



TRABAJO DE GRADO
Opción Seminario-Diplomado.

Aprendizajes clave de servicios en Amazon Web Services

Corporación Universitaria Remington
Facultad de Ingeniería
Ingeniería de Sistemas

Mario Alejandro Rolón Vega
Juan Pablo Berrío López
Opción de Trabajo de grado Seminario-Diplomado
2025

Tabla de Contenidos

Resumen.....	3
Marco conceptual.....	4
AWS:	4
VPC:	4
EC2:	4
Balanceador de carga:	4
Contenedores:	4
Auto Scaling:	4
Security Group:	5
Computación en la nube:	5
AMI:	5
DNS:	5
Marco contextual	5
Desarrollo e implementación del aprendizaje.....	7
Línea de tiempo orígenes de la virtualización y computación en la nube	7
Comparación de nombres de servicios entre AWS, Azure y GCP	8
Implementación de balanceador de carga con auto scaling	9
Implementación de contenedores y prueba de DNS local	16
Conclusiones	21
Referencias.....	22

Resumen

En el presente trabajo durante el seminario de Amazon Web Services (AWS), se abordó la implementación práctica de una arquitectura en la nube utilizando algunos de los principales servicios que ofrece esta plataforma. El objetivo fue comprender el diseño y despliegue de una infraestructura escalable y eficiente para aplicaciones, haciendo uso de componentes clave.

La creación de una Virtual Private Cloud (VPC) como punto de inicio, permitiendo definir una red virtual personalizada para alojar los recursos. Dentro de esta VPC, se desplegaron instancias EC2 que funcionaron como servidores para la aplicación, configurándolas según las necesidades de rendimiento y seguridad. Para garantizar una alta disponibilidad y distribuir el tráfico de manera equitativa, se implementó un Elastic Load Balancer (ELB), que se encargó de redirigir las peticiones entrantes a las instancias activas más óptimas.

Se integró el uso de contenedores de Docker, lo cual facilitó el despliegue eficiente de aplicaciones empaquetadas.

Palabras clave

Amazon Web Services (AWS), Virtual Private Cloud (VPC), Elastic Compute Cloud (EC2), balanceador de cargas, contenedores.

Marco conceptual

AWS: (Amazon Web Services) es una plataforma de computación en la nube que ofrece servicios tecnológicos bajo demanda a través de internet, permitiendo a los usuarios acceder a una infraestructura, herramientas y aplicaciones de forma flexible y con un modelo de pago por uso. (AWS, 2024)

VPC: (Virtual Private Cloud) es un servicio de AWS que permite crear una sección aislada de la nube, donde se pueden lanzar recursos en una red virtual definida, con control sobre aspectos como IP, subredes y seguridad. (AWS, s.f.)

EC2: (Elastic Compute Cloud) es un servicio que permite crear, configurar y gestionar servidores virtuales escalables en la infraestructura de AWS, ofreciendo flexibilidad para ajustar la capacidad según las necesidades. (AWS, 2015)

Balanceador de carga: Es un servicio que actúa como un puente inteligente que redirige solicitudes a recursos saludables como instancias EC2, simplificando la gestión de tráfico en la nube, garantizando que las aplicaciones tengan alta disponibilidad ante los picos de demanda. (AWS, s.f.)

Contenedores: Son paquetes que contienen los recursos necesarios para que una aplicación pueda ejecutarse de manera eficiente en cualquier entorno. (AWS, 2023)

Auto Scaling: Es un servicio que ajusta automáticamente la capacidad de los recursos en la nube, garantizando una alta disponibilidad y reduciendo costos al reemplazar instancias defectuosas. (AWS, 2017)

Security Group: Es un cortafuegos virtual que controla el tráfico hacia los recursos en AWS, usando reglas para permitir o denegar conexiones, funcionando como barrera de seguridad. (AWS, 2015)

Computación en la nube: Es un modelo que ofrece recursos informáticos por medio de internet, permitiendo reducir costos operativos y aprovechar servicios globales, seguros y altamente disponibles. (AWS, 2019)

AMI: (Amazon Machine Image) es una plantilla preconfigurada que contiene los componentes básicos para lanzar instancias EC2, incluidos el sistema operativo, configuraciones y permisos, permitiendo replicar entornos de forma rápida. (AWS, 2015)

DNS: (Domain Name System) traduce nombres de dominio que sean comprensibles, facilitando el acceso a servicios web, usando nombres fáciles en lugar de una secuencia de números. (AWS, 2021)

Marco contextual

En el mundo digital de hoy, la computación en la nube ha transformado por completo cómo las organizaciones crean y escalan sus aplicaciones. Este enfoque moderno permite acceder a recursos tecnológicos de forma remota por medio de internet, dejando así la dependencia de costosas infraestructuras físicas. (AWS, 2019) Para esta práctica, nos enfocamos en Amazon Web Services (AWS), una plataforma líder que ofrece una amplia gama de soluciones para satisfacer las necesidades más exigentes de la industria. Implementaremos algunos de los servicios básicos de su capa gratuita para realizar las pruebas.

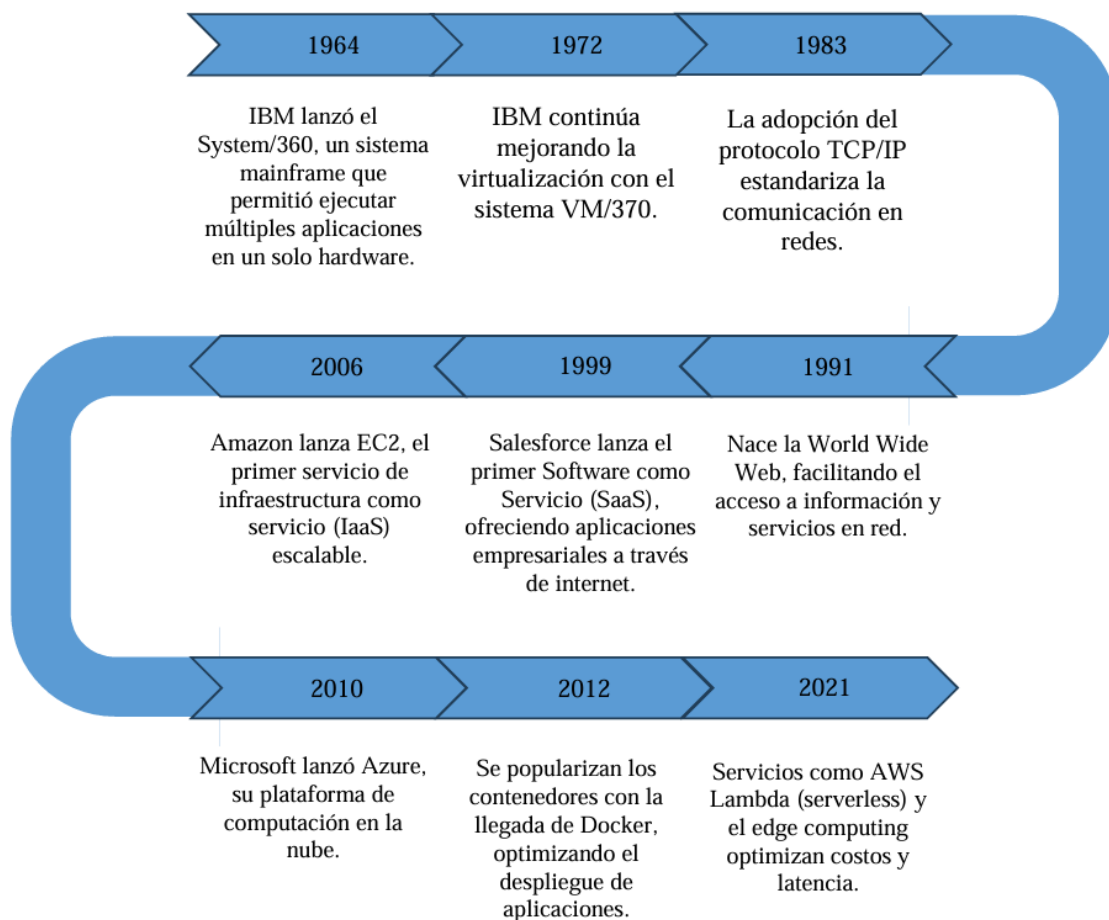
En este entorno, nos adentramos en herramientas esenciales que sustentan arquitecturas modernas y eficientes. Entre estas se destaca Amazon EC2, que ofrece potencia de cómputo flexible a través de instancias virtuales que pueden personalizarse según las circunstancias requeridas. (AWS, 2015) Estas operan dentro de una VPC, una red virtual privada que brinda control y seguridad, ideal para construir entornos complejos y adaptados a cada necesidad. (AWS, s.f.)

Para lograr un alto rendimiento y disponibilidad, integramos un balanceador de carga con auto scaling. Estas dos herramientas reparten el tráfico entre varias instancias EC2 y ajustan los recursos casi en tiempo real según la demanda, lo que permite operar de forma eficiente y rentable. (AWS, 2017)

También adoptamos la filosofía de los contenedores Docker, que permiten empaquetar aplicaciones con todo lo necesario para garantizar su portabilidad. Con herramientas como Amazon ECS (Elastic Container Service), AWS facilita su despliegue y gestión a gran escala, impulsando un desarrollo ágil. (AWS, 2015)

Desarrollo e implementación del aprendizaje

Línea de tiempo orígenes de la virtualización y computación en la nube



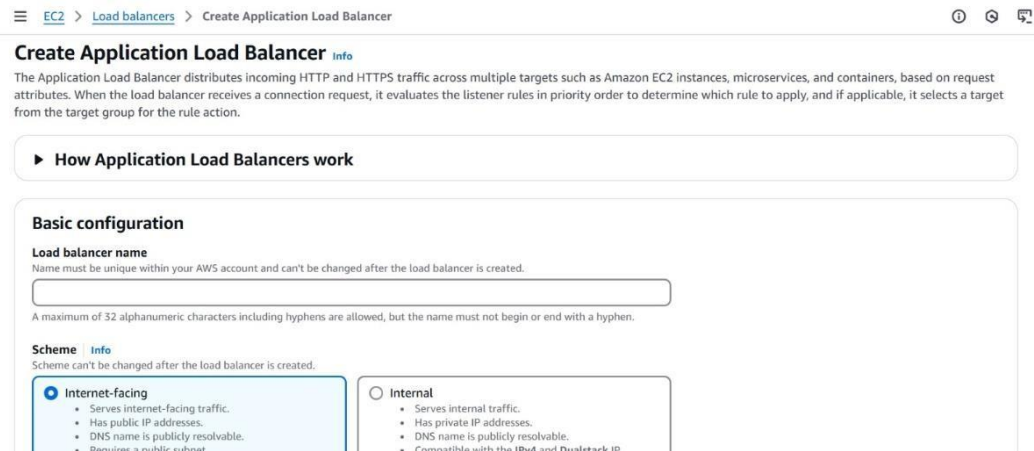
Fuentes consultadas: (WIKIPEDIA, s.f.), (Adaptix Networks, 2025), (Robledano, 2019), (CERN, s.f.)

