



TRABAJO DE GRADO
Opción Seminario-Diplomado.

Seminario AWS: De Noob a Junior

Corporación Universitaria Remington.
Nombre de la facultad: Facultad de Ingeniería.
Nombre del programa académico: Ingeniería de Sistemas.

Yeferson Yezid Camacho Camacho.
Juan Pablo Berrio Lopez
Opción de Trabajo de grado Seminario-Diplomado.
2025.

Tabla de Contenidos

Resumen.....	3
Palabras Claves.....	5
Marco conceptual y contextual	5
❖ Servicios web de Amazon (AWS)	5
❖ Computación en la nube.....	5
❖ Instancias.....	5
❖ Balanceador de carga	5
❖ Kubernetes (Contenedores).....	6
❖ Apache	6
❖ Proxy Reverso.....	6
❖ Nginx.....	6
Desarrollo e implementación del aprendizaje.....	7
1. Creación de cuenta y configuración inicial	7
2. Creación VPC.....	8
3. Creación grupo de seguridad.....	9
4. Creación de Instancias.....	9
4.1. Conexión a la instancia.....	10
5. Instalación Apache y Nginx.....	12
6. Creación de contenido web.....	12
7. Creación de AMI.....	13
8. Balanceador.....	14
9. Escalado Automático (ASG)	15
10. Contenedores.....	16
10.1. Instalación Docker.....	16
10.2. Instalación contenedores.....	16
11. Proxy reverso.....	18
11.1. Balanceo por DNS.....	18
Figuras y tablas	222
Conclusiones.....	23
Referencias.....	24

Resumen

Amazon Web Services (AWS) es la solución en la nube de Amazon que brinda a empresas de todos los tamaños la posibilidad de alojar servidores, servicios e información sin necesidad de infraestructura física. A través de una cuenta registrada, las organizaciones pueden administrar sus recursos de manera remota y eficiente, adaptando sus operaciones a sus necesidades y capacidades económicas. Este enfoque permite una gestión flexible y optimizada de la tecnología, impulsando la productividad sin la complejidad de mantener hardware propio.

Una de las principales ventajas de AWS es su escalabilidad, lo que significa que las empresas pueden ajustar su capacidad según la demanda. Por ejemplo, en períodos de mayor actividad comercial, es posible incrementar el uso de almacenamiento, recursos y disponibilidad sin interrupciones, configurándolo desde la cuenta de Amazon. Esta funcionalidad permite un crecimiento ágil y sostenible, asegurando un rendimiento óptimo sin inversiones innecesarias en infraestructura física.

En el desarrollo de este trabajo, utilizando una cuenta gratuita de Amazon Web Services (AWS), se llevan a cabo una serie de ejemplos prácticos que permiten explorar la creación y configuración de servicios fundamentales en la nube. Se analizan aspectos clave como la disponibilidad de los servicios, el balanceo de carga entre distintos sitios web y la optimización de recursos para garantizar un rendimiento eficiente. A través de estos ejercicios, se destaca la importancia de contar con infraestructura en la nube, evitando los costos y limitaciones de mantener servidores físicos, lo que facilita la escalabilidad y la administración remota de los sistemas.

Además, se demuestra cómo una persona con conocimientos básicos en tecnologías de la información (TI) o computación en la nube puede realizar estas configuraciones de manera intuitiva y accesible. Gracias a la interfaz amigable de AWS y la documentación detallada disponible, los usuarios pueden aprender a implementar soluciones adaptadas a sus necesidades sin requerir experiencia avanzada en desarrollo o administración de servidores. Este enfoque democratiza el acceso a tecnologías cloud, permitiendo que individuos y empresas aprovechen sus beneficios sin grandes inversiones iniciales.

Palabras clave

Computación en la nube, AWS, instancia, balanceador de carga, proxy reverso.

Marco conceptual y contextual

A lo largo de este trabajo, se presentan ejercicios prácticos diseñados para que los usuarios puedan entender y aplicar configuraciones clave dentro del entorno de AWS de manera sencilla y efectiva. Se exploran aspectos fundamentales como la creación y gestión de instancias, la configuración de grupos de seguridad y la implementación de proxies, herramientas esenciales para lograr un balanceo adecuado de servicios. A través de estas prácticas, se simulan escenarios reales, como la sobrecarga de servidores, permitiendo a los participantes visualizar cómo optimizar su infraestructura en la nube y garantizar estabilidad, rendimiento y escalabilidad en sus operaciones sin la necesidad de hardware físico.

Para implementar AWS de manera óptima, es crucial comprender ciertos conceptos técnicos, tanto específicos del proveedor como generales de la computación en la nube, por ello a continuación se detallan:

❖ Servicios web de Amazon (AWS)

Amazon Web Services (AWS) es una plataforma de servicios en la nube de Amazon que ofrece más de 200 servicios completos y bajo demanda, como computación, almacenamiento, bases de datos, análisis, inteligencia artificial, redes y desarrollo de aplicaciones. AWS permite a organizaciones de todos los tamaños escalar su infraestructura de manera segura, flexible y rentable.

❖ Computación en la nube

La computación en la nube es un modelo de entrega de servicios tecnológicos (como almacenamiento, bases de datos, redes y software) a través de internet, bajo demanda y con pago por uso. Este enfoque elimina la necesidad de poseer y mantener infraestructura física propia, brindando escalabilidad, flexibilidad y eficiencia operativa.

❖ Instancias

Una instancia es un servidor virtual que se ejecuta en la nube. En AWS, estas instancias son creadas dentro del servicio Amazon EC2 (Elastic Compute Cloud) y permiten ejecutar aplicaciones en entornos escalables.

❖ Balanceador de carga

Un balanceador de carga distribuye automáticamente el tráfico de entrada entre múltiples instancias para garantizar alta disponibilidad y tolerancia a fallos. AWS ofrece Elastic Load Balancing para distribuir tráfico en instancias EC2, contenedores, direcciones IP, etc.

❖ **Kubernetes (Contenedores)**

Kubernetes es un sistema de orquestación de contenedores que automatiza el despliegue, la gestión y la escalabilidad de aplicaciones en contenedores. En AWS, Amazon EKS (Elastic Kubernetes Service) facilita el uso de Kubernetes en la nube.

❖ **Apache**

Apache HTTP Server es un servidor web de código abierto que permite alojar y servir sitios web mediante los protocolos HTTP y HTTPS. Es altamente configurable y ampliamente utilizado.

❖ **Proxy Reverso**

Un proxy reverso es un servidor que se sitúa frente a uno o más servidores backend y redirige las solicitudes de los clientes a esos servidores. Mejora el rendimiento, seguridad y equilibrio de carga.

❖ **Nginx**

Nginx es un servidor web que también funciona como proxy inverso, balanceador de carga y acelerador de contenido HTTP. Es conocido por su alto rendimiento, escalabilidad y bajo consumo de recursos.

❖ **DNS (Sistema de Nombres de Dominio)**

El DNS traduce los nombres de dominio legibles por humanos (como www.ejemplo.com) en direcciones IP entendibles por las máquinas. Amazon Route 53 es el servicio DNS escalable y confiable de AWS.

❖ **VPC (Nube privada virtual)**

Una VPC es una red virtual privada dentro de AWS que permite lanzar recursos como instancias EC2 con control total sobre el entorno de red, incluidas las subredes, tablas de rutas y gateways.

❖ **EC2 (Nube de computación elástica)**

Amazon EC2 es un servicio que permite lanzar y gestionar instancias de cómputo en la nube. Ofrece flexibilidad en la elección de sistemas operativos, tipos de instancia, almacenamiento y escalabilidad automática.

Desarrollo e implementación del aprendizaje

Para aventurarse en el maravilloso mundo de computación en la nube primero se debe tener en cuenta que se abre un sinfín de posibilidades para individuos y empresas que buscan optimizar sus recursos sin depender de infraestructura física. La capacidad de gestionar servidores, almacenamiento y servicios completamente a través de internet ofrece flexibilidad, escalabilidad y eficiencia, lo que ha llevado a la adopción masiva de esta tecnología. Sin embargo, antes de sumergirse en este universo digital, es esencial comprender que AWS no es el único proveedor disponible en el mercado. Existen diversas compañías que ofrecen soluciones en la nube, cada una con características únicas que pueden ajustarse mejor a diferentes tipos de necesidades y proyectos.

Más allá de determinar cuál servicio es superior, lo realmente importante es identificar cuál se adapta mejor a los objetivos específicos de cada usuario o empresa. La elección dependerá del tipo de implementación que se requiera, la capacidad de procesamiento, almacenamiento y seguridad que se necesite, así como el presupuesto disponible. Algunos proveedores pueden destacar por su facilidad de uso, mientras que otros por su integración con herramientas avanzadas o por su enfoque en modelos híbridos (comparación servicios por proveedor, ver tabla 1). En este sentido, la computación en la nube se convierte en una solución versátil que, bien utilizada, puede transformar la manera en que operamos en el entorno digital.

1. Creación de cuenta y configuración inicial

AWS ofrece una capa gratuita con recursos limitados hasta por un año, en el cual permite a pequeñas empresas, estudiantes o curiosos que quieran aprender o configurar su negocio en los servicios de Amazon de una manera que no genere algún coste adicional (en los recursos especificados, como gratuidad).



Regístrese para obtener una cuenta gratuita de AWS

Crear una cuenta de AWS no tiene costo y le proporciona acceso inmediato al nivel gratuito de AWS



