Documento elaborado por Marcos Hidalgo Baños (GCP) y Alejandro Caro Casado (AWS) como trabajo final de la asignatura Teoría de la Información y Codificación en la Universidad de Málaga.

Google Cloud Platform y Amazon Web Services

Servicios de encriptación y codificación en la nube

Índice de contenidos.

Fundamentos. ¿Cuál es el objetivo del documento?

Nuestra principal meta para la realización de este trabajo es acercar a nuestros compañeros los conceptos propios del desarrollo de aplicaciones en la nube, independientemente de la herramienta en concreto con la que se implemente. La elección de GCP y AWS como elementos de estudio se justifica principalmente por la gran relevancia que ambas empresas tienen en este ámbito, además del propio interés personal por aprender estas tecnologías.

Los dos servicios a presentar son Cloud Key Management Service y Amazon Redshift. Ambos se especializan en la encriptación y codificación (respectivamente) de aquella información sensible, almacenada en las bases de datos, que los desarrolladores alojarán en los servidores de la nube.

Para realizar esta investigación, en primer lugar realizaremos una breve introducción a los conceptos propios de la nube y que pueden ser desconocidos para una parte de la audiencia, para después mostrar el estudio en profundidad de los dos servicios mencionados. Sin embargo, este análisis no será demasiado técnico, puesto que la información disponible es muy superficial y en la mayoría de casos relegan en otros conceptos ajenos a la nube.

Preliminares. ¿Cómo funciona la nube?

Una de las principales ventajas de la nube sobre el almacenamiento local es la capacidad de poder acceder a los recursos informáticos de manera **inmediata** y de forma totalmente **transparente**, en lugar de tener que instalar y administrar directamente las aplicaciones.
Esto se consigue mediante la creación de un **proyecto** al que deberemos asignarle servicios.

Conceptos de servicio y consola.

Todo proyecto en la nube está compuesto por un determinado conjunto de **servicios**. Cada uno de ellos realiza una función concreta dentro del proyecto y deben ser independientes entre sí.

Los servicios son añadidos al proyectos mediante una **consola**, puede ser una interfaz gráfica o interfaz de línea de comandos. Estos pueden clasificarse en secciones o capas donde los servicios pertenecientes a la misma capa introduzcan funcionalidades de un mismo ámbito.
En este estudio, nos centraremos en aquellos pertenecientes a la sección de *Seguridad*.

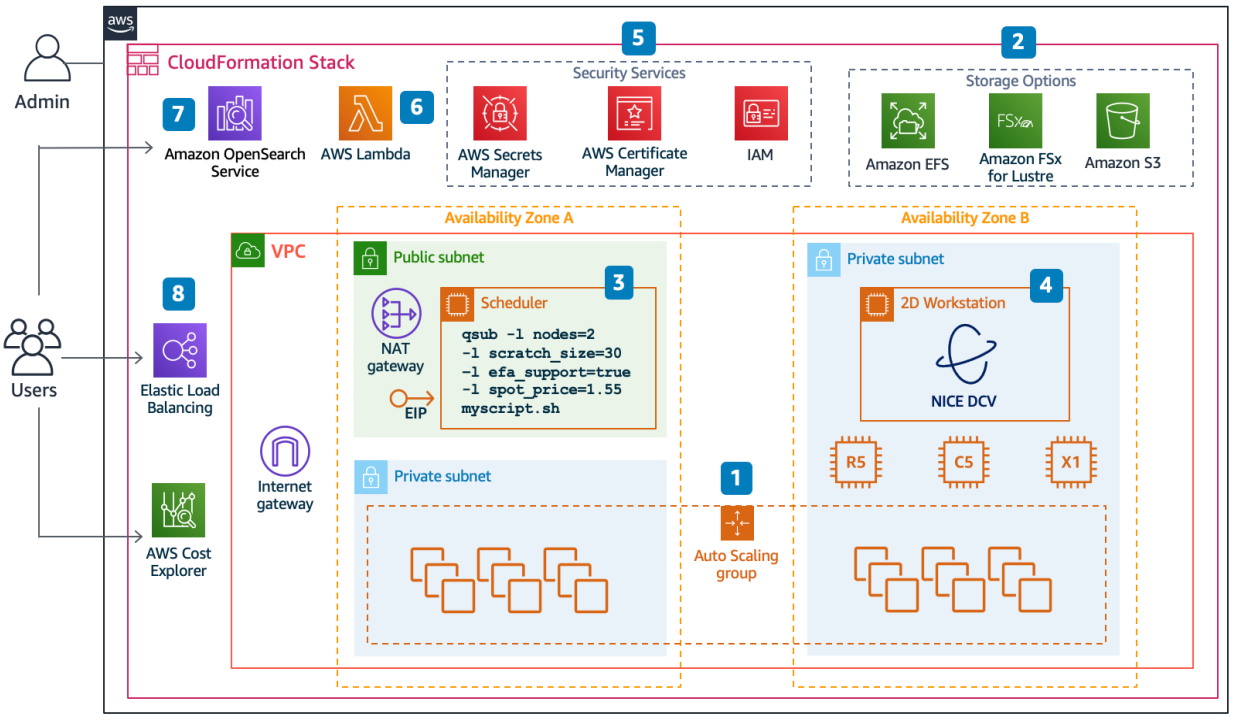

Imagen 1. Ejemplo de un proyecto en la nube de AWS

Caso particular. Google Cloud Platform - GCP

Google es uno de los proveedores de recursos de computación en la nube de mayor relevancia. Sus herramientas se utilizan para desarrollar, implementar y operar aplicaciones en la web.

De todo el abanico de servicios disponibles, destacan **Compute Engine** para la creación de máquinas virtuales, **Cloud Storage** para el almacenamiento de bases de datos, **Cloud SQL** para su administración y la realización de consultas, o **Cloud Run** para el despliegue de aplicaciones.

Cloud Key Management Service (KMS).

Tal y como se cita en la propia descripción del servicio, "Google Cloud KMS permite a los clientes administrar **claves de encriptación** y realizar **operaciones criptográficas** con esas claves".
Es uno de los principales servicios dedicados a la encriptación de material sensible como lo son las claves de encriptación que los usuarios emplearán en sus aplicaciones alojadas en la nube.

Una manera de entender KMS es como la de un **llavero** o Keyring, que almacena las claves de los usuarios y que para poder emplearlas debe ser extraída del conjunto (para no confundirlas). Sin embargo, para conseguir esto debemos pedir permiso al "cerrajero" para que nos la preste durante el tiempo que la estamos utilizando. En esta metáfora, el cerrajero es Google Cloud.

En función del control que este cerrajero tenga sobre el llavero, existen los siguientes servicios.

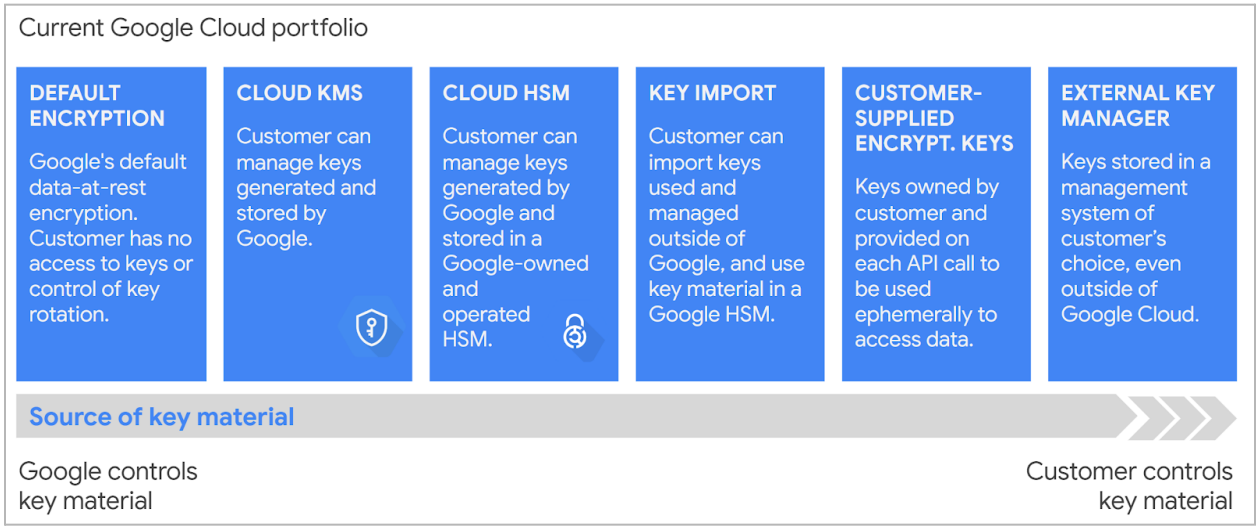

Imagen 2. Opciones para el almacenamiento de las claves que ofrece Google Cloud

Jerarquía de claves de encriptación por defecto en KMS.

Root KMS Master Key

Custodiadas de manera física en los centros seguros de Google, permiten desencriptar y utilizar las KEK. Son extremadamente seguras y el acceso está restringido a unos pocos empleados.

Si no existiera este tipo de claves, cualquier KEK tendría la libertad de crear su propio sistema independiente del resto al no estar controlados por ninguna otra. Esto supondría que dentro de los propios servidores de Google Cloud, habría multitud de KEKs sin posibilidad de administrar, puesto que serían capaces de desencriptar todas sus DEK por sus propios métodos.

Key Encryption Key - KEK

Cada vez que el usuario quiere desencriptar una DEX para utilizarla, debe llamar al servicio KMS para que su KEK correspondiente la desenvuelva. Estas claves están alojadas en el root KMS y deben pedir a su correspondiente Root KMS Master Key que las desenvuelva para poder administrar al conjunto de DEKs que tiene asignado. Así pues, obtienen un mayor control sobre la forma en que los datos se encriptan y cómo se administran las claves de encriptación.

Data Encryption Key - DEK

Son las claves que el usuario usa en sus proyectos para encriptar los datos sensibles de su aplicación. Toda la jerarquía de claves se estructura alrededor de ellas, puesto que en un sistema seguro estas claves deberán estar también cifradas ya que son datos sensibles en sí mismas.

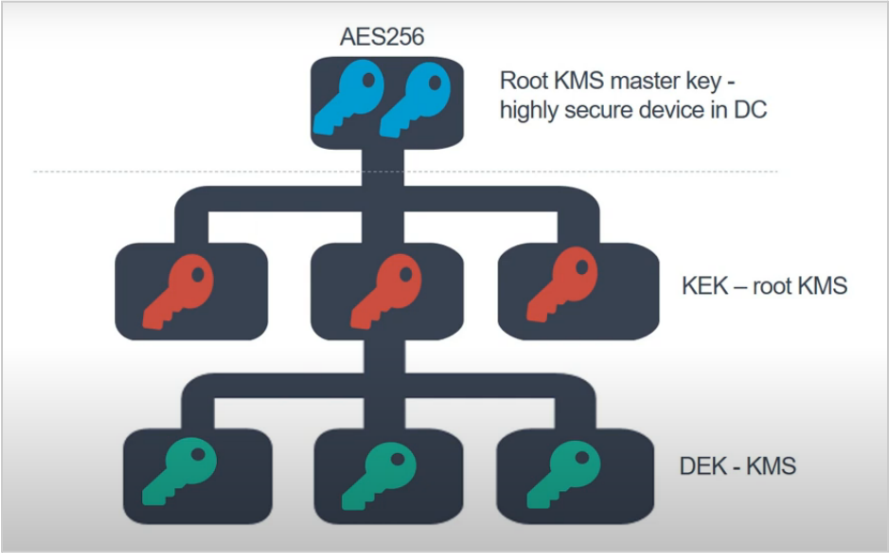

Imagen 3. Esquema de la jerarquía por defecto de KMS

Caso particular. Amazon Web Services - AWS

AWS es una plataforma en la nube que cuenta con una gran cantidad de servicios, más de 200 a nivel global, e incluye secciones de servicios que abarcan ámbitos desde el **almacenamiento** de datos hasta aplicaciones de inteligencia artificial.

Existen algunos servicios capaces de sacarle el máximo partido a la **codificación**, como Amazon Redshift, que introduce el almacenamiento de datos completamente administrado para optimizar el contenido de las columnas utilizando para ello diferentes tipos de codificación según nos convenga. Esta **versatilidad** hace de Redshift uno de los servicios más útiles de AWS.

Amazon Redshift.

Es un servicio de almacenamiento de datos a escala de petabytes totalmente administrativo en la nube, cuenta con columnas altamente **escalable** y performante que se utiliza para análisis de datos a gran escala. Se diseñó específicamente para ser fácil de usar con herramientas de análisis de datos populares como SQL y ofrece una capacidad de almacenamiento y análisis a bajo costo en comparación con otras soluciones de análisis de datos en la nube.

Su principal característica respecto a otros servicios de almacenamiento es el almacenamiento de datos en columnas, ya que a la hora de guardar los datos, Amazon Redshift realiza la comprensión de las columnas mediante el uso de codificación, consiguiendo que se ahorre muchísimo espacio en el almacenamiento de la nube. Esta ventaja es tan grande que hasta otros servicios de Amazon Web Service recurran a Redshift para ahorrar espacio en la nube y poder mejorar su rendimiento frente a la competencia.

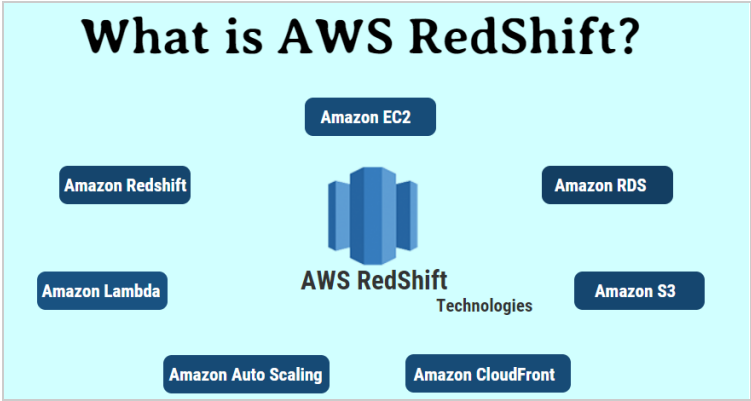

Imagen 4. Servicios que extienden a AWS Redshift

Tipos de codificación empleados por Redshift.

Ya sabemos que se utiliza codificación, pero claro hay distintos tipos de codificación y dependiendo del tipo, se pueden codificar algunos datos y otros no, es destacable que la única codificación que admite todo tipo de datos es la codificación RAW, en la cual los datos se almacenan descomprimidos y sin formato. En la esta tabla se muestra todas las codificaciones:

Tipo de codificación	Tipos de datos
Raw (sin comprimir)	Todos
AZ64	SMALLINT, INTEGER, BIGINT, DECIMAL, DATE, TIMESTAMP, TIMESTAMPTZ
Diccionario de bytes	SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE PRECISION, CHAR, VARCHAR, DATE, TIMESTAMP, TIMESTAMPTZ
Delta	SMALLINT, INT, BIGINT, DATE, TIMESTAMP, DECIMAL INT, BIGINT, DATE, TIMESTAMP, DECIMAL
LZO	SMALLINT, INTEGER, BIGINT, DECIMAL, CHAR, VARCHAR, DATE, TIMESTAMP, TIMESTAMPTZ, SUPER

Tipo de codificación	Tipos de datos
Mostly*n*	SMALLINT, INT, BIGINT, DECIMAL INT, BIGINT, DECIMAL
Run-length	SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE PRECISION, BOOLEAN, CHAR, VARCHAR, DATE, TIMESTAMP, TIMESTAMPTZ
Texto	Solo VARCHAR Solo VARCHAR
Zstandard	SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE PRECISION, BOOLEAN, CHAR, VARCHAR, DATE, TIMESTAMP, TIMESTAMPTZ, SUPER

Imagen 5. Codificaciones en Redshift dependiendo del tipo de datos

A continuación se realizará una breve descripción de cada uno de los elementos descritos en la tabla, comentando sus características principales y medios por los que codifica la información.

Codificación AZ64

Es un algoritmo codificado creado por Amazon que logra una alta relación de compresión y un procesamiento mejorado, comprime grupos de valores de datos más pequeños y utiliza instrucciones SIMD para realizar un procesamiento en paralelo. AZ64 se utiliza sobre todo para ahorrar almacenamiento y conseguir buen rendimiento con datos de tipo numérico, de fecha y de hora.

Codificación por Texto

Tiene dos tipos: text255 y text32k, por su nombre se puede intuir que este tipo de codificación se utiliza para codificar datos tipo VARCHAR. Consiste en crear un diccionario independiente de palabras únicas para cada bloque de los valores de columna.

El diccionario contiene las primeras 245 palabras únicas. Estas palabras se reemplazan en el disco con un valor de índice de un byte que representa uno de los 245 valores, y todas las palabras no representadas en el diccionario se almacenan sin comprimir. Se repite este proceso para cada bloque de 1 MB.

Codificación Delta

Comprime los datos al registrar las diferencias entre los valores que ocurren en una columna. Esta diferencia se escribe en el disco en un diccionario separado para cada bloque de valores de

columna. Tiene dos variaciones:

- DELTA registra las diferencias como valores de 1 byte (enteros de 8 bits)
- DELTA32K registra las diferencias como valores de 2 bytes (enteros de 16 bits)

Codificación Diccionario de Bytes

En este tipo de codificación crea un diccionario independiente de valores únicos para cada bloque de los valores de columna. Este diccionario que se ha creado puede llegar a tener hasta 256 valores de un byte, los cuales se almacenan como índices de los valores de datos originales. Puede llegar a ocurrir que si se almacenan más de 256 valores en el mismo bloque, estos valores sobrantes o adicionales se graban en un bloque descomprimido sin formato. Dicho proceso se va a repetir para cada bloque del disco.

Codificación LZO

Con este tipo de codificación pasa como con la codificación por Texto que se utiliza con las columnas CHAR y VARCHAR sobre los que almacenan cadenas de caracteres muy largas. Funciona particularmente bien con texto de forma libre, como descripciones de productos, reseñas de usuarios o cadenas JSON, ya que proporciona una relación de compresión muy alta con un buen rendimiento.

Codificación Mostlyn

Resulta útil cuando el tipo de datos de la columna es más grande que lo que se necesita para la mayoría de los valores almacenados. Al especificar una codificación principal para este tipo de columna, puede comprimir la mayoría de los valores de la columna en un tamaño de almacenamiento estándar más pequeño. Los valores no comprimibles restantes se mantienen sin procesar.

Codificación Run-length

Reemplaza un valor que se repite de manera consecutiva por un token que consiste en el valor y un recuento de la cantidad de ocurrencias consecutivas. Crea un diccionario independiente de valores únicos para cada bloque de los valores de columna del disco. Este tipo de codificación es una opción ideal para tablas donde los valores de datos a menudo se repiten secuencialmente, como cuando la tabla está ordenada según esos valores.

Codificación Zstandard (ZSTD)

Proporciona una relación de compresión muy alta y un rendimiento muy bueno en diferentes conjuntos de datos. Ocurre algo parecido con la codificación LZO y es que es particularmente adecuada para las columnas CHAR y VARCHAR que almacenan una variedad de cadenas largas y cortas. Aunque los algoritmos como la codificación Delta o la codificación predominante pueden usar más memoria que los datos sin comprimir, es poco probable que ZSTD aumente el uso del disco.

Conclusiones de la investigación.

Para finalizar el proceso de elaboración de este documento y las correspondientes transparencias, realizaremos un repaso de las conclusiones extraídas las cuales clasificaremos según el ámbito general o concreto de los servicios en la nube. Una versión reducida de estas ideas puede ser encontrada en las ya mencionadas diapositivas, alojadas en el repositorio de Github referenciado.

Generales de los servicios en la nube.

- **La nube es una herramienta más que puede emplearse en contextos muy diferentes**. Las diferentes secciones introducidas al comienzo del documento son muestra de la enorme variedad de posibles aplicaciones que emplean la tecnología en la nube.

- **Los servicios en la nube son similares independientemente del proveedor.** Hemos podido comprobar de primera mano lo complejo que resulta encontrar el servicio que mejor satisfaga las necesidades de nuestra aplicación. Esta búsqueda es facilitada mediante la extensa documentación oficial, la cual es de ayuda independientemente del proveedor que ofrezca el servicio. Efectivamente, podemos comprobar que para cada servicio de un proveedor en particular, existe otro equivalente en su alternativa.

- **Los servicios en la nube son versátiles y ayudan a crear otros servicios en la nube.** Tanto con Cloud Key Management Service y Amazon Redshift hemos comprobado que son utilizados en otros esquemas de aplicaciones o servicios de mayor envergadura.

Concretas de los proveedores del servicio.

La administración de datos sensibles en la nube es un asunto de seguridad que no tiene una solución general para todos los casos. Dependiendo de las características propias del sistema, usar un proveedor u otro puede llegar a condicionar el abanico de opciones a elegir:

KMS establece una jerarquía centralizada, pero habilita alternativas menos centralizadas.
La gestión de las claves de los usuarios puede ser realizada dentro de los servidores de Google mediante la jerarquía de claves de KMS o por terceros. Las capas de cifrado de claves ofrecen la seguridad de que ningún indeseado pueda acceder a los datos de la aplicación.

Redshift nos ofrece muchas facilidades gracias a la codificación.
Con este tipo de servicio podemos ver que las posibilidades de la codificación son infinitas y que no solo se tiene que limitar al ámbito de la encriptación, ya que como hemos visto podemos ahorrar almacenamiento y recursos.

Referencias bibliográficas.

La mayor parte de la información ha sido obtenida de la propia documentación que Google Cloud y Amazon Web Services proveen en sus páginas oficiales. Sin embargo, también hemos empleado fuentes de información alternativas como vídeos de Youtube de profesionales.

- cloud.google.com/security-key-management

  cloud.google.com/docs/security/key-management-deep-dive

  Documentación oficial y raíz de otros artículos.

- youtube.com/watch?v=GDECKM9iW0w

  Vídeo sobre la arquitectura de KMS.

- docs.aws.amazon.com

  aws.amazon.com/es/what-is-aws/

  docs.aws.amazon.com/es_es/redshift/latest/dg/welcome.html

  Documentación oficial de AWS y de Redshift.

Enlace al repositorio de Github con los materiales elaborados.

github.com/MarkosHB/Cryptography-in-Cloud

Documento elaborado por Marcos Hidalgo Baños y Alejandro Caro Casado.