Comparación de nombres de servicios entre AWS, Azure y GCP

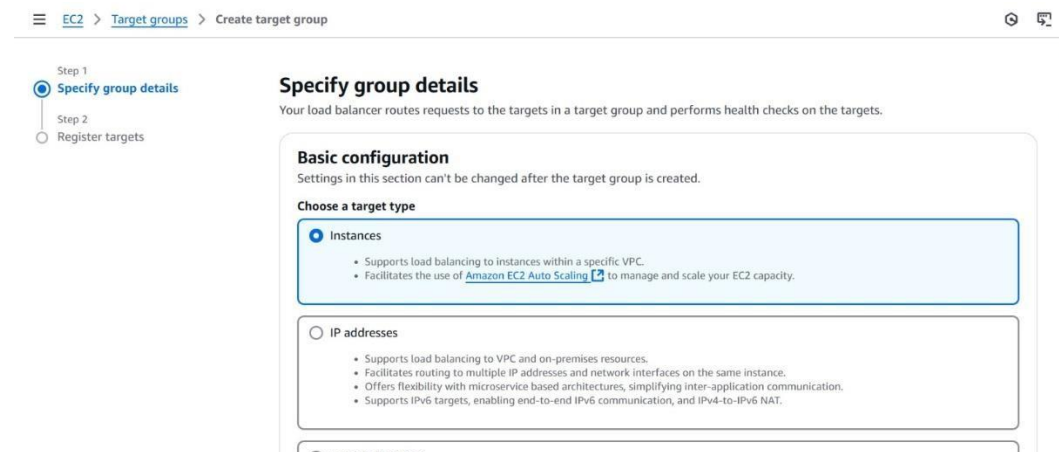
Categoría	AWS	Azure	GCP
Máquinas Virtuales	EC2	Virtual Machines (VMs)	Compute Engine
Contenedores	ECS / EKS	Azure Kubernetes Service (AKS)	Google Kubernetes Engine (GKE)
Funciones Serverless	Lambda	Azure Functions	Cloud Functions
Almacenamiento en bloque	EBS	Managed Disks	Persistent Disks
Almacenamiento en objetos	S3	Blob Storage	Cloud Storage
Base de datos relacional	RDS	Azure SQL Database	Cloud SQL
Base de datos NoSQL	DynamoDB	Cosmos DB	Firestore / Bigtable
Base de datos en memoria	ElastiCache	Azure Cache for Redis	Memorystore
Big Data / Analítica	EMR / Redshift	Synapse Analytics	BigQuery
Mensajería y Colas	SQS / SNS	Service Bus / Event Grid	Pub/Sub
Red Privada	VPC	Virtual Network	VPC
Redes y CDN	CloudFront / Route 53	Azure CDN / Traffic Manager	Cloud CDN / Cloud DNS
Balancede carga	Elastic Load Balancing (ELB)	Azure Load Balancer	Cloud Load Balancing
Identidad y Gestión de Acceso	IAM	Azure Active Directory (AAD)	IAM
Seguridad y Cumplimiento	AWS Shield / AWS WAF	Azure Security Center	Security Command Center
Monitoreo y Logging	CloudWatch / CloudTrail	Azure Monitor / Log Analytics	Cloud Monitoring / Logging

Fuentes consultadas: (Fiz, 2024), (Google Cloud, 2024)

Implementación de balanceador de carga con auto scaling



Creación del balanceador de cargas de aplicación que se encargará de distribuir las consultas de los clientes a los servidores, se configura para que funcione desde internet y se seleccionan los data center en los que van a funcionar los servicios.



Creación del grupo objetivo para las instancias que van a pertenecer al balanceador de carga, se configuran los tiempos para comprobar si las máquinas están funcionando.

EC2 > Target groups > Create target group

Step 1 Specify group details
Step 2 **Register targets**

Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (2/2)

Filter instances

<input checked="" type="checkbox"/>	Instance ID	Name	State	Security group
<input checked="" type="checkbox"/>	i-0f0aeed65bc1347d5	Linux4	Running	SG-ServerLinux
<input checked="" type="checkbox"/>	i-016a6c231acbe1797	Linux3	Running	SG-ServerLinux

2 selected

Ports for the selected instances
Ports for routing traffic to the selected instances.

Selección manual de las instancias que pertenecerán al balanceador.

EC2 > Security Groups > sg-010f740bc8f34bff8 - default

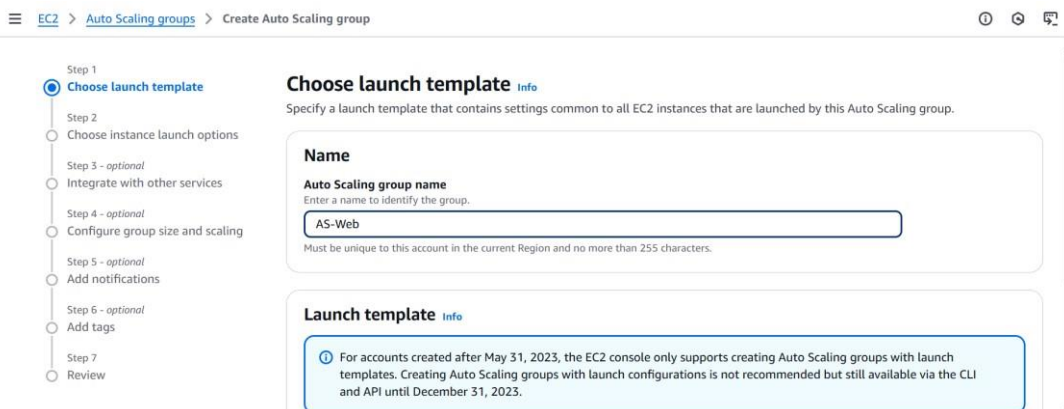
default | sg-010f740bc8f34bff8 | default VPC security group | vpc-0d8b718baa8f7e16e