Tutoriales de 10 minutos

Explore y aprenda con tutoriales sencillos de varios casos de uso

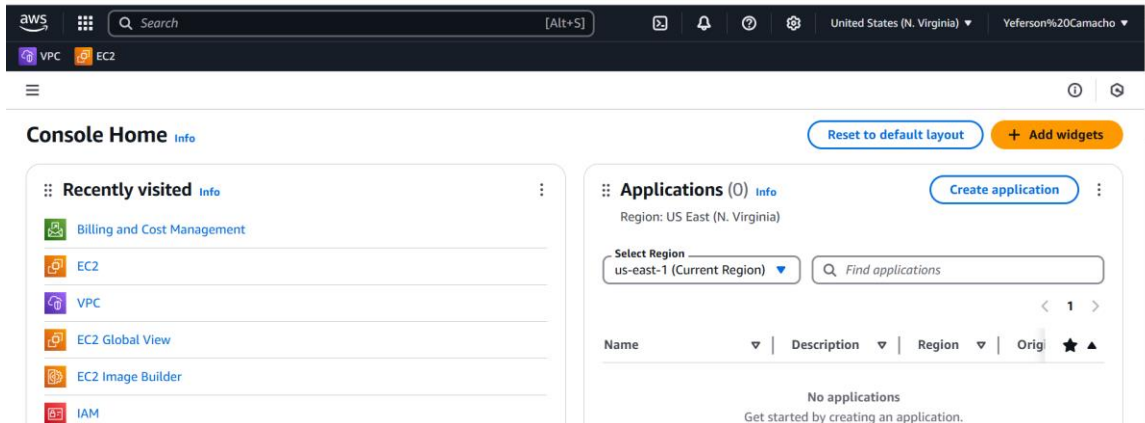


Comience a crear en la consola

Cree su solución de producción de manera rápida y sencilla cuando esté listo

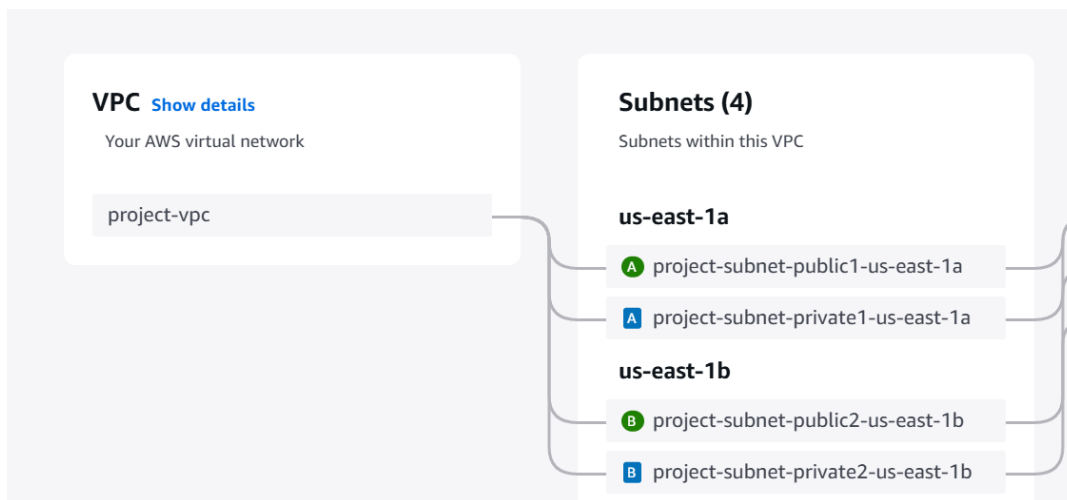
El registro se puede realizar en el portal de signin.aws.amazon.com, importante contar con una tarjeta débito o crédito que avale el registro y los posibles cargos que se puedan generar, se debe leer y conocer los servicios que no generan costo y su capacidad a nivel de performance y almacenamiento, con el fin de evitar cobros inesperados.

Posterior a realizar el registro, el entorno de AWS es grafico y amigable con el usuario, lo que permite ubicar a través de su menú principal los servicios a usar y configurar. En este caso nos centraremos en VPC y EC2



2. Creación VPC

En la creación de la nube privada virtual, es importante definir cuál será su uso, para este caso se requiere poder acceder desde cualquier lugar a través de internet; por lo cual es necesario se habilite la resolución de DNS y redes públicas:



vpc-0e8059926dfe0546a / VPC-SAWS-vpc Actions ▾

Details Info

<p>VPC ID vpc-0e8059926dfe0546a</p> <p>DNS resolution Enabled</p> <p>Main network ACL acl-0f8c98904951f3bd8</p> <p>IPv6 CIDR (Network border group) -</p>	<p>State Available</p> <p>Tenancy default</p> <p>Default VPC No</p> <p>Network Address Usage metrics Enabled</p>	<p>Block Public Access Off</p> <p>DHCP option set dopt-0d2f08e661930f44c</p> <p>IPv4 CIDR 10.0.0.0/16</p> <p>Route 53 Resolver DNS Firewall rule groups -</p>	<p>DNS hostnames Enabled</p> <p>Main route table rtb-0bcc4a5dd078d8930</p> <p>IPv6 pool -</p> <p>Owner ID 759975071259</p>
---	--	---	--

3. Creación grupo de seguridad


Para realizar cualquier tipo de conexión hacia la nube creada o las instancias configuradas, siempre se hace a través de protocolos y puertos; por lo cual es necesario crear grupos de seguridad que permitan regular y controlar dichas conexiones. En la opción de creación de grupos de seguridad, en sus reglas de entrada, se debe configurar el puerto SSH (22) para poder realizar conexiones de escritorio remoto al servidor o instancia creados y para configurar sitios web se debe definir una conexión de tipo http o https (conexión segura) y el puerto por el cual se van a realizar las peticiones:


Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Destination <small>Info</small>	Description - optional <small>Info</small>	
SSH ▾	TCP	22	Cu... ▾ 0.0.0.0/0 ✕		Delete
Custom TCP ▾	TCP	80	An... ▾ 0.0.0.0/0 ✕ 0.0.0.0/0 ✕		Delete
Add rule					


