

UNIVERSIDAD  
COMPLUTENSE  
DE MADRID



# BigData con Google Cloud

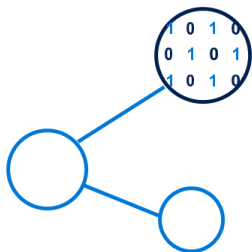
Autor: Pedro Pablo Malagón Amor

Actualizado Enero 2021

# INTRODUCCIÓN.

## Big Data y Cloud Computing:

Tres tendencias están transformando el Mundo



Big Data



Cloud



Computación  
Cognitiva

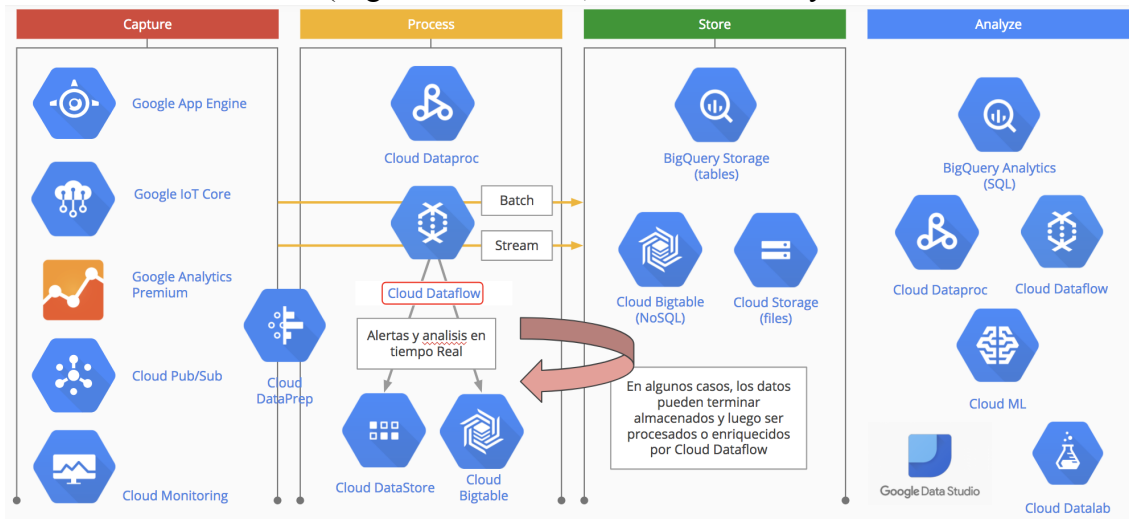
Hoy en día, el mundo que nos rodea es cada vez más cambiante, más rápido que en cualquier otro punto en la historia. Este es un indicador de la cuarta revolución industrial que surge, en gran parte impulsada por la subida de Big Data, el crecimiento de la nube y una nueva era de capacidades de Machine Learning.

Gracias a la proliferación exponencial de procesadores, sensores y chips cada vez más y más pequeños, así como de más bajo coste, las computadoras son omnipresentes, así como el número de personas que los utilizan. Desde los tradicionales ordenadores, pasando por las tabletas, smartphones, sensores y wearables; las máquinas están en todas partes, constantemente creando, recogiendo y dando sentido a los datos en nuestro medio.

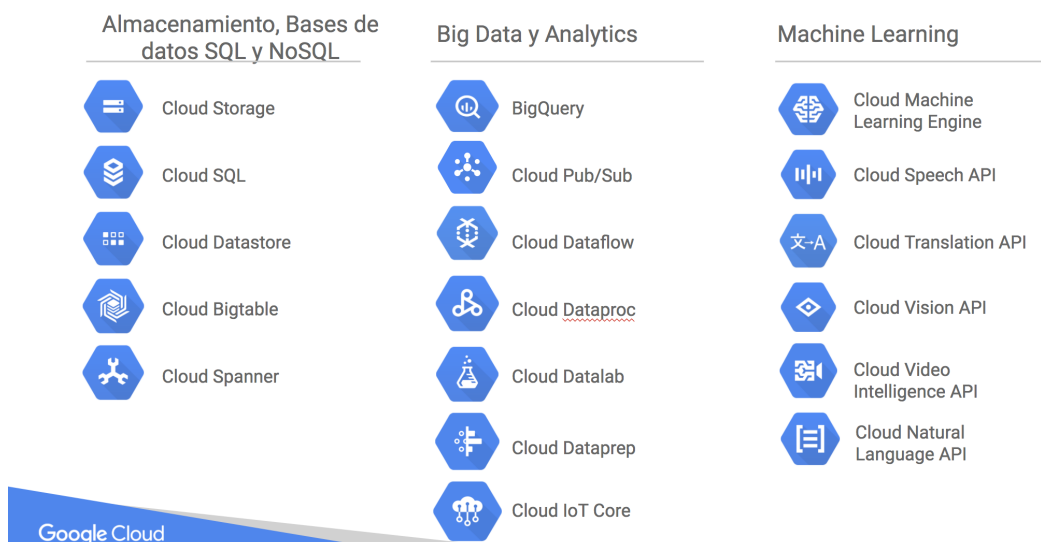
Con la “magia” de Machine Learning y la potencia de cálculo sin límites de la nube, estos datos están dando lugar a poder aumentar las capacidades humanas con la utilización de la computación cognitiva de nuevas formas.

Juntas, estas tecnologías tienen el poder de cambiar casi todas las industrias, razón por la cual esta era está siendo anunciada como la cuarta revolución industrial. La amplitud y profundidad de estos cambios está dando como resultado la transformación de sistemas completos de producción, gestión y gobierno.

La nube de Google y sus Servicios de Datos nos proporcionan la forma de aprovechar estas tres tendencias (Big Data, Cloud y Machine Learning).



Google con Google Cloud, lleva años de investigación e innovación – que abarcan la tecnología y la infraestructura para la analítica avanzada, incluyendo las capacidades tales como Google Cloud Machine Learning, BigData, procesamiento y computación en la nube, capacidades de la Machine Learning como visión, reconocimiento de rostros, reconocimiento de voz e integración con el objetivo de ayudar a los clientes a tomar mejores decisiones y más rápidas para acelerar su ritmo de negocios.



Como habéis visto en el video de presentación del módulo, los servicios de Google Cloud alrededor de los datos, contemplan todo el ciclo de vida de un proyecto, donde realicemos la ingesta de datos en tiempo real con piezas como IoT Core o de datos en modo batch con Pub/Sub, independizándose de los tipos de datos (json, binario, imágenes, texto, etc) y de los protocolos para recibirlos.

Después de su clasificación nos permite el almacenamiento de la información en sistemas orientados a BigData como DataProc y en sistemas de procesamiento masivo como BigQuery.

Para poder pasar a transformar los datos, enriquecerlos y procesarlos ya sea con herramientas como DataFlow para el escenario de tiempo real o con herramientas como hadoop DataProc para procesos batch o mini batch.

Así mismo podemos analizar y enriquecer la información con servicios como Google Cloud machine learning para su análisis estadístico basado en modelos.

Y finalmente podremos visualizar la información desde entornos de BI o apps teniendo en cuenta el contexto de los usuarios finales en tiempo real a través de la utilización de elementos de Google Cloud machine learning y las Apis de Machine Learning.