Owner: 257394464676 | Inbound rules count: 2 Permission entries | Outbound rules count: 1 Permission entry

Inbound rules (2)

<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol
<input type="checkbox"/>	-	sgr-009dae57189dc8a41	IPv4	HTTP	TCP
<input type="checkbox"/>	-	sgr-07aba9cd24de2a709	-	All traffic	All

En el security group del balanceador se agrega una regla que permita el puerto 80 donde estará escuchando las peticiones.



EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1: Choose launch template

Step 2: Choose instance launch options

Step 3 - optional: Integrate with other services

Step 4 - optional: Configure group size and scaling

Step 5 - optional: Add notifications

Step 6 - optional: Add tags

Step 7: Review

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.

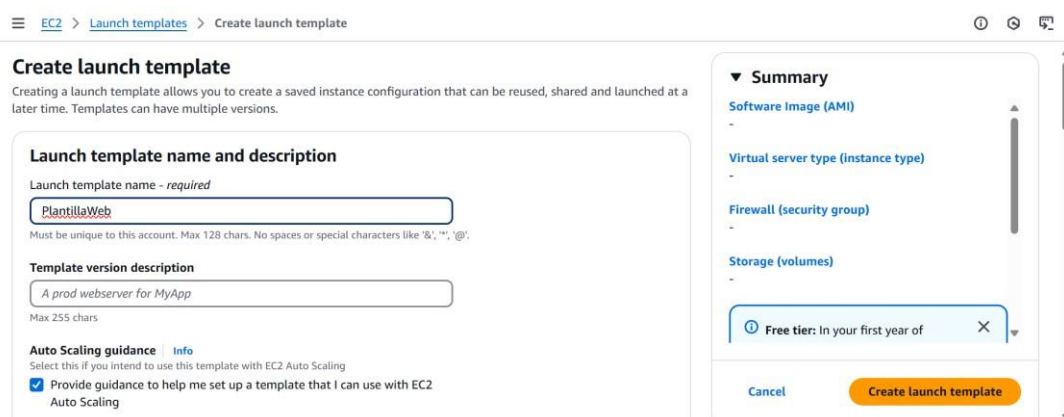
AS-Web

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

Launch template Info

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.

Creación del auto scaling group que lanzará instancias de forma automática.



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*

PlantillaWeb

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

Template version description

A prod webservers for MyApp

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

Summary

Software Image (AMI)

Virtual server type (instance type)

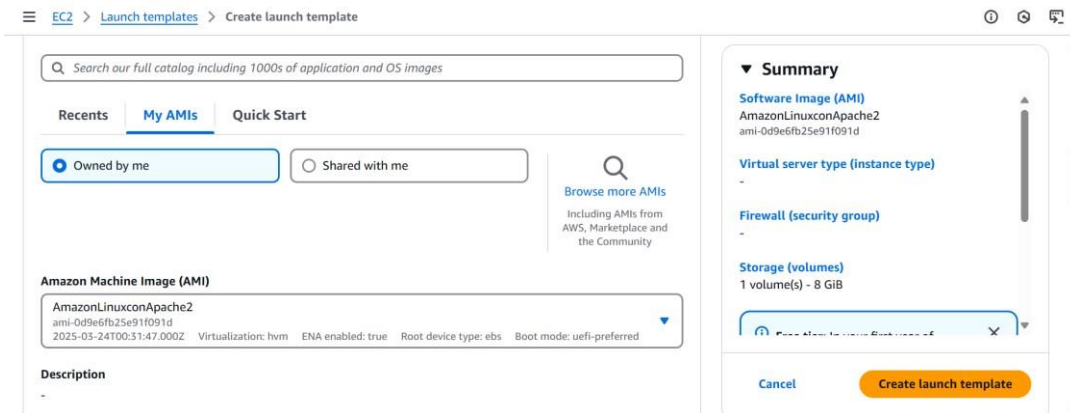
Firewall (security group)

Storage (volumes)

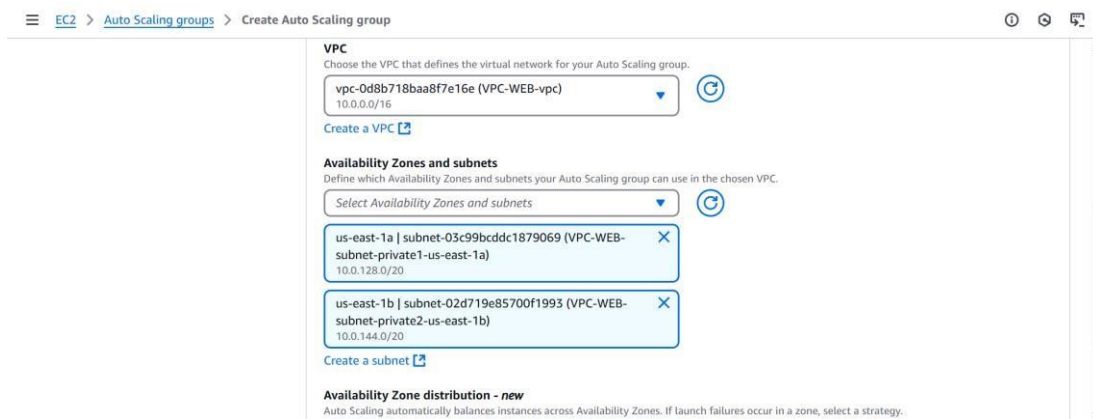
Free tier: In your first year of

Cancel Create launch template

Se crea una plantilla de lanzamiento con la configuración que el auto scaling utilizará para crear las nuevas instancias.



