ANÁLISIS Y DISEÑO – PROYECTO 2

JUAN CAMILO GUERRERO ALARCÓN
DANIEL GARCÍA GARCÍA
CÉSAR ANDRÉS GARCÍA POSADA

TÓPICOS ESPECIALES EN TELEMÁTICA
UNIVERSIDAD EAFIT
MEDELLÍN
2021-1

ROLES Y RESPONSABILIDADES

Juan Camilo (jcguerrera@eafit.edu.co): Encargado de asegurar la disponibilidad del sistema desde el diseño hasta la implementación y seguridad del sistema.

César (cagarciap@eafit.edu.co): Encargado de apoyar en la implementación de los componentes que garanticen la disponibilidad y seguridad del sistema.

Daniel (dgarciag@eafit.edu.co): Encargado de garantizar la seguridad del sistema desde el diseño hasta la implementación y disponibilidad del sistema.

GITHUB

https://github.com/cagarciap/practicaTelematica2

REQUISITOS NO FUNCIONALES

1. El uso de la CPU de las instancias web no debe superar el 60% y en caso de que el uso supere este valor, se deberá crear otra instancia para disminuir la saturación.
2. El sistema debe tener una disponibilidad del 99.99%
3. El sistema debe desplegar como máximo 3 instancias cuando la CPU supere el 60% de uso con las instancias creadas.
4. El sistema debe desplegar como mínimo 2 instancias cuando la CPU esté por debajo del 60% de uso con las instancias creadas.
5. El tiempo máximo de respuesta del sistema debe ser máximo de 2 segundos.
6. El sistema debe soportar 20000 usuarios activos.
7. El sistema implementará un certificado SSL para garantizar la seguridad del mismo.
8. Todos los usuarios administradores deberán iniciar sesión para tener acceso al sistema como administrador.
9. El sistema permitirá a todos los usuarios el inicio de sesión con la plataforma Google, de manera que se utilice el concepto de Single Sign on (SSO).
10. El sistema contará con una infraestructura robusta, que cuente con los componentes necesarios proporcionados por AWS, para garantizar la seguridad a todos los usuarios que hagan uso de él.

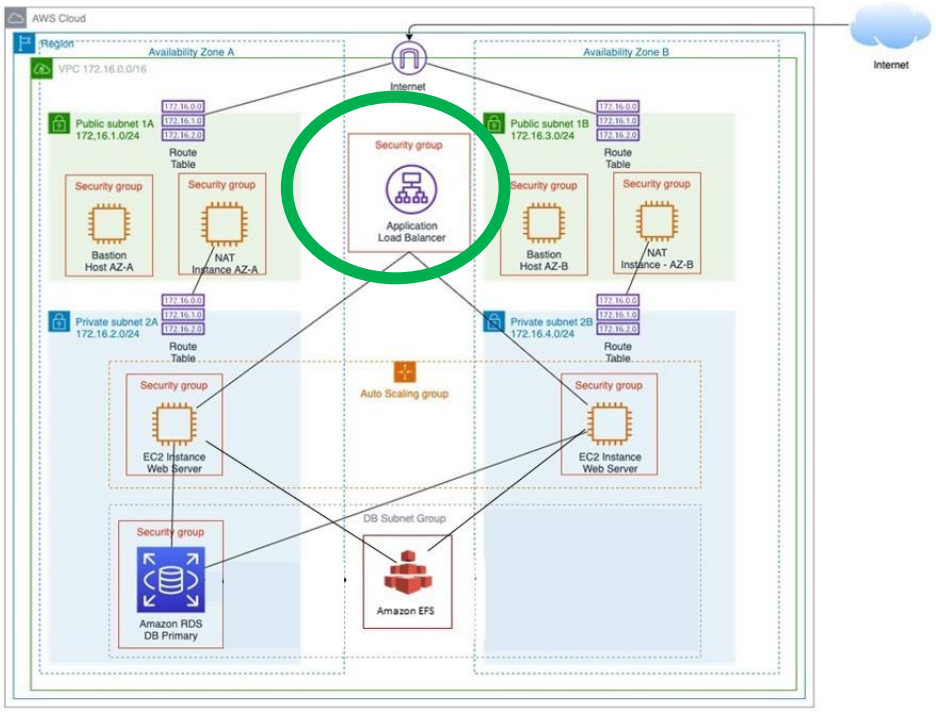
DISEÑO PARA LA ESCALABILIDAD

ALTA DISPONIBILIDAD

En el requisito no funcional de disponibilidad se tienen los siguientes elementos:

1. **Balanceador de carga**

Un balanceador de carga es una herramienta, la cual direcciona a un cliente al servidor web que se encuentre con mayor disponibilidad entre los que cuentan con el mismo contenido.



En este caso el balanceador de carga direcciona a un cliente para el NAT de la región que tenga mayor disponibilidad, es decir, para el NAT de la región A o para el NAT de la región B. Cuando se habla de disponibilidad en un servidor web, es aquel atributo el cual permite que el servicio se pueda ser consumido ininterrumpidamente y de la mejor manera. De esta forma el balanceador de carga evaluara el estado de cada una de las instancias de los servidores con el objetivo de determinar cuál de estas puede proporcionar un mejor hosting al usuario que desea acceder al servicio.

Una vez el balanceador de carga, determina cuál de los dos NATs está disponible, este (el NAT seleccionado) procede a recibir la petición y/o solicitud del cliente.

A su vez el balanceador de carga permite una distribución de las peticiones con el objetivo de que el sistema no tenga una carga mayor que pueda generar una caída del sistema.

- **Crecimiento horizontal**

Para el crecimiento horizontal se tiene un servicio de Auto Scaling, este servicio se usa en AWS por medio de las imágenes de AMI, estas imágenes permiten crear una réplica de las instancias del servidor web, estas instancias tienen las mismas características tanto a nivel de hardware y software. El sistema está en la capacidad de crear tantas instancias como sea necesario, por este motivo se requiere determinar la cantidad máxima de instancias a crear.

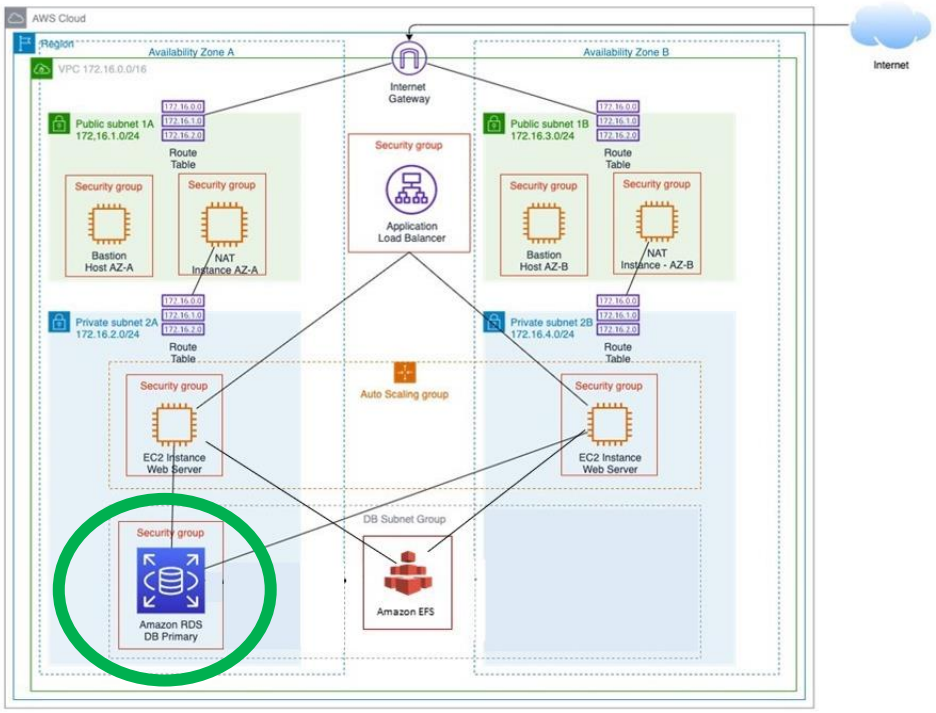
Estas instancias se utilizan en caso de que una instancia, se caiga y con el sistema de archivos se logra tener una trazabilidad de los datos.

Las instancias se van creando y disminuyendo a medida que crecen o disminuyen las peticiones al sistema.

- **Disponibilidad en la capa de persistencia de datos para el manejo de archivos y disponibilidad en la capa de bases de datos para el manejo de la información almacenada en el motor definido para esto**

Otra parte fundamental a la hora de tocar la disponibilidad de un sistema es la gestión de los archivos compartidos, este sistema se denomina EFS (Elastic File System). Este sistema permite compartir todos los archivos de la aplicación y permite el manejo de la concurrencia de estos. De igual forma permite compartir los archivos estáticos que se manejan en un servidor CMS (Sistema de gestión de contenidos) tales como imágenes, PDFs, videos, entre otros.

Y por otro lado se tiene el servicio RDS (Relational Database Service) el cual es el encargado de la gestión y la administración de la base de datos. De igual forma permite manejar la concurrencia de la base de datos para tener una disponibilidad en ambas regiones del sistema.



SEGURIDAD

Para la implementación de nuestro proyecto, utilizaremos una serie de componentes que nos proporciona AWS para garantizar la seguridad de nuestro sistema. Estos elementos que nos van a garantizar específicamente la seguridad hacen parte de la infraestructura de nuestra aplicación y son los siguientes:

VPC: La parte más exterior de nuestra infraestructura, es la nube virtual privada, podemos observar que la tenemos creada en la Ilustración 3, está nube virtual privada es de gran importancia para tener mayor control, privacidad y seguridad, ya que nos permite conectarnos como red privada virtual protegiendo nuestros datos y permitiendo también controlar los aspectos de nuestro entorno, como la selección del propio rango de direcciones IP, creación de subredes, configuración de tablas de enrutamiento y los gateways. En esta VPC podemos utilizar direcciones IP tanto versión 4 como versión 6 (IPv4 e IPv6 respectivamente).

Grupos de seguridad: Un grupo de seguridad lo que nos permite es tener un tipo de firewall virtual para controlar el tráfico saliente y entrante de nuestras instancias de AWS, por ende, cuando añadimos reglas lo que estamos haciendo es permitir un tráfico específico ya sea entrante o saliente de nuestras instancias, garantizando de esta manera la seguridad en el tráfico.

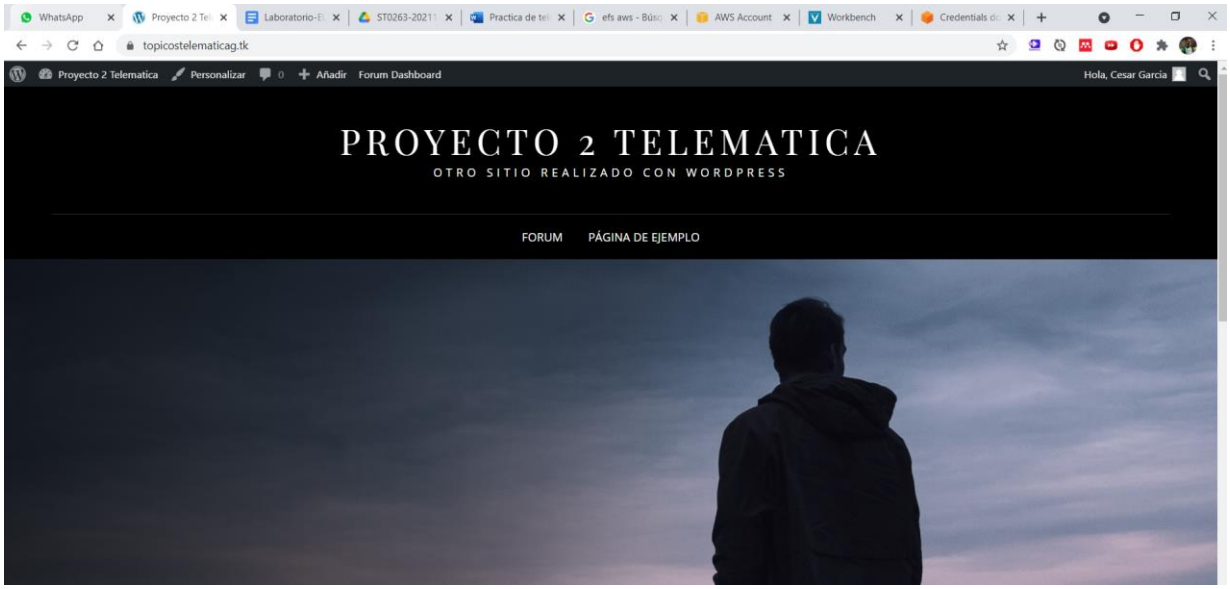
Subnets: Las subnets son redes las cuales provienen de la división de una red mayor, teniendo entonces un rango de direcciones, esto nos sirve para tener una comunicación más dirigida, para reducir los dominios broadcast y para reducir las tablas de enrutamiento. Al tener subnets públicas y privadas garantizamos la seguridad de nuestro sistema ya que estamos restringiendo el acceso desde y hacia las direcciones de nuestras subnets.

NAT Gateway: Este componente está relacionado directamente con las subnets de las que acabamos de hablar, ya que a través de él es que podemos conectar a internet las instancias de nuestras subnets privadas, de manera que podemos coenctarnos a internet manteniendo la seguridad del sistema.

Bastion Host: Es uno de los elementos más importantes de nuestro sistema, ya que actúa como un puente protector antes de acceder a nuestra red, de manera que recibe directamente los ataques y funciona también como un servidor proxy.

Aparte de los anteriores elementos propios de la nube de AWS, también implementamos otros componentes para darle más seguridad a nuestro sistema, uno de ellos es el Single Sign on y el otro fue utilizar un certificado SSL que es un estándar de seguridad a nivel global que nos permite autenticar nuestro sitio web para evitar sitios web que puedan vulnerar la seguridad del usuario y para la transferencia de datos cifrados entre un navegador y un servidor web.

A continuación, se muestra la ventana principal después de haber implementado todos los componentes como el SSL que nos garantizan la escalabilidad.


WhatsApp
Proyecto 2 Tel
Laboratorio-El
ST0263-2021
Practica de tel
efs aws - Búsc
AWS Account
Workbench
Credentials d
topicostelematicag.tk
Proyecto 2 Telematica
Personalizar
0
Añadir
Forum Dashboard
Hola, Cesar Garcia
PROYECTO 2 TELEMATICA
OTRO SITIO REALIZADO CON WORDPRESS
FORUM
PÁGINA DE EJEMPLO