4. Creación de Instancias


Crear una instancia es como configurar o lanzar una maquina virtual con un sistema operativo a elección y recursos de acuerdo con las necesidades en un espacio “virtual”, se debe tener claro el tipo de alcance y uso que se le va dar a la instancia, con ello en mente se puede definir el sistema y capacidades que mejor se adapten a los requerido. Para este ejercicio practico usaremos recursos mínimos y el SO será Linux AWS


Recents | My AMIs | **Quick Start**


Amazon Linux



macOS


Ubuntu


Windows


Red Hat


SUSE Linux



[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI Free tier eligible ▼

ami-00a929b66ed6e0de6 (64-bit (x86), uefi-preferred) / ami-05f417c208be02d4d (64-bit (Arm), uefi)

Virtualization: hvm ENA enabled: true Root device type: ebs

▼ **Configure storage** [Info](#) Advanced

1x GiB Root volume, 3000 IOPS, Not encrypted

i Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage ✕

Para que la instancia sea accesible desde internet, se debe habilitar la ip publica, estar en una subnet publica y que su grupo de seguridad permita conexiones de entrada por los puertos que está usando la instancia; En este caso se uso el grupo de seguridad creado previamente.

Auto-assign public IP [Info](#)

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) [Info](#)


A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group
 Select existing security group

Common security groups [Info](#)

Select security groups ▼

launch-wizard-1 sg-08adf8106a9ec895a ✕
VPC: vpc-004a969175b0c00d1

 [Compare security group rules](#)

Security groups that you add or remove here will be added to or removed from all your network interfaces.

4.1. Conexión a la instancia

Hay dos formas de conectarse a una instancia, una a través de escritorio remoto (Windows) o usando un programa de conexión remota (Linux). Para conectarse a la

Para tener privilegio de usuario root o administrador, una vez se establezca la conexión se debe ejecutar el comando *sudo su*

5. Instalación Apache y Nginx

Una vez se tenga acceso a la instancia a través de la consola de administración, las posibilidades son variadas, en este caso se realizó la instalación de servidores web Apache y Nginx, los dos a través de líneas de comandos

- Apache: *sudo systemctl install httpd*
- Nginx: *sudo amazon-linux-extras install nginx1*

```

No packages marked for update
[root@ip-172-31-20-33 ec2-user]# sudo amazon-linux-extras install nginx1 -y
Installing nginx
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-kernel-5.10 amzn2extra-nginx1
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 3.6 kB 00:00:00
amzn2extra-docker | 2.9 kB 00:00:00
amzn2extra-kernel-5.10 | 3.0 kB 00:00:00
amzn2extra-nginx1 | 2.9 kB 00:00:00
(1/9): amzn2-core/2/x86_64/group_gz | 2.7 kB 00:00:00
(2/9): amzn2-core/2/x86_64/updateinfo | 1.0 MB 00:00:00
(3/9): amzn2extra-docker/2/x86_64/updateinfo | 22 kB 00:00:00
(4/9): amzn2extra-nginx1/2/x86_64/updateinfo | 3.9 kB 00:00:00
(5/9): amzn2extra-docker/2/x86_64/primary_db | 120 kB 00:00:00
(6/9): amzn2extra-kernel-5.10/2/x86_64/updateinfo | 119 kB 00:00:00
(7/9): amzn2extra-nginx1/2/x86_64/primary_db | 61 kB 00:00:00
(8/9): amzn2extra-kernel-5.10/2/x86_64/primary_db | 35 MB 00:00:00
(9/9): amzn2-core/2/x86_64/primary_db | 75 MB 00:00:01
Resolving Dependencies
--> Running transaction check
--> Package nginx.x86_64 1:1.26.3-1.amzn2.0.1 will be installed
--> Processing Dependency: nginx-core = 1:1.26.3-1.amzn2.0.1 for package: 1:nginx-1.26.3-1.amzn2.0.1.x86_64

```

6. Creación de contenido web

Se puede cargar o crear contenido web en el servidor apache a través de su archivo index.html o cargando plantillas creadas previamente u obtenidas de sitios web especializados en la generación de dichas plantillas. Para el ejercicio práctico, se descargó una plantilla del sitio web html5up (se debe copiar la ruta de descarga para usarla en línea de comandos y poder descargar su contenido en la instancia)