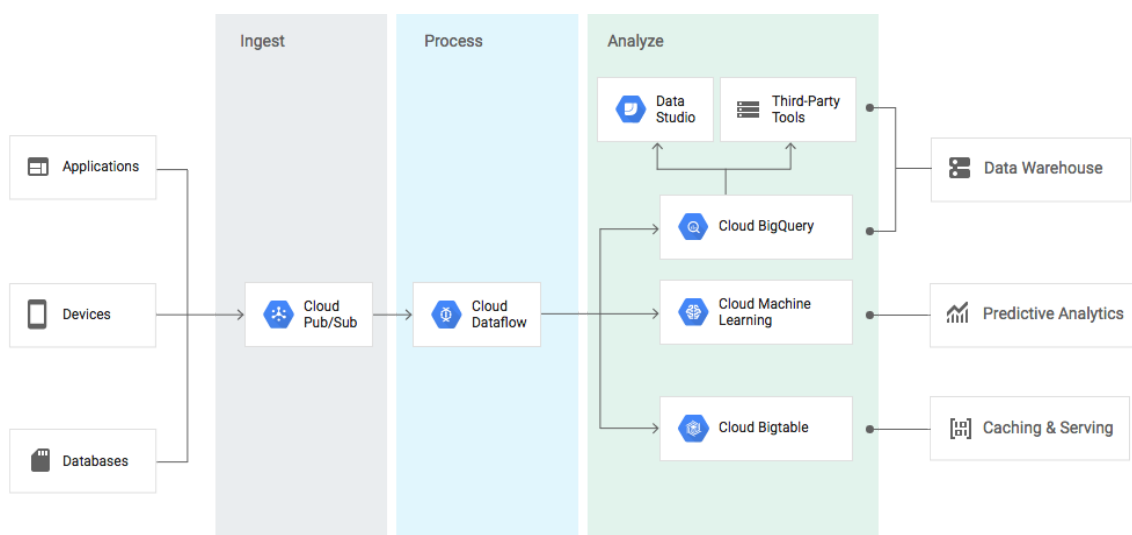
- **Google Cloud Pub/Sub.**

- **Broker de mensajes PaaS:**

## **Google Cloud Pub / Sub: servicio de mensajería a escala Google**

El servicio se basa en un componente básico de la infraestructura de Google en el que muchos productos de Google han confiado durante más de una década.

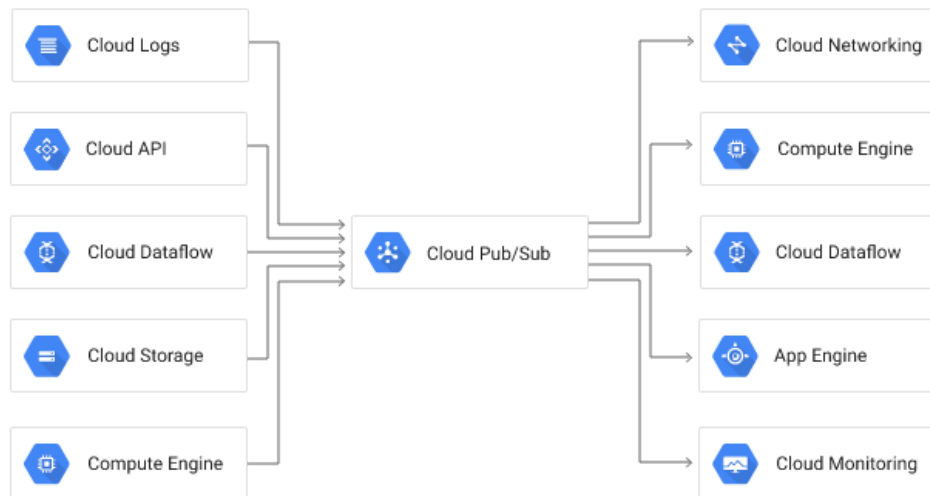
Los productos de Google, incluidos Ads, Search y Gmail, utilizan Pub / Sub para enviar más de 100 millones de mensajes por segundo, con un total de más de 300 GB / s de datos.



### **Los fundamentos de un servicio de publicación / suscripción**

Google Cloud Pub/Sub, es un broker de mensajes para la ingesta de secuencias de eventos desde cualquier lugar, a cualquier escala, para análisis de flujo en tiempo real simples y confiables

Google Cloud Pub / Sub brinda la escalabilidad, flexibilidad y confiabilidad del middleware orientado a mensajes empresariales a la nube. Al proporcionar mensajería asíncrona de muchos a muchos que desacopla a remitentes y receptores, permite una comunicación segura y de alta disponibilidad entre aplicaciones escritas de forma independiente. Google Cloud Pub / Sub ofrece mensajería duradera y de baja latencia que ayuda a los desarrolladores a integrar rápidamente los sistemas alojados en Google Cloud Platform y externamente.



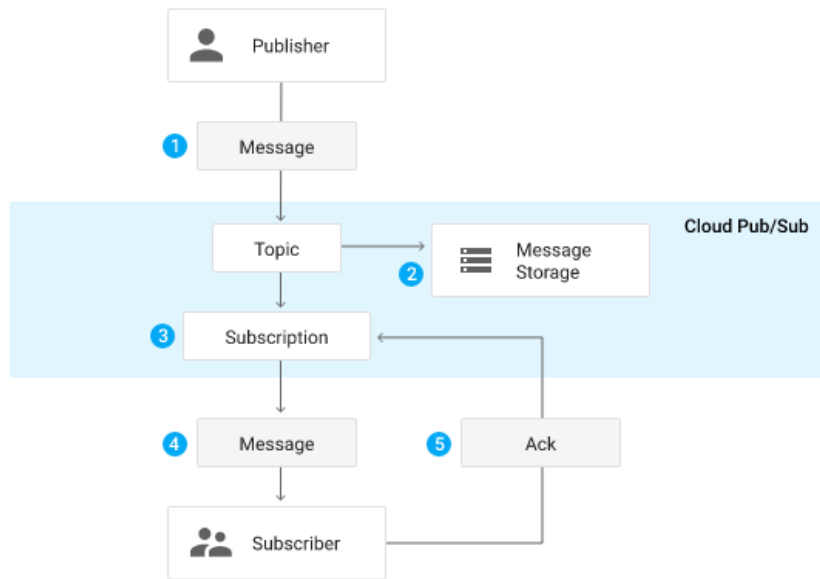
Google Cloud Pub / Sub es un servicio de publicación / suscripción (Pub / Sub): un servicio de mensajería en el que los remitentes de mensajes están desacoplados de los receptores de mensajes.

Hay varios conceptos clave en un servicio Pub / Sub:

- Mensaje: la información que se mueve a través del servicio.
- Tema: una entidad con nombre que representa un feed de mensajes.
- Suscripción: una entidad con nombre que representa un interés en recibir mensajes sobre un tema en particular.
- Editor (también llamado productor): crea mensajes y los envía (pública) al servicio de mensajería sobre un tema específico.
- Suscriptor (también llamado consumidor): recibe mensajes en una suscripción específica.

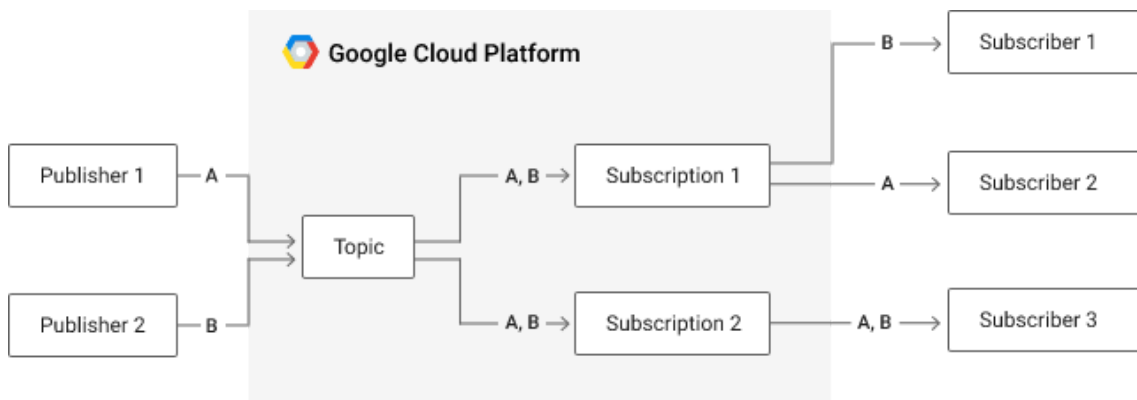
### Pub / Sub conceptos y flujo de mensajes

