

	Soluciones GeoEspaciales	Analisis del sistema web - servidor	Fecha de actualización: 03/10/2023
		version: 02	página 1 de 4

RECURSOS Y COMPONENTES (AWS) NECESARIOS PARA LA PUBLICACIÓN DE SERVICIOS

01 – INTRODUCCIÓN

Para la publicación y distribución de software desarrollado por SGE, se han explorado diversas alternativas con las cuales llevar a cabo el objetivo.

Principalmente se analizó en profundidad los recursos de Cloud y Hosting que ofrece AWS, y cómo se podría implementar en base a los distintos entornos del software desarrollado y lograr una mayor escalabilidad cumpliendo los estandares de software actuales.

02 – IMPLEMENTACIÓN EN EL ÁREA FRONT-END

En este apartado el Host y almacenamiento de la aplicación web / Pagina web, se plantea de la siguiente forma:

>Almacenar los archivos (Codigo Fuente, recursos audiovisuales , entre otros) en la nube desde un bucket en : [Amazon S3 – Web Services](#)

>Distribución de la aplicación a la web con Cloud Front el cual permitira que este disponible al publico a traves de un dominio (*) y que este responda a los protocolos de seguridad que se establecen para la proteccion de datos y seguridad: [AWS Cloud Front](#)

Aqui un diagrama de como funcionaria:



*Obtener un dominio desde [AWS Route 53](#) o en caso de querer conservar el dominio anterior Cloud Front ofrece la opción de usar dominios en otros servicios de host

Documento:	Edición:	Revisión:	Aprobación:
SGE_web_server_impl02.pdf	Departamento Operaciones		

	Soluciones GeoEspaciales	Analisis del sistema web - servidor	Fecha de actualización: 03/10/2023
		version: 02	página 2 de 4

03 – GEOPORTAL

Esta aplicación construida sobre OpenLayers puede seguir el mismo camino que el punto anterior siguiendo el diagrama hecho, se almacenaria en un Bucket y luego se distribuiria por medio de Amazon Cloud Front. Esto significa que funcionaria como una web a parte de la que esta en el Front end.

[idea] En caso de querer integrar ambos lados existen diversas maneras, desde una simple redireccion de un lugar a otro, hasta un cambio estructural que permita compartir material embebido y no necesariamente ir de web en web. Claro que este cambio puede significar que requiera de otro servicio para su publicación. Si se piensa en seguir una estructura que responda al protocolo HTTP y permita la transferencia de datos, necesitaría el servicio de [AWS Elastic beanstalk](#) que puede con la compilación de este tipo de microservicios que se construye a partir de Node y javaScript.

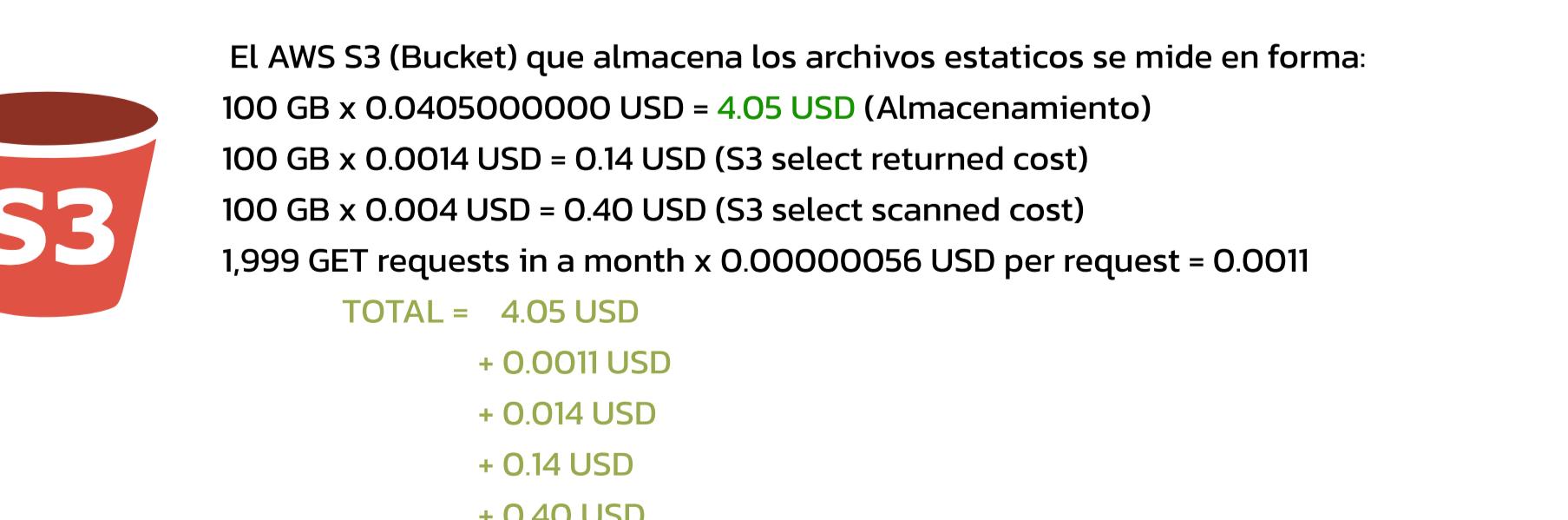
04 – IMPLEMENTACIÓN EN EL SERVICIO DE USUARIOS / WEB (API Rest)

Se plantea un cambio queriendo descartar la máquina virtual debido a su complejidad y al hecho de que aumentaría la complejidad del servicio y prolongaría el tiempo de producción. Por lo tanto, se encontró una versión más simplificada y de bajo costo dentro de AWS que permitirá la distribución de este servicio y su consumo a nivel global por las aplicaciones que lo requieran.