wget https://html5up.net/hyperspace/download

```

2 html]# rm index.html
file 'index.html'? y
2 html]# ls
2 html]# wget https://html5up.net/eventually/download
45-- https://html5up.net/eventually/download
et (html5up.net)... 172.67.195.190, 104.21.76.136, 2606:4700:3036::ac43:c3be, ...
up.net (html5up.net)|172.67.195.190|:443... connected.
awaiting response... 200 OK
[application/x-zip]
d'

[ <=> ] 1.72M 8.84MB/s in 0.2s

(8.84 MB/s) - 'download' saved [1799395]

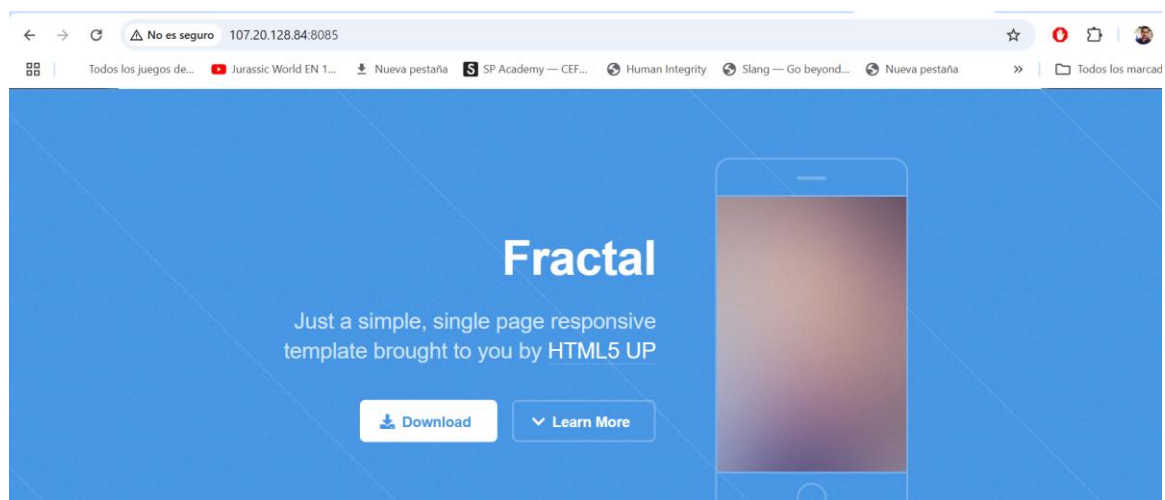
```

Una vez descargado el sitio, se debe descomprimir para poder visualizar el contenido html, *unzip download*

```
2025-03-29 15:42:48 (20.6 MB/s) - 'download' saved [1472285]
```

```
[root@ip-172-31-20-33 app2]# unzip download
Archive:  download
```

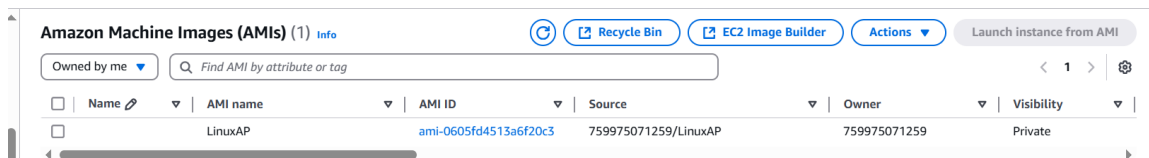
El contenido html es accesible a través de la ip publica de la instancia



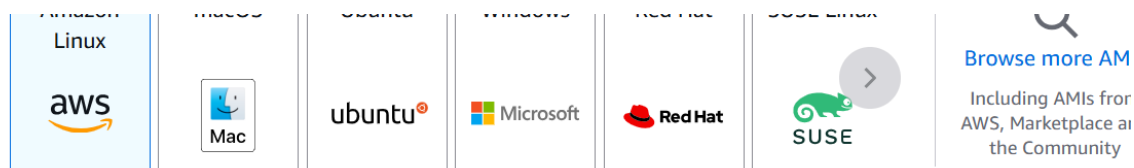
7. Creación de AMI

Otra de las funcionalidades útiles de amazon es la creación de imágenes de máquinas, esto permite ahorrarse tiempo a la hora de configurar una instancia adicional, ya no se realizará desde cero, si no que se toma una instancia creada previamente con su capacidad, configuración y aplicativos instalados.

En este caso se creo una imagen de la instancia previamente configurada con Amazon Linux y el servidor web apache instalados



En el flujo de creación de una nueva instancia, la imagen (AMI) estará disponible para seleccionarse como una versión más de sistema/ imagen, permitiendo crear la instancia con configuraciones y aplicativos precargados en la imagen



Amazon Machine Image (AMI)

Amazon Linux 2023 AMI

ami-08b5b3a93ed654d19 (64-bit (x86), uefi-preferred) / ami-0eae2a0fc13b15fce (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

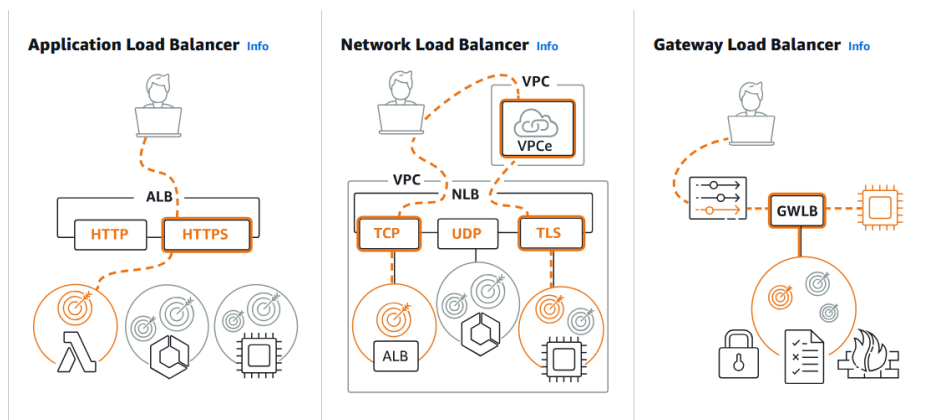
Free tier eligible

Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

8. Balanceador

Para configurar un balanceador de carga es necesario definir qué tipo de balanceo se requiere y cuál será su alcance.