Aquí veis una descripción general de los componentes en el sistema Pub / Sub y cómo fluyen los mensajes entre ellos:



- Una aplicación de editor crea un tema en el servicio Pub / Sub de Google Cloud y envía mensajes al tema. Los mensajes contienen los datos y atributos opcionales que describen el contenido.
- Los mensajes se conservan en un almacén de mensajes hasta que son entregados y confirmados por los suscriptores.
- El servicio Pub / Sub reenvía mensajes de un tema a todas sus suscripciones, de forma individual. Cada suscripción recibe mensajes de Pub / Sub moviéndolos al punto final elegido por el suscriptor, o por el suscriptor que los retira del servicio.
- El suscriptor recibe mensajes pendientes de su suscripción y reconoce cada uno al servicio Pub / Sub.
- Cuando el suscriptor recoge un mensaje, se elimina de la cola de mensajes de la suscripción.

El flujo básico de mensajes a través de Google Cloud Pub / Sub se puede resumir en el siguiente diagrama:



En este escenario, hay dos editores que publican mensajes sobre un solo tema.

Hay dos suscripciones al tema, donde la primera suscripción tiene dos suscriptores, y la segunda suscripción tiene un suscriptor.

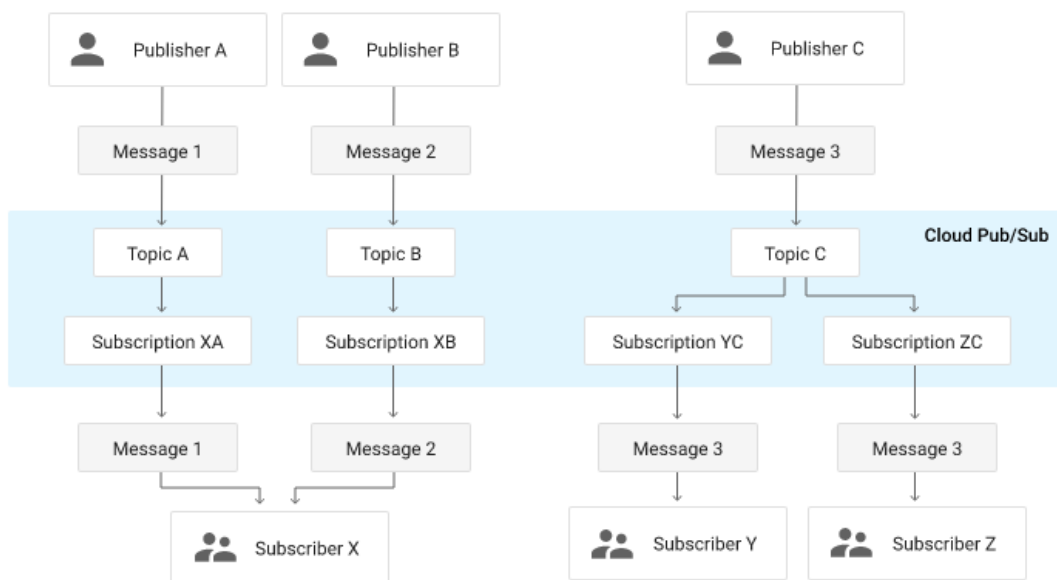
Las letras en **negrita** representan mensajes.

El mensaje A proviene del publicador 1 y se envía al suscriptor 2 a través de la suscripción 1 y al suscriptor 3 a través de la suscripción 2.

El mensaje B proviene del publicador 2 y se envía al suscriptor 1 a través de la suscripción 1 y al suscriptor 3 a través de la suscripción 2.

La lógica de Google Cloud Pub / Sub, la aplicación de editor crea y envía mensajes a un tema y las aplicaciones del suscriptor crean una suscripción a un tema para recibir mensajes de él.

La comunicación puede ser de uno a muchos (fan-out), de muchos a uno (fan-in) y de muchos a muchos.



#### Resumen de Beneficios y características

1. Mensajería unificada: durabilidad y entrega de baja latencia en un solo producto
2. Presencia global: conecte servicios ubicados en cualquier parte del mundo
3. Opciones de entrega flexible: admite suscripciones de estilo push y pull
4. Fiabilidad de los datos: almacenamiento replicado y entrega de mensajes garantizada al menos una vez
5. Fiabilidad de extremo a extremo: reconocimiento explícito de nivel de aplicación
6. Seguridad y protección de datos: cifrado de datos en el cable y en reposo
7. Control de flujo: limitación de velocidad dinámica implementada por el sistema Pub / Sub
8. Simplicidad: API REST / JSON fácil de usar





### **Juzgar el rendimiento de un servicio de mensajería**

Un servicio de mensajería como Google Cloud Pub / Sub se puede juzgar por su rendimiento en tres aspectos:

1. escalabilidad,
2. disponibilidad
3. latencia.

Estos tres factores a menudo están en desacuerdo entre sí, lo que requiere compromisos en uno para mejorar los otros dos.

Los términos "escalabilidad", "disponibilidad" y "latencia" pueden referirse a diferentes propiedades de un sistema, por lo que las siguientes secciones describen cómo se definen en Google Cloud Pub / Sub.

### **Escalabilidad**

Un servicio escalable debería poder manejar aumentos en la carga sin una degradación notable de la latencia o disponibilidad.

"Carga" puede referirse a varias dimensiones de uso en Google Cloud Pub / Sub:

- Número de temas
- Número de editores
- Número de suscripciones
- Número de suscriptores
- Cantidad de mensajes
- Tamaño de los mensajes
- Tasa de mensajes (rendimiento) publicados o consumidos
- Tamaño del log de cualquier suscripción determinada

### **Disponibilidad**

En un sistema distribuido, los tipos y la gravedad de los problemas pueden variar mucho. La disponibilidad de un sistema se mide en función de cómo de bien se ocupe de los diferentes tipos de problemas y de forma elegante de una manera que los usuarios finales no lo perciban.

Pueden ocurrir fallos de hardware (por ejemplo, unidades de disco que no funcionan o problemas de conectividad de red), en el software o debido a la carga.

Los fallos debidos a la carga pueden ocurrir cuando un aumento repentino en el tráfico en el servicio (o en otros componentes de software que se ejecutan en el mismo hardware o en dependencias de software) resulta en escasez de recursos.

La disponibilidad también puede degradarse debido a un error humano, donde uno comete errores al construir o implementar software o configuraciones.

### **Latencia**

La latencia es una medida basada en el tiempo del rendimiento de un sistema. Un servicio generalmente quiere minimizar la latencia siempre que sea posible. Para Google Cloud Pub / Sub, las dos métricas de latencia más importantes son:

1. La cantidad de tiempo que lleva reconocer un mensaje publicado.
2. La cantidad de tiempo que lleva entregar un mensaje publicado a un suscriptor.

## Arquitectura Básica de Google Cloud Pub / Sub

Esta sección explica el diseño de Google Cloud Pub / Sub para mostrar cómo el servicio logra su escalabilidad y baja latencia a la vez que mantiene la disponibilidad.

El sistema está diseñado para ser escalable horizontalmente, donde se puede manejar un aumento en el número de temas, suscripciones o mensajes aumentando el número de instancias de servidores en ejecución.

Los servidores de Google Cloud Pub / Sub se ejecutan en varios centros de datos de Google, que se distribuyen por todo el mundo. Cada centro de datos contiene una o más instancias de un clúster, una agrupación lógica de máquinas que generalmente comparten el mismo dominio de falla (por ejemplo, red local compartida y potencia compartida).

