

Food FastConnected

Corporación Universitaria Remington.
Facultad de Ingeniería
Ingeniería de Sistemas/ Especialización en Seguridad de la
Información

Elkin David Zuluaga
Mary Luz Hernández Henao
Docente: Juan David Berrio
Opción de Trabajo de grado Seminario-Diplomado.
2025

Tabla de Contenidos

Resumen.....	3
Palabras clave.....	4
Marco conceptual y contextual	5
Entrega 1	6
Primera entrega	6
Parte 1:	6
1. Diagrama de arquitectura	6
2. Descripción de la arquitectura	6
3. Configuraciones Realizadas.....	7
4. Procedimiento de acceso	9
5. Configuración del servidor web.....	10
Parte 2	11
1. Crear La infraestructura	11
2. Configurar Grupos de seguridad.....	12
3. Acceder a Las instancias.....	14
4. Instalar y configurar los servidores web	15
5. Prueba de conectividad	16
6. Validación de acceso web	17
Entrega 2	18
Entrega Final.....	18
1. Balanceador de Carga:	19
2. Instancias EC2	20
3. Instancias proxy reverso	20
4. Implementacion del servicio de docker de forma manual con aplicación de prueba.....	21
5. AutoEscalado	21
6. Diagrama de arquitectura.....	24
Desarrollo e implementación del aprendizaje.....	25
Figuras y tablas	26
Conclusiones.....	28

Resumen

Este documento tiene como finalidad describir el diseño y la implementación de un entorno productivo de infraestructura basada en nube utilizando recursos de Amazon Web Services(AWS); Orientada a cumplir el requerimiento de crecimiento para la plataforma de Fast FoodConnect.

Con este principio de expansión, se crea una red VPC (Nube privada) donde se lanza un entorno en instancias de tipo Windows Server y Linux con grupos de seguridad que permiten controlar el tráfico entrante y saliente, la preparación de contenedores con una aplicación de prueba.

Está enfocado en desplegar servicios para operatividad en tiempo real y aplicaciones garantizando la eficiencia y seguridad de los servicios en nube con base a la demanda que requiera la organización garantizando disponibilidad, alto rendimiento y bajos costos.

Palabras clave

- AWS: (Amazon Web Services) especialista en servicios de nube.
- Nube: Maneja servicios informáticos como servidores con disco y cpu, bases de datos entre otros para crear una red de acuerdo con las necesidades.
- Seguridad: Se basa en proteger la información, ayuda a salvaguardar los datos.
- EC2: (Elastic Compute Cloud): Máquinas virtuales con Sistema operativo (Windows, Linux)
- VPC: Virtual private Cloud, red virtual aislada

Marco conceptual y contextual

¿Qué es la Nube? En términos de computación, forma parte de recursos como lo es almacenamiento, bases de datos, servidores, software, redes, entre otros que es entregado a través de internet.

AWS (Amazon Web Services):

Plataforma que permite construir e implementar servicios como aplicaciones sin necesidad de tener un equipo físico.

Instancias EC2 (Elastic Compute Cloud): Equipos virtuales sobre los que se instalan diferentes Sistemas Operativos como Windows-Linux, Ubuntu entre otros. Estos son necesarios para la operación de aplicaciones.

VPC (Virtual Private Cloud): permite crear red virtual aislada donde se definen tablas de ruteo, direcciones ip y subredes.

Grupos de seguridad: hacen la función de firewalls virtuales que controla lo que entra y sale de las instancias (EC2)

Balanceador de carga: Distribuye el tráfico o peticiones que llegan entre varios servidores o instancias

Contenedor Docker: Automatiza el despliegue y ejecución de aplicaciones.

EBS (Elastic Block Store): Suministra almacenamiento, tienen la función de discos duros virtuales.

Target Group: Define a que servicio se debe dirigir el tráfico. Contiene las instancias que forman parte del balanceador de carga.

Proxy Inverso: Es un intermediario entre el cliente y la aplicación, puede funcionar como balanceador de carga.

Entrega 1

Primera entrega

Implementación de una red en AWS con servidores Windows y Linux accesibles desde Internet

Objetivo General:

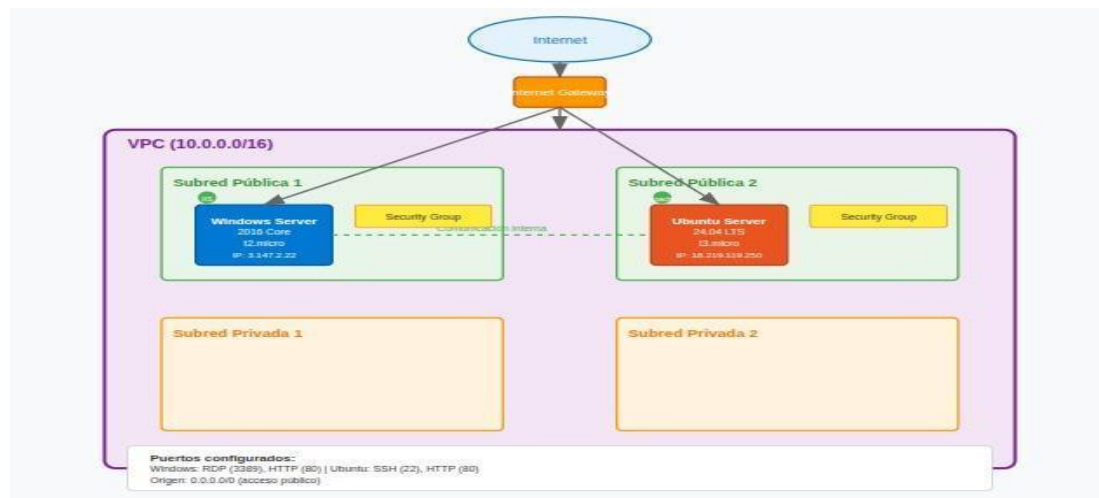
Diseñar, desplegar y documentar una red en AWS que incluya dos instancias EC2 (una Windows y una Linux), asegurando su accesibilidad pública, conectividad entre ellas y la instalación de un servidor web funcional en cada instancia.

Parte 1:

1. Diagrama de arquitectura

Figura 1

Diagrama de arquitectura AWS



2. Descripción de la arquitectura

La arquitectura implementada tiene los siguientes elementos, una VPC, un gateway, EC2 y grupos de seguridad

la VPC (10.0.0.0/16) cuenta con 4 subredes tiene 2 dos públicas y dos privadas, además también viene configurado por defecto un Gateway.

Se crearon con instancias e EC2 una en Windows Server 2016 core base y otra en Ubuntu 4.04 lts las dos el tipo de instancia en Ubuntu la t3.micro en Windows t2.micro cada uno

se le asignaron unas ip públicas del Vpc creado anteriormente, además se configuro un grupo de seguridad

Justificación de las configuraciones de red- VPC

En la infraestructura de AWS se ha creado una red Virtual con recursos gratuitos, llevando a pruebas controladas.

Uso VPC: Se ha implementado con el fin de tener control acerca del direccionamiento, políticas de seguridad y ruteo.

Subredes públicas: Se han definido para alojar recursos que necesitan acceso a internet (Instancias EC2) permitiendo verificaciones básicas, como de conectividad.

Internet Gateway: Se agrega un Internet Gateway a la VPC y se asocian las rutas a internet desde las subredes. Esta configuración ayuda a que los recursos ubicados en las subredes, establezcan tráfico hacia afuera.

3. Configuraciones Realizadas

Pasos para creación instancias de EC2:

Se configura VPC con IPv4 10.0.0.0/16 y 4 subredes con zonas de disponibilidad en (**us-east-2a**)(**us-east-2b**)

Windows

Se realiza búsqueda de EC2, se selecciona instancias y se da la opción de crear Instancias.

-Se coloca nombre, sistema operativo y la imagen, para Windows se selecciona (Windows server core 2016).

- Tipo de instancia (t2.micro) este proporciona rendimiento de CPU, y se selecciona subred.

-Se coloca ip publica por defecto, y se crean los grupos de seguridad (RDP)

-Se configura almacenamiento (30 Gib GP2)

En la creación de la instancia, se define su arquitectura, memoria y Nombre de usuario “**Administrator**” el cual es el que permite conectarse por RDP.

-Una vez creada la instancia se realiza prueba de conexión por el puerto 3389 configurado previamente, desde la opción Conectar en la instancia se busca Usuario y se descarga la clave privada con Extensión .pem

- El grupo de seguridad, donde se define la IP-publica además de seleccionar la clave .en el archivo generado .PEM que es otro de los datos requeridos para ingresar en Windows de forma de RDP
- Luego selecciona la unidad de almacenamiento donde se selecciona la memoria de almacenamiento

Linux

- Se ingresa en la instancia y se da clic en opción de Lanzar Instancias.
- Se selecciona imagen ubuntu Server 24.04 LTS o Amazon Linux.
- Se hace configuración de red como en Windows, en él se define la IP-publica, tambien se selecciona la clave .pem en el archivo generado con antelación para la instancia de Windows. Este dato es requerido para ingresar a ubuntu via SSH.
- Se crea grupo de seguridad y reglas de seguridad: Puerto 80 y puerto 22 (SSH).
- Luego selecciona la unidad de almacenamiento donde se selecciona la memoria de almacenamiento.

Tabla 1

Reglas de grupo de seguridad

Nombre	Protocolo	Puerto	Origen
Windows RDP	TCP	3389	0.0.0.0/0
Windows HTTP	TCP	80	0.0.0.0/0
Linux HTTP	TCP	80	0.0.0.0/0
Linux SSH	TCP	22	0.0.0.0/0
Ping	ICMP	N/A	N/A

Nota. Estas aplican a grupos de seguridad de instancias Windows y Linux.

Tabla 2

Asignacion de Ip-publicas y privadas

Instancia	IP pública	IP privada
Windows Server 2016 Core	3.147.2.22	10.0.3.153
Ubuntu Server 24.04 LTS	18.219.119.250	172.31.34.147

Nota. Las IP son tomadas de forma automática por AWS en el despliegue de las instancias en distintas subredes.

4. Procedimiento de acceso

Windows server 2016 core

-En “ubuntu” se usó *Remmina* para el acceso a Windows

usuario: administrator

Contraseña: es la que se obtiene en AWS por Acciones->Seguridad-> obtener la contraseña de Windows, se sube el archivo .pem que creamos al momento de crear la instancia de windows y se da en la opción de Descifrar Contraseña

ubuntu Server LTS

ssh -i (Ruta archivo)/ubuntu.pem ubuntu@ippublica

Consideraciones de seguridad

- La clave .pem se guardó en una carpeta segura. Protege información sensible por medio de cifrado
- La contraseña se genera de manera automática y única lo que evita un riesgo de vulnerabilidades de claves.

5. Configuración del servidor web

Windows Server

-Se ubica el archivo .pem descargado previamente.
 -Por medio de la máquina local, se abre conexión de escritorio remoto, y con la contraseña .pem descargada previamente se accede al seridor. Abrimos **Server Manager** en el menu luego seleccionamos Addy o Roles and Features → seleccionamos Web Server(IIS)→ terminamos los pasos, ya luego en Sites en la parte derecha seleccionamos Browser *:80 y nos debería de cargar el navegador con todo en la siguiente url <http://3.147.2.22/>

Ubuntu Server

-Luego de ingresar mediante via ssh `ssh -i (Ruta archivo)/ubuntu.pem ubuntu@ippublica` ejecutamos el comando `sudo apt install apache2` -y luego

`sudo systemctl enable apache2`

`sudo systemctl start apache2`

Se verifica:

`curl localhost` o la url <http://18.219.119.250>

Figura 2

Pruebas de acceso web Windows server

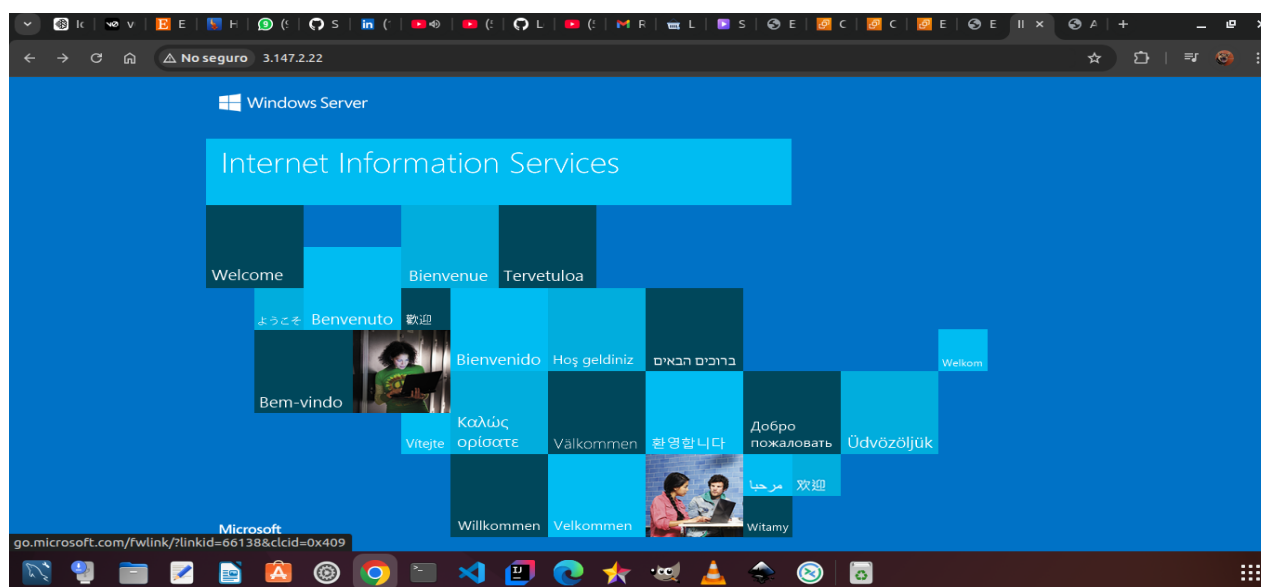
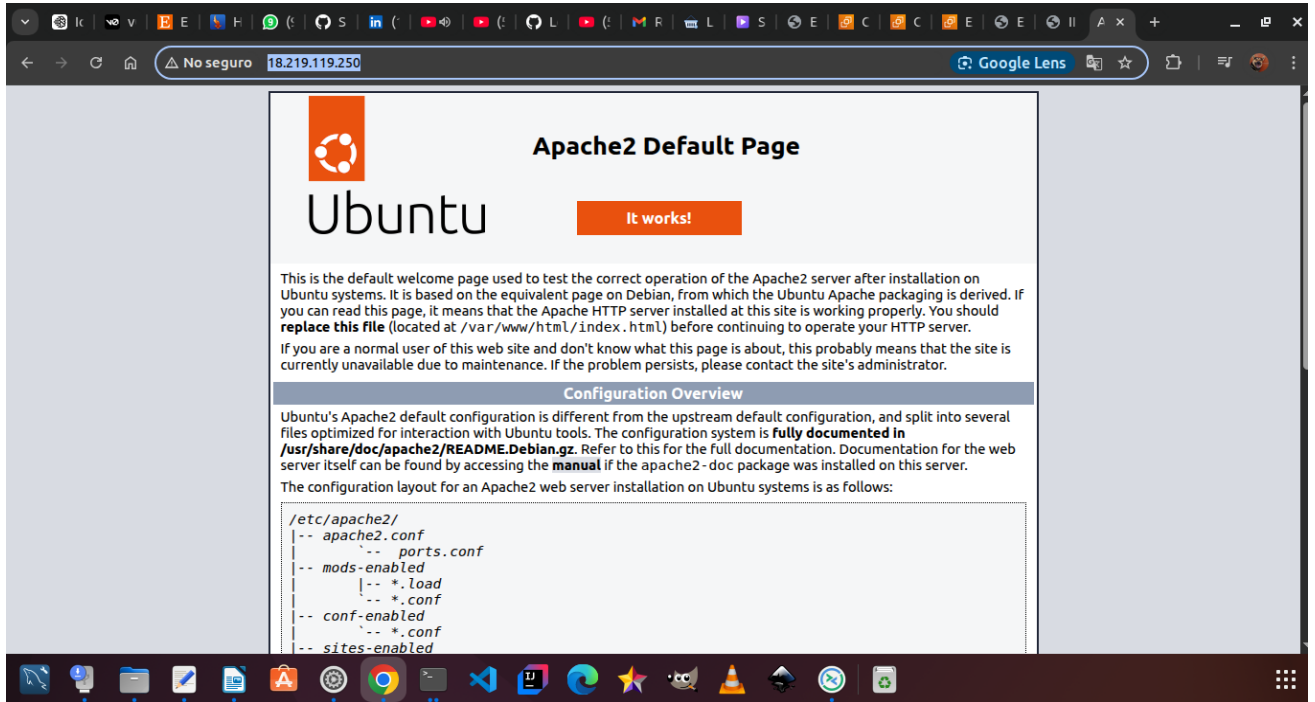