Para el ejercicio se configuro un balanceo por zonas, el cual permite cambiar de zona de acuerdo con la disponibilidad de las mismas (network load balancer)

Select Availability Zones and subnets your Auto Scaling group can use in the chosen VPC

Select Availability Zones and subnets ▼ ↻

us-east-1b | subnet-07c8918cccb67f752 (VPC-SAWS-subnet-public2-us-east-1b) ✕

10.0.16.0/20

[Create a subnet](#) ↗

Availability Zone distribution - new

Auto Scaling automatically balances instances across Availability Zones. If launch failures occur

Balanced best effort

If launches fail in one Availability Zone, Auto Scaling will attempt to launch in another healthy Availability Zone.

Balance

If launch to attempt balance

9. Escalado Automático (ASG)

El escalado automático permite aumentar los recursos o las instancias de manera controlada cuando se cumplan ciertas condiciones que se definen de manera personalizada, tales como disponibilidad de las zonas, consumo de recursos.

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability ...
Balanceo	Temp1 Version Default	1	-	1	1	2	us-east-1b

Para configurar un escalado automático, es necesario configurar previamente una plantilla o template, el cual contendrá las condiciones del auto escalado; Se usa una versión AMI del sistema y un grupo de seguridad existente

EC2 > Launch templates > Modify template (Create new version)

linuxami Template value ▼ ↻ Create new key pair

Network settings Info

Subnet Info

Don't include in launch template ▼ ↻ Create new subnet

When you specify a subnet, a network interface is automatically added to your template.

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Select existing security group Create security group

Common security groups Info

Select security groups ▼ ↻ Compare security group rules

SG-Linux3 sg-03eead7601c339ba ✕

VPC: vpc-0e8059926dfe0546a

Security groups that you add or remove here will be added to or removed from all your network interfaces.

▶ **Advanced network configuration**

10. Contenedores

10.1. Instalación Docker

Docker al ser una aplicación de código abierto, nos permite crear y configurar contenedores de manera practica y eficiente. *yum install docker*

```

[ec2-user@ip-172-31-20-33 ~]$ sudo su
[root@ip-172-31-20-33 ec2-user]# yum install docker
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
----> Package docker.x86_64 0:25.0.8-1.amzn2.0.1 will be installed
--> Processing Dependency: containerd >= 1.3.2 for package: docker-25.0.8-1.amzn2.0.1.x86_64
--> Processing Dependency: libcgrou >= 0.40.rc1-5.15 for package: docker-25.0.8-1.amzn2.0.1.x86_64
--> Processing Dependency: runc >= 1.0.0 for package: docker-25.0.8-1.amzn2.0.1.x86_64
--> Processing Dependency: pigz for package: docker-25.0.8-1.amzn2.0.1.x86_64
--> Running transaction check
----> Package containerd.x86_64 0:1.7.25-1.amzn2.0.1 will be installed
----> Package libcgrou.x86_64 0:0.41-21.amzn2 will be installed
----> Package pigz.x86_64 0:2.3.4-1.amzn2.0.1 will be installed
----> Package runc.x86_64 0:1.2.4-1.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

```

yum install Docker.io

```

5/5

Installed:
  docker.x86_64 0:25.0.8-1.amzn2.0.1

Dependency Installed:
  containerd.x86_64 0:1.7.25-1.amzn2.0.1      libcgrou.x86_64 0:0.41-21.amzn2
  pigz.x86_64 0:2.3.4-1.amzn2.0.1
  runc.x86_64 0:1.2.4-1.amzn2

Complete!
[root@ip-172-31-20-33 ec2-user]# yum install docker.io
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Existing lock /var/run/yum.pid: another copy is running as pid 3517.
Another app is currently holding the yum lock; waiting for it to exit...
The other application is: yum
Memory ..: 222 M RSS (443 MB VSZ)
Started:  Sat Mar 29 15:03:37 2025 - 00:06 ago
State ..:  Running, pid: 3517

```

10.2. Instalación contenedores

Los contenedores permiten cargar aplicaciones o servicios de manera más rápida y eficiente que las instancias, esta ventaja los hace muy prácticos y permite tener múltiples contenedores con información diferente dentro de un mismo espacio.

```
docker run -dit --name appdkr1 -p 8085:80 -v /home/ec2-user/appdk1/:/usr/local/apache2/htdocs/httpd
```

Appdkr1: nombre del contenedor

/home/ec2-user/appdk1/: ruta del contendor

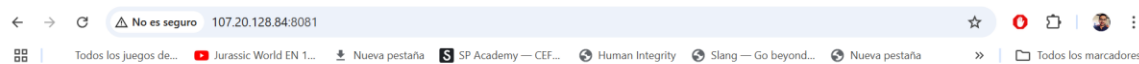
/usr/local/apache2/htdocs/httpd: ruta del servidor web

```
download.2
[root@ip-172-31-20-33 appdk1]# docker run -dit --name appdkr1 -p 8085:80 -v
/home/ec2-user/appdk1:/usr/local/apache2/htdocs/httpd
```