Google Cloud Pub / Sub es un servicio global: los clientes pueden publicar y suscribirse desde cualquier parte del mundo a cualquier parte del mundo.

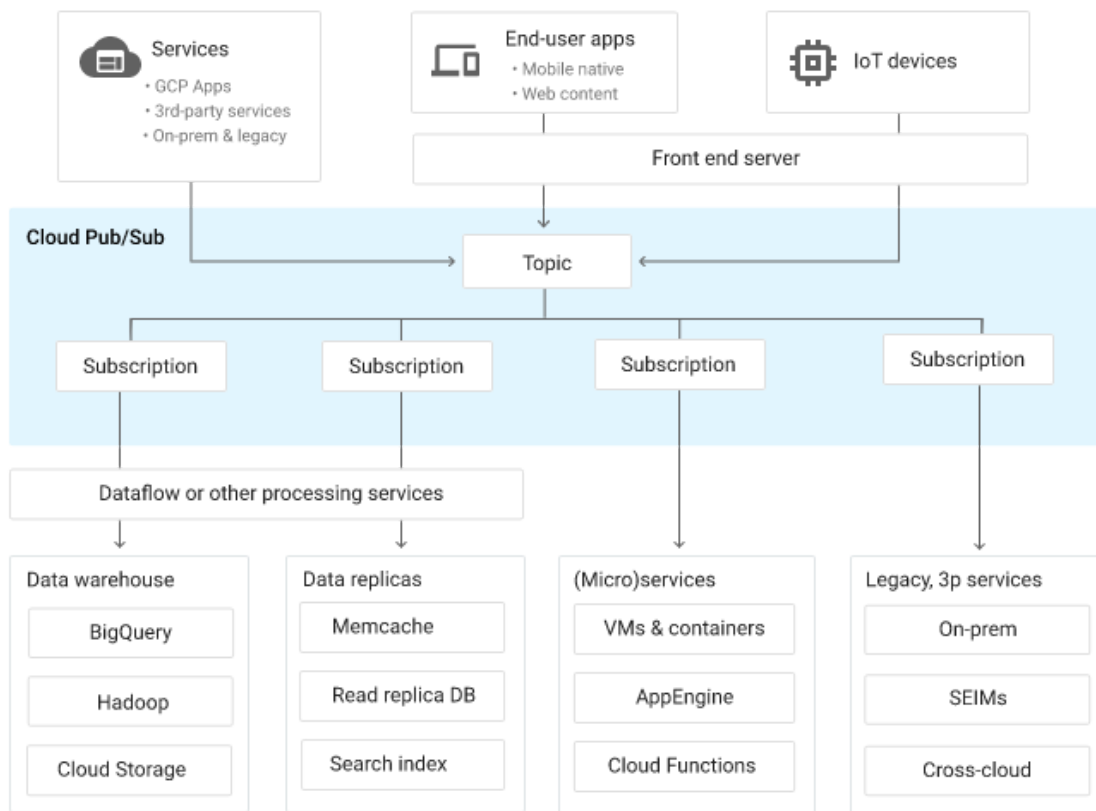
### Editor y puntos de publicación para el suscriptor

Los editores pueden ser cualquier aplicación que pueda realizar solicitudes HTTPS a [googleapis.com](https://googleapis.com):

- una aplicación de Google Cloud App Engine,
- un servicio web alojado en Google Compute Engine o cualquier otra red de terceros,
- una aplicación instalada para un ordenador de escritorio o dispositivo móvil,
- o incluso un navegador.

Los suscriptores de extracción también pueden ser cualquier aplicación que pueda realizar solicitudes de HTTPS a [googleapis.com](https://googleapis.com).

Actualmente, los suscriptores push deben ser puntos finales Webhook que puedan aceptar solicitudes POST a través de HTTPS.



## Modelo de datos

### Tipos de recursos y modelos

Los siguientes son los tipos de objetos utilizados por la API de Pub / Sub.

Los temas y las suscripciones son los únicos tipos de recursos expuestos como colecciones REST.

#### Tema

Un recurso con nombre al que los editores envían los mensajes.

#### Suscripción

Un recurso con nombre que representa la secuencia de mensajes de un único tema específico para ser entregado a la aplicación suscriptora.

#### Mensaje

La combinación de datos y atributos (opcionales) que un editor envía a un tema y finalmente, se entrega a los suscriptores.

#### Atributo de mensaje

Un par clave-valor que un editor puede definir para un mensaje. Por ejemplo, la clave *iana.org/language\_tag* y *value* en podría agregarse a los mensajes para marcarlos como leíbles por un suscriptor que hable castellano.

## Nomenclatura de los recursos

Los nombres de Pub / Sub *tema* y *suscripción* deben tener un alcance de proyecto:

projects/**project-identifier**/**collection**/**resource-name**

El **project-identifier** debe ser el ID de proyecto, disponible desde Google Cloud Platform Console.

Por ejemplo, projects / myproject / topics / mytopic.

La **collection** debe ser una de suscripciones o temas.

El **resource-name** debe comenzar con una letra y contener solo letras ([A-Za-z]), números ([0-9]), guiones (-), guiones bajos (\_), puntos (.), Tildes (~), más (+) o signos de porcentaje (%). Debe tener entre 3 y 255 caracteres de longitud, y no puede comenzar con la cadena goog.

### Google Cloud Pub / Sub se divide en dos partes principales:

1. el plano de datos, que maneja los mensajes en movimiento entre editores y suscriptores,
2. el plano de control, que maneja la asignación de editores y suscriptores a servidores en el plano de datos.

Los servidores en el plano de datos se llaman reenviadores, y los servidores en el plano de control se llaman enrutadores. Cuando los editores y suscriptores están conectados a sus reenviadores asignados, no necesitan ninguna información de los enrutadores. Por lo tanto, es posible actualizar el plano de control de Google Cloud Pub / Sub sin afectar a los clientes que ya están conectados y enviando o recibiendo mensajes.

### Plano de control

El plano de control Google Cloud Pub / Sub distribuye clientes a reenviadores de una manera que proporciona escalabilidad, disponibilidad y baja latencia para todos los clientes. Cualquier reenviador puede atender a los clientes por cualquier tema o suscripción. Cuando un cliente se conecta a Google Cloud Pub / Sub, el enrutador decide los centros de datos a los que el cliente debe conectarse en función de la distancia de red más corta, una medida de la latencia en la conexión entre dos puntos. Dentro de cualquier centro de datos dado, el enrutador trata de distribuir la carga general a través del conjunto de reenviadores disponibles.

El enrutador debe equilibrar dos objetivos diferentes al realizar esta asignación:

(a) uniformidad de la carga (es decir, idealmente, cada reenviador está igualmente cargado);

y

(b) la estabilidad de las asignaciones (es decir, idealmente un cambio en la carga o un cambio en el conjunto de reenviadores disponibles cambia el menor número de asignaciones existentes).

El enrutador usa una variante de hash consistente desarrollada por Google Research para lograr un equilibrio sintonizable entre consistencia y uniformidad. El enrutador proporciona al cliente una lista ordenada de reenviadores a los que puede considerar conectarse. Esta lista ordenada puede cambiar en función de la disponibilidad del reenviador y la forma de la carga del cliente.