Figura 3

Prueba de conectividad Linux server



Parte 2

1. Crear La infraestructura

VPC y subredes

Figura 4

Configuración de VPC

The screenshot shows the AWS Management Console for a VPC named 'seminario-vpc' in the us-east-2 region. The console displays the following details:

- Detalles:** ID de la VPC: vpc-0b93e550e8e9e612d, Estado: Disponible, Resolución de DNS: Habilitada, ACL de red principal: acl-02a8a258194c0f552, CIDR IPv6: -.
- Estado:** Disponible, Tenencia: default, VPC predeterminada: No, Métricas de uso de direcciones de red: Desactivado.
- Bloquear el acceso público:** Desactivado, Conjunto de opciones de DHCP: dhcp-0837a3d57ef62aa22, CIDR IPv4: 10.0.0.0/16, Grupos de reglas del firewall de DNS de Route 53: Resolver.
- Mapa de recursos:** Muestra un diagrama de la VPC con 4 subredes (us-east-2a, us-east-2b) y 4 tablas de enrutamiento (seminario-rtb-private1-us-east-2a, seminario-rtb-private1-us-east-2b, seminario-rtb-public, seminario-rtb-private2-us-east-2b). También muestra 2 conexiones de red (seminario-igw, seminario-vpc-e3).

Figura 5
Lanzamiento de Instancias EC2

The screenshot shows the AWS Management Console for EC2 Instances in the us-east-2 region. The console displays the following details:

- Instancias (1/2):** Lista de instancias con columnas: Name, ID de la Instancia, Estado de la Instancia, Tipo de Instancia, Comprobación de estado, Estado de la instancia, Zona de disponibilidad, DNS de IPv4 pública, Dirección IP pública, Dirección IP privada.
- Instancia seleccionada (i-02a80cb246a8793b2):**
 - Resumen de instancia:** ID de la instancia: i-02a80cb246a8793b2, Dirección IPv6: -, Tipo de nombre de anfitrión: Nombre de IP: ip-10-0-3-153.us-east-2.compute.internal.
 - Dirección IPv4 pública:** 3.147.2.22 | dirección abierta.
 - Direcciones IPv4 privadas:** 10.0.3.153.
 - Estado de la instancia:** En ejecución.
 - DNS público:** ec2-3-147-2-22.us-east-2.compute.amazonaws.com | dirección abierta.
 - Nombre DNS de IP privada (solo IPv4):** ip-10-0-3-153.us-east-2.compute.internal.

2. Configurar Grupos de seguridad

Figura 6
Grupos de seguridad -Windows Server

The screenshot shows the AWS Management Console interface for a security group named 'sg-0fe581a229523d1c0 - windowsServer'. The 'Reglas de entrada' (Inbound Rules) section is active, displaying a table with two rules:

Nombre	ID de la regla del grupo...	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción
-	sgr-07eaacd871b41af26	IPv4	RDP	TCP	3389	0.0.0.0/0	-
-	sgr-06546e9eff39bc16f	IPv4	HTTP	TCP	80	0.0.0.0/0	pagina

Nota. Captura con evidencia de las reglas configuradas

Figura 7
Grupos de seguridad -Linux Server

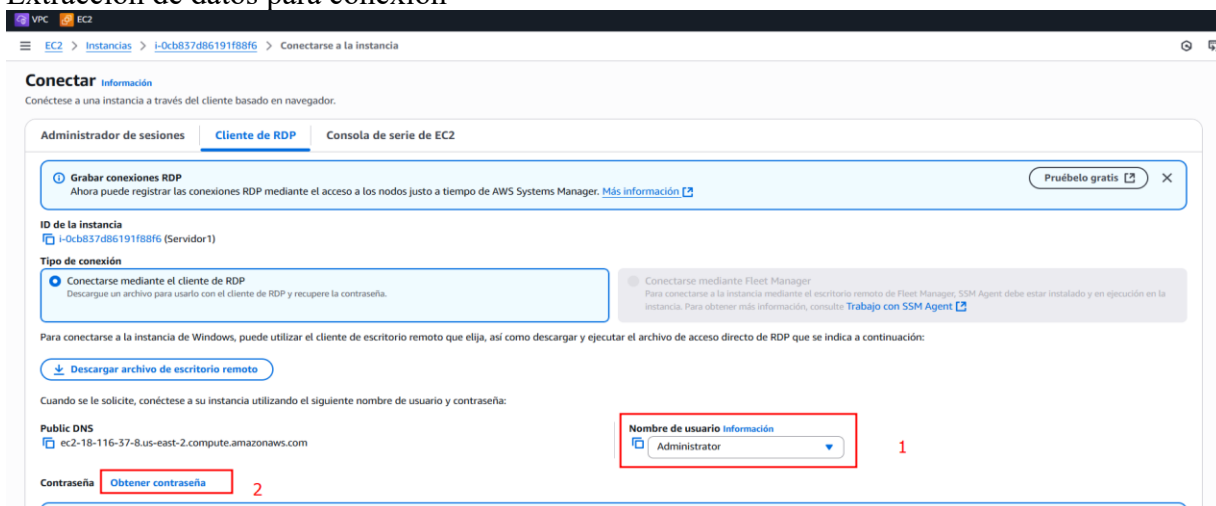
The screenshot shows the AWS Management Console interface for a security group named 'sg-07a853b66addd7b6a'. The 'Reglas de entrada' (Inbound Rules) section is active, displaying a table with three rules:

Nombre	ID de la regla del grupo d...	Intervalo de pu...	Protocolo	Origen	Grupos de seguridad	Descripción
-	sgr-09588027b4b69bdb8	22	TCP	0.0.0.0/0	default	ssh
-	sgr-0cbe67b65c5124f1a	Todo	Todo	sg-07a853b66addd7b6a	default	-
-	sgr-0b0445e42df9d90d8	80	TCP	0.0.0.0/0	default	-

Nota. Se observan puertos habilitados para permisos de acceso.

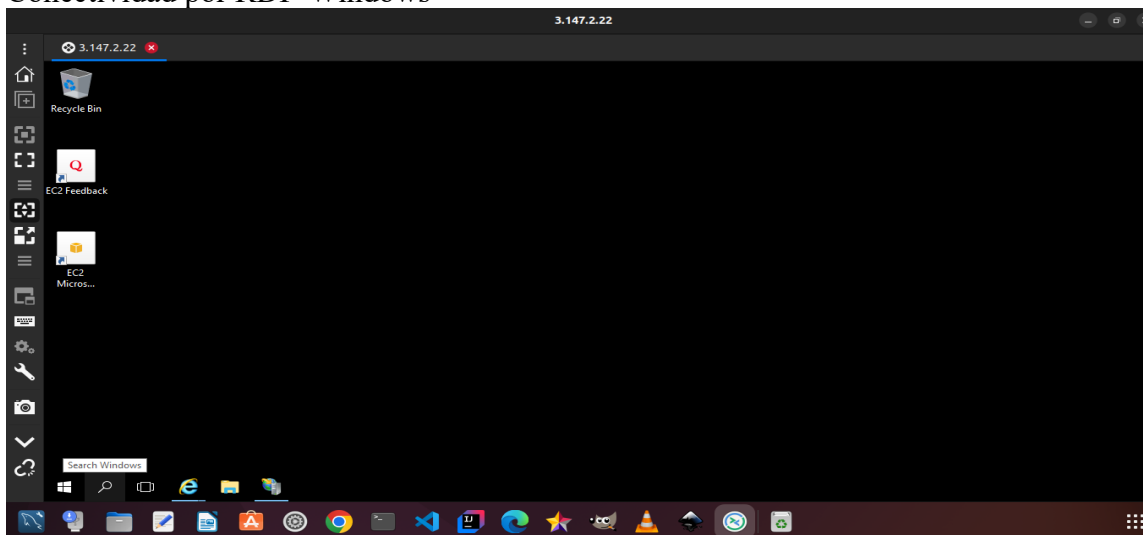
3. Acceder a Las instancias

Figura 8
Extracción de datos para conexión



Nota. En la imagen se identifica forma de obtener usuario y contraseña para ingresar a la instancia

Figura 9
Conectividad por RDP-Windows



Nota. Se visualiza el ingreso exitoso

Figura 10
Conectividad por SSH - Ubuntu Server

```

ubuntu@ip-172-31-34-147: ~
ivan@ivan-X555LB:~/Descargas/seminarioAWS$ ssh -i ~/Descargas/seminarioAWS/ubuntu.pem ubuntu@18.219.119.250
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1029-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Fri Jul  4 06:24:57 UTC 2025

System load:  0.0          Processes:      108
Usage of /:   34.1% of 6.71GB   Users logged in:  0
Memory usage: 30%          IPv4 address for enX0: 172.31.34.147
Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

5 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

*** System restart required ***
Last login: Thu Jul  3 05:51:02 2025 from 181.135.192.7
ubuntu@ip-172-31-34-147:~$

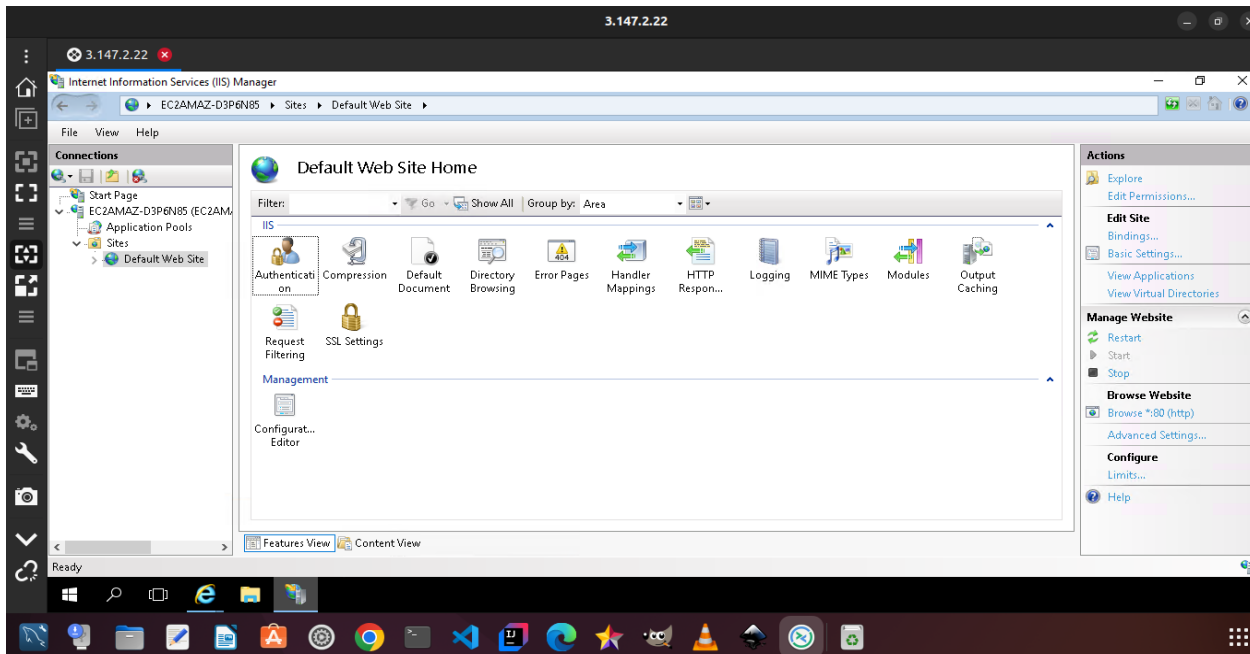
```

Nota. Se observa adecuado ingreso a la instancia Linux Server

4. Instalar y configurar los servidores web

Figura 11

Instalación IIS - Windows server



Nota. Captura de pantalla del Windows server para instalación de rol de IIS con el fin de subir sitio Web.

Figura 12
Instalación Apache - Ubuntu Server

```

directories (when enabled) and <tt>/usr/share</tt> (for web
applications). If your site is using a web document root
located elsewhere (such as in <tt>/srv</tt>) you may need to whitelist your
document root directory in <tt>/etc/apache2/apache2.conf</tt>.

</p>
<p>
The default Ubuntu document root is <tt>/var/www/html</tt>. You
can make your own virtual hosts under /var/www.

</p>
</div>

<div class="section_header">
<div id="bugs"></div>
Reporting Problems
</div>
<div class="content_section_text">
<p>
Please use the <tt>ubuntu-bug</tt> tool to report bugs in the
Apache2 package with Ubuntu. However, check <a
href="https://bugs.launchpad.net/ubuntu/+source/apache2"
rel="nofollow">existing bug reports</a> before reporting a new bug.

</p>
<p>
Please report bugs specific to modules (such as PHP and others)
to their respective packages, not to the web server itself.

</p>
</div>
</div>
<div class="validator">
</div>
</body>
</html>
ubuntu@ip-172-31-34-147: ~$

```

5. Prueba de conectividad

Figura 13
Prueba ping desde Windows a Windows Server

```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> netsh advfirewall firewall add rule name="Permitir ICMPv4 Entrante" protocol=icmpv4:8,any dir=in action=allow
Ok.

PS C:\Users\Administrator> netsh advfirewall firewall add rule name="Permitir ICMPv4 Entrante" protocol=icmpv4:8,any dir=in action=allow
Ok.

PS C:\Users\Administrator> ping 10.0.5.104

Pinging 10.0.5.104 with 32 bytes of data:
Reply from 10.0.5.104: bytes=32 time=1ms TTL=64
Reply from 10.0.5.104: bytes=32 time=1ms TTL=64
Reply from 10.0.5.104: bytes=32 time=1ms TTL=64
Reply from 10.0.5.104: bytes=32 time=1ms TTL=64

Ping statistics for 10.0.5.104:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milliseconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
PS C:\Users\Administrator>

```

Nota. En la imagen se identifica ping exitoso de respuesta hacia el servidor Linux desde Windows

Figura 14
Prueba de Ping desde Ubuntu a Windows

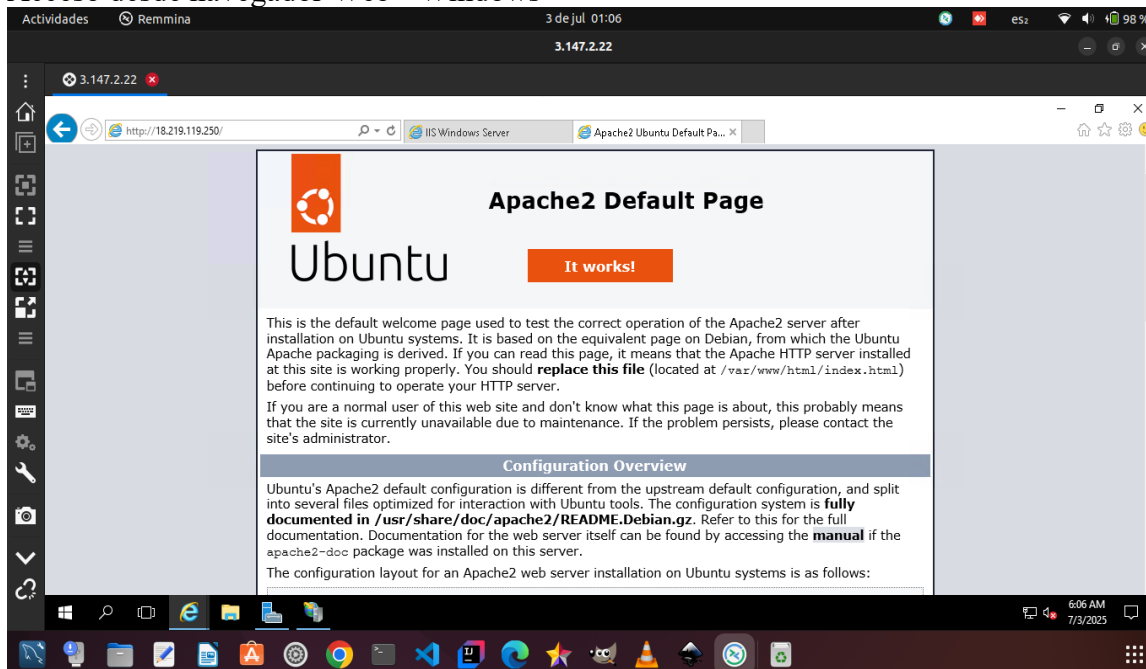
```

ubuntu@ip-10-0-5-104:~$ ping 18.224.45.119
PING 18.224.45.119 (18.224.45.119) 56(84) bytes of data:
64 bytes from 18.224.45.119: icmp_seq=1 ttl=127 time=0.484 ms
64 bytes from 18.224.45.119: icmp_seq=2 ttl=127 time=0.443 ms
64 bytes from 18.224.45.119: icmp_seq=3 ttl=127 time=1.12 ms
64 bytes from 18.224.45.119: icmp_seq=4 ttl=127 time=0.507 ms
64 bytes from 18.224.45.119: icmp_seq=5 ttl=127 time=0.443 ms
64 bytes from 18.224.45.119: icmp_seq=6 ttl=127 time=0.471 ms
64 bytes from 18.224.45.119: icmp_seq=7 ttl=127 time=0.545 ms
64 bytes from 18.224.45.119: icmp_seq=8 ttl=127 time=0.526 ms
64 bytes from 18.224.45.119: icmp_seq=9 ttl=127 time=0.509 ms
64 bytes from 18.224.45.119: icmp_seq=10 ttl=127 time=0.478 ms
64 bytes from 18.224.45.119: icmp_seq=11 ttl=127 time=0.530 ms
64 bytes from 18.224.45.119: icmp_seq=12 ttl=127 time=0.469 ms
64 bytes from 18.224.45.119: icmp_seq=13 ttl=127 time=0.412 ms
64 bytes from 18.224.45.119: icmp_seq=14 ttl=127 time=3.16 ms
64 bytes from 18.224.45.119: icmp_seq=15 ttl=127 time=0.612 ms
64 bytes from 18.224.45.119: icmp_seq=16 ttl=127 time=0.467 ms
64 bytes from 18.224.45.119: icmp_seq=17 ttl=127 time=0.476 ms
64 bytes from 18.224.45.119: icmp_seq=18 ttl=127 time=0.502 ms
64 bytes from 18.224.45.119: icmp_seq=19 ttl=127 time=0.543 ms
64 bytes from 18.224.45.119: icmp_seq=20 ttl=127 time=0.584 ms
64 bytes from 18.224.45.119: icmp_seq=21 ttl=127 time=0.480 ms
64 bytes from 18.224.45.119: icmp_seq=22 ttl=127 time=0.542 ms
64 bytes from 18.224.45.119: icmp_seq=23 ttl=127 time=0.664 ms
64 bytes from 18.224.45.119: icmp_seq=24 ttl=127 time=0.670 ms
64 bytes from 18.224.45.119: icmp_seq=25 ttl=127 time=0.544 ms
64 bytes from 18.224.45.119: icmp_seq=26 ttl=127 time=0.485 ms
64 bytes from 18.224.45.119: icmp_seq=27 ttl=127 time=0.509 ms
64 bytes from 18.224.45.119: icmp_seq=28 ttl=127 time=0.476 ms
64 bytes from 18.224.45.119: icmp_seq=29 ttl=127 time=0.550 ms
64 bytes from 18.224.45.119: icmp_seq=30 ttl=127 time=0.478 ms
64 bytes from 18.224.45.119: icmp_seq=31 ttl=127 time=0.532 ms
64 bytes from 18.224.45.119: icmp_seq=32 ttl=127 time=0.499 ms
64 bytes from 18.224.45.119: icmp_seq=33 ttl=127 time=0.448 ms
64 bytes from 18.224.45.119: icmp_seq=34 ttl=127 time=0.522 ms
64 bytes from 18.224.45.119: icmp_seq=35 ttl=127 time=0.522 ms
64 bytes from 18.224.45.119: icmp_seq=36 ttl=127 time=0.522 ms
64 bytes from 18.224.45.119: icmp_seq=37 ttl=127 time=0.522 ms
64 bytes from 18.224.45.119: icmp_seq=38 ttl=127 time=0.522 ms
64 bytes from 18.224.45.119: icmp_seq=39 ttl=127 time=0.522 ms
64 bytes from 18.224.45.119: icmp_seq=40 ttl=127 time=0.522 ms
64 bytes from 18.224.45.119: icmp_seq=41 ttl=127 time=0.522 ms
64 bytes from 18.224.45.119: icmp_seq=42 ttl=127 time=0.522 ms
64 bytes from 18.224.45.119: icmp_seq=43 ttl=127 time=0.522 ms
64 bytes from 18.224.45.119: icmp_seq=44 ttl=127 time=0.522 ms
64 bytes from 18.224.45.119: icmp_seq=45 ttl=127 time=0.522 ms
64 bytes from 18.224.45.119: icmp_seq=46 ttl=127 time=0.522 ms
64 bytes from 18.224.45.119: icmp_seq=47 ttl=127 time=0.522 ms
64 bytes from 18.224.45.119: icmp_seq=48 ttl=127 time=0.522 ms
64 bytes from 18.224.45.119: icmp_seq=49 ttl=127 time=0.522 ms
64 bytes from 18.224.45.119: icmp_seq=50 ttl=127 time=0.522 ms

```

6. Validación de acceso web

Figura 15
Acceso desde navegador Web - Windows



Nota. En la imagen se identifica el acceso desde el navegador local al sitio de la instancia Windows

Figura 16

Acceso desde navegador Web - Ubuntu

```

located elsewhere (such as in <tt>/srv</tt>) you may need to whitelist your
document root directory in <tt>/etc/apache2/apache2.conf</tt>.
</p>
<p>
The default Ubuntu document root is <tt>/var/www/html</tt>. You
can make your own virtual hosts under /var/www.
</p>
</div>
<div class="section_header">
<div id="bugs"></div>
Reporting Problems
</div>
<div class="content_section_text">
<p>
Please use the <tt>ubuntu-bug</tt> tool to report bugs in the
Apache2 package with Ubuntu. However, check <a
href="https://bugs.launchpad.net/ubuntu/+source/apache2"
rel="nofollow">existing bug reports</a> before reporting a new bug.
</p>
<p>
Please report bugs specific to modules (such as PHP and others)
to their respective packages, not to the web server itself.
</p>
</div>
</div>
</div>
<div class="validator">
</div>
</div>
</body>
</html>
ubuntu@ip-172-31-34-147:~$ nc -zv 3.147.2.22 3389
Connection to 3.147.2.22 3389 port [tcp/ms-wbt-server] succeeded!
ubuntu@ip-172-31-34-147:~$ █

```

Nota. En la imagen se identifica el acceso desde el navegador local al sitio de la instancia Ubuntu Server

Entrega 2

Entrega Final

Trabajo Práctico: Implementación de Arquitectura en AWS con Balanceador de Carga y Contenedores

Introducción: La startup " Food FastConnected "

Bienvenidos al proyecto de implementación. En esta tarea, ustedes asumirán el rol de arquitectos en la nube para una startup llamada Food FastConnected, una plataforma innovadora que conecta a restaurantes con clientes mediante entregas rápidas. La empresa ha experimentado un rápido crecimiento en los últimos meses y ahora necesita escalar su infraestructura tecnológica para manejar una mayor demanda, garantizar la disponibilidad y mejorar los tiempos de respuesta.

El CTO de Food FastConnected ha diseñado una arquitectura preliminar y necesita de su ayuda para implementarla en **Amazon Web Services (AWS)**. La solución debe ser altamente disponible, escalable y estar diseñada para manejar una gran cantidad de tráfico de manera eficiente.

Objetivo del Trabajo

Implementar una arquitectura en AWS que cumpla con los siguientes requisitos:

1. *Balancedador de Carga:*

Buscamos tener una alta disponibilidad, especialmente en horas pico (1:00pm a 2:30 pm y 6:00 pm a 9:30pm). Las peticiones que recibe y las divide equitativamente entre todas las instancias EC2 configuradas del balanceador de carga, el punto de inicio en la arquitectura usada para Food FastConnected es el que recibe las peticiones HTTP y que este es el encargado de distribuir equitativamente entre las instancias de EC2

características de la configuración:

- **protocolo y puerto:** Http/80
- **Esquema:** internet-facing
- **Zona de disponibilidad:** us-east-2a y ua-east-2b
- **Grupo de destino:** TargetGroupDocker
- **estado:** activo

Figura 17

Composición del Balanceador para el contenedor Docker

The screenshot displays the configuration details for the 'BalanceadorDocker' load balancer. Key information includes:

- Estado:** Activo
- Esquema:** Internet-facing
- Zona hospedada:** Z3AADJGKKTLL2
- VPC:** vpc-0b93e550e8e9e612d
- Zonas de disponibilidad:** us-east-2a (use2-az1) and us-east-2b (use2-az2)
- Tipo de dirección IP del equilibrador de carga:** IPv4
- Fecha creada:** 18 de julio de 2025, 23:11 (UTC-05:00)
- ARN del equilibrador de carga:** arn:aws:elasticloadbalancing:us-east-2:237431694113:loadbalancer/app/BalanceadorDocker/a21f759b629077a5
- Nombre de DNS:** BalanceadorDocker-966095825.us-east-2.elb.amazonaws.com (Registro A)

Below the details, the 'Agentes de escucha y reglas' section shows a single listener configuration:

Protocolo:Port	Acción predeterminada	Reglas	ARN	Política de seguridad	Certificado SSL/TLS predet...	mTLS	Tru
HTTP:80	Reenviar al grupo de destino	1 regla	ARN	No aplicable	No aplicable	No aplicable	No

The target group for the listener is 'TargetGroupDocker' with a weight of 1 (100%) and is currently 'Desactivada'.

Nota. En la captura se identifica el Balanceador de carga en AWS configurado para enrutar el tráfico HTTP por medio del puerto 80 hacia los contenedores Docker

2. Instancias EC2

Al momento de crear el balanceador de carga se realizó la configuración que ayudará a tener una mejor disponibilidad debido a que va permitir tener la instancia en dos lugares físicos diferentes y de esta manera una alta disponibilidad.

Si falla un data center la cual la probabilidad es baja, menos del 1%, se debe tener otro de respaldo en una zona totalmente diferente, como se muestra en la imágenes a continuación, ya que están en zona diferentes

Figura 18
Implementación de instancia Multizona 1

Resumen de instancia de i-0967da729346f7f09 (UbuntuDockerNginx) Información

Se ha actualizado hace 3 minutos

ID de la instancia
i-0967da729346f7f09

Dirección IPv6
-

Tipo de nombre de anfitrión
Nombre de IP: ip-10-0-9-139.us-east-2.compute.internal

Responder al nombre DNS de recurso privado
-

Dirección IP asignada automáticamente
3.138.37.155 [IP pública]

Rol de IAM
-

IMDSv2
Opcional
⚠️ EC2 recomienda configurar IMDSv2 como obligatorio | Más información

Operador
-

Dirección IPv4 pública
3.138.37.155 | dirección abierta

Estado de la instancia
En ejecución

Nombre DNS de IP privada (solo IPv4)
ip-10-0-9-139.us-east-2.compute.internal

Tipo de instancia
t2.micro

ID de VPC
vpc-0b93e550e8e9e612d (seminario-vpc)

ID de subred
subnet-0f8fd4f165baf65b (seminario-subnet-public1-us-east-2a)

ARN de instancia
arn:aws:ec2:us-east-2:237431694113:instance/i-0967da729346f7f09

Direcciones IPv4 privadas
10.0.9.139

DNS público
ec2-3-138-37-155.us-east-2.compute.amazonaws.com | dirección abierta

Direcciones IP elásticas
-

Hallazgo de AWS Compute Optimizer
Suscribirse a AWS Compute Optimizer para recibir recomendaciones. | Más información

Nombre del grupo de Auto Scaling
-

Administradas
falso

Figura 19
Implementación de instancia Multizona 2

Resumen de instancia de i-06a86878f05b54514 Información

Se ha actualizado hace 4 minutos

ID de la instancia
i-06a86878f05b54514

Dirección IPv6
-

Tipo de nombre de anfitrión
Nombre de IP: ip-10-0-155-177.us-east-2.compute.internal

Responder al nombre DNS de recurso privado
-

Dirección IP asignada automáticamente
-

Rol de IAM
-

IMDSv2
Opcional
⚠️ EC2 recomienda configurar IMDSv2 como obligatorio | Más información

Operador
-

Dirección IPv4 pública
-

Estado de la instancia
En ejecución

Nombre DNS de IP privada (solo IPv4)
ip-10-0-155-177.us-east-2.compute.internal

Tipo de instancia
t2.micro

ID de VPC
vpc-0b93e550e8e9e612d (seminario-vpc)

ID de subred
subnet-0a452b3d95dfc736f (seminario-subnet-private2-us-east-2b)

ARN de instancia
arn:aws:ec2:us-east-2:237431694113:instance/i-06a86878f05b54514

Direcciones IPv4 privadas
10.0.155.177

DNS público
-

Direcciones IP elásticas
-

Hallazgo de AWS Compute Optimizer
Suscribirse a AWS Compute Optimizer para recibir recomendaciones. | Más información

Nombre del grupo de Auto Scaling
AutoScalingSeminario

Administradas
falso

3. Instancias proxy reverso

Lo que realiza el proxy inverso(reverso) es que al momento recibir una petición en el caso de nosotros “Nginx” este lo direcciona internamente a un contenedor para este caso se ha usado Docker. Nginx direcciona de manera equitativa como un balanceador de carga.

Figura 20
Implementación de proxy reverso

```

GNU nano 7.2 nginx.conf
events {}

http {
    upstream seminario {
        server localhost:81;
        server localhost:82;
        server localhost:84;
    }

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

```

Nota. En la imagen se muestra como se configura el proxy inverso y los puertos a los que va realizar el balanceo.

4. Implementación del servicio de Docker de forma manual con aplicación de prueba

Para el proceso de contenerización de la aplicación se realizó la configuración y la instalación de Docker en la instancia y se buscó la aplicación que se requería. Para esta situación se instaló apache. En la imagen se muestra como tenemos tres contenedores con aplicaciones funcionando y corriendo.

Figura 21
Contenedores Docker con NGINX

```

ubuntu@ip-10-0-0-36:~$ cd /etc/nginx/
ubuntu@ip-10-0-0-36:/etc/nginx$ nano nginx.conf
ubuntu@ip-10-0-0-36:/etc/nginx$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
09c60e444696   httpd    "httpd-foreground"      18 hours ago   Up 17 hours   0.0.0.0:84->80/tcp, [::]:84->80/tcp   app84
aeab39dd5949   httpd    "httpd-foreground"      18 hours ago   Up 17 hours   0.0.0.0:82->80/tcp, [::]:82->80/tcp   app82
e9c8ec9fdfce   httpd    "httpd-foreground"      18 hours ago   Up 17 hours   0.0.0.0:81->80/tcp, [::]:81->80/tcp   app81
ubuntu@ip-10-0-0-36:/etc/nginx$
ubuntu@ip-10-0-0-36:/etc/nginx$ █

```

Nota. En la imagen se muestra como tenemos tres contenedores con aplicaciones funcionando y corriendo.

5. AutoEscalado

Se realizó la configuración que permita a la aplicación cumplir con la alta disponibilidad en horas pico para que si requiere crear nuevas instancias lo haga, para mantener la operatividad, Se configuró que al superar el 80% de CPU cree una nueva instancia.

Figura 22
Configuración de política de autoescalado

AutoScalingSeminaro

AutoScalingSeminaro Descripción general de la capacidad Editar

aws:autoscaling:us-east-2:237431694113:autoScalingGroup:1a26f133-bc2b-44ef-926c-e037d3e994bd:autoScalingGroupName/AutoScalingSeminaro

Capacidad deseada 2	Límites de escalamiento (Min. - Máx.) 2 - 4	Tipo de capacidad deseado Unidades (número de instancias)	Estado Actualización de capacidad
-------------------------------	---	---	---

Fecha de creación
Thu Jul 17 2025 00:30:21 GMT-0500 (hora estándar de Colombia)

Detalles | Integraciones - nueva | **Escalado automático** | Administración de instancias | Actualización de instancias | Actividad | Monitoreo

Las políticas de escalado ajustan el tamaño del grupo de escalado automático para satisfacer los cambios de demanda. Con las políticas de escalado dinámico reactivas, puede realizar un seguimiento de métricas específicas de CloudWatch y tomar medidas cuando se alcance el umbral de alarma de CloudWatch. Utilice políticas de escalado predictivo junto con políticas de escalado dinámico en las siguientes situaciones: cuando la demanda de la aplicación cambie rápidamente, pero con un patrón recurrente, o cuando las instancias de EC2 necesiten más tiempo para inicializarse.

Políticas de escalado dinámico (1) Acciones Crear una política de escalado dinámico

80 por ciento servidores

Tipo de política
Escalado de seguimiento de destino

Habilitado o deshabilitado
Habilitado

Ejecutar la política cuando
Según sea necesario para mantener Utilización promedio de la CPU en 30

Realizar la acción
Agregar o eliminar unidades de capacidad según sea necesario

Las instancias necesitan
300 segundos para prepararse antes de incluirse en la métrica

Escalado descendente
Habilitado

Nota. Se identifica puesta de política al 80%

En el siguiente caso al llegar al límite permitido, se crean las instancias de manera automática. Se evidencia creación de las instancias, las cuales en para este caso fueron 4.

Figura 23
Creación de las instancias según la carga

Estado	Descripción	Causa	Hora de inicio	Hora de finalización
Correcto	Terminating EC2 instance: i-08a1686c41cfa1234	At 2025-07-20T06:42:01Z a monitor alarm TargetTracking-AutoScalingSeminaro-AlarmLow-cd381657-b223-41d1-a71a-f30452e04854 in state ALARM triggered policy 80 por ciento servidores changing the desired capacity from 3 to 2. At 2025-07-20T06:42:05Z 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 2025-07-20T06:42:05Z instance i-08a1686c41cfa1234 was selected for termination.	2025 July 20, 01:42:05 AM -05:00	2025 July 20, 01:47:07 AM -05:00
Correcto	Terminating EC2 instance: i-07f7fb177a3e61ced	At 2025-07-20T06:40:55Z a monitor alarm TargetTracking-AutoScalingSeminaro-AlarmLow-cd381657-b223-41d1-a71a-f30452e04854 in state ALARM triggered policy 80 por ciento servidores changing the desired capacity from 4 to 3. At 2025-07-20T06:41:07Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 4 to 3. At 2025-07-20T06:41:07Z instance i-07f7fb177a3e61ced was selected for termination.	2025 July 20, 01:41:07 AM -05:00	2025 July 20, 01:46:29 AM -05:00
Correcto	Launching a new EC2 instance: i-07e675cd127b8af10	At 2025-07-20T06:33:34Z an instance was launched in response to an unhealthy instance needing to be replaced.	2025 July 20, 01:33:36 AM -05:00	2025 July 20, 01:34:08 AM -05:00
Correcto	Terminating EC2 instance: i-08f609c10e80e1404	At 2025-07-20T06:33:34Z an instance was taken out of service in response to an EC2 health check indicating it has been terminated or stopped.	2025 July 20, 01:33:34 AM -05:00	2025 July 20, 01:38:36 AM -05:00
Correcto	Launching a new EC2 instance: i-06a86878f05b54514	At 2025-07-20T06:21:44Z an instance was launched in response to an unhealthy instance needing to be replaced.	2025 July 20, 01:21:46 AM -05:00	2025 July 20, 01:21:53 AM -05:00
Correcto	Terminating EC2 instance: i-08ed08f78dccc890b2	At 2025-07-20T06:21:44Z an instance was taken out of service in response to an EC2 health check indicating it has been terminated or stopped.	2025 July 20, 01:21:44 AM -05:00	2025 July 20, 01:26:46 AM -05:00

Nota. Se evidencia creación de las instancias, las cuales en para este caso fueron 4.

Evidencia Funcionamiento

A continuación, se muestra la operatividad de un contenedor con sus respectivos puertos en funcionamiento.

Figura 24

Identificación de uso de contenedores

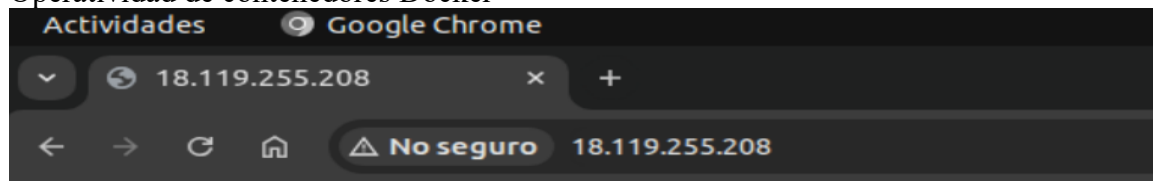
```

ubuntu@ip-10-0-0-36:~$ cd /etc/nginx/
ubuntu@ip-10-0-0-36:~$ nano nginx.conf
ubuntu@ip-10-0-0-36:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS                               NAMES
09c60e444696   httpd    "httpd-foreground"     18 hours ago    Up 17 hours    0.0.0.0:84->80/tcp, [::]:84->80/tcp    app84
aeb39dd5949   httpd    "httpd-foreground"     18 hours ago    Up 17 hours    0.0.0.0:82->80/tcp, [::]:82->80/tcp    app82
e9c0ec9dfce   httpd    "httpd-foreground"     18 hours ago    Up 17 hours    0.0.0.0:81->80/tcp, [::]:81->80/tcp    app81
ubuntu@ip-10-0-0-36:~$
ubuntu@ip-10-0-0-36:~$

```

Figura 25

Operatividad de contenedores Docker

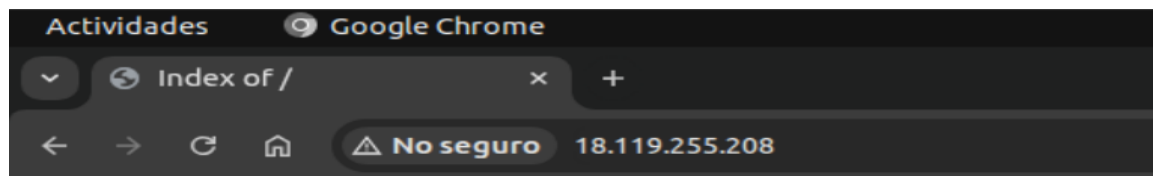


contenedor 82

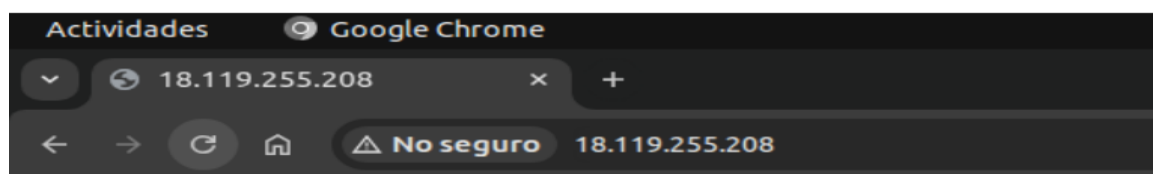
Nota. Se evidencia el funcionamiento en el contenedor con la ip 19.119.255.208 que llama al contenedor con el puerto puerto 82

Figura 26

Operatividad de contenedores Docker puerto 84

**Index of /****Figura 27**

Operatividad de contenedores Docker puerto 81

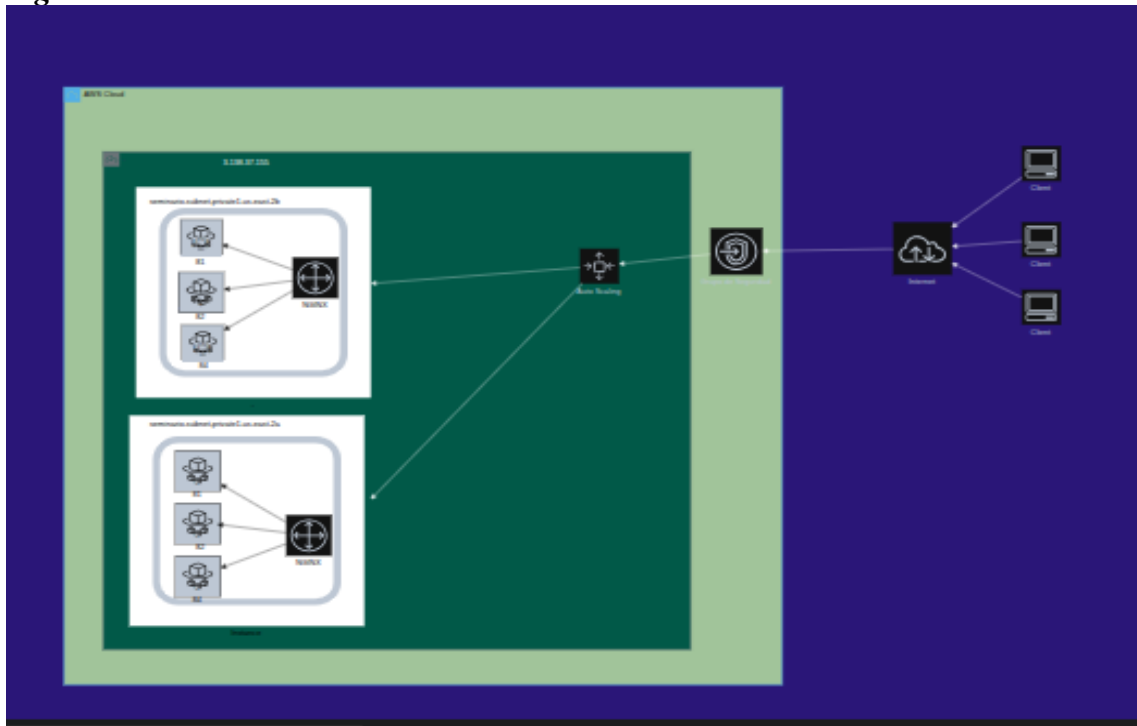


contenedor 81

Nota. En la imagen, se rectifica el funcionamiento del puerto al ingresar el balanceador de cargar distribuirlo entre todos, en este caso el 81.

6. Diagrama de arquitectura

Figura 28



Desarrollo e implementación del aprendizaje

Durante el desarrollo del seminario se han obtenido conocimientos teóricos y prácticos en la implementación de servicios e infraestructura en la nube aplicándolos en entornos reales permitiendo consolidar los conocimientos durante el seminario de AWS.

Se ha emulado la implementación de aplicaciones con una robustez segura, accesible y resistente al fallo.

Inicialmente se crea un aislamiento lógico con control de tráfico con una VPC(Red virtual privada) segmentada por subredes para luego crear las instancias EC2 con sistema Operativo Windows y Linux, seguidamente de la configuración de grupos de seguridad permitiendo el acceso principalmente al Ping y los puertos 22(SSH) y 80(HTTP).

La configuración de Docker ha permitido la gestión de contenedores y el despliegue de aplicaciones utilizando NGINX como proxy inverso llevando a la creación de rutas para el enrutamiento de tráfico por medio del ELB(Balanced Load Controller) hacia las instancias.

Un elemento a destacar ha sido configurar la automatización en las políticas de autoescalado, permitiendo automatizar ante posibles aumentos o disminuciones de cargas la capacidad de CPU su capacidad de respuesta.

Por último, por medio de imágenes se puede demostrar los resultados de los laboratorios realizados, evidenciando la conexión exitosa a la aplicación desde internet a través de los balanceadores de carga, avalando una eficiente distribución del tráfico entre las instancias que estaban disponibles. Así mismo se ha rectificado la capacidad de mantener la disponibilidad sin intervención humana siendo de manera automática.

Figuras y tablas

Figura 1	6
Figura 2	10
Figura 3	11
Figura 4	11
Figura 5	12
Figura 6	13
Figura 7	13
Figura 8	14
Figura 9	14
Figura 10	14
Figura 11	15
Figura 12	16
Figura 13	16
Figura 14	17
Figura 15	17
Figura 16	18
Figura 17	19
Figura 18	20
Figura 19	20
Figura 20	21
Figura 21	21
Figura 22	21
Figura 23	22
Figura 24	23
Figura 25	23
Figura 26	23
Figura 27	23
Figura 28	24

Tabla 1. Reglas de grupo de seguridad

<i>Nombre</i>	<i>Protocolo</i>	<i>Puerto</i>	<i>Origen</i>
<i>Windows RDP</i>	<i>TCP</i>	<i>3389</i>	<i>0.0.0.0/0</i>
<i>Windows HTTP</i>	<i>TCP</i>	<i>80</i>	<i>0.0.0.0/0</i>
<i>Linux HTTP</i>	<i>TCP</i>	<i>80</i>	<i>0.0.0.0/0</i>
<i>Linux SSH</i>	<i>TCP</i>	<i>22</i>	<i>0.0.0.0/0</i>
<i>Ping</i>	<i>ICMP</i>	<i>N/A</i>	<i>N/A</i>

Tabla 2. Asignación de Ip-públicas y privadas

<i>Instancia</i>	<i>IP pública</i>	<i>IP privada</i>
<i>Windows Server 2016 Core</i>	<i>3.147.2.22</i>	<i>10.0.3.153</i>
<i>Ubuntu Server 24.04 LTS</i>	<i>18.219.119.250</i>	<i>172.31.34.147</i>

Conclusiones

Como resultado del desafío propuesto del diseño y despliegue efectuado de la infraestructura en Amazon Web Services(AWS) queda demostrada la competencia de la plataforma para la puesta en producción sistemas altamente escalables, seguros y robustos en la nube. La arquitectura implementada optimiza la disponibilidad, rendimiento y eficiencia al escalar recursos solo cuando son requeridos.

En el proyecto se ha permitido integrar conocimientos técnicos y teóricos como la utilización y despliegue de instancias EC2 Windows y Linux, puesta en operación balanceadores de carga, contenedores Docker, además un Proxy inverso con la aplicación de NGINX lo que demuestra la flexibilidad de la plataforma para soportar diferentes arquitecturas de aplicación.

AWS es una herramienta versátil y potente para la operación de servicios de TI resilientes y modernos con una base sólida para afrontar los desafíos más grandes en términos de computación en nube.

Referencias

Amazon Elastic Compute Cloud Documentation. (n.d.). AWS Documentation.

Retrieved July 20, 2025, from <https://docs.aws.amazon.com/ec2/>

Amazon web service. (n.d.). *Default security groups.* guía de seguridad. Retrieved

07 20, 2025, from https://docs.aws.amazon.com/es_es/vpc/latest/userguide/vpc-security-groups.html

Computación en la nube segura y de tamaño ajustable - Amazon EC2 - Amazon

Web Services. (n.d.). AWS. Retrieved July 20, 2025, from

<https://aws.amazon.com/es/ec2/>

Elastic Load Balancing Documentation. (n.d.). AWS Documentation. Retrieved

July 20, 2025, from <https://docs.aws.amazon.com/elasticloadbalancing/>

Escalado automático de su servicio de Amazon ECS - Amazon Elastic Container

Service. (n.d.). AWS Documentation. Retrieved July 20, 2025, from

https://docs.aws.amazon.com/es_es/AmazonECS/latest/developerguide/service-auto-scaling.html

Guides. (n.d.). Docker Docs. Retrieved July 20, 2025, from

<https://docs.docker.com/guides/>

Guides. (n.d.). Docker Docs. Retrieved July 20, 2025, from

<https://docs.docker.com/guides/>

Install Docker Engine on Ubuntu. (n.d.). Docker Docs. Retrieved July 20, 2025,

from <https://docs.docker.com/engine/install/ubuntu/>

Microsoft. (2025, July 13). *ping* | *Microsoft Learn*. Learn Microsoft. Retrieved July 20, 2025, from <https://learn.microsoft.com/es-es/windows-server/administration/windows-commands/ping>

Mikołajczyk, N. (n.d.). *Install and configure Nginx*. Ubuntu. Retrieved July 20, 2025, from <https://ubuntu.com/tutorials/install-and-configure-nginx#2-installing-nginx>

OpenSSH server - Ubuntu Server documentation. (n.d.). Ubuntu documentation. Retrieved July 20, 2025, from <https://documentation.ubuntu.com/server/how-to/security/openssh-server/index.html>

¿Qué es Amazon EC2? - Amazon Elastic Compute Cloud. (n.d.). AWS Documentation. Retrieved July 20, 2025, from https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/concepts.html

¿Qué es Amazon VPC? - Amazon Virtual Private Cloud. (n.d.). AWS Documentation. Retrieved July 20, 2025, from https://docs.aws.amazon.com/es_es/vpc/latest/userguide/what-is-amazon-vpc.html

Ruiz, P. (2017, May 10). *Uso del comando ping en Ubuntu*. SomeBooks.es -. Retrieved July 20, 2025, from <https://somebooks.es/uso-del-comando-ping-ubuntu/>

SSH en Terminal Windows | *Microsoft Learn*. (2024, May 30). Learn Microsoft. Retrieved July 20, 2025, from <https://learn.microsoft.com/es-es/windows/terminal/tutorials/ssh>

What is a reverse proxy? | Proxy servers explained. (n.d.). Cloudflare. Retrieved July 20, 2025, from <https://www.cloudflare.com/learning/cdn/glossary/reverse-proxy/>

Windows Server documentation | Microsoft Learn. (n.d.). Learn Microsoft. Retrieved July 20, 2025, from <https://learn.microsoft.com/en-us/windows-server/>