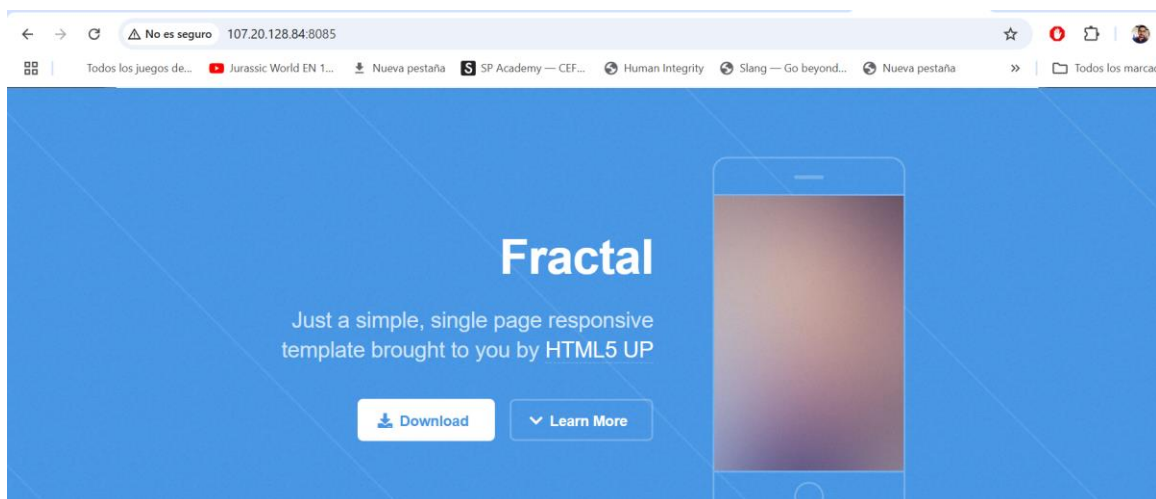
Se usa la misma sentencia o comando para crear los demás contenedores, reemplazando el nombre del contenedor y el puerto que va usar el mismo.

```
[root@ip-172-31-20-33 ec2-user]# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS                               NAMES
469e06c1db87   httpd    "httpd-foreground"     2 hours ago Up 2 hours 0.0.0.0:8083->80/tcp, :::8083->80/tcp appdk3
125da8a47ec9   httpd    "httpd-foreground"     4 hours ago Up 2 hours 0.0.0.0:8086->80/tcp, :::8086->80/tcp appdkr2
616093e5a990   httpd    "httpd-foreground"     4 hours ago Up 2 hours 0.0.0.0:8085->80/tcp, :::8085->80/tcp appdkr1
f0aa7a25e8e7   httpd    "httpd-foreground"     5 hours ago Up 2 hours 0.0.0.0:8080->80/tcp, :::8080->80/tcp apache1
```

Para acceder a la información contenida o cargada en cada uno de los contenedores se hace a través de la ip publica de la instancia y el puerto del contendor



It works!



11. Proxy reverso

Para este ejercicio se usó el Nginx como proxy reverso, al configurar el archivo `nginx.conf`, se puede balancear el tráfico entre contenedores a partir del puerto usado por cada uno.

```
events {}

http {
    upstream aplicaciones {
        server localhost:8080;
        server localhost:8081;
        server localhost:8085;
    }

    server {
        listen 80;
        server_name nginx;
        location / {
            proxy_pass http://aplicaciones;
        }
    }
}
```

De esta manera al acceder a la ip pública de la instancia, se carga de manera aleatoria la información que se encuentra en los contenedores, es una opción práctica cuando se tienen servicios balanceados o de alta disponibilidad (si uno falla, el otro contenedor puede mostrar la misma información).

11.1. Balanceo por DNS

Este tipo de balanceo permite que una sola ip pública cargue varios sitios a partir de varios DNS, o que varios DNS carguen un mismo sitio.

Para el ejercicio, al no contar con dominios propios, se configuraron dominios reales en el archivo `host` local para simular el apuntamiento a la ip pública de la instancia creada previamente

C:\Windows\System32\drivers\etc\hosts - Notepad++

Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Macro Ejecutar Compleme

Pestañas ?

change.log hosts

```

1 107.20.128.84 www.exito.com
2 107.20.128.84 www.dl.com.co
3 107.20.128.84 www.carulla.com
4
5
6
7
8
9
10

```



Fractal

Just a simple, single page responsive template brought to you by [HTML5 UP](#)

- [Download](#)
- [Learn More](#)



En el archivo nginx.conf, se cambia la configuración server, para que a través del nginx de acuerdo con el server_name (url) redireccie a un contendedor diferente, usando el puerto del contenedor en el apuntamiento del proxy_pass