Selección del sistema que van a tener las instancias nuevas, el tipo de instancia, la clave para conectarse.



Configuración de red con el VPC que se ha usado para las instancias y selección de los data center en los que se van a crear las instancias nuevas.

The screenshot shows the 'Health checks' configuration page in the AWS Management Console. The breadcrumb navigation is 'EC2 > Auto Scaling groups > Create Auto Scaling group'. The page title is 'Health checks'. Below the title, there is a brief explanation: 'Health checks increase availability by replacing unhealthy instances. When you use multiple health checks, all are evaluated, and if at least one fails, instance replacement occurs.' Under 'EC2 health checks', there is a link 'Always enabled'. The 'Additional health check types - optional' section includes three options: 'Turn on Elastic Load Balancing health checks' (checked, with a 'Recommended' badge), 'Turn on VPC Lattice health checks' (unchecked), and 'Turn on Amazon EBS health checks' (unchecked). A blue callout box contains the text: 'EC2 Auto Scaling will start to detect and act on health checks performed by Elastic Load Balancing. To avoid unexpected terminations, first verify the settings of these health checks in the Load Balancer console.' The 'Turn on Elastic Load Balancing health checks' option has a sub-description: 'Elastic Load Balancing monitors whether instances are available to handle requests. When it reports an unhealthy instance, EC2 Auto Scaling can replace it on its next periodic check.'

Habilitamos la revisión del chequeo de salud para las máquinas del balanceador.

The screenshot shows the 'Scaling limits' configuration page in the AWS Management Console. The breadcrumb navigation is 'EC2 > Auto Scaling groups > Create Auto Scaling group'. On the left, there is a progress indicator with steps: 'Step 6 - optional', 'Add tags', 'Step 7', and 'Review'. The main section is titled 'Scaling limits' and includes the text: 'You can resize your Auto Scaling group manually or automatically to meet changes in demand.' Below this, there are two input fields: 'Min desired capacity' with the value '1' and 'Max desired capacity' with the value '5'. The 'Min desired capacity' field has a sub-description 'Equal or less than desired capacity' and the 'Max desired capacity' field has 'Equal or greater than desired capacity'. At the bottom, the 'Automatic scaling - optional' section has a heading 'Choose whether to use a target tracking policy' and two radio button options: 'No scaling policies' (selected) and 'Target tracking scaling policy'.

Definimos la capacidad deseada de instancias, para este ejemplo se estableció en 2, el máximo de instancias que tendrá el auto scaling y un mínimo cuando se tiene un bajo consumo de CPU, para que empiece a eliminar instancias que no se están necesitando.

The screenshot shows the AWS Management Console for EC2 Instances. The left sidebar contains navigation options like Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, and Images. The main content area displays a table of 5 instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
Linux4	i-0f0aee65bc1347d5	Running	t2.micro	2/2 checks passed	View alarms
Server1	i-012fb58a48c695aec	Stopped	t2.micro	-	View alarms
Linux3	i-016a6c231acbe1797	Running	t2.micro	2/2 checks passed	View alarms
	i-Odb65c7c8e32ca788	Running	t2.micro	Initializing	View alarms
	i-Oae815dd2c6cb9ce5	Running	t2.micro	Initializing	View alarms

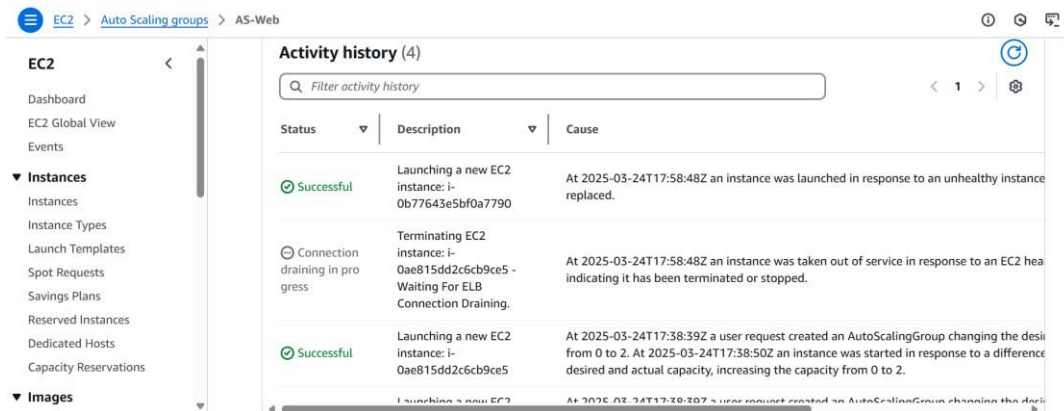
El auto scaling comenzará a crear instancias nuevas, independientes de las que se crearon manualmente, para lograr su capacidad deseada establecida en 2.

