

Universidad San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ciencias y Sistemas  
Redes de Computadoras 2  
Aux. Adriana Marié Gómez Dávila



# Manual Tecnico

Proyecto Fase 2

Grupo 10

## **Glosario:**

**Load Balancer:** El load balancing o balanceo de carga es una tecnología diseñada para distribuir la carga de trabajo entre distintos servidores o aplicaciones. El objetivo es optimizar las prestaciones globales de la infraestructura, así como su rendimiento y su capacidad.

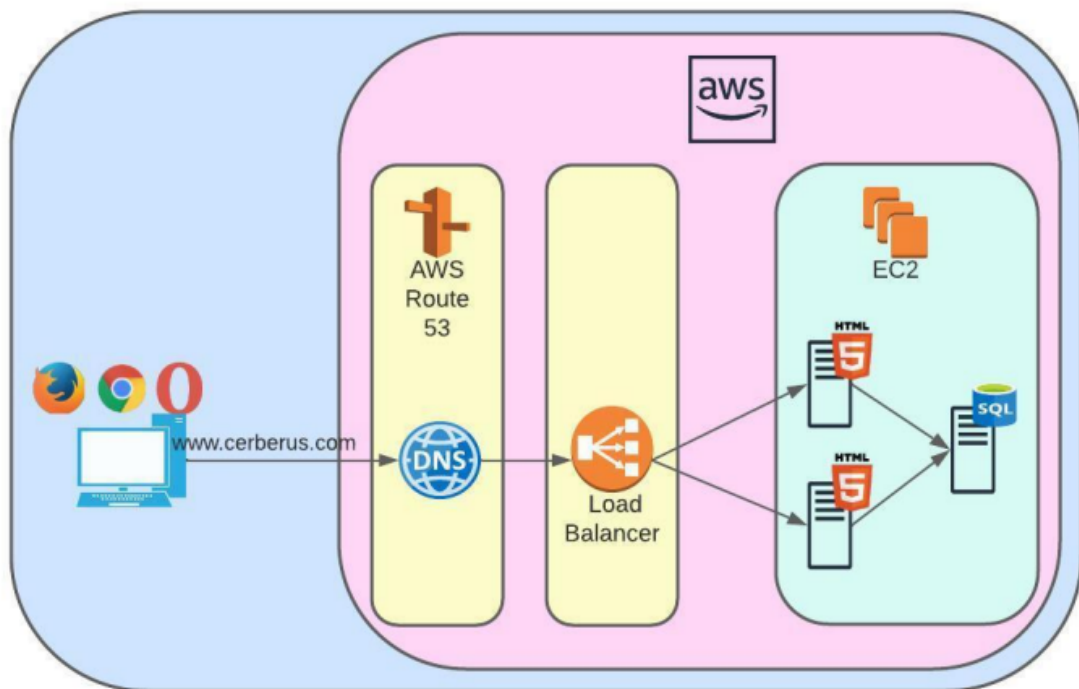
**DNS:** El sistema de nombres de dominio (DNS) es el directorio telefónico de Internet. Las personas acceden a la información en línea a través de nombres de dominio como nytimes.com o espn.com. Los navegadores web interactúan mediante direcciones de Protocolo de Internet (IP). El DNS traduce los nombres de dominio a direcciones IP para que los navegadores puedan cargar los recursos de Internet.

**Registro NS de DNS:** NS significa "servidor de nombres". El registro del servidor de nombres indica qué servidor DNS es autoritativo para un dominio. Los registros NS indican a Internet a dónde ir para buscar la dirección IP de un dominio. A menudo, el dominio tendrá múltiples registros NS que pueden indicar servidores de nombres primarios y auxiliares de ese dominio.

**Registro SOA de DNS:** El registro de "inicio de autoridad" (SOA) de DNS almacena información importante sobre un dominio o una zona, como la dirección de correo electrónico del administrador, cuándo se actualizó el dominio por última vez y cuánto tiempo debe esperar el servidor entre actualizaciones.

**Registro A de DNS:** Un registro A de DNS contiene la dirección IP del equipo anfitrión (o nombre de host) que desea asociar a su nombre de dominio.

## Arquitectura:

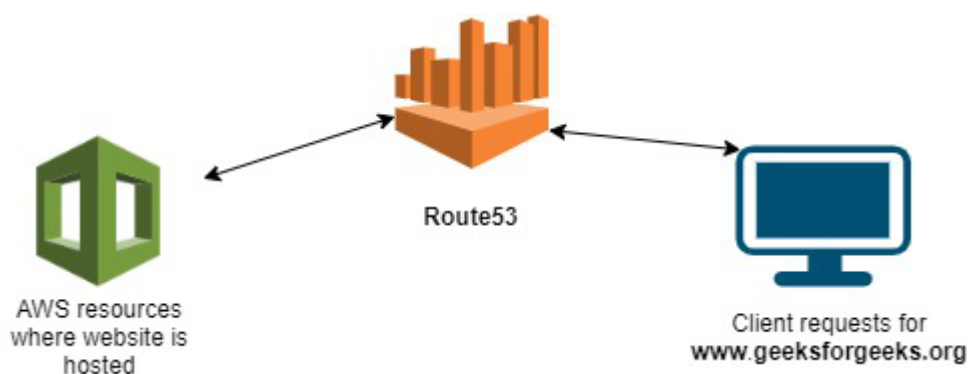


## Dominio:

El dominio de la plataforma fue adquirido por medio de namecheap teniendo `cerberus-grupo10.xyz` como la dirección para acceder a la aplicación web. Para su configuración se ingresaron los servidores del Registro NS de DNS obtenidos de Amazon Route 53 ingresando los nameserver.

## Amazon Route 53:

Amazon Route 53 es un servicio web de sistema de nombres de dominio (DNS) escalable y de alta disponibilidad. Route 53 conecta las solicitudes de los usuarios con las aplicaciones de Internet que se ejecutan en AWS o en las instalaciones.



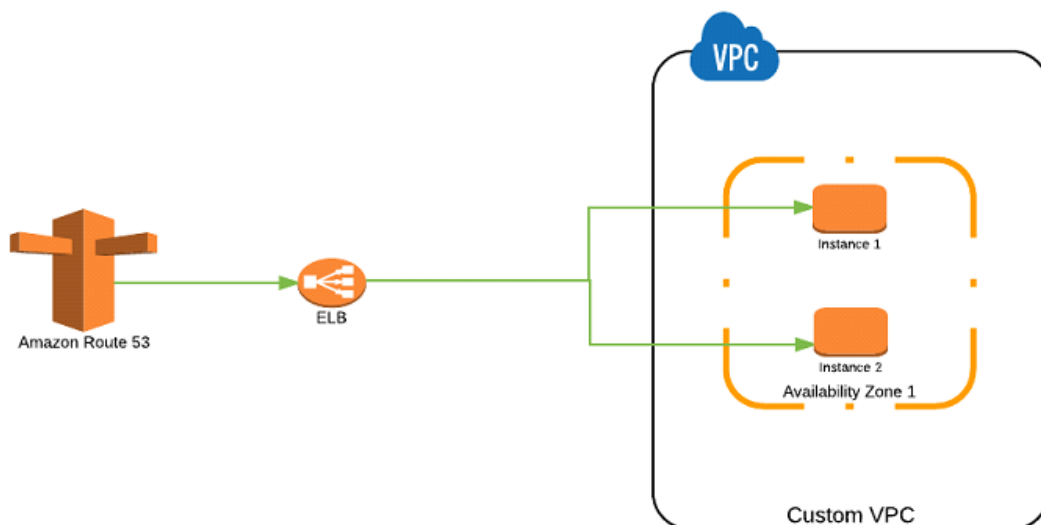
Para la configuración del dominio DNS dentro de Amazon Route 53 se configuró un administrador de DNS con el dominio obtenido por namecheap. Con los siguientes Registros:

Nombre del registro ▼	Tipo ▲
cerberus-grupo10.xyz	A
www.cerberus-grupo10.xyz	A
cerberus-grupo10.xyz	NS
cerberus-grupo10.xyz	SOA

Donde:

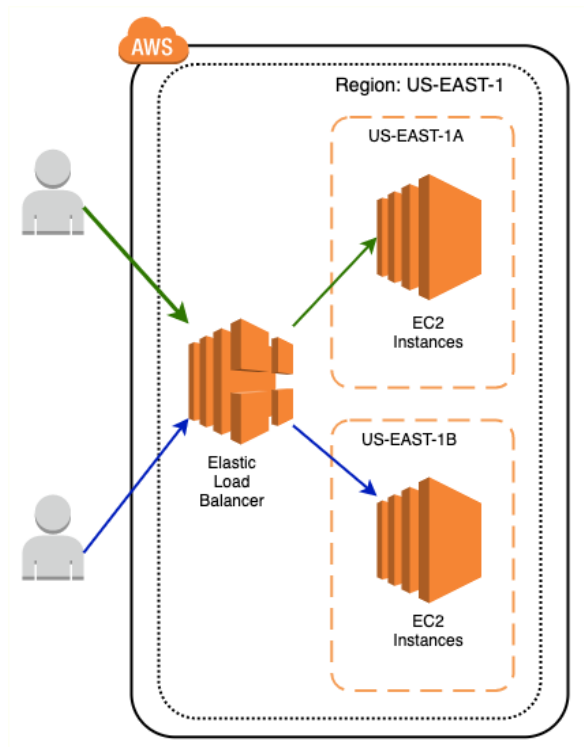
El registro NS fue utilizado en namecheap para redirigir las peticiones del dominio al Administrador de DNS en AWS.

Los Registros A fueron utilizados para que el tráfico recibido se dirija al Elastic Load Balancer implementado para la obtención de la página web.



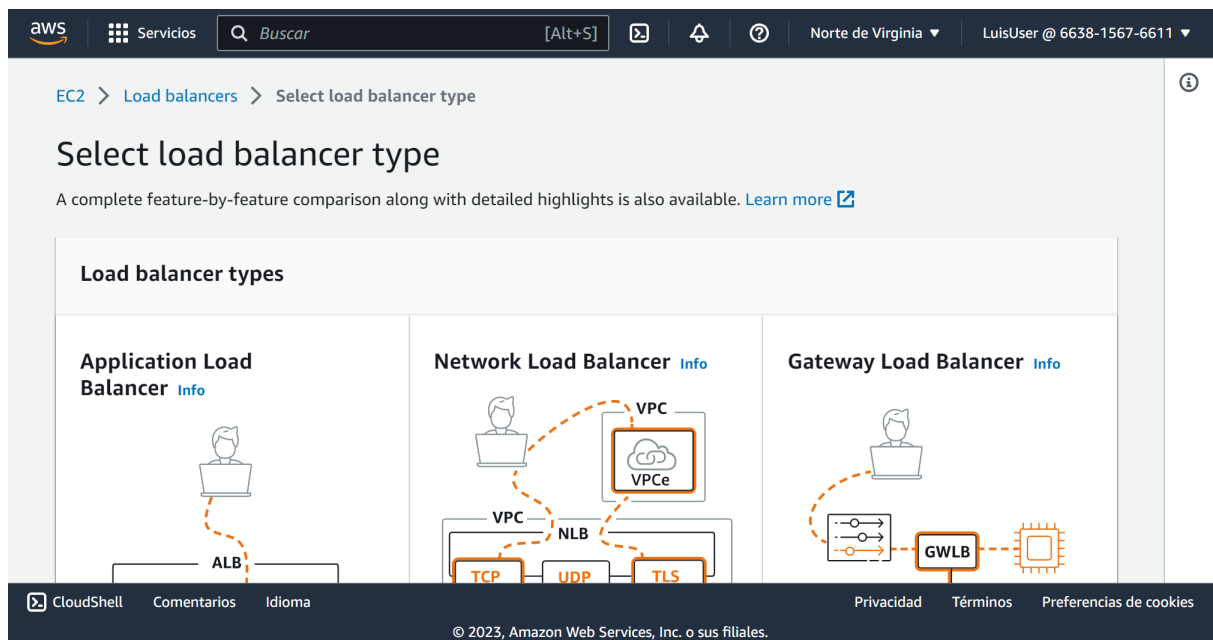
### Elastic Load Balancing:

Servicio que distribuye automáticamente el tráfico de red entre varias instancias de Amazon EC2 (Elastic Compute Cloud) para mejorar la disponibilidad, la escalabilidad y la resistencia de una aplicación.



Configuración:



- dirigirse a la sección de EC2 de la consola de AWS.
- seleccionar Balanceadores de Carga.
- seleccionar Create load balancer






- se selecciona el tipo de Balanceador que necesitaremos en este caso será un Application Load Balancer
- Se ingresa el nombre que le pondremos
- La VPC a utilizar
- Las zonas de Disponibilidad por lo menos 2

- El security group a Utilizar
- El puerto de escucha en este caso sera el "80"
- Seleccionar el Target Group
- Dar click en Crear

**Load balancers (1)**  
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

 **Actions**  **Create load balancer**

 *Filter by property or value*

<input type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones
<input type="checkbox"/>	PruebaRedes2	 PruebaRedes2-498795263...	 Active	vpc-0c666499ccdabb994	<a href="#">2 Availability Zones</a>

## EC2:

Grupos de Seguridad:

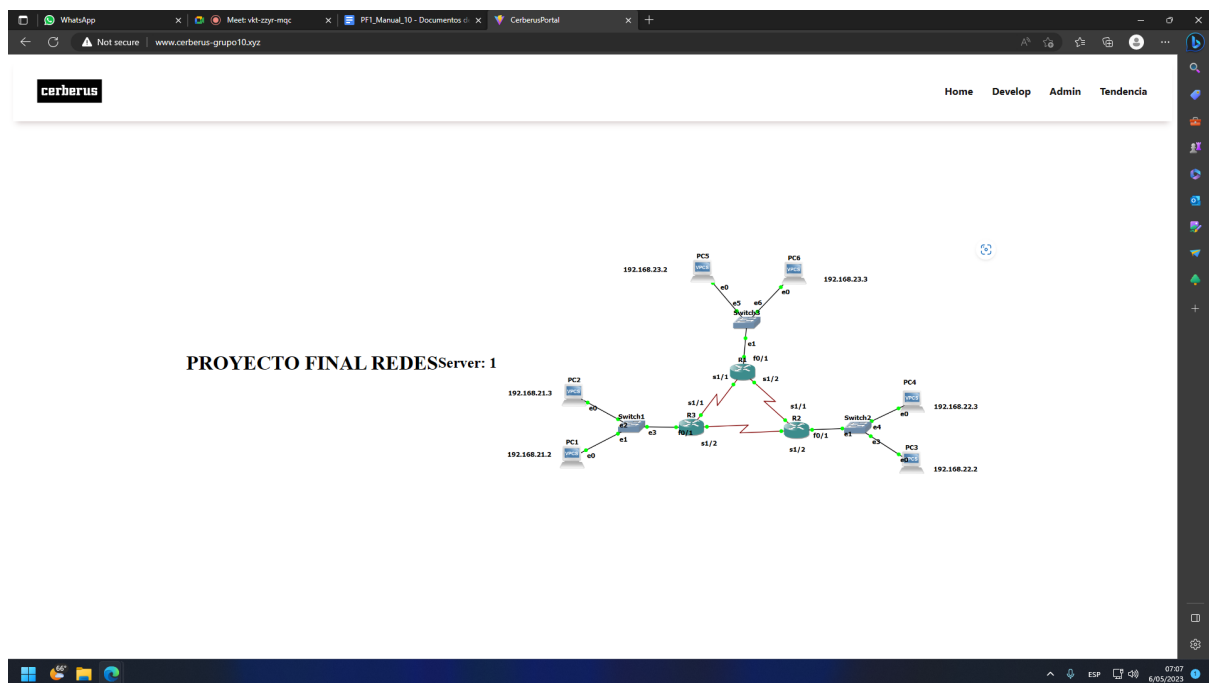
- EC2 Base de datos:

Type	Info	Protocol	Info	Port range	Info	Source	Info	Description
All ICMP - IPv4		ICMP		All		Custom		ping sv2-db
							3.237.182.173/32	
MYSQL/Aurora		TCP		3306		Custom		mongo sv2
							3.237.182.173/32	
MYSQL/Aurora		TCP		3306		Custom		mongo sv1
							54.82.51.49/32	
SSH		TCP		22		Custom		ssh
							190.56.117.56/32	
All ICMP - IPv4		ICMP		All		Custom		ping sv1-db
							54.82.51.49/32	

- EC2 Aplicación:

Type	Protocol	Port range	Source	Description
Custom TCP	TCP	5000	Custom	api port
SSH	TCP	22	Custom	ip Ozmar
HTTP	TCP	80	Custom	http port
HTTPS	TCP	443	Custom	https port
All ICMP - IPv4	ICMP	All	Custom	ping

## Aplicación:



## Base de datos:

Se decidió utilizar MySQL 8 como base de datos por la facilidad que se tiene para desplegar una instancia funcional y empezar a utilizarla con rapidez.

Se desplegó utilizando la [imagen oficial de docker hub](#) y con docker compose:

```

13 lines (13 sloc) | 225 Bytes

1  version: '3'
2  services:
3    base:
4      image: "mysql"
5      container_name: 'MYSQL_Base'
6      environment:
7        MYSQL_ROOT_PASSWORD: mysql123
8      volumes:
9        - base:/var/lib/mysql
10     ports:
11       - "3306:3306"
12  volumes:
13    base:

```

docker-compose.yml

El comando para desplegar un container de docker con esta configuración es  
> docker compose up -d

El puerto a utilizar es el default, TCP 3306, y el acceso se limitará desde los grupos de seguridad de AWS, para la ip del estudiante Ozmar Escobar, y para las 2 máquinas de cerberus que contienen el front-end para que las aplicaciones puedan consumir la información de la DB.

Las sentencias DDL de la DB son las siguientes:

```

16 lines (14 sloc) | 414 Bytes

1  CREATE DATABASE CerberusDB;
2  USE CerberusDB;
3
4  CREATE TABLE `CerberusDB`.`Administrador` (
5    `Id` INT NOT NULL,
6    `Nombre` VARCHAR(100) NULL,
7    `Ocupacion` VARCHAR(100) NULL,
8    `Descripcion` VARCHAR(500) NULL,
9    PRIMARY KEY (`Id`));
10
11 CREATE TABLE `CerberusDB`.`Developer` (
12   `Id` INT NOT NULL,
13   `Nombre` VARCHAR(100) NULL,
14   `Ocupacion` VARCHAR(100) NULL,
15   `Descripcion` VARCHAR(500) NULL,
16   PRIMARY KEY (`Id`));

```