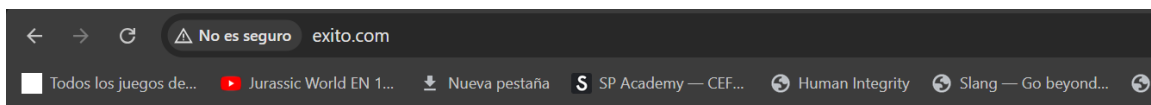
```
server {
    listen 80;
    server_name www.exito.com;

    location / {
        proxy_pass http://localhost:8083;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

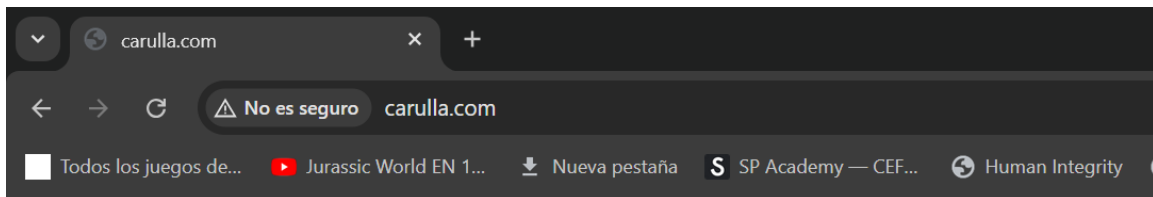
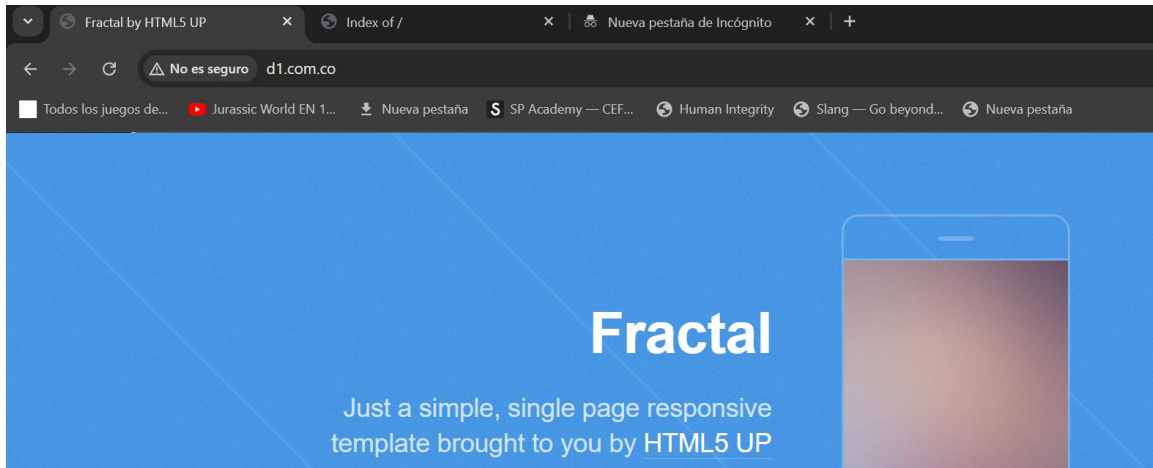
```
server {
    listen 80;
    server_name www.d1.com.co;

    location / {
        proxy_pass http://localhost:8085;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

Al tener dicha configuración en el nginx, si se consulta a través del navegador cada una de las 3 páginas, carga la información que se encuentra en el index.html del contenedor al cual corresponde el puerto



Index of /



It works!

De esta manera, el nginx actúa como balanceador y enruta el tráfico de acuerdo con el Dominio que realice la consulta.

Figuras y tablas

Tabla 1. Comparación de servicios de computación en la nube por proveedor

Servicio	AWS	AZURE	GCP
Redes	VPC	Azure Virtual Network	Google Virtual Private Cloud
Almacenamiento	S3	Azure Blob Storage	Google Cloud Storage
BD	RDS	Azure SQL Database	Cloud SQL
Virtualización	EC2	Azure Virtual Machines	Google Compute Engine
Contenedores	EKS	Azure Kubernetes Service	Google Kubernetes Engine (GKE)
Seguridad - Acceso	IAM	Azure Active Directory	Cloud Identity

Conclusiones

- La computación en la nube avanza a paso agigantados y entender su funcionamiento nos permite adaptarnos al mercado y generar soluciones prácticas, escalables y confiables.
- Afianzar conocimientos en AWS aumenta las posibilidades de gestionar arquitecturas basadas en la nube.
- Pasar de novato (noob) a junior no es tan difícil si se aplican los conocimientos adquiridos en el seminario
- La documentación y los ejercicios prácticos son la base fundamental para seguir avanzando en arquitectura en la nube.

Referencias

- Servicios web de Amazon. (2023). *¿Qué es AWS?* <https://aws.amazon.com/es/what-is-aws/>
- ¿Qué es la computación en la nube? (2024) <https://cloud.google.com/learn/what-is-cloud-computing?hl=es-419>
- Servicios web de Amazon. (2023). https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/Instances.html
- Amazon Web Services. (2023). https://docs.aws.amazon.com/es_es/elasticloadbalancing/
- Kubernetes.io. (2023). <https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>
- Servicios web de Amazon. (2023). Amazon EKS. <https://aws.amazon.com/eks/>
- Fundación de Software Apache. (2023). <https://httpd.apache.org/>
- Nginx, Inc. (2023). <https://www.nginx.com/resources/glossary/reverse-proxy/>
- Nginx, Inc. (2023). Documentación NGINX. <https://docs.nginx.com/>
- Servicios web de Amazon. (2023). <https://aws.amazon.com/route53/IETF> .
- RFC 1035 - Nombres de dominio - Implementación y especificación. (1987). <https://www.rfc-editor.org/rfc/rfc1035>
- Servicios web de Amazon. (2023). <https://docs.aws.amazon.com/vpc/>
- Documentación de Amazon Elastic Compute Cloud. (2025). https://docs.aws.amazon.com/es_es/ec2/#amazon-ec2