>Para la distribución del servicio, se utilizaría: [AWS Elastic beanstalk](#)

>También se necesitaría otro complemento aparte, que sería la base de datos, la cual estan las opciones: [Azure SQL Database \(Microsoft\)](#) o tambien [AWS RDS for MySQL](#)

La base de datos generara las credenciales necesarias con las cuales se conectara la API en su configuración y asi poder tener un ecosistema que retorne los datos almacenados y sea reutilizable en diversos proyectos, conservando los mismos generados y todo lo que se suba a esta base de datos (archivos / componentes / embeduras / etc).



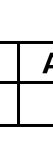
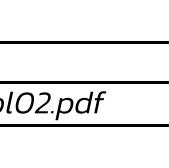
Documento:	Edición:	Revisión:	Aprobación:
SGE_web_server_impl02.pdf	Departamento Operaciones		

05 – COSTOS

El costo de hacer este sistema web es relativo, debido a que en servicios como AWS elastic BeanStalk no tiene una tarifa definida, sino que solo se paga por los recursos de AWS que utiliza realmente para almacenar y ejecutar la aplicación.

Base de datos (Relacional):

La selección de una instancia de base de datos se hizo a partir un requerimiento minimo para la gestión de usuarios y otras operaciones, las opciones y precios son:



AWS RDS for MySQL

Características:

- db.t2.micro => (nombre de instancia)
- vCPU: 1
- Memory: 1 GiB | Storage: 10 GB

Esta opción es de uso moderado para volúmenes no tan grandes de personas que soliciten los datos, sin embargo es funcional y completa si tratamos los requisitos actuales.

*No incluye impuestos

27.74 USD

[VER OPCIÓN](#)

Azure SQL database

Características:

- Tipo: Single Database | Model DTU Standard
- Performance: 10 DTU's
- Storage: 250 GB

Tambien de uso moderado, con una distribución de recursos distinta comparado con aws, conllevo migrar la API a SQL server y testear (baja dificultad), precio sin respaldo de datos.

*No incluye impuestos

18.19 USD

[Ver opción](#)

Cálculo de precio: instancia (AWS)

1 instance(s)
x 0.035 USD hourly
x 730 hours in a month =

(Precio mensual)

Cálculo de precio: Almacenamiento (AWS)

Cantidad de GB
x 0.437 USD
x 1 instances
(precio mensual)

[Cálculos hechos en [AWS Pricing Calculator](#) en la region de South America (Sao Paulo)]

Cálculo de precio: (Microsoft)

17.81 => 1 instancia (Single/DTU)
+ 0.38 (Point-In-Time Restore)
x 1 month

(Precio mensual)

Cálculo de precio: Almacenamiento (Microsoft)

Precio incluido con el nivel de performance

[Cálculos hechos en [Calculadora de precios Microsoft Azure](#) en la region de South America (Sao Paulo)]

Documento:	Edición:	Revisión:	Aprobación:
SGE_web_server_impl02.pdf	Departamento Operaciones		

06 – CONCLUSIÓN

Una vez con todos los servicios publicados se puede esperar un ecosistema como el siguiente:

Documento:	Edición:	Revisión:	Aprobación:
SGE_web_server_impl02.pdf	Departamento Operaciones		

07 – COSTOS

El costo de hacer este sistema web es relativo, debido a que en servicios como AWS elastic BeanStalk no tiene una tarifa definida, sino que solo se paga por los recursos de AWS que utiliza realmente para almacenar y ejecutar la aplicación.

Base de datos (Relacional):

La selección de una instancia de base de datos se hizo a partir un requerimiento minimo para la gestión de usuarios y otras operaciones, las opciones y precios son:

AWS RDS for MySQL

Características:

- db.t2.micro => (nombre de instancia)
- vCPU: 1
- Memory: 1 GiB | Storage: 10 GB

Esta opción es de uso moderado para volúmenes no tan grandes de personas que soliciten los datos, sin embargo es funcional y completa si tratamos los requisitos actuales.

*No incluye impuestos

27.74 USD

[VER OPCIÓN](#)

Azure SQL database

Características:

- Tipo: Single Database | Model DTU Standard
- Performance: 10 DTU's
- Storage: 250 GB

Tambien de uso moderado, con una distribución de recursos distinta comparado con aws, conllevo migrar la API a SQL server y testear (baja dificultad), precio sin respaldo de datos.

*No incluye impuestos

18.19 USD

[Ver opción](#)

Cálculo de precio: instancia (AWS)

1 instance(s)
x 0.035 USD hourly
x 730 hours in a month =

(Precio mensual)

Cálculo de precio: Almacenamiento (AWS)

Cantidad de GB
x 0.437 USD
x 1 instances
(precio mensual)

[Cálculos hechos en [AWS Pricing Calculator](#) en la region de South America (Sao Paulo)]

Cálculo de precio: (Microsoft)

17.81 => 1 instancia (Single/DTU)
+ 0.38 (Point-In-Time Restore)
x 1 month

(Precio mensual)

Cálculo de precio: Almacenamiento (Microsoft)

Precio incluido con el nivel de performance

[Cálculos hechos en [Calculadora de precios Microsoft Azure](#) en la region de South America (Sao Paulo)]

Documento:	Edición:	Revisión:	Aprobación:
SGE_web_server_impl02.pdf	Departamento Operaciones		

08 – CONCLUSION

Una vez con todos los servicios publicados se puede esperar un ecosistema como el siguiente:

Documento:	Edición:	Revisión:	Aprobación: