

ANÁLISIS Y DISEÑO DEL SISTEMA

Por:

- Yhoan Alejandro Guzmán García (yaguzmang@eafit.edu.co)
 - Juan Sebastián Pérez Salazar (jsperezs@eafit.edu.co)
 - Daniel Felipe Gómez Martínez (dfgomezm@eafit.edu.co)

Profesor:

Edwin Nelson Montoya Munera

17/05/2021

Universidad EAFIT

Requisitos no funcionales

1. El sistema debe tener una disponibilidad del 99.9999%.
2. El uso de la CPU de las instancias web no debe superar el 60% y en caso de que el uso sea mayor a esta cifra se crea otra instancia para disminuir la saturación.
3. El sistema debe desplegar como máximo 3 instancias cuando se reporte el uso del 60% de las instancias creadas.
4. El sistema debe desplegar como mínimo 2 instancias cuando se reporte el uso inferior al 60% de las instancias creadas.
5. El tiempo máximo de respuesta es de 2 segundos.
6. Las páginas web deben tener certificado de seguridad SSL.

Arquitectura del proyecto

Como se puede ver en la siguiente imagen, la arquitectura propuesta para el despliegue de nuestro proyecto se caracteriza por tres requisitos no funcionales (escalabilidad, rendimiento y seguridad), cada uno de estos requisitos serán explicados de forma más específica en la continuidad de este artículo.

Antes de empezar con la explicación de cada uno de los requisitos no funcionales del sistema, cabe mencionar que a pesar de que en la imagen describe de forma explícita cada uno de los componentes del sistema en materia de hardware, es propio recalcar la importancia de servicios y herramientas de software por el cual no solo es fundamental para la comunicación en el sistema, sino que además ayudan a que el sistema tenga un mejor robustez, escalabilidad y seguridad.

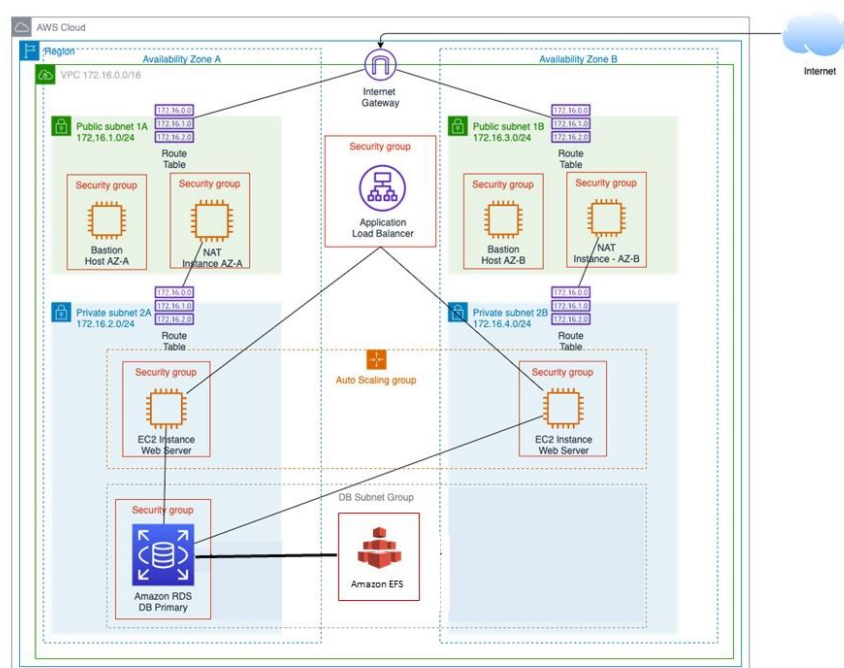


Imagen 1 – Arquitectura general del sistema

Seguridad

En el aspecto de seguridad se tuvieron en cuenta distintos aspectos. Se gestionó la implementación de un certificado SSL a través de cloudflare con dominio de Freenom, para autenticar la identidad del sitio web, garantizando a los visitantes que no están en un sitio falso y para cifrar la información transmitida.

En cuanto a seguridad a través de la arquitectura, se implementó una VPC, con subredes públicas y privadas para sectorizar los accesos a las instancias. Además, se implementó un BastionHost para que el acceso a las instancias sea seguro. Además, por cada una de las instancias desplegadas en el sistema (NAT, BastionHost, página web y base de datos) se crearon grupos de seguridad independientes, en los cuales se especifican los puertos de entrada y salida de cada instancia.

En cuanto a la seguridad en el acceso al CMS, se implementó una autenticación de dos factores haciendo uso de Google Authenticator, y un Single Sign On con un login con la cuenta de google.

Disponibilidad

Como se puede ver en la imagen 1, la arquitectura cuenta con un balanceador de cargas. A través de este servicio administrado por AWS se cuenta con un servicio de monitoreo y balance de peticiones que se le genera a una instancia, por lo cual este servicio es fundamental para solucionar problemas de disponibilidad y rendimiento, porque por el lado de la disponibilidad este servicio en nuestro sistema está distribuyendo las peticiones a través de dos instancias que están ubicadas en dos zonas distintas y por el lado del rendimiento este distribuye las peticiones de tal forma que no se presente una sobre carga en alguna de las instancias y que esta rebose los tiempos de respuestas (generando una posible caída del sistema).

Por otro lado, el sistema cuenta con el servicio de auto scaling group el cual se encarga del crecimiento horizontal del sistema, ese servicio bien acompañado con un recurso de AWS el cual se llama AMI, uno a través de este recurso crea una imagen de las instancias, la funciona como un molde para crear nuevas instancias con las mismas características a nivel de hardware y con las mismas configuraciones a nivel de software. Este servicio de auto escalabilidad replica la cantidad de instancias a nivel que el sistema necesite más instancias de procesamiento y despliegue de la página web, por esta razón en el momento de configurar este servicio es fundamental asignarle una cantidad máxima y mínima de estas instancias, las cuales van a ir creando o eliminando a medida que aumente o disminuyan las peticiones al sistema.

Otros servicios que también se utilizaron los servicios administrados EFS (Elastic File System) y RDS (Relational Database Service), por parte del servicio de EFS se alojan y se distribuyen los archivos (datos estáticos y dinámicos) alojados en la base de datos, este servicio es muy útil y necesario para el sistema porque es el encargado de manejar la concurrencia de los datos en las instancias desplegadas en el sistema. Por otro lado, el servicio de RDS proporciona la administración y gestión de la base de datos, por lo que es necesario especificar en su configuración el motor de base de datos.