The screenshot shows the AWS Management Console for EC2 Instances. The left sidebar contains navigation options like Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, and Snapshots. The main content area displays a table of 1/5 instances:

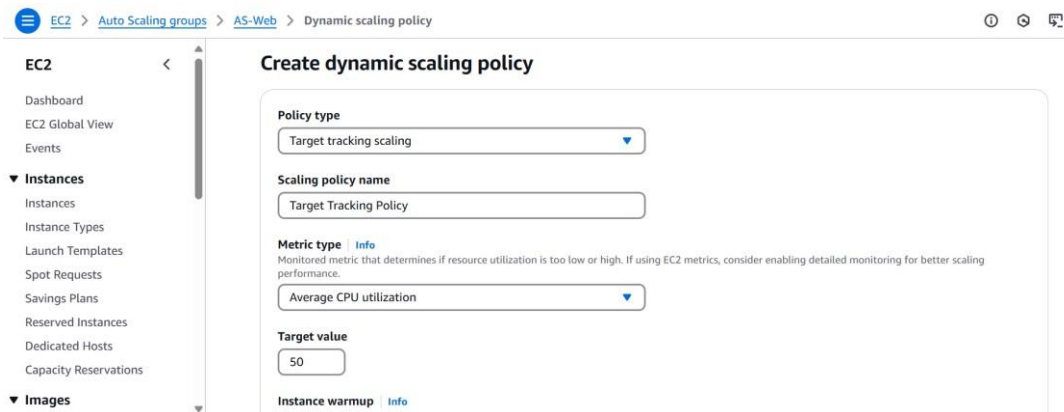
Name	Instance ID	Instance state	Instance type	Status check	Alarm status
Server1	i-012fb58a48c695aec	Stopped	t2.micro	-	View alarms
Linux3	i-016a6c231acbe1797	Running	t2.micro	2/2 checks passed	View alarms
	i-Odb65c7c8e32ca788	Running	t2.micro	2/2 checks passed	View alarms
	i-Oae815dd2c6cb9ce5	Shutting-down	t2.micro	2/2 checks passed	View alarms

The instance **i-Oae815dd2c6cb9ce5** is selected, and its details are shown below the table. The details include tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. The Instance summary is also visible.

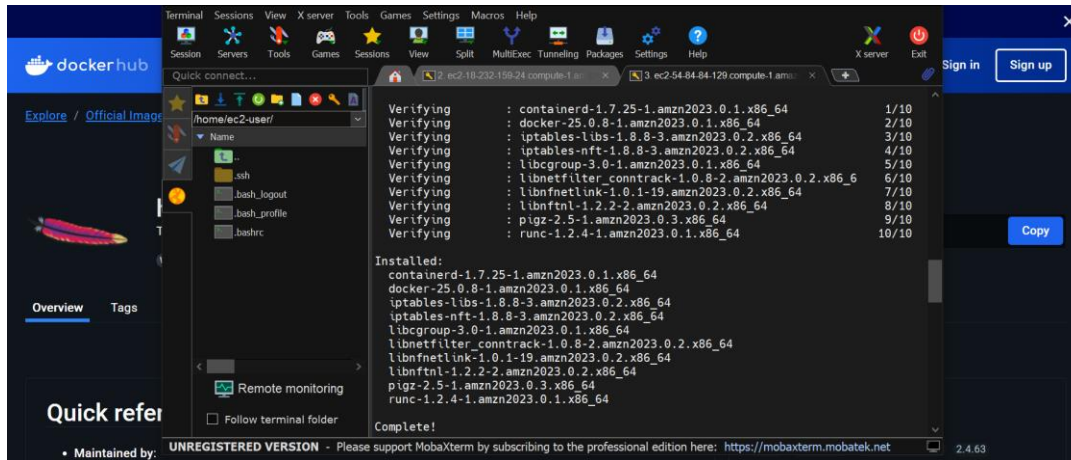
Eliminación de una instancia nueva de forma manual, al que el balanceador ya no podrá llegar y el target group la marcará como no saludable.



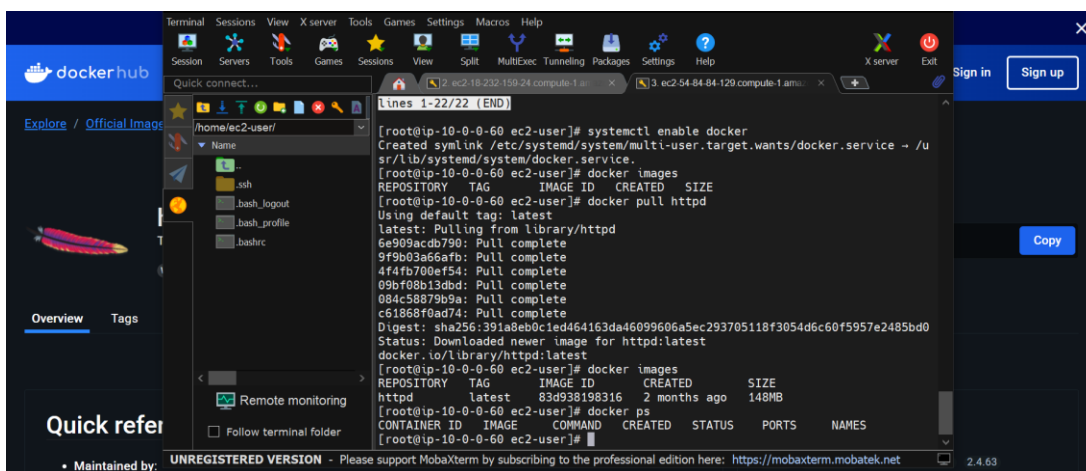
Al definir el valor deseado en 2 y como se ha eliminado una de esas instancias, automáticamente procederá a crear una nueva para mantener dicho valor.



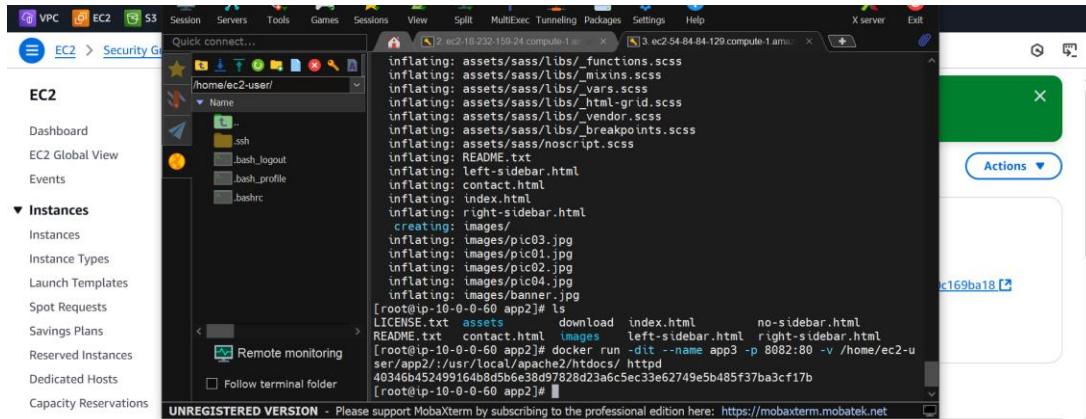
Creación de una política dinámica para un aumento de CPU de un umbral del 50%.



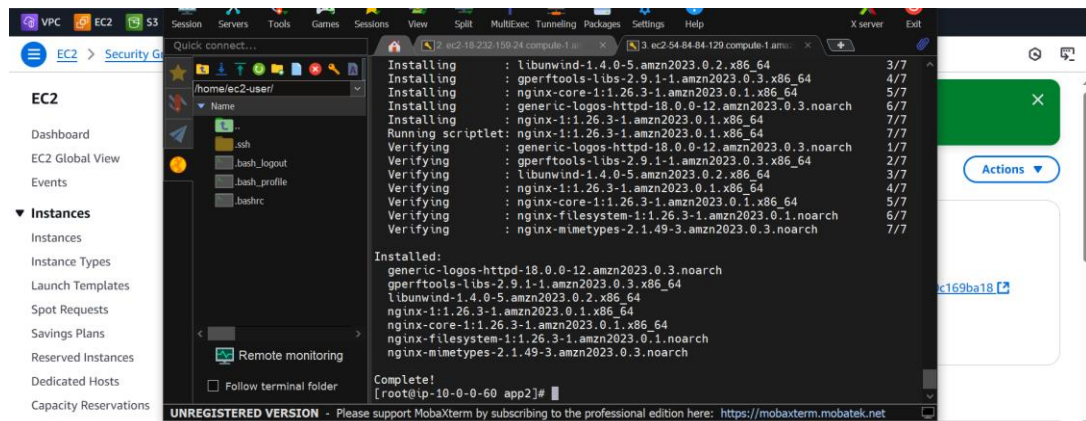
Procedemos a instalar Docker, una plataforma con la que podremos crear y desplegar aplicaciones en contenedores.



Descargamos la imagen de la aplicación httpd, una plantilla con la que ejecutaremos un servidor web en los contenedores.



Creación de los contenedores y asignación de sus puertos específicos, cada contenedor tendrá un puerto diferente dentro de la instancia, dado que cada uno es independiente.



Instalación de nginx, un servidor web que es usado de proxy, como intermediario entre un usuario y sitio web.

```

server {
    listen 80;
    server_name www.tienda1.com;

    location / {
        proxy_pass http://54.84.84.129:8080;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}

server {
    listen 80;
    server_name www.d1.com;

    location / {
        proxy_pass http://54.84.84.129:8080;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}

```

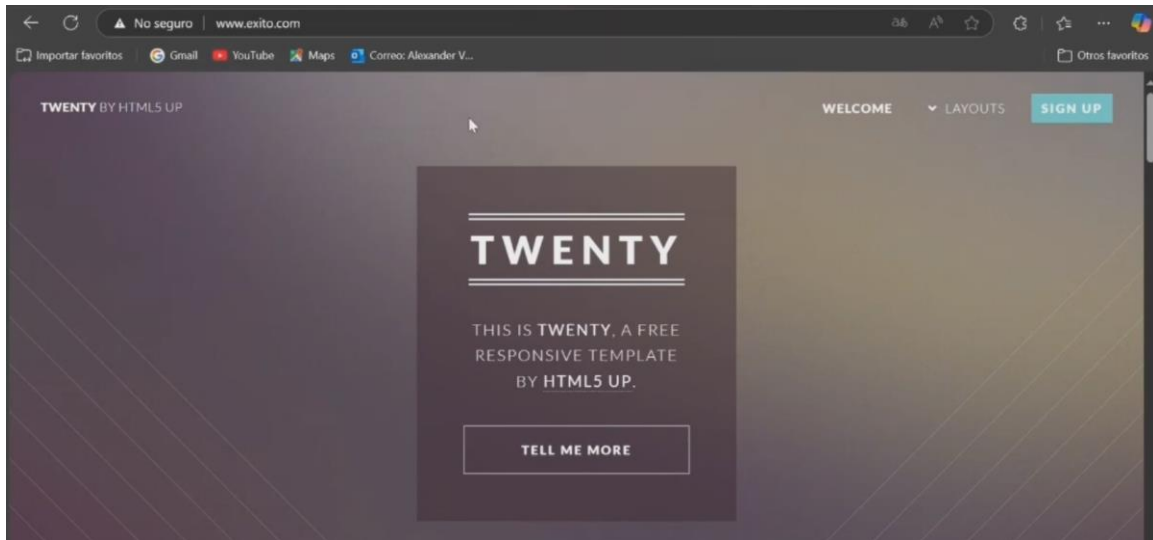
Dentro de los archivos de la carpeta de nginx, creamos un nuevo archivo para la configuración de los DNS y asociamos el contenedor que queremos que se muestre al ingresar un dominio en el navegador. Se especifica un nombre de ejemplo para cada contenedor, para apuntar a dicho nombre y que nos enseñe el contenido deseado, en lugar de escribir su dirección IP.

```

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 54.84.84.129 www.tienda1.com www.d1.com www.exito.com
23

```

En el archivo hosts de nuestro computador, agregamos una línea que contenga los nombres de dominio que definimos anteriormente para realizar la prueba localmente.



Probamos con alguno de los nombres que configuramos en el archivo host. Con el dominio exito.com nos muestra la aplicación del contenedor que estaba asociado a ese dominio. Esta configuración solo podrá realizarse como prueba de manera local desde nuestro equipo.

Conclusiones

Las instancias no solo nos ofrecieron potencia de cómputo bajo demanda, sino que también representaron una forma de pensar la infraestructura como algo vivo, algo que puede adaptarse, crecer o reducirse según nuestras decisiones, como un cuadro sobre el que pintamos nuestras soluciones.

Implementar un balanceador de cargas nos enseñó que la eficiencia no es solo cuestión de velocidad, sino de equilibrio. Distribuir adecuadamente el tráfico no solo mejora el rendimiento, sino que también refleja una arquitectura bien pensada, donde cada componente cumple su función sin saturarse.

Trabajar con contenedores cambió nuestra perspectiva del desarrollo. Pasamos de crear sistemas pesados a construir módulos livianos, portables y más organizados.

La computación en la nube no es simplemente un cambio tecnológico, sino una evolución conceptual, nos olvidamos de pensar en máquinas físicas para empezar a diseñar entornos dinámicos, donde las ideas se despliegan tan rápido como se crean.

Este seminario dejó claro que AWS no es solo una plataforma, sino un ecosistema que impulsa la innovación. A medida que exploramos sus servicios, entendimos que el verdadero poder no está solo en la tecnología, sino en cómo la usamos para resolver problemas reales de forma escalable y creativa.

Referencias

- Adaptix Networks. (2025). *Antecedentes de la Nube*. Obtenido de Adaptix Networks: <https://www.adaptixnetworks.com/antecedentes-la-nube/>
- AWS. (s.f.). *¿Qué es Elastic Load Balancing?* Obtenido de AWS: https://docs.aws.amazon.com/es_es/elasticloadbalancing/latest/userguide/what-is-load-balancing.html
- AWS. (2015). *¿Qué es Amazon EC2?* Obtenido de AWS: https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/concepts.html
- AWS. (2015). *Control de acceso con grupos de seguridad*. Obtenido de AWS: https://docs.aws.amazon.com/es_es/AmazonRDS/latest/UserGuide/Overview.RDSSecurityGroups.html
- AWS. (2015). *Creación de una imagen de contenedor para utilizarla en Amazon ECS*. Obtenido de AWS: https://docs.aws.amazon.com/es_es/AmazonECS/latest/developerguide/create-container-image.html
- AWS. (2015). *Imágenes de máquina de Amazon (AMI) en Amazon EC2*. Obtenido de AWS: https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/AMIs.html
- AWS. (2017). *¿Qué es Amazon EC2 Auto Scaling?* Obtenido de AWS: https://docs.aws.amazon.com/es_es/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html
- AWS. (2019). *¿Qué es la computación en nube?* Obtenido de AWS: https://docs.aws.amazon.com/es_es/whitepapers/latest/aws-overview/what-is-cloud-computing.html
- AWS. (2021). *Descripción de la terminología y conceptos de DNS*. Obtenido de AWS: https://docs.aws.amazon.com/es_es/amplify/latest/userguide/understanding-dns-terminology-and-concepts.html
- AWS. (2023). *¿Qué son los contenedores en la nube?* Obtenido de AWS: <https://aws.amazon.com/what-is/cloud-containers>
- AWS. (2024). *Descripción general de Amazon Web Services*. Obtenido de AWS: https://docs.aws.amazon.com/es_es/whitepapers/latest/aws-overview/introduction.html
- AWS. (s.f.). *Uso de Amazon VPC con otros Servicios de AWS*. Obtenido de AWS: https://docs.aws.amazon.com/es_es/vpc/latest/userguide/related-services.html
- CERN. (s.f.). *Una breve historia de la Web*. Obtenido de CERN: https://home-cern.translate.google/science/computing/birth-web/short-history-web?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=wa
- Fiz, J. (2024). *La guerra de las nubes: AWS vs. Azure vs. GCP*. Obtenido de Paradigma Digital: <https://www.paradigmadigital.com/dev/guerra-de-nubes-aws-azure-gcp/>
- Google Cloud. (2024). *Compara los servicios de AWS y Azure con Google Cloud*. Obtenido de Google Cloud: <https://cloud.google.com/docs/get-started/aws-azure-gcp-service-comparison?hl=es-419>
- Robledano, A. (2019). *¿Qué es TCP/IP?* Obtenido de OpenWebinars: <https://openwebinars.net/blog/que-es-tcpip/>

WIKIPEDIA. (s.f.). *Hipervisor*. Obtenido de WIKIPEDIA:
<https://es.wikipedia.org/wiki/Hipervisor>