

TRABAJO DE GRADO
Opción Seminario-Diplomado.

Trabajo final

Servicios de AWS

Corporación Universitaria Remington.
Nombre de la facultad: De ingenierías
Nombre del programa académico: Ingeniería de sistemas, Especialización en seguridad informática, Tecnología en desarrollo de software

Nombres de los estudiantes autores del trabajo de grado.
Javier Antonio Tabares
Luis Fernando Palacio
Jeimy Catherine Giraldo

Nombre del Tutor del trabajo de grado (Juan Pablo Berrio).
Opción de Trabajo de grado Seminario-Diplomado.
Año 2024.

Tabla de Contenidos

Resumen.....	3
Palabras claves.....	4
Entrega 1.....	5
Entrega 2	7
Entrega 3.....	14
Entrega final.....	16
Video final.....	23
Conclusiones.....	24
Referencias.....	25

Resumen

Este trabajo se realizó con el fin de aprender acerca de AWS se va a colocar en práctica cada uno de los ítems, como crear instancias en ec2 como realizar diferentes tipos en vpc Y poderla ejecutar en la consola de WSL, Crea colecciones de instancias EC2, designadas grupos de Auto Scaling. Se va a validar y especificar el número mínimo de instancias en cada grupo de escalado automático EC2 Auto Scaling responderá que el grupo nunca tenga menos de esas instancias. Puede especificar el número máximo de instancias en cada grupo de escalado automático y luego en EC2 y Auto Scaling garantizará que el grupo nunca tenga más de esas instancias. Si evidencia de manera detalla el volumen deseado, en EC2 Auto Scaling avalará que el grupo tenga ese número de instancias. C2 Auto Scaling supervisa automáticamente el estado y la disponibilidad de las instancias mediante demostraciones de estado de EC2 y reemplaza las instancias canceladas o deterioradas para mantener la capacidad deseada. Se va a probar el acceso a la aplicación a través de la URL del Load Balance.

Palabras clave

En este trabajo final se realizo con las siguientes herramientas como AWS, WSL, EC2, auto Scaling Y EC3 entre otras

Entrega 1

Los orígenes

Desde la década de los 60 y al principio de los 70 ARPANET creó una red de computadores que se conectaban entre dos PC desde ese entonces se originó el primer servicio en la nube, años más tarde principio de la década de los 80 llegó la virtualización para mejorar la tecnología que esto permite tener varios sistemas operativos en un mismo sistema que facilitó a los usuarios compartir

En el 2006 Jeff Bezos fue uno de los creadores de AWS, Marc Benioff dueño de SALESFORCE que es una empresa de software de nube para ofrecer a los clientes en relaciones de ventas y marketing

Cloud, M. (2024, 23 febrero). *Origen e historia de la computación en la nube.*

Mainframe Cloud. <https://mainframecloud.co/origen>

Características	AWS	Azure	Gcp
lanzamiento	2006	2010	2008
empresa	Amazon	Microsoft	Google
Composición con Materiales de Mejora	Extensa afinidad con compuestos lenguajes y marcos de trabajo	Excelente integración con Visual Studio, .Net, Azure Devops	Buen soporte para Kubernetes y Docker, Cloud Funciones para serverless
Seguridad y Desempeño	Amplia escala de certificaciones y materiales de seguridad, IAM, AWS Shield	Alto guía en desempeño y gobernanza, Active Directory, Security Center	Certificaciones fuertes, IAM, Google Cloud Armor
Almacenamiento para archivado	Acceso poco frecuente	Almacenamiento fresco y archivo	Línea fresca y línea fría
Almacenamiento híbrido	Puerta enlace de almacenamiento	Almacenamiento simple	N/A
Bases no racionales	es una BD clave valor y de títulos que es ideal para aplicaciones	Es una asistencia de base de datos NoSQL distribuido globalmente que acepta varios modelos de datos	servicio de base de datos de columnas amplias y crecidamente escalable.

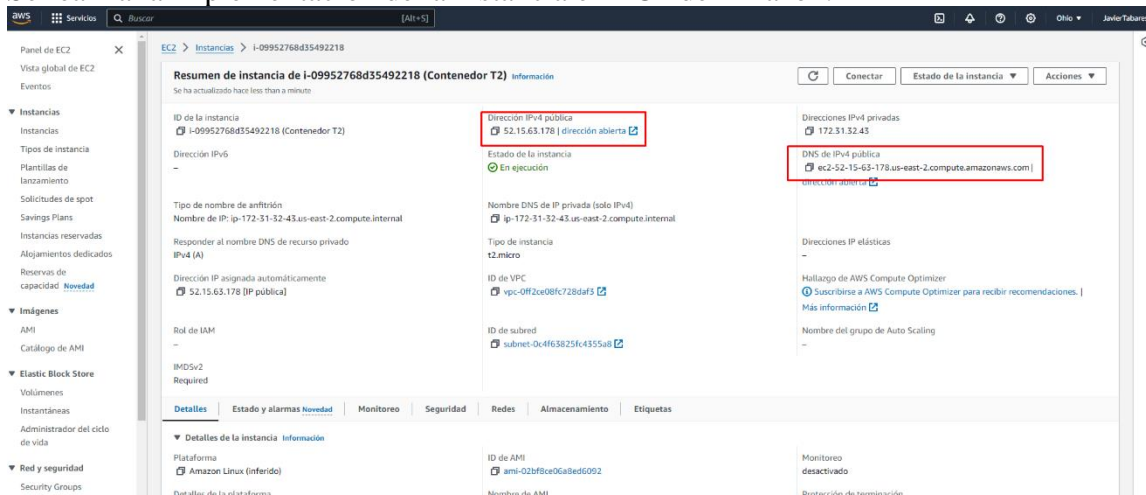
Almacenamiento en caché	Almacenamiento de archivos: Amazon Elastic File System (EFS) .	Almacenamiento de objeto, Almacenamiento de bloques	Almacenamiento de objetos y bloques
--------------------------------	--	---	-------------------------------------



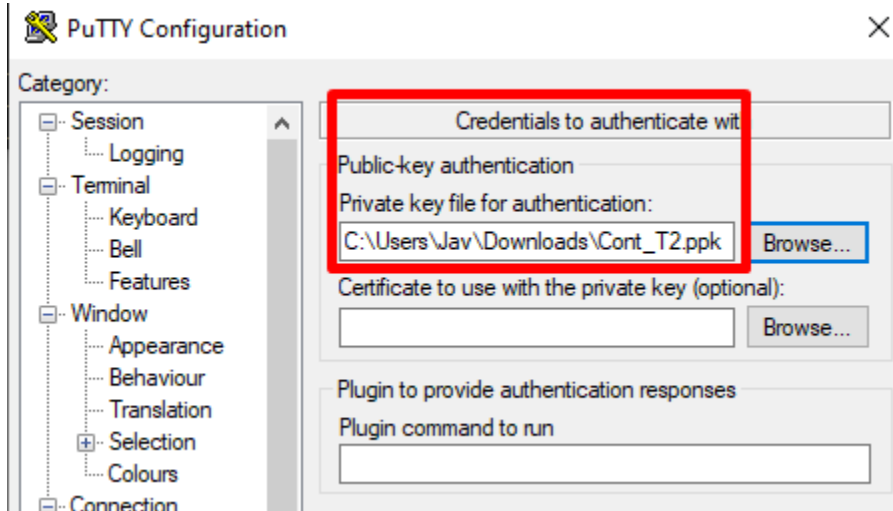
<https://prezi.com/p/kqsoydompsjt/?present=1>

entrega 2

Se realiza la implementación de la instancia en EC2 de Amazon.



Se integra archivo de conexión .pem en la consola de putty. Y se establece conexión con la EC2




```

root@ip-172-31-32-43:/home/ec2-user
[root@ip-172-31-32-43 ec2-user]#
[root@ip-172-31-32-43 ec2-user]# systemctl docker status
Unknown command verb docker.
[root@ip-172-31-32-43 ec2-user]# service docker start
Redirecting to /bin/systemctl start docker.service
[root@ip-172-31-32-43 ec2-user]# systemctl docker status
Unknown command verb docker.
[root@ip-172-31-32-43 ec2-user]# service docker status
Redirecting to /bin/systemctl status docker.service
• docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; preset: disabled)
   Active: active (running) since Mon 2024-05-27 02:45:00 UTC; 21s ago
   TriggeredBy: • docker.socket
     Docs: https://docs.docker.com
    Process: 26665 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
    Process: 26666 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
   Main PID: 26667 (dockerd)
     Tasks: 8
    Memory: 31.9M
     CPU: 335ms
   CGroup: /system.slice/docker.service
           └─26667 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=3

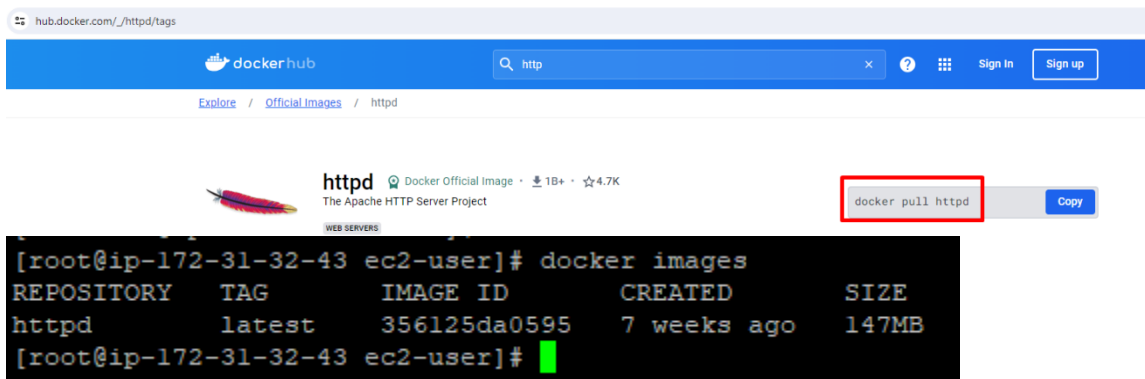
May 27 02:44:59 ip-172-31-32-43.us-east-2.compute.internal systemd[1]: Starting docker.service - Docker Application Co
May 27 02:45:00 ip-172-31-32-43.us-east-2.compute.internal dockerd[26667]: time="2024-05-27T02:45:00.013961946Z" leve
May 27 02:45:00 ip-172-31-32-43.us-east-2.compute.internal dockerd[26667]: time="2024-05-27T02:45:00.078027610Z" leve
May 27 02:45:00 ip-172-31-32-43.us-east-2.compute.internal dockerd[26667]: time="2024-05-27T02:45:00.594859133Z" leve
May 27 02:45:00 ip-172-31-32-43.us-east-2.compute.internal dockerd[26667]: time="2024-05-27T02:45:00.649672286Z" leve
May 27 02:45:00 ip-172-31-32-43.us-east-2.compute.internal dockerd[26667]: time="2024-05-27T02:45:00.650012908Z" leve
May 27 02:45:00 ip-172-31-32-43.us-east-2.compute.internal dockerd[26667]: time="2024-05-27T02:45:00.697582664Z" leve
May 27 02:45:00 ip-172-31-32-43.us-east-2.compute.internal systemd[1]: Started docker.service - Docker Application Co
lines 1-22/22 (END)

[root@ip-172-31-32-43 ec2-user]#
[root@ip-172-31-32-43 ec2-user]# docker
docker          docker-containerd-shim  docker-init          docker-runc
docker-containerd  docker-ctr          docker-proxy          dockerd
[root@ip-172-31-32-43 ec2-user]# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
[root@ip-172-31-32-43 ec2-user]#
[root@ip-172-31-32-43 ec2-user]#
[root@ip-172-31-32-43 ec2-user]# docker info
Client:
 Version:      25.0.3
 Context:      default
 Debug Mode:  false
 Plugins:
  buildx: Docker Buildx (Docker Inc.)
   Version:  v0.0.0+unknown
   Path:     /usr/libexec/docker/cli-plugins/docker-buildx

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 0
 Server Version: 25.0.3
 Storage Driver: overlay2
  Backing Filesystem: xfs
  Supports d_type: true
  Using metacopy: false
  Native Overlay Diff: true
  userxattr: false
 Logging Driver: json-file
 Cgroup Driver: systemd
 Cgroup Version: 2
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local splunk syslog
 Swarm: inactive
 Runtimes: io.containerd.runc.v2 runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: 64b8a811b07ba6288238eefc14d898ee0b5b99ba
 runc version: 4bcb38cc9cf198d52bebf2b3a90cd14e7af8c06

```

Una vez ya funcional el Docker se realiza uso del PULL de la imagen httpd. Para hacer la publicación de los sitios web a consumir.



hub.docker.com/_/httpd/tags

dockerhub

Explore / Official Images / httpd

httpd Docker Official Image · 1B+ · 4.7K
The Apache HTTP Server Project

docker pull httpd

```
[root@ip-172-31-32-43 ec2-user]# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
httpd latest 356125da0595 7 weeks ago 147MB
[root@ip-172-31-32-43 ec2-user]#
```

Con la ayuda de esta imagen se procede a la parametrización de los sitios web que se iban a exponer en cada contenedor. “App1” y “App2”

```
[root@ip-172-31-32-43 ec2-user]# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
72aba696659d httpd "httpd-foreground" 3 hours ago Up 3 hours 0.0.0.0:8888->80/tcp, :::8888->80/tcp App2
7991b22c06eb httpd "httpd-foreground" 3 hours ago Up 3 hours 0.0.0.0:8080->80/tcp, :::8080->80/tcp App1
[root@ip-172-31-32-43 ec2-user]#
```

Los servicios son expuestos sobre la instancia por los puertos 8080 y 8888.

Se procede a realizar el ajuste de permisos sobre los security Group con el fin de permitir el tráfico expuesto:

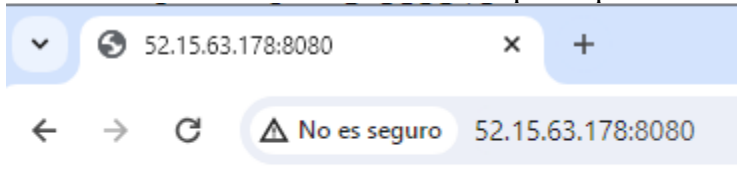
Se permite la conexión por el puerto 22 para la administración, el puerto 8080 servicio App1, el puerto 8888 para el servicio App2 y el puerto 80 para la exposición de la instancia.

Reglas de entrada (4)

Buscar

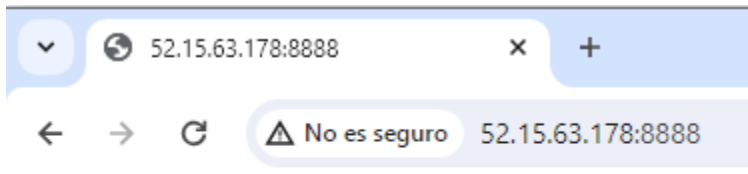
<input type="checkbox"/>	Name	ID de la regla del gr...	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Origen
<input type="checkbox"/>	-	sgr-00fa3a89438b9cd0d	IPv4	SSH	TCP	22	0.0.0.0/0
<input type="checkbox"/>	-	sgr-09518d79536777...	IPv4	TCP personalizado	TCP	8080	0.0.0.0/0
<input type="checkbox"/>	-	sgr-074318a2669309...	IPv4	HTTP	TCP	80	0.0.0.0/0
<input type="checkbox"/>	-	sgr-0766776b2fe7485...	IPv4	TCP personalizado	TCP	8888	0.0.0.0/0

Se evidencia el consumo del contenedor por el puerto 8080



Bienvenido a la APP 1

Se evidencia el consumo del contenedor por el puerto 8888



Bienvenido a la APP 2

Para realizar la publicación de los contenedores sin que el usuario deba de mapear el puerto se investigan las opciones posibles; dentro de las cuales se encontró hacer uso de un balanceador de carga de AWS conocido como Elastic Load Balancing (ELB). Sobre el cual se debe realizar varias configuraciones previas con el fin de garantizar su funcionalidad tales como:

- Configurar Listeners: Definir el puerto que va a escuchar la instancia.
- Configurar Reglas de Enrutamiento: Crear las reglas para redirigir el tráfico esperado al destino publicado. pej: 80 -> 8080

Otra opción para realizar la configuración solicitada es haciendo uso de un proxy reverso servicio que corre desde la misma instancia publicada.

Para ello se realizan los siguientes pasos:

Se realiza la instalación del servicio:

```
sudo yum update -y
sudo yum install nginx -y
```

Una vez instalado se procede con la modificación del archivo de configuración “nginx.conf” ubicado en la ruta `/etc/nginx/nginx.conf` o en `/etc/nginx/sites-available/default`

```
sudo nano /etc/nginx/nginx.conf
```

```
[ec2-user@ip-172-31-32-43 ~]$ cd /etc/nginx/
[ec2-user@ip-172-31-32-43 nginx]$ ls
conf.d      fastcgi.conf      fastcgi_params    koi-utf  mime.types      nginx.conf
default.d  fastcgi.conf.default  fastcgi_params.default  koi-win  mime.types.default  nginx.conf.bk
[ec2-user@ip-172-31-32-43 nginx]$
```

```
54 # Configuración adicional para appl.local
55 server {
56     listen 80;
57     server_name appl.local;
58     location / {
59         proxy_pass http://52.15.63.178:8080;
60         proxy_set_header Host $host;
61         proxy_set_header X-Real-IP $remote_addr;
62         proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
63         proxy_set_header X-Forwarded-Proto $scheme;
64     }
65 }
66 # Configuración adicional para app2.local
67 server {
68     listen 80;
69     server_name app2.local;
70     location / {
71         proxy_pass http://52.15.63.178:8888;
72         proxy_set_header Host $host;
73         proxy_set_header X-Real-IP $remote_addr;
74         proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
75         proxy_set_header X-Forwarded-Proto $scheme;
76     }
77 }
```

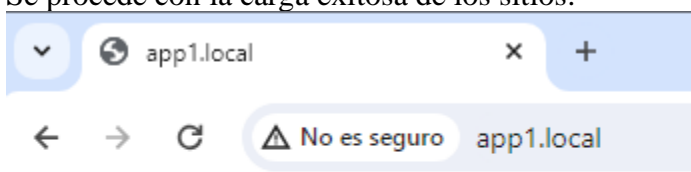
Luego de guardar los cambios se realiza reinicio del servicio para aplicar las configuraciones, se procede a probar el servicio por el nombre y sin especificar el puerto. Con el fin de resolver DNS se realiza ajuste en local Host para garantizar la resolución de nombres.

```

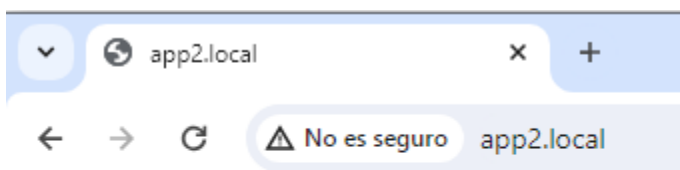
hosts: Bloc de notas
Archivo Edición Formato Ver Ayuda
# be placed in the first column followed by the correspo
# The IP address and the host name should be separated b
# space.
#
# Additionally, comments (such as these) may be inserted
# lines or following the machine name denoted by a '#' s
#
# For example:
#
#     102.54.94.97      rhino.acme.com          # source
#     38.25.63.10     x.acme.com              # x clie
# localhost name resolution is handled within DNS itself
#     127.0.0.1       localhost
#     ::1             localhost
# Added by Docker Desktop
192.168.1.12 host.docker.internal
192.168.1.12 gateway.docker.internal
# To allow the same kube context to work on the host and
127.0.0.1 kubernetes.docker.internal
# End of section
52.15.63.178 app1.local
52.15.63.178 app2.local

```

Se procede con la carga exitosa de los sitios:



Bienvenido a la APP 1



Bienvenido a la APP 2

Entrega 3

Explicar y adjuntar evidencia de cómo se hace la conexión de una instancia AWS en Linux para que pueda cargar archivos en S3 y además pueda conectarse a una base de datos RDS MySQL.

“Antes de conectarse a una instancia de base de datos, debe crear la instancia de base de datos. Para obtener más información, consulte Creación de una instancia de base de datos de Amazon RDS. Después de que Amazon RDS aprovisiona su instancia de base de datos, use cualquier ventaja o estudio cliente estándar para que su motor de base de datos se conecte a la instancia de base de datos. En la cadena de conexión, especifique la dirección DNS del punto de conexión de la instancia de base de datos como parámetro del host. Asimismo, especifique el número de puerto del punto de enlace de la instancia de base de datos como el parámetro del puerto”

RDS > Databases > mydb

mydb

Summary

DB identifier mydb	CPU 2.33%
Role Instance	Current activity 0 Connection:

Connectivity & security | Monitoring | Logs & events | Configuración

Connectivity & security

Endpoint & port

Endpoint mydb.██████████.us-east-1.rds.amazonaws.com
Port 3306

El conjunto de controladores de AWS se ha diseñado para permitir tiempos de “conmutación y conmutación por error más rápidos y autenticarse con AWS Secretos Manager, AWS Identity and Access Management (IAM) e identidad federada. Los

controladores de AWS se basan en la supervisión del estado de la instancia de base de datos y en el conocimiento de la topología de la instancia para determinar la nueva instancia principal.”

- Conexión a una instancia de base de datos de Amazon RDS - Amazon Relational Database Service. (s. f.).
https://docs.aws.amazon.com/es_es/AmazonRDS/latest/UserGuide/CHAP_CommOnTasks.Connect.html

Entrega 4

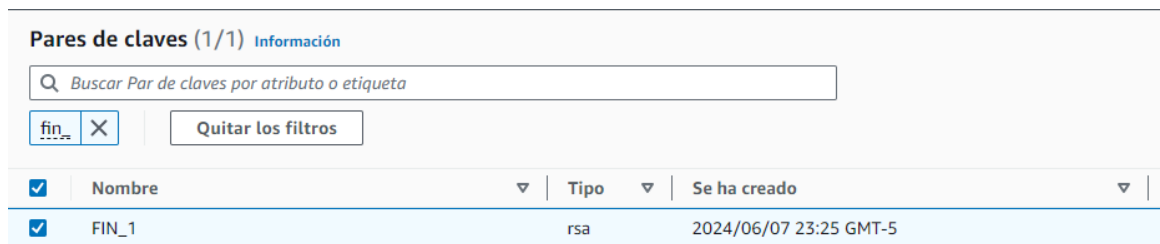
Entrega Final

1. Preparación

- Crear una cuenta en AWS Educate (si no tienen una cuenta de AWS ya).
- Configurar el entorno AWS CLI y SDK si fuera necesario.

2. Lanzamiento de Instancias EC2

- Crear un par de claves SSH para acceder a las instancias.



Pares de claves (1/1) [Información](#)

Q Buscar Par de claves por atributo o etiqueta

fin_ X Quitar los filtros

<input checked="" type="checkbox"/>	Nombre	Tipo	Se ha creado
<input checked="" type="checkbox"/>	FIN_1	rsa	2024/06/07 23:25 GMT-5

- Lanzar una instancia EC2 inicial utilizando una AMI Windows.

EC2 > Instancias > i-069be7a4387ce659f

Resumen de instancia de i-069be7a4387ce659f (FINAL) Información

Se ha actualizado hace less than a minute

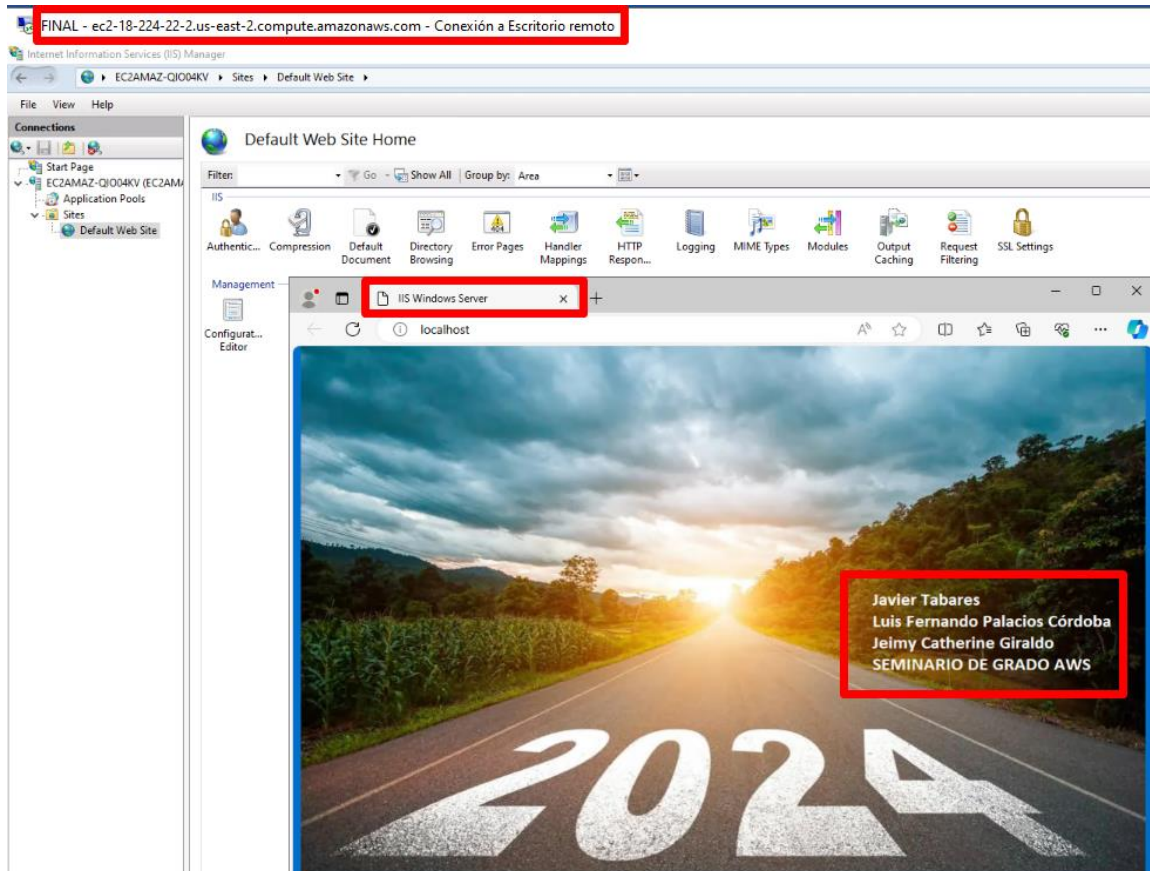
ID de la instancia i-069be7a4387ce659f (FINAL)	Dirección IPv4 pública 18.224.22.2 dirección abierta
Dirección IPv6 -	Estado de la instancia En ejecución
Tipo de nombre de anfitrión Nombre de IP: ip-172-31-26-198.us-east-2.compute.internal	Nombre DNS de IP privada (solo IPv4) ip-172-31-26-198.us-east-2.compute.internal
Responder al nombre DNS de recurso privado IPv4 (A)	Tipo de instancia t2.medium
Dirección IP asignada automáticamente 18.224.22.2 [IP pública]	ID de VPC vpc-00441918281fdc41b
Rol de IAM -	ID de subred subnet-02b4d2ce7170d08b4
IMDSv2 Required	Instance ARN arn:aws:ec2:us-east-2:730335422361:instance/i-069be7a4387ce659f

Detalles | Estado y alarmas | Monitoreo | Seguridad | Redes | Almacenamiento | Etiquetas

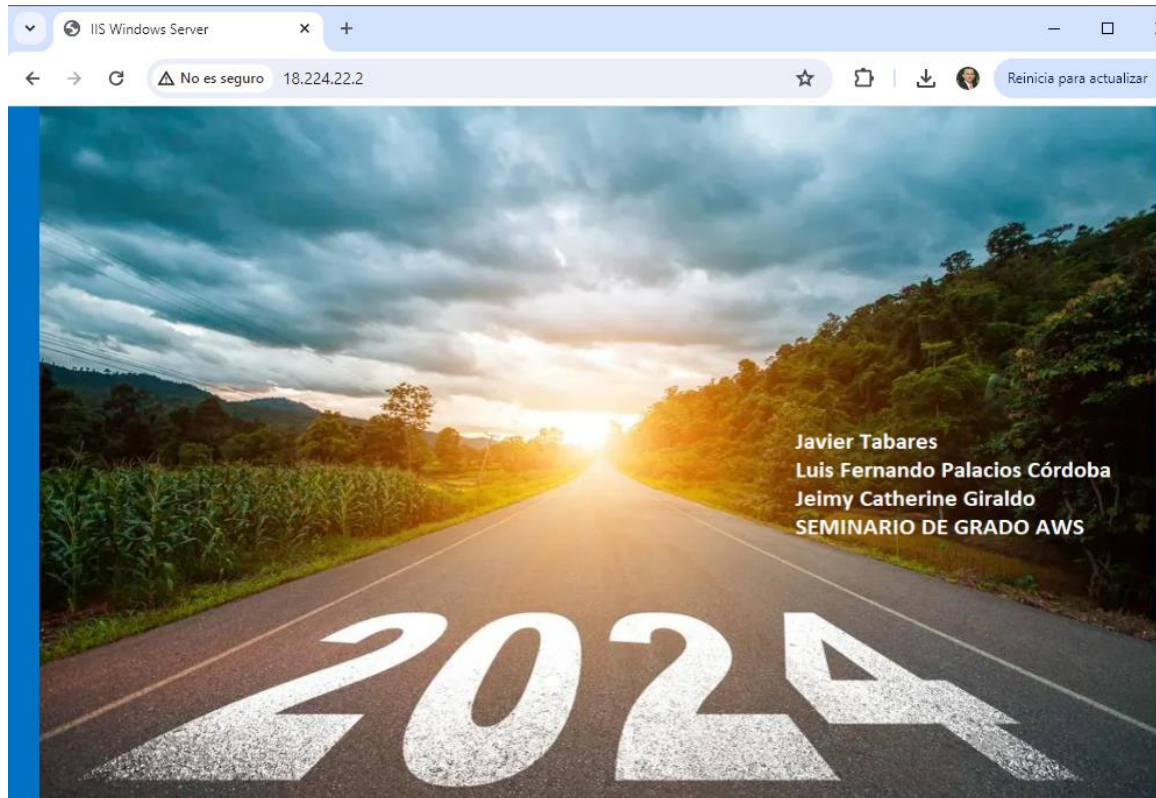
▼ **Detalles de la instancia** Información

Plataforma windows	ID de AMI ami-087fc1aada380df36
Detalles de la plataforma Windows	Nombre de AMI Windows_Server-2022-English-Full-Base-2024.05.15
Detener la protección desactivado	Hora de lanzamiento Fri Jun 07 2024 23:32:16 GMT-0500 (hora estándar de Colombia) (about 1 hour)

- Configurar la instancia con un servidor web y desplegar la aplicación web en esta instancia.
3. **Configuración de la Aplicación Web**
- Utilizar una aplicación web simple que responda a las solicitudes HTTP.
 - Probar el acceso a la aplicación web desde el navegador local.



Alcanzo desde la Publica <http://18.224.22.2/>



4. Configuración del Balanceador de Carga (ELB)

- Crear un Load Balancer (ALB o ELB) en AWS.

Balanceadores de carga (1/1) Acciones Crear balanceador

Elastic Load Balancing escala automáticamente la capacidad del equilibrador de carga en respuesta a los cambios en el tráfico entrante.

<input checked="" type="checkbox"/>	Nombre	Nombre de DNS	Estado	ID de VPC	Zonas de disponibi...	Tipo	Fecha creada
<input checked="" type="checkbox"/>	ELB-FINAL	ELB-FINAL-291354147.us-...	Activo	vpc-00441918281fdc4...	2 Zonas de disponibilidad.....	application	8 de junio de 2024, 00:05 (UTC-05:00)

- Registrar la instancia EC2 en el Load Balancer.

Win-DEST Acciones ▾

Detalles
 am.aws.elasticloadbalancing.us-east-2:730355422361:targetgroup/Win-DEST/2d4841ddfa45f0f2

Tipo de destino Instancia	Protocolo : Puerto HTTP: 80	Versión del protocolo HTTP1	VPC vpc-00441918281fdc41b
Tipo de dirección IP IPv4	Balanceador de carga ELB-FINAL		

1	🟢 1 En buen estado	🔴 0 En mal estado	⚪ 0 Sin utilizar	⚪ 0 Inicial	⚪ 0 Vaciado
Destinos totales	0 Anómalo				

► **Distribución de destinos por zona de disponibilidad (AZ)**
 Seleccione los valores de esta tabla para ver los filtros correspondientes aplicados a la tabla Destinos registrados que aparece a continuación.

Destinos | Monitorización | Comprobaciones de estado | Atributos | Etiquetas

Destinos registrados (1) info Mitigación de anomalías: No aplicable Anular el registro Registrar destinos

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

<input type="checkbox"/>	ID de instancia	Nombre	Puerto	Zona	Estado	Detalles del estado	Hora d...	Resultado de la detección de anomalías
<input type="checkbox"/>	i-069be7a4387c659f	FINAL	80	us-east-2b	🟢 Healthy	-	7 de junio...	🟢 Normal

- Configurar los listeners y reglas de enrutamiento para que el tráfico HTTP se dirija a las instancias EC2 registradas.

Agentes de escucha y reglas | Mapeo de red | Mapa de recursos - nuevo | Seguridad | Monitorización | Integraciones | Atributos | Etiquetas

Agentes de escucha y reglas (1) info Administrar reglas ▾ Administrar agente de escucha ▾ Agregar agente de escucha

Un agente de escucha comprueba las solicitudes de conexión en su protocolo y puerto configurados. El tráfico recibido por el agente de escucha se enruta de acuerdo con la acción predeterminada y cualquier regla adicional.

<input type="checkbox"/>	Protocolo:Port	Acción predeterminada	Reglas	ARN	Política de seguridad	Certificado SSL/TLS predet...	mTLS
<input type="checkbox"/>	HTTP:80	Reenviar al grupo de destino <ul style="list-style-type: none"> Win-DEST 1 (100%) Persistencia de nivel de grupo: Desactivada 	1 regla	ARN	No aplicable	No aplicable	No aplicable

- Probar el acceso a la aplicación a través de la URL del Load Balancer.

<http://elb-final-291354147.us-east-2.elb.amazonaws.com/>

IP <http://3.134.176.58/>

Video

https://remingtonedu-my.sharepoint.com/:v:/g/personal/jeimy_giraldo_9109_miremington_edu_c/EcUBait_10tBgQOdxdcqFK8Bg_9HijZBZKilxiIg0F7Mbg?referrer=Teams.TEAMS-ELECTRON&referrerScenario=MeetingChicletGetLink.view.view

Conclusiones

En resumen, este seminario se presenta como una herramienta esencial y versátil para el desarrollo de máquinas virtuales. Su origen como un proyecto de evolución hacia una virtualización para mejorar la tecnología que nos permite tener varios sistemas operativos en todo el mundo. AWS ofrece una amplia gama de componentes de diseño y una estructura coherente, facilitando la creación de máquinas virtuales, con altos estándares de calidad y funcionales. Fácilmente adaptables a distintos sistemas operativos.

Contamos con dos métodos principales para trabajar con AWS: a nivel local y en línea. Ambos enfoques tienen sus ventajas y desventajas, como se ilustra en una base de datos. Trabajar a nivel local proporciona mayor control y facilidad para personalizar los archivos, aunque requiere una gestión manual de actualizaciones y espacio en disco. Ofreciendo automáticamente las versiones más recientes de AWS, pero depende del acceso constante a Internet y puede ser más complejo en términos de versiones actualizadas. En definitiva, la elección entre trabajar localmente o en línea dependerá de las necesidades específicas del proyecto y de las preferencias del desarrollador. Sin embargo, ambos métodos aseguran que los desarrolladores puedan aprovechar al máximo las capacidades de una máquina virtual.

Referencias

- *Tutorial: Configuración de una aplicación con escalado y balanceo de carga aplicados - Amazon EC2 Auto Scaling.* (s. f.).
https://docs.aws.amazon.com/es_es/autoscaling/ec2/userguide/tutorial-ec2-auto-scaling-load-balancer.html
- *Establecimiento de una política de mantenimiento de instancias en el grupo de escalado automático - Amazon EC2 Auto Scaling.* (s. f.).
https://docs.aws.amazon.com/es_es/autoscaling/ec2/userguide/set-instance-maintenance-policy-on-group.html?icmpid=docs_ec2as_help_panel
- *Establecimiento de una política de mantenimiento de instancias en el grupo de escalado automático - Amazon EC2 Auto Scaling.* (s. f.-b).
https://docs.aws.amazon.com/es_es/autoscaling/ec2/userguide/set-instance-maintenance-policy-on-group.html?icmpid=docs_ec2as_help_panel