaDB, PostgreSQL, SQL Server, Oracle. (Amazon AWS, s.f.)

De acuerdo con las indicaciones dadas durante el seminario procedemos a crear la Base de datos con el servicio RDS

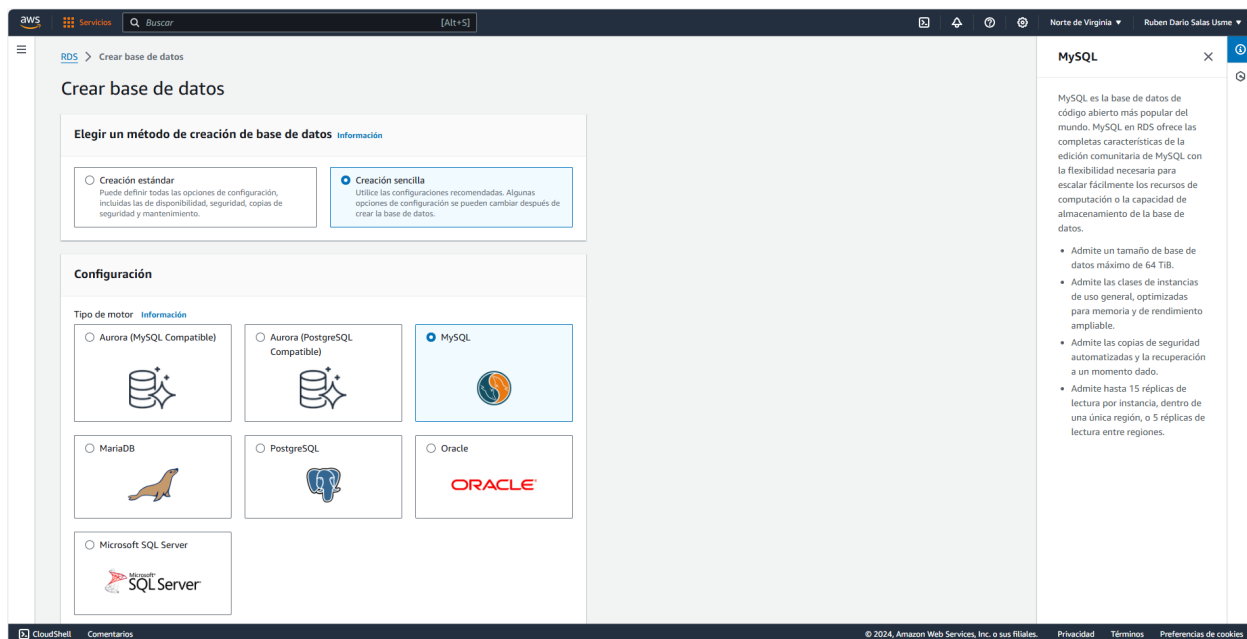


Ilustración 27 Creación Base de Datos

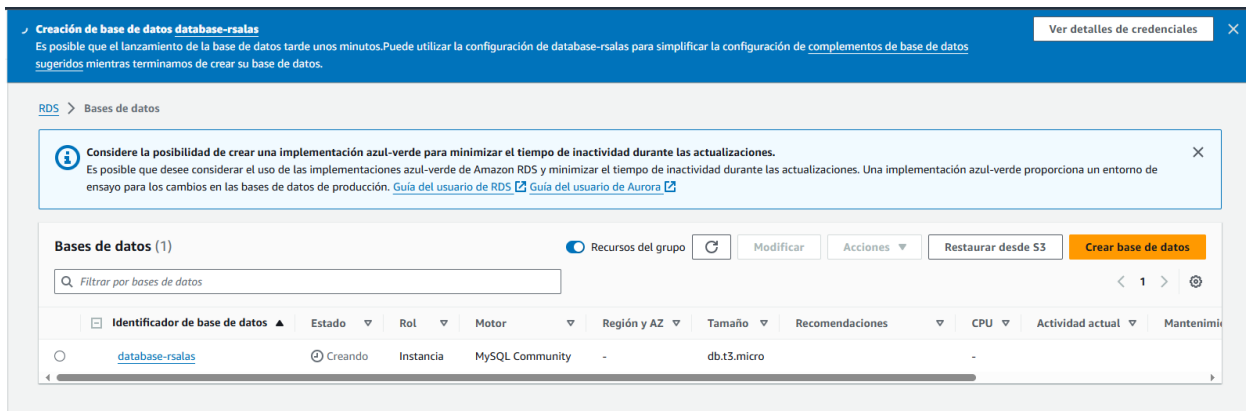


Ilustración 28 Creación de base de datos

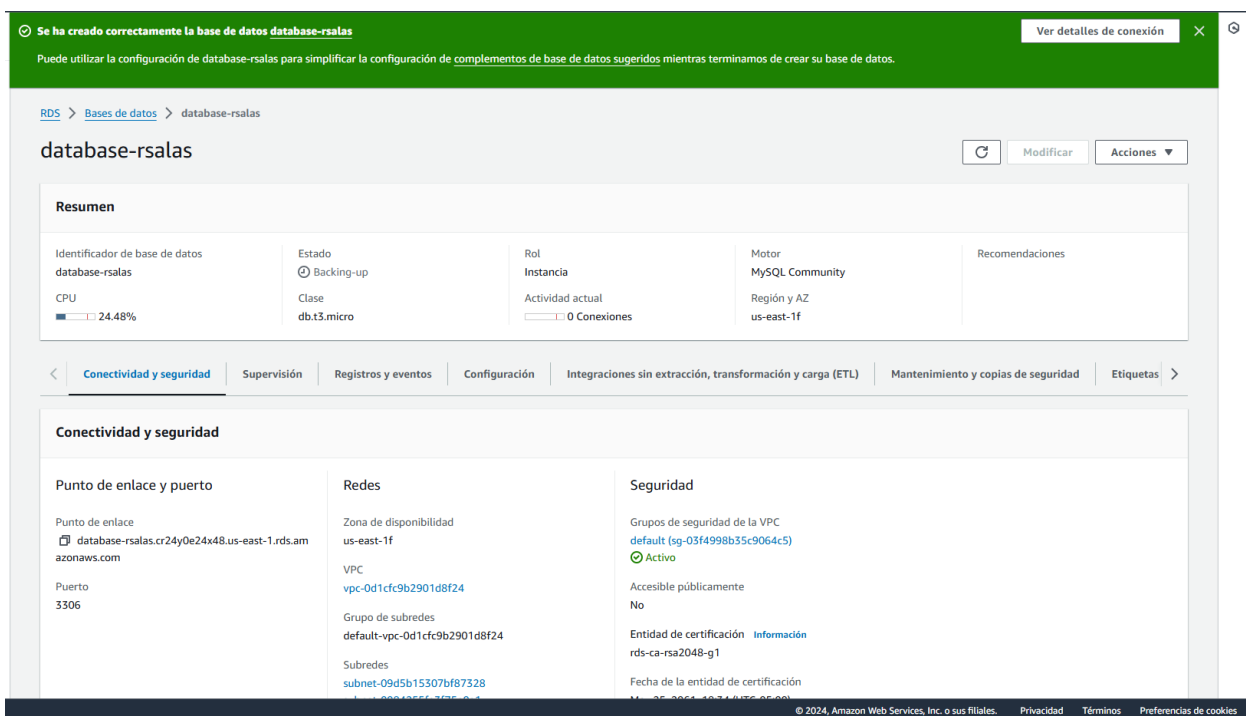


Ilustración 29 Proceso de creación terminado

A continuación, nos conectaremos por SSH desde la instancia EC2 a la base de datos que acabamos de crear

```
mysql -h database-rsalas.cr24y0e24x48.us-east-1.rds.amazonaws.com -u admin -p
```



Ilustración 30 Conexión a base de datos

Crearemos una base de datos, una tabla e insertaremos registros

Iniciamos como root

```
mysql -u root -p
```

Creamos la base de datos

```
CREATE DATABASE seminarioaws;
```

Mostramos la base de datos

Entrega 4

Amazon Elastic Load Balancing (ELB):

Se encarga de distribuir automáticamente el tráfico de aplicaciones entrantes entre varios destinos y dispositivos virtuales en una o varias zonas de disponibilidad (AZ). (Amazon AWS, s.f.)

AWS Auto Scaling:

Sirve para monitorizar las aplicaciones y ajustar automáticamente la capacidad para mantener un desempeño predecible y estable.

Amazon EC2 Auto Scaling

Sirve aumentar o reducir la capacidad de computación para satisfacer los cambios en la demanda, permitiendo mantener la disponibilidad de la aplicación, permite agregar o eliminar automáticamente instancias de EC2 según las condiciones que se definan. Las políticas dinámicas o predictivas le permiten aumentar o reducir la capacidad de instancias de EC2 para prestar servicio a patrones de demanda establecidos o en tiempo real. (Amazon AWS, s.f.)

Actividad

De acuerdo con las indicaciones compartidas en el seminario se realiza la creación de una instancia EC2 T2 micro de la capa gratuita con el sistema operativo Windows Server 2022, una vez la instancia se encuentra disponible haciendo uso del Access key se procede a descifrar la contraseña para el acceso a través de RDP.

Se inicia sesión RDP y se procede a instalar el servidor web IIS, este proceso llevara algunos minutos.

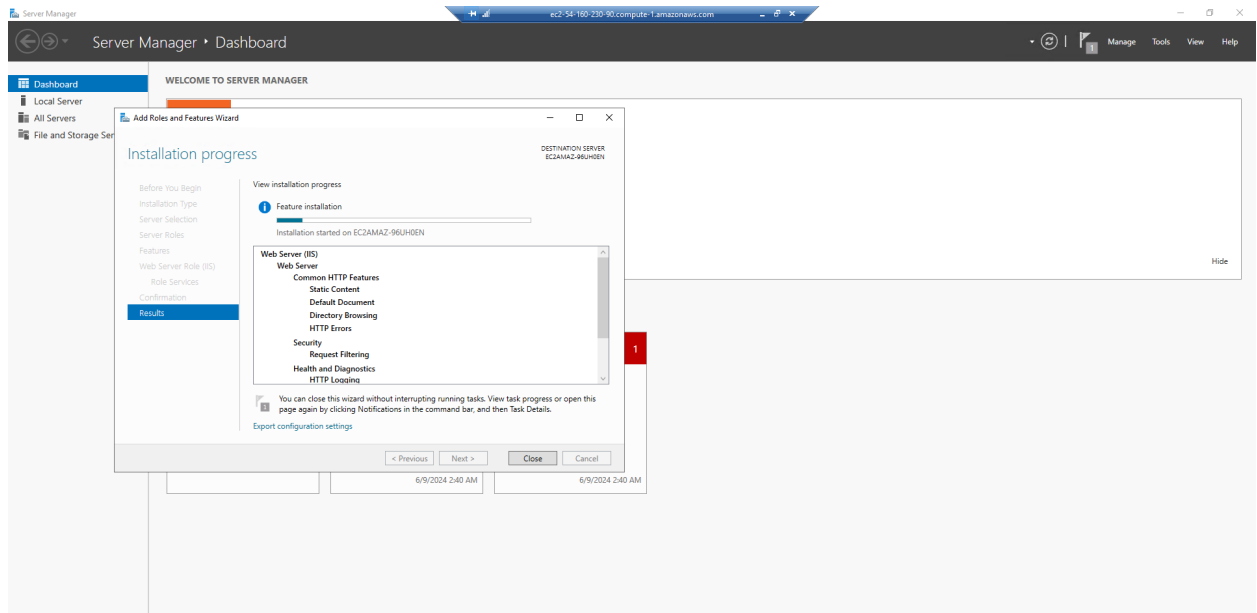


Ilustración 33 Instalación Web server IIS

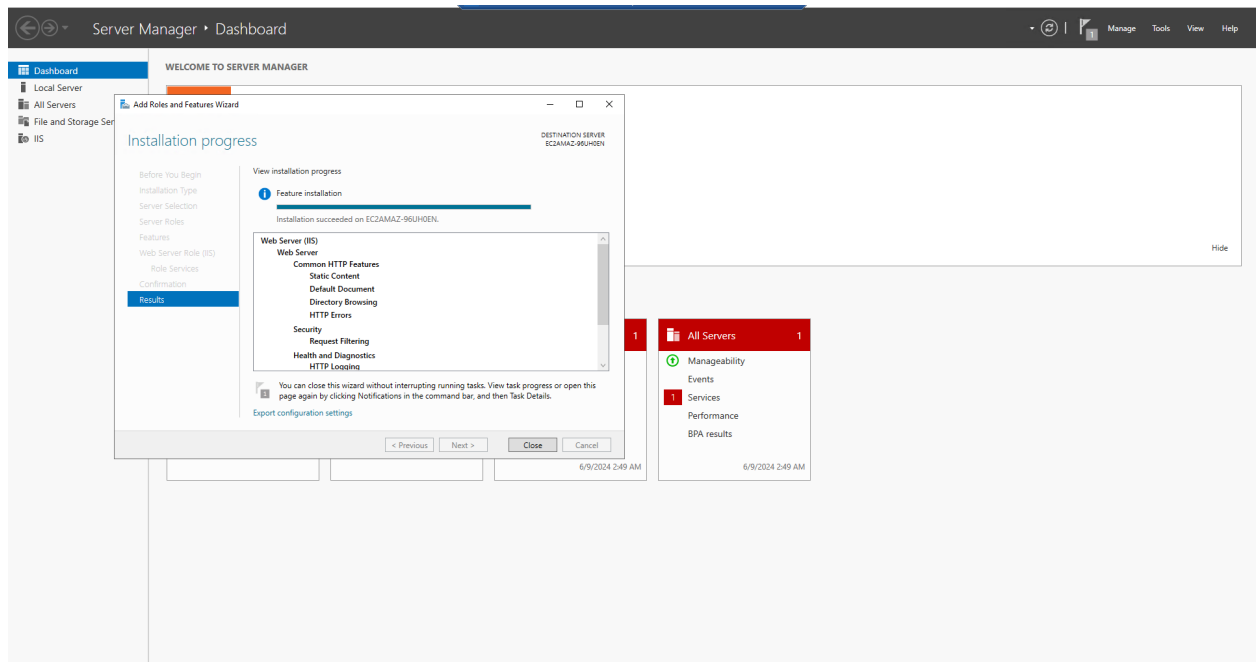


Ilustración 34 Instalación finalizada IIS

Se valida que el servidor web IIS este respondiendo por el puerto 80 a través de la dirección publica, en este caso se accedió a través del nombre DNS

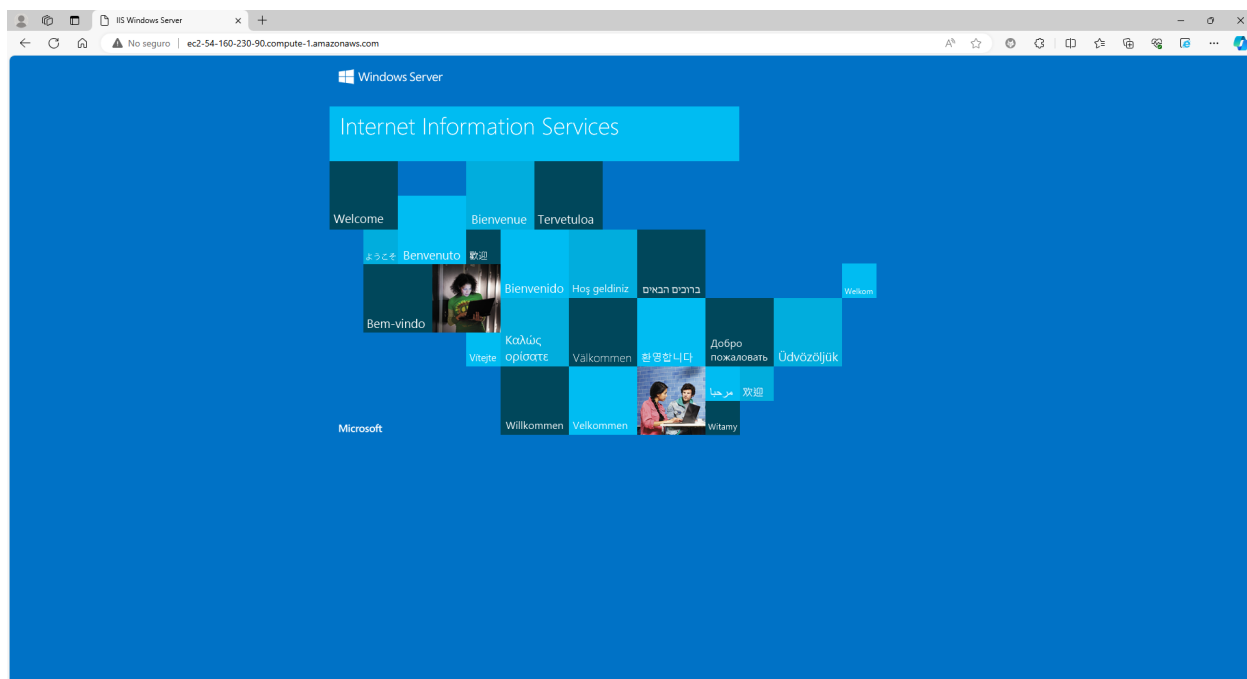


Ilustración 35 Respuesta servidor web IIS

Con la prueba realizada confirmamos que la instancia y el servidor web se encuentran funcionando.

Realizaremos la configuración del Balanceador de carga de acuerdo con las indicaciones compartidas en el seminario, para este escogeremos un tipo de balanceador de aplicaciones.

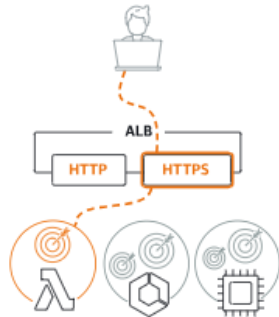
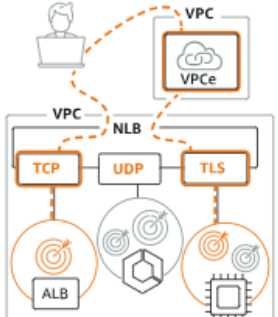

aws Services Search [Alt+S]

EC2 > Load balancers > Compare and select load balancer type

Compare and select load balancer type

A complete feature-by-feature comparison along with detailed highlights is also available. [Learn more](#)

Load balancer types

Application Load Balancer Info	Network Load Balancer Info	Gateway Load Balancer Info
		
<p>Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.</p> <p>Create</p>	<p>Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.</p> <p>Create</p>	<p>Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.</p> <p>Create</p>

▶ [Classic Load Balancer - previous generation](#)

[Close](#)

Ilustración 36 Creación Balanceador

Pondremos su respectivo nombre al balanceador, para este caso, el esquema del balanceador será de cara a internet, usando el protocolo IPv4, escogemos grupo de seguridad del cual hará parte, para este caso el protocolo de escucha es HTTP en el puerto TCP 80, el cual será enrutado a las instancias EC2.

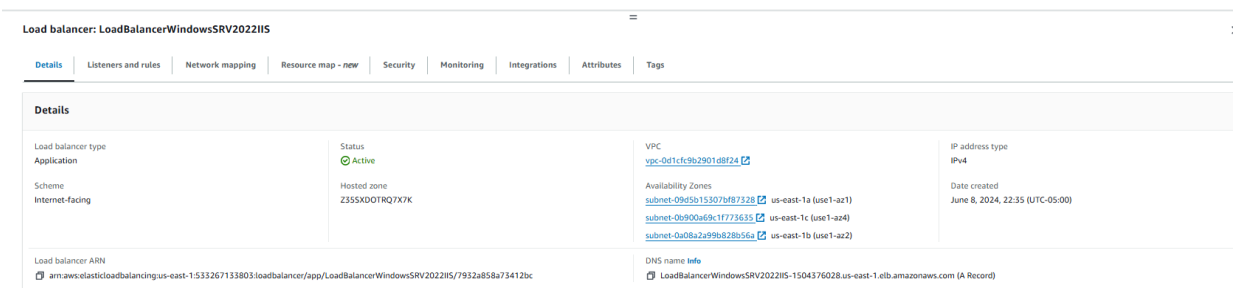


Ilustración 37 Resumen de Balanceador

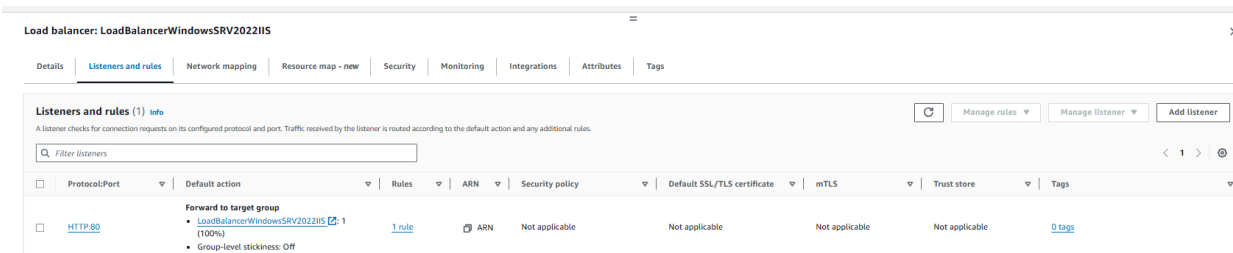


Ilustración 38 Listener y ruta

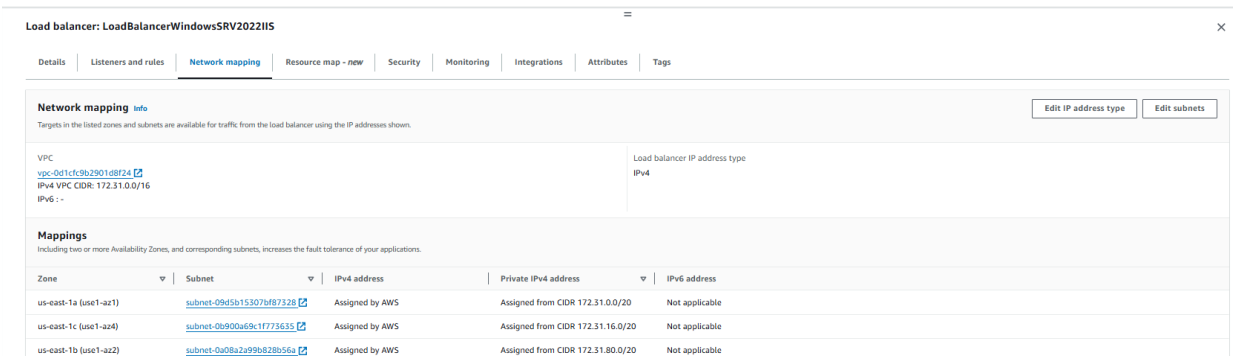


Ilustración 39 VPC y Zonas de disponibilidad

Load balancer: LoadBalancerWindowsSRV2022IIS

Details | Listeners and rules | Network mapping | Resource map - new | **Security** | Monitoring | Integrations | Attributes | Tags

Security groups (1) Edit

A security group is a set of firewall rules that control the traffic to your load balancer.

Security Group ID	Name	Description
sg-03f4998b35c9064c5	default	default VPC security group

Ilustración 40 Grupo de Seguridad

Load balancer: LoadBalancerWindowsSRV2022IIS

Details | Listeners and rules | Network mapping | **Resource map - new** | Security | Monitoring | Integrations | Attributes | Tags

Resource map Info Give feedback

View, explore, and troubleshoot your load balancer's architecture.

Overview | Unhealthy target map | Show resource details

LoadBalancerWindowsSRV2022IIS Last fetched seconds ago Export

Listeners (1)

- HTTP:80 (1 rule)

Rules (1)

- Priority default
- Forward to target group
- Conditions (1) If no other rule applies

Target groups (1) info

- Instance LoadBalancerWindowsSRV2022IIS (1 target)

Targets (1)

- i-062b7139c3b4003a1 Port 80 (Healthy)

Ilustración 41 Mapa de recursos

Ahora probaremos el balanceador ingresando por la ruta DNS que suministra el balanceador para nuestro caso

<http://loadbalancerwindowssrv2022iis-1504376028.us-east-1.elb.amazonaws.com/>

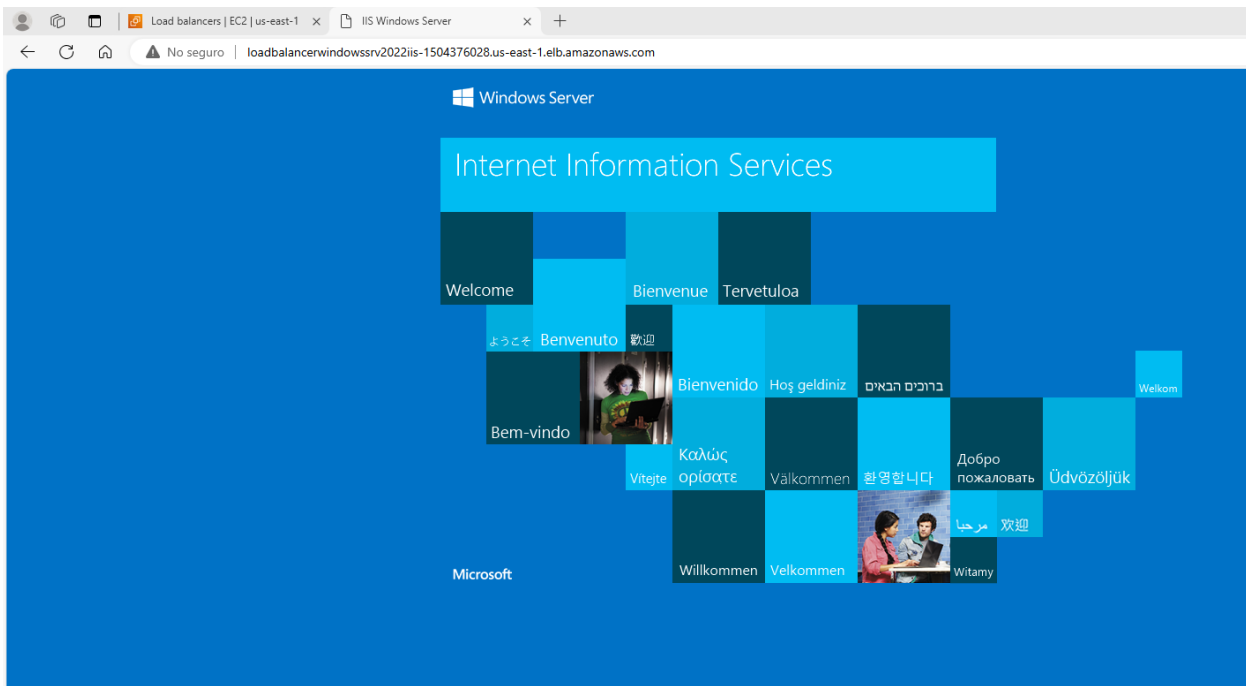


Ilustración 42 Respuesta del balanceador

Crearemos un Snapshot (instantánea) del volumen de esta instancia EC2

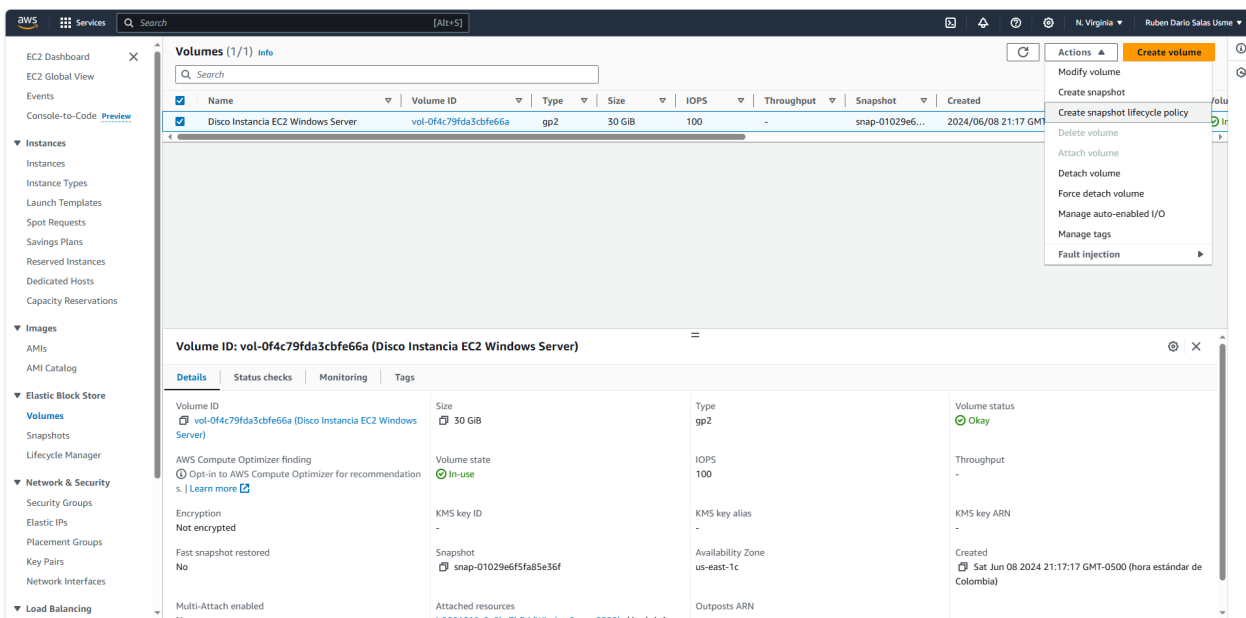


Ilustración 43 Creación Snapshot

Snapshots (1) Info									
Owned by me <input type="text" value="Search"/>									
Name	Snapshot ID	Volume size	Description	Storage tier	Snapshot status	Started	Progress	Encryption	
-	snap-05c89730d73f892a1	30 GiB	SnapWin-SRV-2022-IIS	Standard	Completed	2024/06/08 22:00 GMT-5	Available (100%)	Not encrypted	

Ilustración 44 Snapshot creada con éxito

Creamos una AMI a partir del Snapshot de la EC2

[EC2](#) > [Snapshots](#) > [snap-05c89730d73f892a1](#) > [Create image from snapshot](#)

Create image from snapshot [Info](#)

Create a new image from a snapshot taken from the root device volume of an instance.

Image settings

Snapshot ID

Image name
 A descriptive name for the image.

 3 - 128 characters. Valid characters are a-z, A-Z, 0-9, spaces, and - _ . / () [] ' ' @.

Description
 A description for the image.

 255 characters maximum

Architecture [Info](#)
 Select i386 for 32-bit or x86_64 for 64-bit.

Root device name [Info](#)
 The device name that is reserved for the root volume.

Virtualization type [Info](#)
 The virtualization type to be used by instances launched from this image.

Kernel ID [Info](#)
 The operating system kernel for the AMI.

Ilustración 45 Creación AMI del Snapshot

Como resultado tendremos una AMI personalizada basada en la instancia EC2 que previamente habíamos configurado

The screenshot shows the Amazon Machine Images (AMIs) console. At the top, there's a header for "Amazon Machine Images (AMIs) (1/1) Info" with a search bar and several action buttons: "Recycle Bin", "EC2 Image Builder", "Actions", and "Launch instance from AMI". Below the header is a table listing AMIs. The table has columns for Name, AMI name, AMI ID, Source, Owner, Visibility, Status, and Creation date. One AMI is listed: "AMI-WinSRV2022-IIS" with AMI ID "ami-0f11467ef7a772778", Source "533267133803/AMI-WinSRV2022-IIS", Owner "533267133803", Visibility "Private", Status "Available", and Creation date "2024/06/08 22:22".

Below the table, the details for the selected AMI are shown. The AMI ID is "ami-0f11467ef7a772778". There are tabs for "Details", "Permissions", "Storage", and "Tags". A notification banner at the top of the details section says "Enable fast launch for your Windows AMI instances. - Recommended" and "Improve launch times for Windows Server AMIs by up to 65%. You only pay for the storage of the snapshots created to speed up the instance launch times. Learn more". There is an "Enable fast launch" button.

AMI ID	ami-0f11467ef7a772778	Image type	machine	Platform details	Windows	Root device type	EBS
AMI name	AMI-WinSRV2022-IIS	Owner account ID	533267133803	Architecture	x86_64	Usage operation	RunInstances:0002
Root device name	/dev/sda1	Status	Available	Source	533267133803/AMI-WinSRV2022-IIS	Virtualization type	hvm
Boot mode	-	State reason	-	Creation date	-	Kernel ID	-

Ilustración 46 AMI Personalizada

Crearemos un Launch Template, esto con el fin de usarla para la creación de las instancias que usaremos en el auto escalado

[EC2](#) > [Launch templates](#) > Create launch template

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - *required*

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description

Max 255 chars

Auto Scaling guidance | [Info](#)
Select this if you intend to use this template with EC2 Auto Scaling

Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

► **Template tags**

► **Source template**

Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

▼ **Application and OS Images (Amazon Machine Image) - required** [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Ilustración 47 Creación Launch Template

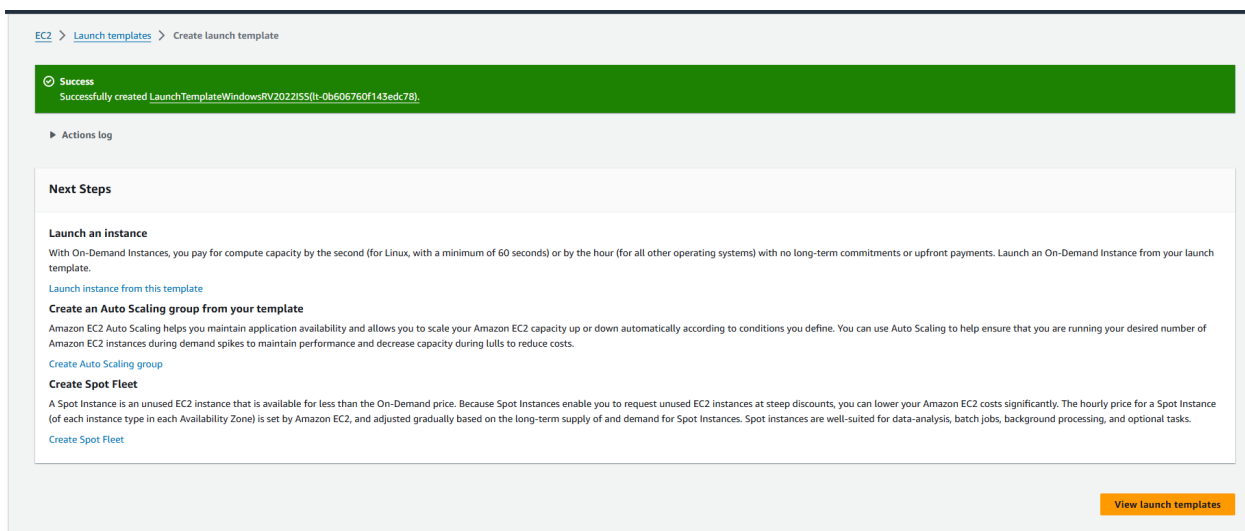


Ilustración 48 Proceso terminado creación Launch Template

Launch Templates (1) [Info](#) 🔄 Actions ▾ Create launch template

< 1 > ⚙️

	Launch Template ID ▾	Launch Template Name ▾	Default Version ▾	Latest Version ▾	Create Time ▾	Created By ▾
○	it-0b606760f143edc78	LaunchTemplateWindowsRV2...	1	1	2024-06-09T03:20:25.000Z	arn:aws:iam::533267133803:root

Ilustración 49 Template disponible

Creamos un Grupo de Auto Escalado, y se escoge Launch Template que previamente se había creado

Choose launch template Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

Name

Auto Scaling group name
Enter a name to identify the group.

Must be unique to this account in the current Region and no more than 255 characters.

Launch template Info

i For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

[Create a launch template](#)

Version

[Create a launch template version](#)

Description -	Launch template LaunchTemplateWindowsRV2022ISS lt-0b606760f143edc78	Instance type t2.micro
AMI ID ami-0f11467ef7a772778	Security groups -	Request Spot Instances No
Key pair name EC2SeminararioAWS	Security group IDs sg-03f4998b35c9064c5	
Additional details		
Storage (volumes) -	Date created Sat Jun 08 2024 22:20:25 GMT-0500 (hora estándar de Colombia)	

sole/home?region=us-east-1

Ilustración 50 Creación Auto Escalado

Escogeremos la VPC y las zonas de disponibilidad

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
[Choose launch template](#)

Step 2
Choose instance launch options

Step 3 - optional
[Configure advanced options](#)

Step 4 - optional
[Configure group size and scaling](#)

Step 5 - optional
[Add notifications](#)

Step 6 - optional
[Add tags](#)

Step 7
[Review](#)

Choose instance launch options [Info](#)

Choose the VPC network environment that your instances are launched into, and customize the instance types and purchase options.

Instance type requirements [Info](#) Override launch template

You can keep the same instance attributes or instance type from your launch template, or you can choose to override the launch template by specifying different instance attributes or manually adding instance types.

Launch template	Version	Description
LaunchTemplateWindowsRV2022ISS lt-0b606760f143edc78	Default	-

Instance type
t2.micro

Network [Info](#)

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-0d1cfc9b2901d8f24 (VPC-LAB)
172.31.0.0/16 Default Refresh

[Create a VPC](#)

Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets Refresh

- us-east-1a | subnet-09d5b15307bf87328 ×
172.31.0.0/20 Default
- us-east-1b | subnet-0a08a2a99b828b56a ×
172.31.80.0/20 Default
- us-east-1c | subnet-0b900a69c1f773635 ×
172.31.16.0/20 Default

[Create a subnet](#)

Cancel Skip to review Previous Next

Ilustración 51 Opciones de launch template

Asociaremos el auto scaling a un balanceador existente que previamente hemos creado

Create Auto Scaling group

Configure advanced options - *optional* [Info](#)

Integrate your Auto Scaling group with other services to distribute network traffic across multiple servers using a load balancer or to establish service-to-service communications using VPC Lattice. You can also set options that give you more control over health check replacements and monitoring.

Load balancing [Info](#)

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.

Attach to an existing load balancer
Choose from your existing load balancers.

Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.

Attach to an existing load balancer

Select the load balancers that you want to attach to your Auto Scaling group.

Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.

Choose from Classic Load Balancers

Existing load balancer target groups
Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups ▼ ↻

LoadBalancerWindowsSRV2022IIS | HTTP

Application Load Balancer: LoadBalancerWindowsSRV2022IIS

✕

VPC Lattice integration options [Info](#)

To improve networking capabilities and scalability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between AWS services and helps you connect and manage your applications across compute services in AWS.

Select VPC Lattice service to attach

No VPC Lattice service
VPC Lattice will not manage your Auto Scaling group's network access and connectivity with other services.

Attach to VPC Lattice service
Incoming requests associated with specified VPC Lattice target groups will be routed to your Auto Scaling group.

[Create new VPC Lattice service](#) [↗](#)

Ilustración 52 Asociación a balanceador existente

Continuamos en el asistente definimos la cantidad de instancias que vamos a tener, la cantidad mínima y máxima

> Create Auto Scaling group

Configure group size and scaling - *optional* [Info](#)

Define your group's desired capacity and scaling limits. You can optionally add automatic scaling to adjust the size of your group.

Group size [Info](#)

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances) ▼

Desired capacity

Specify your group size.

1

Scaling [Info](#)

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits

Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity	Max desired capacity
1	3
Equal or less than desired capacity	Equal or greater than desired capacity

Automatic scaling - *optional*

Choose whether to use a target tracking policy | [Info](#)

You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

No scaling policies
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

Target tracking scaling policy
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

Ilustración 53 Cantidad de instancias en ejecución

The screenshot displays the 'Auto Scaling Windows SRV 2022 ISS' page in the AWS Management Console. It is divided into three main sections: Group details, Launch template, and Network.

Group details:

- Auto Scaling group name: Auto Scaling Windows SRV 2022 ISS
- Date created: Sat Jun 08 2024 22:35:19 GMT-0500 (hora estándar de Colombia)
- Desired capacity: 1
- Minimum capacity: 1
- Maximum capacity: 3
- Desired capacity type: UNITS (number of instances)
- Status: -
- Amazon Resource Name (ARN): arn:aws:autoscaling:us-east-1:533267133803:autoScalingGroup:0ba1faac-c008-4c7a-bbf0-e8894052f0ab:autoScalingGroupName/Auto Scaling Windows SRV 2022 ISS

Launch template:

- Launch template: lt-0a609c760f145edc78 (LaunchTemplateWindowsSRV2022ISS)
- Version: Default
- Description: -
- AMI ID: ami-0f11467ef7a772778
- Instance type: t2.micro
- Security group IDs: sg-03f4998b35c9064c5
- Key pair name: EC2SeminarioAWS
- Owner: amaws:iam::533267133803:root
- Create time: Sat Jun 08 2024 22:20:25 GMT-0500 (hora estándar de Colombia)
- Request Spot Instances: No

Network:

- Availability Zones: us-east-1a, us-east-1b, us-east-1c
- Subnet ID: subnet-0945d153079f87528, subnet-06900a69c1f773655, subnet-0a0ba22990828056a

Ilustración 54 Resumen de Auto Scaling

Comprobamos la cantidad de instancias en ejecución en el auto scaling

The screenshot shows the 'Auto Scaling groups' page in the AWS Management Console. It features a search bar and a table listing the groups.

Name	Launch template/configuration	Version	Instances	Status	Desired capacity	Min	Max	Availability Zones
Auto Scaling Windows SRV 2022 ISS	LaunchTemplateWindowsSRV2022ISS	Version Default	1	-	1	1	3	us-east-1a, us-east-1b, us-east-1c

Ilustración 55 Instancias en ejecución Auto Scaling

Estableceremos como política de auto escalado el uso de procesador y tiempo, para este caso uso de CPU al 70% durante 120 segundos.

EC2 > Auto Scaling groups > Auto Scaling Windows SRV 2022 ISS

Edit dynamic scaling policy

Policy type

Target tracking scaling

Scaling policy name

Target Tracking Policy

Metric type [Info](#)

Monitored metric that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.

Average CPU utilization

Target value

70

Instance warmup [Info](#)

120 seconds

Disable scale in to create only a scale-out policy

Cancel Update

Ilustración 56 Política de auto escalado basado en métricas

Procedemos a realizar prueba de stress sobre la instancia en ejecución, para comprobar el funcionamiento del auto escalado

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs
WindowServer2022	i-0661618c8d89c7b5d	Stopped	t2.micro		View alarms +	us-east-1c				
	i-062b7139e3b40b3a1	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-44-212-220-99.compute-1.amazonaws.com	44.212.220.99		

Ilustración 57 Instancias en ejecución

Lo haremos a través de abrir una página web en el navegador, la cual se encargará de aumentar el consumo de CPU.

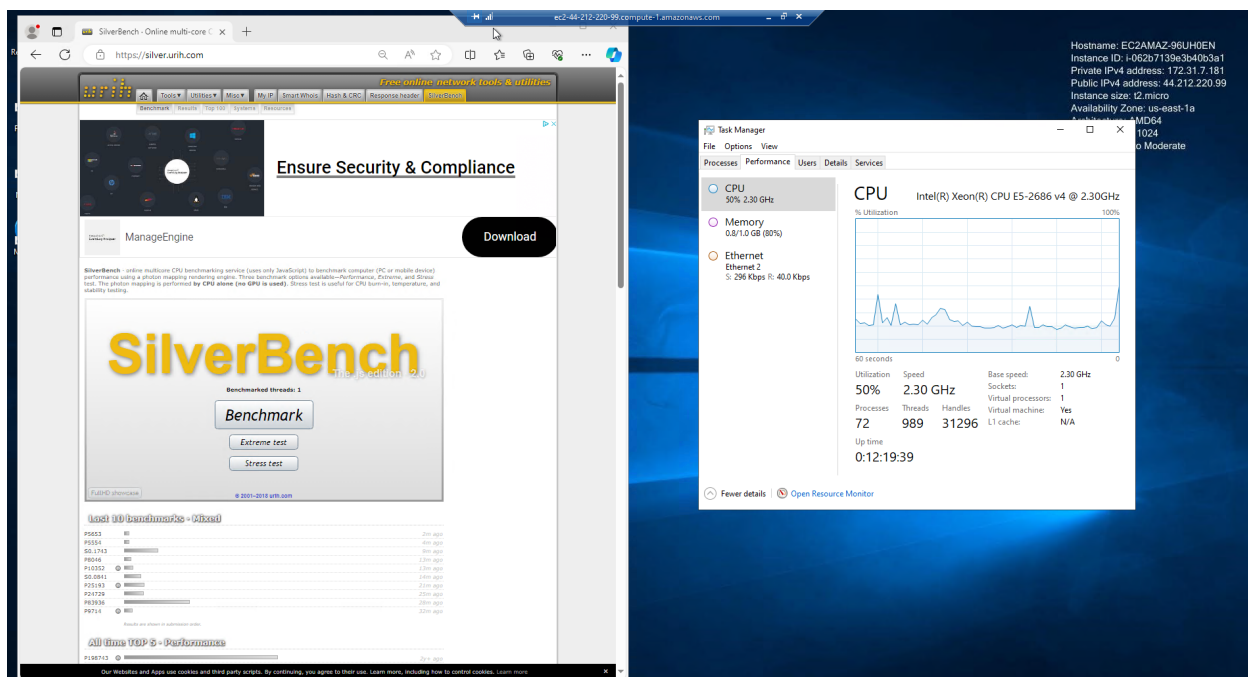


Ilustración 58 Antes de prueba de estrés de CPU

Inicio de la prueba

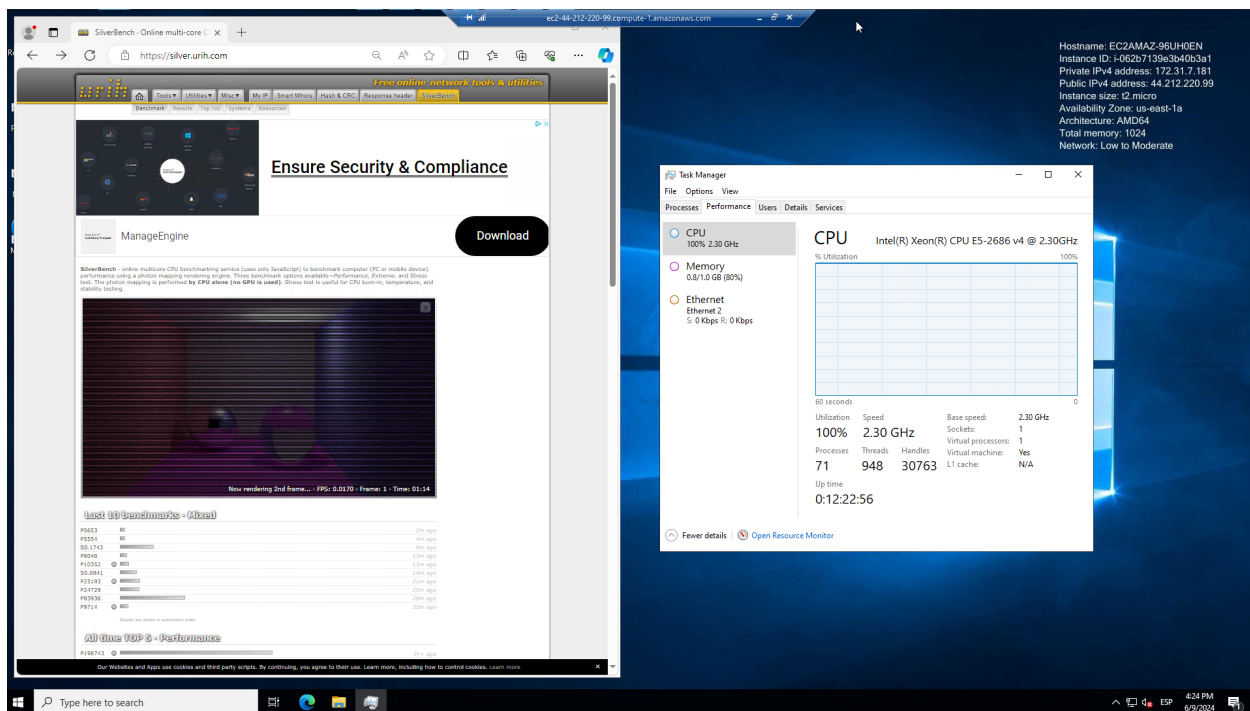


Ilustración 59 Inicio de la prueba de estrés

Al cumplirse las condiciones de aumento de CPU se comienza a preparar la segunda instancia

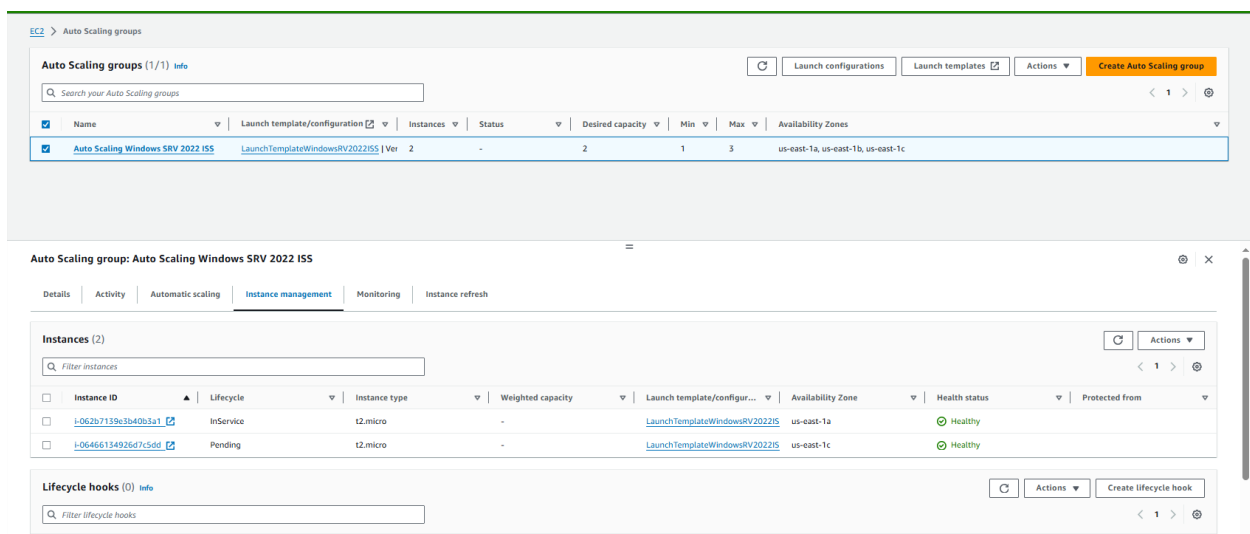


Ilustración 60 Segunda instancia en preparación para atender el auto escalado

Dos instancias en ejecución por el funcionamiento del auto escalado

Auto Scaling group: Auto Scaling Windows SRV 2022 ISS

Details | Activity | Automatic scaling | **Instance management** | Monitoring | Instance refresh

Instances (2)

Filter instances

Instance ID	Lifecycle	Instance type	Weighted capacity	Launch template/configur...	Availability Zone	Health status	Protected from
i-062b7139e3b40b3a1	InService	t2.micro	-	LaunchTemplateWindowsRV2022IS	us-east-1a	Healthy	
i-06466134926d7c5dd	InService	t2.micro	-	LaunchTemplateWindowsRV2022IS	us-east-1c	Healthy	

Ilustración 61 Segunda instancia en ejecución

Al continuar la prueba de estrés el auto escalado lanza la tercera instancia, observamos de forma adicional que se está cumpliendo la política de acuerdo con las zonas que habíamos escogido las está lanzando en diferentes zonas de disponibilidad

EC2 > Auto Scaling groups

Auto Scaling groups (1/1) Info

Search your Auto Scaling groups

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
Auto Scaling Windows SRV 2022 ISS	LaunchTemplateWindowsRV2022IS Ver	3	-	3	1	3	us-east-1a, us-east-1b, us-east-1c

Auto Scaling group: Auto Scaling Windows SRV 2022 ISS

Details | Activity | Automatic scaling | **Instance management** | Monitoring | Instance refresh

Instances (3)

Filter instances

Instance ID	Lifecycle	Instance type	Weighted capacity	Launch template/configur...	Availability Zone	Health status	Protected from
i-062b7139e3b40b3a1	InService	t2.micro	-	LaunchTemplateWindowsRV2022IS	us-east-1a	Healthy	
i-06466134926d7c5dd	InService	t2.micro	-	LaunchTemplateWindowsRV2022IS	us-east-1c	Healthy	
i-0f87536388a73d5fe	InService	t2.micro	-	LaunchTemplateWindowsRV2022IS	us-east-1b	Healthy	

Lifecycle hooks (0) Info

Ilustración 62 Auto escalado con 3 instancias en ejecución

Verificamos el balanceador y nos muestra que tiene 3 instancias para balancear la carga

The screenshot displays the AWS Management Console interface for a Load Balancer. At the top, it shows 'Load balancers (1/1)' with a search bar and a table listing the load balancer 'LoadBalancerWindowsSRV2022IIS'. Below this, the 'Resource map' section provides a visual overview of the load balancer's architecture. It includes a 'Listeners (1)' box for 'HTTP:80', a 'Rules (1)' box for 'Priority default', and a 'Target groups (1) info' box for 'LoadBalancerWindowsSRV2022IIS'. The 'Targets (3)' box shows three EC2 instances: 'i-062b7139c3b40b3a1', 'i-06466134926d7c5dd', and 'i-0f87536388a73d5fe', all marked as 'Healthy'.

Ilustración 63 Balanceador de carga en funcionamiento con 3 instancias

Observamos el historial de actividad donde se registran los aumentos de capacidad

The screenshot shows the 'Auto Scaling groups (1/1) Info' section in the AWS Management Console. Below the group overview, the 'Activity history' table provides a detailed log of capacity changes. The table has columns for Status, Description, Cause, Start time, and End time.

Status	Description	Cause	Start time	End time
Successful	Launching a new EC2 instance: i-0f87536388a73d5fe	At 2024-06-09T16:36:55Z a monitor alarm TargetTracking-Auto Scaling Windows SRV 2022 ISS-AlarmHigh-64d744fe-3a4d-47ff-9891-9f558263272e in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 2 to 3. At 2024-06-09T16:37:07Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 2 to 3.	2024 June 09, 11:37:09 AM -05:00	2024 June 09, 11:39:40 AM -05:00
Successful	Launching a new EC2 instance: i-06466134926d7c5dd	At 2024-06-09T16:31:55Z a monitor alarm TargetTracking-Auto Scaling Windows SRV 2022 ISS-AlarmHigh-64d744fe-3a4d-47ff-9891-9f558263272e in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 1 to 2. At 2024-06-09T16:32:06Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2.	2024 June 09, 11:32:08 AM -05:00	2024 June 09, 11:34:39 AM -05:00

Ilustración 64 Registros de actividad

Detenemos la prueba de estrés

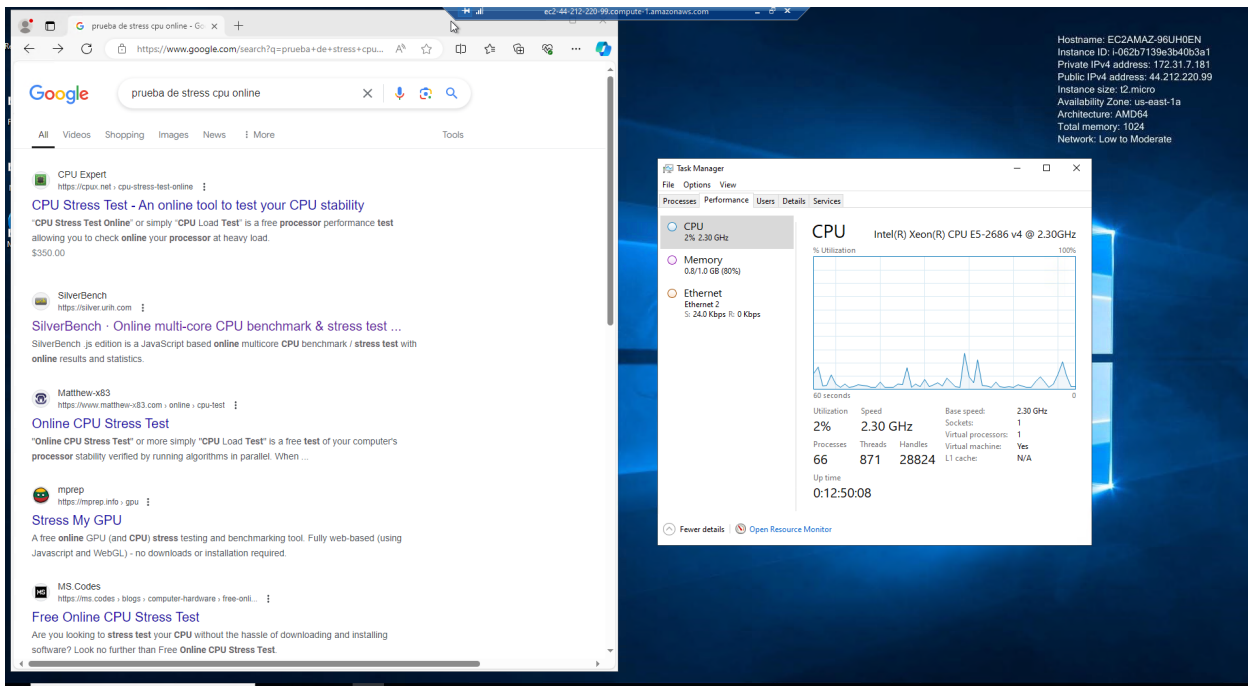


Ilustración 65 Detener prueba de estrés

Pasado el tiempo de espera se comienza a des escalar en el auto scaling

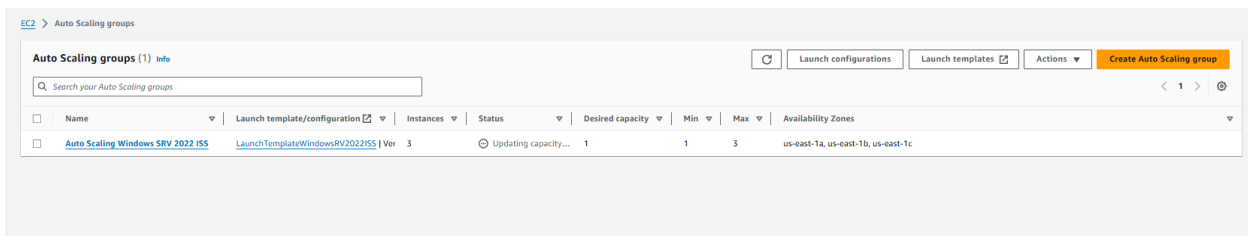


Ilustración 66 Cambio de estado en el auto scaling

Pasado el tiempo de espera se comienza a des escalar en el balanceador

Load balancers (1/1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Filter load balancers

Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
LoadBalancerWindowsSRV2022I55	LoadBalancerWindowsSRV2022I55	Active	vpc-0d1cf902901d8f24	us-east-1a, us-east-1b	application	June 8, 2024, 22:35 (UTC-05:00)

Load balancer: LoadBalancerWindowsSRV2022I55

Listeners (1)

- HTTP-80 (1 rule)

Rules (1)

- Priority default: Forward to target group
- Conditions (1): If no other rule applies

Target groups (1) info

- Instance: LoadBalancerWindowsSRV2022I55 (3 targets)
- Health: 1 Healthy, 0 Draining, 0 Pending, 0 Unavailable, 2 Deleted

Targets (3)

- i-03f377c8242b25428 (Port 80): Healthy
- i-062b7139e3b40b3a1 (Port 80): Draining: Target deregistration is in progress
- i-07d3a2e8df76effb (Port 80): Draining: Target deregistration is in progress

Ilustración 67 Desmonte de instancias en el balanceador

Auto Scaling groups (1/1) info

Search your Auto Scaling groups

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
Auto Scaling Windows SRV 2022 I55	LaunchTemplateWindowsRV2022I55 Ver	3	Updating capacity...	1	1	3	us-east-1a, us-east-1b, us-east-1c

Auto Scaling group: Auto Scaling Windows SRV 2022 I55

Details | Activity | Automatic scaling | **Instance management** | Monitoring | Instance refresh

Instances (3)

Instance ID	Lifecycle	Instance type	Weighted capacity	Launch template/configur...	Availability Zone	Health status	Protected from
i-062b7139e3b40b3a1	InService	t2.micro	-	LaunchTemplateWindowsRV2022I55	us-east-1a	Healthy	
i-0646e134926d7c5dd	Terminating	t2.micro	-	LaunchTemplateWindowsRV2022I55	us-east-1c	Healthy	
i-0f87536388a73d5fe	Terminating	t2.micro	-	LaunchTemplateWindowsRV2022I55	us-east-1b	Healthy	

Lifecycle hooks (0) info

Filter lifecycle hooks

Name	Lifecycle transition	Default result	Heartbeat timeout (seconds)	Notification target ARN	Role ARN
No lifecycle hooks are currently configured.					

Lifecycle hooks help you perform custom actions on instances as they launch and before they terminate.

Create lifecycle hook

Ilustración 68 Proceso de reducción de instancias

Gradualmente va terminando las instancias

The screenshot shows the AWS Management Console interface for an Auto Scaling group named "Auto Scaling Windows SRV 2022 ISS". The group is currently in the "Instance management" tab. The "Instances" section shows two instances:

Instance ID	Lifecycle	Instance type	Weighted capacity	Launch template/configur...	Availability Zone	Health status	Protected from
i-062b7139e3b40b3a1	InService	t2.micro	-	LaunchTemplateWindowsRV2022IS	us-east-1a	Healthy	
i-0f87536388a73d5fc	Terminating	t2.micro	-	LaunchTemplateWindowsRV2022IS	us-east-1b	Healthy	

The "Lifecycle hooks" section shows that no lifecycle hooks are currently configured. The "Warm pool" section also shows that no warm pool is currently configured.

Ilustración 69 2 instancias corriendo

Regresa al mínimo de instancias

The screenshot shows the same AWS Management Console interface for the "Auto Scaling Windows SRV 2022 ISS" group. The "Instances" section now shows only one instance:

Instance ID	Lifecycle	Instance type	Weighted capacity	Launch template/configur...	Availability Zone	Health status	Protected from
i-062b7139e3b40b3a1	InService	t2.micro	-	LaunchTemplateWindowsRV2022IS	us-east-1a	Healthy	

The "Lifecycle hooks" and "Warm pool" sections remain the same as in the previous screenshot.

Ilustración 70 Estado mínimo de instancias

Confirmamos el proceso en el historial de actividad

Auto Scaling group: Auto Scaling Windows SRV 2022 ISS

Activity history (11)

Filter activity history

Status	Description	Cause	Start time	End time
Successful	Terminating EC2 instance: i-0f87556388a73d5fe	At 2024-06-09T17:07:09Z a monitor alarm TargetTracking-Auto Scaling Windows SRV 2022 ISS-AlarmLow-6bda4933-bafe-4ef0-93a1-740c62a972e6 in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 2 to 1. At 2024-06-09T17:07:20Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 2 to 1. At 2024-06-09T17:07:20Z instance i-0f87556388a73d5fe was selected for termination.	2024 June 09, 12:07:20 PM -05:00	2024 June 09, 12:13:03 PM -05:00
Successful	Terminating EC2 instance: i-06466134926d7c5dd	At 2024-06-09T17:06:10Z a monitor alarm TargetTracking-Auto Scaling Windows SRV 2022 ISS-AlarmLow-6bda4933-bafe-4ef0-93a1-740c62a972e6 in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 3 to 2. At 2024-06-09T17:06:18Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 3 to 2. At 2024-06-09T17:06:18Z instance i-06466134926d7c5dd was selected for termination.	2024 June 09, 12:06:18 PM -05:00	2024 June 09, 12:12:00 PM -05:00
Successful	Launching a new EC2 instance: i-0f87556388a73d5fe	At 2024-06-09T16:36:55Z a monitor alarm TargetTracking-Auto Scaling Windows SRV 2022 ISS-AlarmHigh-64d744fe-3a4d-47ff-9891-9f5582d3272e in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 2 to 3. At 2024-06-09T16:37:07Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 2 to 3.	2024 June 09, 11:37:09 AM -05:00	2024 June 09, 11:39:40 AM -05:00
Successful	Launching a new EC2 instance: i-06466134926d7c5dd	At 2024-06-09T16:31:55Z a monitor alarm TargetTracking-Auto Scaling Windows SRV 2022 ISS-AlarmHigh-64d744fe-3a4d-47ff-9891-9f5582d3272e in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 1 to 2. At 2024-06-09T16:32:06Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2.	2024 June 09, 11:32:08 AM -05:00	2024 June 09, 11:34:39 AM -05:00

Ilustración 71 historial de actividad del auto scaling

Índice de Tablas

Tabla 1 Comparativa Nombre de Servicios Cloud 19

Índice de ilustraciones

<i>Ilustración 1 Línea de tiempo Hitos Computación en la nube</i>	18
<i>Ilustración 2 Resumen de instancia Creada</i>	22
<i>Ilustración 3 VPC creada</i>	23
<i>Ilustración 4 Grupo de Seguridad</i>	24
<i>Ilustración 5 Configuración SSH -1</i>	25
<i>Ilustración 6 Configuración SSH -2</i>	26
<i>Ilustración 7 Ingreso por SSH</i>	26
<i>Ilustración 8 Ejecución del comando docker ps -a</i>	28
<i>Ilustración 9 Contenido de los archivos de configuración de apache</i>	30
<i>Ilustración 10 Acceso por IP y puerto al contenedor1</i>	32
<i>Ilustración 11 Acceso por nombre de dominio al contenedor1</i>	32
<i>Ilustración 12 Acceso por IP y puerto al contenedor2</i>	32
<i>Ilustración 13 Acceso por nombre de dominio al contenedor2</i>	33
<i>Ilustración 14 Ruta archivo hosts</i>	33
<i>Ilustración 15 Contenido archivo hosts</i>	34
<i>Ilustración 16 Usuario creado IAM Identity Center</i>	35
<i>Ilustración 17 Grupo creado</i>	36
<i>Ilustración 18 Asociación de cuenta AWS al grupo</i>	36
<i>Ilustración 19 Ingreso a la cuenta creada</i>	37
<i>Ilustración 20 Bucket creado</i>	37
<i>Ilustración 21 Instancia EC2</i>	38
<i>Ilustración 22 Instalación AWS CLI</i>	38
<i>Ilustración 23 Confirmación de instalación AWS CLI</i>	39
<i>Ilustración 24 Access Keys</i>	40

<i>Ilustración 25 variables de conexión y prueba de listar el contenido del S3</i>	<i>41</i>
<i>Ilustración 26 Archivo subido con éxito</i>	<i>41</i>
<i>Ilustración 27 Creación Base de Datos.....</i>	<i>42</i>
<i>Ilustración 28 Creación de base de datos</i>	<i>43</i>
<i>Ilustración 29 Proceso de creación terminado.....</i>	<i>43</i>
<i>Ilustración 30 Conexión a base de datos.....</i>	<i>44</i>
<i>Ilustración 31 Creación de base de datos</i>	<i>45</i>
<i>Ilustración 32 Creación de tabla, inserción de datos</i>	<i>46</i>
<i>Ilustración 33 Instalación Web server IIS.....</i>	<i>48</i>
<i>Ilustración 34 Instalación finalizada IIS.....</i>	<i>48</i>
<i>Ilustración 35 Respuesta servidor web IIS.....</i>	<i>49</i>
<i>Ilustración 36 Creación Balanceador</i>	<i>50</i>
<i>Ilustración 37Resumen de Balanceador.....</i>	<i>51</i>
<i>Ilustración 38 Listener y ruta</i>	<i>51</i>
<i>Ilustración 39 VPCy Zonas de disponibilidad.....</i>	<i>51</i>
<i>Ilustración 40 Grupo de Seguridad.....</i>	<i>52</i>
<i>Ilustración 41 Mapa de recursos.....</i>	<i>52</i>
<i>Ilustración 42 Respuesta del balanceador</i>	<i>53</i>
<i>Ilustración 43 Creación Snapshot.....</i>	<i>53</i>
<i>Ilustración 44 Snapshot creada con éxito</i>	<i>54</i>
<i>Ilustración 45 Creación AMI del Snapshot</i>	<i>55</i>
<i>Ilustración 46 AMI Personalizada</i>	<i>55</i>
<i>Ilustración 47 Creación Launch Template.....</i>	<i>56</i>
<i>Ilustración 48 Proceso terminado creación Launch Template</i>	<i>57</i>
<i>Ilustración 49 Template disponible</i>	<i>57</i>

<i>Ilustración 50 Creación Auto Escalado.....</i>	<i>58</i>
<i>Ilustración 51 Opciones de launch template.....</i>	<i>59</i>
<i>Ilustración 52 Asociación a balanceador existente.....</i>	<i>60</i>
<i>Ilustración 53 Cantidad de instancias en ejecución.....</i>	<i>61</i>
<i>Ilustración 54 Resumen de Auto Scaling.....</i>	<i>62</i>
<i>Ilustración 55 Instancias en ejecución Auto Scaling.....</i>	<i>62</i>
<i>Ilustración 56 Política de auto escalado basado en métricas.....</i>	<i>63</i>
<i>Ilustración 57 Instancias en ejecución.....</i>	<i>63</i>
<i>Ilustración 58 Antes de prueba de estrés de CPU.....</i>	<i>64</i>
<i>Ilustración 59 Inicio de la prueba de estrés.....</i>	<i>65</i>
<i>Ilustración 60 Segunda instancia en preparación para atender el auto escalado.....</i>	<i>65</i>
<i>Ilustración 61 Segunda instancia en ejecución.....</i>	<i>66</i>
<i>Ilustración 62 Auto escalado con 3 instancias en ejecución.....</i>	<i>66</i>
<i>Ilustración 63 Balanceador de carga en funcionamiento con 3 instancias.....</i>	<i>67</i>
<i>Ilustración 64 Registros de actividad.....</i>	<i>67</i>
<i>Ilustración 65 Detener prueba de estrés.....</i>	<i>68</i>
<i>Ilustración 66 Cambio de estado en el auto scaling.....</i>	<i>68</i>
<i>Ilustración 67 Desmonte de instancias en el balanceador.....</i>	<i>69</i>
<i>Ilustración 68 Proceso de reducción de instancias.....</i>	<i>69</i>
<i>Ilustración 69 2 instancias corriendo.....</i>	<i>70</i>
<i>Ilustración 70 Estado mínimo de instancias.....</i>	<i>70</i>
<i>Ilustración 71 historial de actividad del auto scaling.....</i>	<i>71</i>

Conclusiones

El seminario de grado nos ha llevado a descubrir el Extenso universo de AWS y sus servicios, revelando cómo estos pueden convertirse en aliados indispensables para las empresas modernas. AWS no es solo un conjunto de herramientas tecnológicas; es una plataforma que facilita la creación, gestión y expansión de nuestras ideas y proyectos, permitiéndonos soñar en grande y ejecutar con precisión.

Durante el seminario, exploramos cómo Amazon EC2 nos ofrece la capacidad de desplegar servidores virtuales de manera flexible, ajustándose a nuestras necesidades específicas, ya sea para un proyecto pequeño o una aplicación de gran escala. Con Amazon VPC, tuvimos la oportunidad de aprender a crear redes virtuales aisladas, brindándonos un control total sobre nuestro entorno de red, asegurando que nuestros datos y aplicaciones estén protegidos.

La seguridad es un tema que nunca podemos subestimar, y los Security Groups nos mostraron cómo establecer reglas claras y precisas para proteger nuestros recursos contra accesos no autorizados. A través del IAM Identity Center, comprendimos la importancia de gestionar identidades y accesos de manera centralizada, garantizando que solo las personas correctas tengan acceso a la información adecuada.

Amazon S3 nos impresionó con su capacidad de almacenamiento prácticamente ilimitada, ideal para guardar y recuperar datos de manera segura y eficiente. Al conocer Amazon RDS, vimos cómo gestionar bases de datos relacionales se simplifica, permitiéndonos enfocarnos más en el análisis y menos en la administración.

Una de las revelaciones más importantes fue entender el poder del autoescalado con Amazon EC2 Auto Scaling. Este servicio asegura que nuestras aplicaciones siempre tengan los recursos necesarios para funcionar de manera óptima, ajustándose automáticamente a las variaciones en la demanda. Y, por supuesto, el Elastic Load Balancing nos enseñó a distribuir el tráfico de manera equitativa entre nuestras instancias, asegurando una experiencia de usuario constante y sin interrupciones.

En resumen, AWS nos ofrece una base robusta y adaptable que no solo se alinea con las necesidades actuales de las empresas, sino que también las prepara para enfrentar los desafíos futuros. Este seminario ha demostrado que con las herramientas adecuadas, podemos construir y escalar aplicaciones de manera eficiente y segura, siempre listos para innovar y crecer en el dinámico entorno digital de hoy.

Referencias

Amazon. (n.d.). Retrieved from AWS: Acerca de AWS: <https://aws.amazon.com/es/about-aws/>

Amazon. (2024, 05 19). Retrieved from <https://aws.amazon.com/es/what-is/compute/>

Amazon AWS. (n.d.). Retrieved from <https://aws.amazon.com/es/rds/>

Amazon AWS. (n.d.). Retrieved from <https://aws.amazon.com/es/s3/>

Amazon AWS. (n.d.). Retrieved from <https://aws.amazon.com/es/ec2/>

Amazon AWS. (n.d.). Retrieved from <https://aws.amazon.com/es/amazon-linux-2>

Amazon AWS. (n.d.). Retrieved from <https://aws.amazon.com/es/vpc/>

Amazon AWS. (n.d.). Retrieved from https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/ec2-security-groups.html

Amazon AWS. (n.d.). Retrieved from <https://aws.amazon.com/es/elasticloadbalancing/>

Amazon AWS. (n.d.). Retrieved from <https://aws.amazon.com/es/ec2/autoscaling/>

Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., & Zaharia, M. (2010). *A view of cloud computing* (Vol. 53(4)). Communications of the ACM.

AWS. (2024). Retrieved from https://aws.amazon.com/es/products/?nc2=h_ql_prod_fs_f&aws-products-all.sort-by=item.additionalFields.productNameLowercase&aws-products-all.sort-order=asc&awsf.re%3AInvent=*all&awsf.Free%20Tier%20Type=*all&awsf.tech-category=*all

AZURE. (2024). Retrieved from <https://azure.microsoft.com/en-us/products/>

Barroso, L. A., & Cidon, I. (2009). *The Datacenter as a Computer: An Introduction to the Development of Cloud Computing*. Morgan Kaufmann.

- BBC*. (2018, junio 02). Retrieved from <https://www.bbc.com/mundo/noticias-44290222>
- Cerf, V. G. (1982). *Introduction to the internet* (Vol. 70(10)). Proceedings of the IEEE.
- Cloudflare*. (n.d.). Retrieved from <https://www.cloudflare.com/es-es/learning/access-management/what-is-ssh/>
- Creasy, R. (1981). The Origin of the VM/370 Time-Sharing System. *BM Journal of Research and Development*, 483-490.
- Docker*. (n.d.). Retrieved from <https://www.docker.com/>
- Garcia Pombo, K., & Fernandez Romero, Y. (2011). Virtualización. *Revista Telematica*, 61-63. Retrieved from <http://148.202.167.116:8080/jspui/bitstream/123456789/2281/1/Virtualizaci%C3%B3n.pdf>
- Gartner*. (2021). Retrieved from Gartner Forecasts Worldwide Public Cloud End-User Spending to Grow 18% in 2021: <https://www.gartner.com/en/newsroom/press-releases/2021-04-21-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-grow-18-percent-in-2021>
- google cloud*. (2024). Retrieved from https://cloud.google.com/products?hl=es_419
- Google Cloud*. (2024, 05 19). Retrieved from <https://cloud.google.com/learn/what-is-cloud-computing?hl=es-419>
- Google Cloud Platform*. (2023). Retrieved from <https://cloud.google.com/>
- Google Workspace*. (2023). Retrieved from <https://workspace.google.com/>
- hpe*. (2024, 05 19). Retrieved from <https://www.hpe.com/lamerica/es/what-is/virtualization.html#:~:text=El%20origen%20de%20la%20virtualizaci%C3%B3n,un%20proceso%20a%20la%20vez.>

Microsoft Azure. (2023). Retrieved from <https://azure.microsoft.com/en-us>

redhat. (2023, enero 25). Retrieved from <https://www.redhat.com/es/topics/virtualization/what-is-virtualization>

Subrahmanyam, R., & Buyya, R. (2010). *Mastering Cloud Computing: Concepts, Technologies, & Architectures.* Tata McGraw-Hill Education.

The Apache Software Foundation. (n.d.). Retrieved from https://httpd.apache.org/docs/2.4/howto/reverse_proxy.html

The Apache Software Foundation. (n.d.). Retrieved from <https://httpd.apache.org/>

VMware. (2021). Retrieved from VMware ESX Server: <https://www.vmware.com/products/esxi-and-esx.html>

zippia. (2024, 05 19). Retrieved from [www.vmware.com: https://www.zippia.com/vmware-careers-12537/history/](https://www.zippia.com/vmware-careers-12537/history/)