### Plano de datos: la vida de un mensaje

El plano de datos recibe mensajes y los envía a los clientes. Tal vez la mejor forma de entender el plano de datos de Google Cloud Pub / Sub sea mirando la vida de un mensaje, desde el momento en que el servicio lo recibe hasta que ya no está presente en el servicio. Permítanos rastrear los pasos de procesar un mensaje. Suponemos que el tema sobre el que se publica el mensaje tiene al menos una suscripción adjunta. En general, un mensaje sigue estos pasos:

1. Un editor envía un mensaje.
2. El mensaje se escribe en el almacenamiento.
3. Google Cloud Pub / Sub envía un reconocimiento al editor de que recibió el mensaje y garantiza su entrega a todas las suscripciones adjuntas.
4. Al mismo tiempo que se escribe el mensaje en el almacenamiento, Google Cloud Pub / Sub lo entrega a los suscriptores.
5. Los suscriptores envían un reconocimiento a Google Cloud Pub / Sub de que han procesado el mensaje.
6. Una vez que al menos un suscriptor por cada suscripción ha reconocido el mensaje, Google Cloud Pub / Sub elimina el mensaje del almacenamiento.

Primero, un editor envía un mensaje sobre un tema a Google Cloud Pub / Sub. Se cifra mediante la capa de proxy y se envía a un reenviador de publicación, un reenviador al que está conectado el editor.

Para garantizar la entrega, el mensaje se escribe inmediatamente en el almacenamiento.

El reenviador escribe inicialmente el mensaje en  $N$  clusters (donde  $N$  es un número impar) y considera que el mensaje persiste cuando se escribió en al menos  $\lceil N / 2 \rceil$  clusters. Una vez que persiste un mensaje, el reenviador de publicación acusa recibo del mensaje al editor, momento en el que Google Cloud Pub / Sub garantiza que el mensaje se entregará a todas las suscripciones adjuntas.

Un proceso en segundo plano escribe regularmente cualquier mensaje que no está en todos los  $N$  clusters en los clusters que faltan los mensajes.

Dentro de cada clúster, el mensaje se escribe en  $M$  discos independientes (donde  $M$  es un número impar), requiriendo que los datos estén en discos  $\lceil M / 2 \rceil$  antes de que se considere persistente en ese clúster. En total, cualquier mensaje publicado se escribirá en al menos  $\lceil M / 2 \rceil$  discos independientes en  $\lceil N / 2 \rceil$  clusters antes de que se considere persistente y eventualmente se replicará en los discos  $N * M$ .

El reenviador de publicación tiene una lista de todas las suscripciones que están asociadas a un tema. Es responsable de mantener tanto los mensajes publicados como los metadatos que describen qué mensajes han sido reconocidos por cada suscripción. El conjunto de mensajes recibidos y almacenados por un

reenviador de publicación para un tema en particular, junto con este seguimiento de los mensajes reconocidos, se denomina "fuente de mensaje de publicación".

Dependiendo de los requisitos de rendimiento para el tema, un único editor puede enviar sus mensajes a múltiples reenviadores de publicación y almacenar mensajes en múltiples fuentes de mensajes de publicación. Diferentes editores para el mismo tema también pueden enviar mensajes a diferentes agentes de publicación. Google Cloud Pub / Sub ajusta dinámicamente el número de reenviadores de publicación que reciben mensajes para un tema en particular a medida que cambia el rendimiento.

Los suscriptores reciben mensajes al conectarse a los reenviadores suscriptores, reenviadores a través de los cuales los mensajes llegan a los suscriptores de los editores.

"Conectar" en el caso de un suscriptor de extracción significa emitir una solicitud de extracción. "Conectar" en el caso de un suscriptor de inserción significa tener el extremo de inserción registrado con Google Cloud Pub / Sub.

Una vez que se crea una suscripción, se garantiza que todos los mensajes publicados después de ese punto se entregarán a esa suscripción, lo que llamamos una garantía de punto de sincronización.

Cada reenviador de suscripción necesita solicitar mensajes de los reenviadores de publicación que tienen fuentes de publicación de publicación para el tema. Al igual que los editores, los suscriptores pueden conectarse a más de un agente de suscripción para recibir mensajes. De esta forma, no todos los reenviadores suscriptores deben tener en cuenta o recibir mensajes de cada origen de mensaje de publicación para un tema, una propiedad importante para que Google Cloud Pub / Sub pueda escalar horizontalmente.

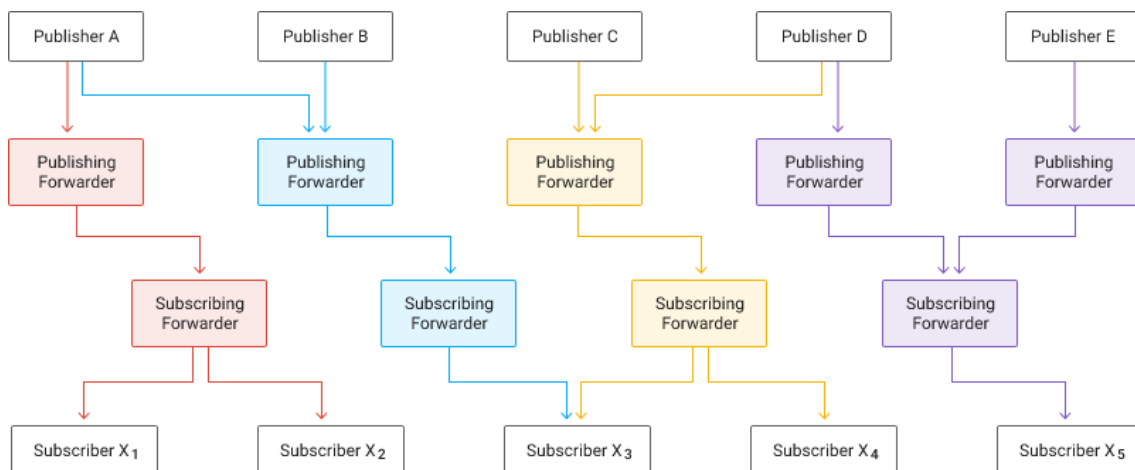
En función del rendimiento de los mensajes entregados a los suscriptores, Google Cloud Pub / Sub sintoniza dinámicamente la cantidad de reenviadores de suscripción a través de los cuales los suscriptores reciben mensajes para un tema en particular a medida que cambia el rendimiento.

Un reenviador de suscripción hace que las solicitudes a uno o más reenviadores de publicación que tienen fuentes de publicación de publicación para un tema soliciten los mensajes que necesita.

El reenviador de publicación envía los mensajes no reconocidos al reenviador suscriptor, que luego retransmite los mensajes a un suscriptor.

Una vez que un suscriptor procesa un mensaje, envía un acuse de recibo al reenviador suscriptor. El promotor suscriptor transfiere este acuse de recibo al reenviador de publicación, que almacena el acuse de recibo en el origen del mensaje de publicación. Una vez que todas las suscripciones de un tema han reconocido un mensaje, el mensaje se elimina de forma asíncrona del origen del mensaje de publicación y del almacenamiento.

La vida de un mensaje es bastante compleja, con varias conexiones involucradas para enviar mensajes de los editores a los suscriptores. El flujo de mensajes a través de conexiones entre editores, suscriptores y reenviadores es el siguiente:



## Mantener Google Cloud Pub / Sub Up en ejecución

Asegurar que un sistema distribuido como Google Cloud Pub / Sub pueda mantenerse en funcionamiento y servir efectivamente a todos los clientes requiere una gran visibilidad y control del sistema.

El mantenimiento del servicio es responsabilidad de nuestros Site Reliability Engineers (SRE).

Para Google Cloud Pub / Sub, estos ingenieros tienen su sede en múltiples ubicaciones en todo el mundo para proporcionar cobertura 24/7.

## Entornos

La primera parte de mantener un sistema como Google Cloud Pub / Sub es tener la capacidad de probar el software antes de que sea utilizado por los clientes.

Para hacer esto posible, hay tres entornos Google Cloud Pub / Sub:

1. prueba,
2. preproducción,
3. producción.

La prueba y preproducción no contienen ningún tráfico de clientes, sólo contienen nuestras pruebas de funcionamiento continuo y monitoreo que ayudan a encontrar cualquier problema con las versiones. Estos entornos reciben nuevas versiones del software antes de la producción.

La diferencia entre prueba y preproducción es que esta última es una réplica exacta de lo que está en (o muy pronto estará) el entorno de producción, incluida la versión del software.

El primero puede tener características habilitadas que los desarrolladores están trabajando actualmente y planean lanzar en el futuro.

## Supervisión y Monitorización



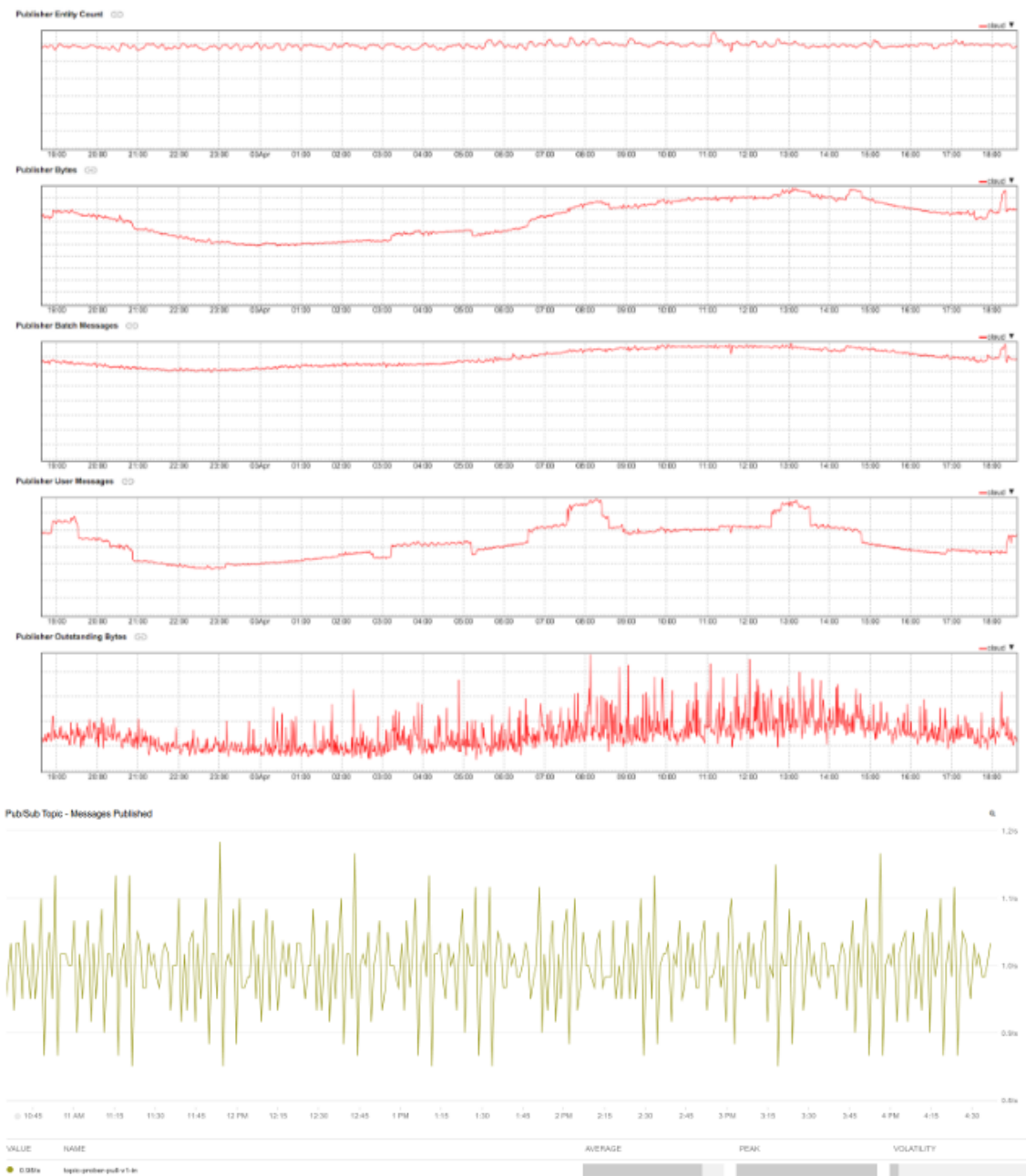
La clave para mantener Google Cloud Pub / Sub funcionando es detectar y mitigar problemas automáticamente antes de que sean visibles para los clientes. Para lograr esto, se requiere una amplia supervisión del sistema. Los SRE mantienen un conjunto de indicadores de nivel de servicio (SLI), métricas bien definidas que describen el comportamiento del sistema. Las métricas pueden incluir "la cantidad de tiempo que lleva completar una solicitud de CreateSubscription" o "la tasa de errores generados por las solicitudes de publicación". Estas métricas se miden de diversas maneras. Algunos de ellos son estrictamente internos para nuestros reenviadores y enrutadores. Por ejemplo, miden cuánto tiempo lleva escribir mensajes en el disco. En total, hay diez SLI y cientos de métricas adicionales utilizadas para supervisar el estado de Google Cloud Pub / Sub.

Todas estas medidas ayudan a definir objetivos de nivel de servicio interno (SLO), objetivos específicos para los SLI. Por ejemplo, "una solicitud CreateSubscription no debería tomar más de cinco segundos para completarse". Los SRE reciben alertas de violaciones de SLO y deben atender las alertas en cinco minutos.

Un acuerdo de nivel de servicio (SLA) enumera los SLO que definen nuestras garantías de rendimiento para nuestros clientes y las consecuencias si no los cumplimos.

Mantenemos un conjunto de tareas que actúan como clientes y publican y suscriben predeciblemente llamados probadores. Existen probadores tanto para el plano de datos como para el plano de control. Cada uno de nuestros diez tipos de probadores realiza acciones específicas tal como lo haría un cliente y mide cuánto tiempo duran las operaciones. Por ejemplo, tenemos un probador que crea una nueva suscripción, publica un mensaje y ve cuánto tiempo le tomó crear la suscripción y recibir el mensaje. Si los probadores determinan que cualquiera de las treinta mediciones medidas no es lo que se espera, los SRE son alertados.

Las métricas para nuestros servidores y probadores se resumen en varios paneles internos, los SRE de primer lugar se ven cada vez que se diagnostican posibles problemas. Estas páginas proporcionan acceso rápido a estadísticas y gráficos de todo el servicio, como se muestra a continuación. También se pueden desglosar por tema, centro de datos o tarea individual.



## Escenarios comunes

Estos son algunos casos de uso clásicos para Google Cloud Pub / Sub:

1. Equilibrar las cargas de trabajo en los clústeres de red. Por ejemplo, una gran cantidad de tareas se puede distribuir de manera eficiente entre varios trabajadores, como las instancias de Google Compute Engine.

2. Implementando flujos de trabajo asíncronos. Por ejemplo, una aplicación de procesamiento de pedidos puede realizar un pedido de un tema, desde el cual uno o más trabajadores pueden procesarlo.
3. Distribución de notificaciones de eventos. Por ejemplo, un servicio que acepta registros de usuario puede enviar notificaciones cada vez que un nuevo usuario se registra, y los servicios descendentes pueden suscribirse para recibir notificaciones del evento.
4. Actualización de cachés distribuidos. Por ejemplo, una aplicación puede publicar eventos de invalidación para actualizar los ID de los objetos que han cambiado.
5. Iniciando sesión en múltiples sistemas. Por ejemplo, una instancia de Google Compute Engine puede escribir registros en el sistema de supervisión, en una base de datos para consultas posteriores, y así sucesivamente.
6. Transmisión de datos desde varios procesos o dispositivos. Por ejemplo, un sensor residencial puede transmitir datos a servidores de back-end alojados en la nube.
7. Mejora de la confiabilidad. Por ejemplo, un servicio de Compute Engine de una sola zona puede operar en zonas adicionales al suscribirse a un tema común, para recuperarse de fallos en una zona o región.

Glosario de términos de Google Cloud Pub/Sub

| Term                                | Description  |
|-------------------------------------|--|
| cluster                             | A logical grouping of machines that generally share the same failure domain (e.g., shared local network and shared power).                 |
| control plane                       | The layer of Google Cloud Pub/Sub that handles the assignment of publishers and subscribers to servers on the data plane.                  |
| data plane                          | The layer of Google Cloud Pub/Sub that handles moving messages between publishers and subscribers.   |
| forwarder                           | A server in the data plane.  |
| global service                      | A service that does not require clients to know or specify the region (or data center) to which to connect to use the service.             |
| horizontally scalable               | The ability of a service to seamlessly handle more load by increasing the number of instances of components of the service.                |
| message                             | The data that moves through Google Cloud Pub/Sub.  |
| network distance                    | A measure of the latency on the connection between two points.   |
| prober                              | A task that acts as a client and predictably performs one or more actions on the Google Cloud Pub/Sub servers.                             |
| publish message source              | A set of messages received and stored by a publishing forwarder and the set of IDs of messages acknowledged by all attached subscriptions. |
| publish/subscribe (pub/sub) service | A messaging service where the senders of messages are decoupled from the receivers of messages   |
| publisher                           | A client of Google Cloud Pub/Sub that creates messages and sends (publishes) them on a specified topic.                                    |
| router                              | A server in the control plane.   |
| routing constraints                 | A list of rules indicating which forwarders should or should not be sent by routers to clients as possible endpoints to connect to.        |
| service level agreement (SLA)       | A list of SLOs that define a system's performance guarantees to customers and outlines the consequences if they are not met.               |
| service level indicator (SLI)       | A well-defined metric that describes the behavior of the system.   |
| service level objective (SLO)       | A specific target for a service level indicator.   |
| subscriber                          | A client of Google Cloud Pub/Sub that receives messages on a specified subscription.   |
| subscription                        | A named entity that represents an interest in receiving all messages on a particular topic.  |
| sync-point guarantee                | The time at which a subscriber is created, where all subsequent messages published will be delivered to that subscriber.                   |
| topic                               | A named entity that represents a feed of messages.   |



- **Google Cloud IoT Core.**

- **Internet de las Cosas como servicio PaaS:**

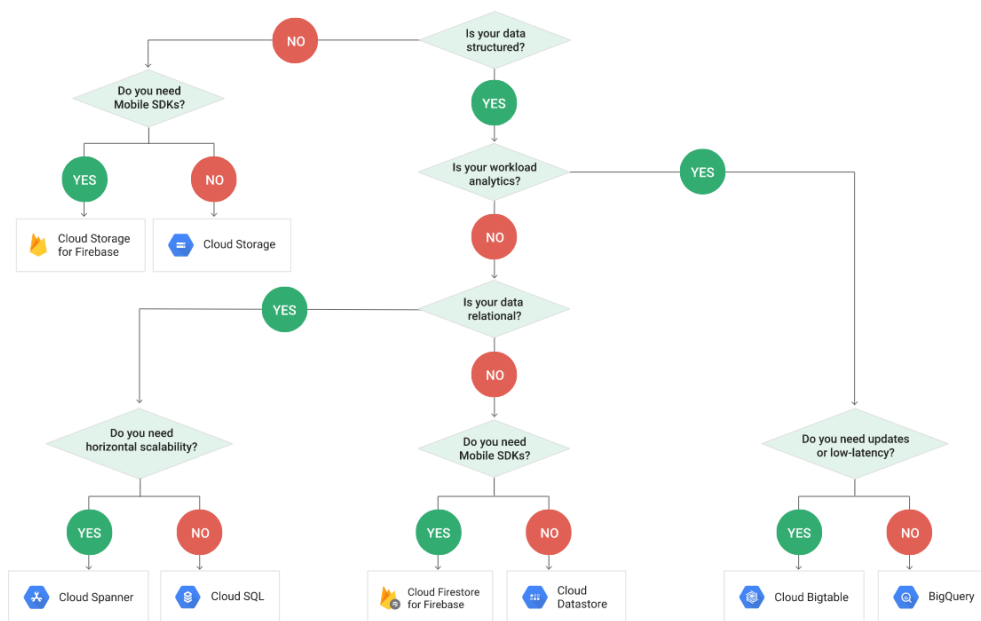
Google Cloud Internet of Things (IoT) Core es un servicio completamente administrado para conectar y administrar de forma segura dispositivos, desde unos pocos hasta millones. Obtenga datos de dispositivos conectados y cree aplicaciones completas que se integren con otros servicios de big data de Google Cloud Platform.

## **Google Cloud Cloud Storage.**

- **Data Lake PaaS:**
- Google Cloud Cloud Storage.

### **Google Cloud Data Lake : Tus opciones de almacenamiento**

Cada aplicación y carga de trabajo requiere una solución de almacenamiento y base de datos diferente. Ofrecemos un conjunto completo de servicios de almacenamiento líderes en el sector, con una buena relación precio-rendimiento, y que satisfacen tus necesidades de datos estructurados, sin estructurar, transaccionales y relacionales. La información de esta página te puede ayudar a identificar las soluciones óptimas para tus circunstancias, tanto si se trata de aplicaciones móviles como del alojamiento de software comercial, pasando por canalizaciones de datos y almacenamiento de copias de seguridad.



## Opciones de almacenamiento en Google Cloud:

| PRODUCTO                     | DESCRIPCIÓN  | ADECUADO PARA  | CARGAS DE TRABAJO HABITUALES  |
|------------------------------|--|--|---|
| <u>Disco persistente</u>     | Almacenamiento en bloques totalmente administrado y con una buena relación precio-rendimiento, adecuado para máquinas virtuales y contenedores | <ul style="list-style-type: none"> <li>Almacenamiento en bloques para Google Compute Engine y Google Container Engine</li> <li>Capturas de copias de seguridad de datos</li> </ul>           | <ul style="list-style-type: none"> <li>Discos para máquinas virtuales</li> <li>Datos de solo lectura compartidos en varias máquinas virtuales</li> <li>Copias de seguridad rápidas y duraderas de máquinas virtuales en ejecución</li> </ul>          |
| <u>Google Cloud Storage</u>  | Almacén de blobs y objetos escalable, totalmente administrado, muy fiable y económico  | <ul style="list-style-type: none"> <li>Imágenes, fotografías y vídeos</li> <li>Objetos y blobs</li> <li>Datos sin estructurar</li> </ul>   | <ul style="list-style-type: none"> <li>Almacenamiento y transmisión de datos multimedia</li> <li>Almacenamiento de canalizaciones de análisis de datos personalizados</li> <li>Archivado, copia de seguridad y recuperación ante desastres</li> </ul> |
| <u>Google Cloud Bigtable</u> | Base de datos de columnas anchas NoSQL escalable y totalmente administrada, apta para cargas de trabajo de análisis y acceso en tiempo real    | <ul style="list-style-type: none"> <li>Acceso de lectura y escritura de baja latencia</li> <li>Análisis de alto rendimiento</li> <li>Compatibilidad con series temporales nativas</li> </ul> | <ul style="list-style-type: none"> <li>Internet de las Cosas, finanzas y tecnología de anuncios</li> <li>Personalización y recomendaciones</li> <li>Supervisión</li> <li>Conjuntos de datos geoespaciales</li> <li>Gráficos</li> </ul>                |

|   |  |   |  |
|---|--|---|--|
| <a href="#"><u>Google Cloud Datastore</u></a> | Base de datos de documentos NoSQL escalable y totalmente administrada para aplicaciones web y móviles  | <ul style="list-style-type: none"> <li>• Datos de aplicaciones semiestructurados</li> <li>• Datos jerárquicos</li> <li>• Datos de clave-valor duraderos</li> </ul>  | <ul style="list-style-type: none"> <li>• Perfiles de usuarios</li> <li>• Catálogos de productos</li> <li>• Estado del juego</li> </ul>   |
| <a href="#"><u>Google Cloud SQL</u></a>       | Servicio de base de datos MySQL y PostgreSQL totalmente administrado, basado en la potencia y fiabilidad de la infraestructura de Google           | <ul style="list-style-type: none"> <li>• Frameworks web</li> <li>• Datos estructurados</li> <li>• Cargas de trabajo de procesamiento de transacciones online (OLTP)</li> </ul>  | <ul style="list-style-type: none"> <li>• Sitios web, blogs y sistemas de administración de contenido (CMS)</li> <li>• Aplicaciones de inteligencia empresarial (BI)</li> <li>• Aplicaciones de planificación de recursos empresariales (ERP), administración de la relación con los clientes (CRM) y comercio electrónico</li> <li>• Aplicaciones geoespaciales</li> </ul> |
| <a href="#"><u>Google Cloud Spanner</u></a>   | Servicio de base de datos relacional para aplicaciones y servicios fundamentales con coherencia transaccional, escala global y alta disponibilidad | <ul style="list-style-type: none"> <li>• Aplicaciones críticas</li> <li>• Gran volumen de transacciones</li> <li>• Requisitos de escala y consistencia</li> </ul>   | <ul style="list-style-type: none"> <li>• Tecnología de anuncios</li> <li>• Servicios financieros</li> <li>• Cadena de suministro mundial</li> <li>• Tiendas</li> </ul>   |
| <a href="#"><u>Google BigQuery</u></a>        | Almacén de datos empresariales (EDW) escalable y totalmente administrado con SQL y tiempos de respuesta rápidos                                    | <ul style="list-style-type: none"> <li>• Cargas de trabajo de procesamiento analítico online (OLAP) hasta nivel de petabytes</li> <li>• Exploración y procesamiento de Big Data</li> <li>• Creación de informes mediante herramientas de inteligencia empresarial (BI)</li> </ul> | <ul style="list-style-type: none"> <li>• Informes analíticos sobre grandes volúmenes de datos</li> <li>• Ciencia de datos y análisis avanzados</li> <li>• Procesamiento de Big Data mediante SQL</li> </ul>  |
| <a href="#"><u>Google Drive</u></a>           | Espacio de colaboración para almacenar, compartir y editar archivos, incluido Documentos de Google   | <ul style="list-style-type: none"> <li>• Interacción con documentos y archivos por parte del usuario final</li> <li>• Creación y edición en modo colaborativo</li> <li>• Sincronización de archivos entre la nube y los dispositivos locales</li> </ul>                           | <ul style="list-style-type: none"> <li>• Acceso a archivos desde cualquier lugar a través de distintos clientes: web, de aplicaciones y de sincronización</li> <li>• Creación y modificación de documentos junto con compañeros de trabajo</li> <li>• Copia de seguridad de fotos y archivos multimedia</li> </ul>   |

En este apartado vamos a centrarnos en Google Cloud Storage es el sistema de almacenamiento de objetos unificado para desarrolladores y empresas, que abarca desde el suministro de datos activos hasta el aprendizaje automático y el análisis de datos, pasando por el archivado.



Google Cloud Storage trabaja con el concepto de almacenamiento online mediante un producto unificado que responde a las necesidades en todo el espectro de disponibilidad: desde los datos activos que emplean las aplicaciones más exigentes actualmente hasta las soluciones de archivado en la nube Nearline y Coldline. Google Cloud storage es una de las causas del por qué la infraestructura de Google, ofrece niveles uniformes de APIs, latencia y velocidad en todas las clases de almacenamiento, es el mejor sistema de almacenamiento online en la nube para tus datos más importantes.

### **Duradero**

Google Cloud Storage está diseñado para ofrecer un 99,999999999% de durabilidad. se puede almacenar los datos de forma redundante, con sumas de comprobación automáticas para garantizar su integridad. Incluso con el almacenamiento multirregional, los datos se conservan en distintas ubicaciones geográficas.

### **Disponible**

Todas las clases de almacenamiento ofrecen una gran disponibilidad. se puede acceder a los datos siempre que lo necesites. Según sus acuerdos de nivel de servicio, el almacenamiento multirregional ofrece una disponibilidad mensual del 99,95% y el almacenamiento regional, del 99,9%. La disponibilidad mensual del almacenamiento Nearline y Coldline es de un 99%.

### **Escalable**

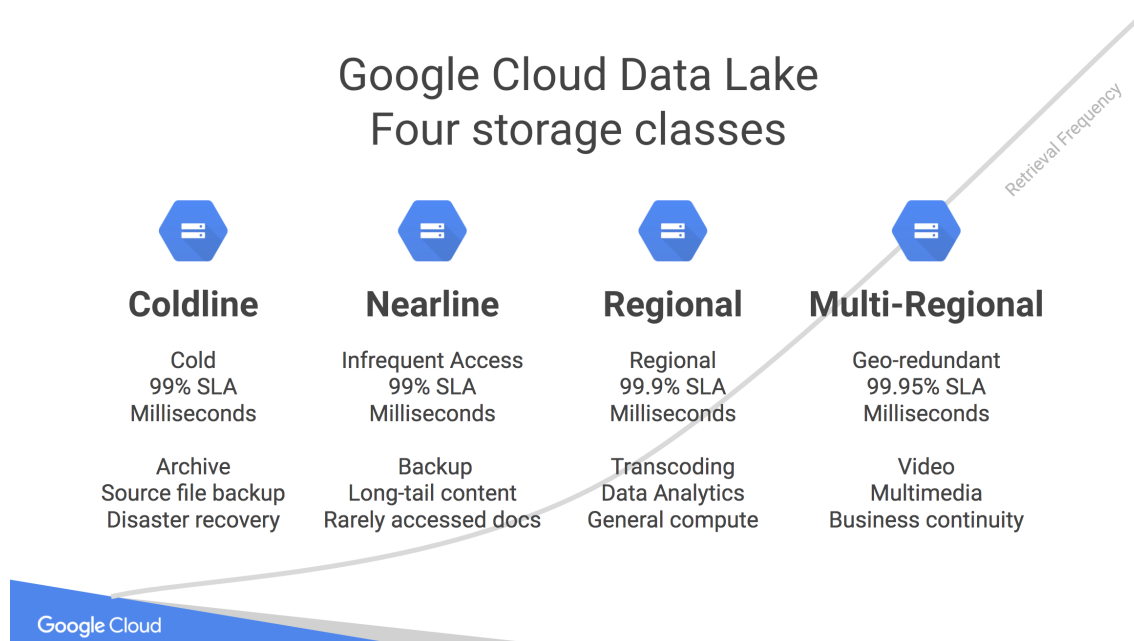
Google Cloud Storage ofrece una escalabilidad prácticamente infinita. Tanto si se suministra una aplicación pequeña como si quiere crear un sistema de gran tamaño con exabytes de datos, Cloud Storage puede hacerse cargo.

### **Escala de Google Cloud Storage**

Google Cloud Storage se basa en la misma infraestructura de almacenamiento central que utiliza la Búsqueda de Google, YouTube y los demás servicios de Google. La infraestructura de almacenamiento de Google está construida para admitir algunos de los volúmenes de datos más grandes del mundo, y su capacidad de almacenamiento crece cada año en cantidades masivas.

Para usar YouTube como un ejemplo de almacenamiento y escala de red: cada minuto, de promedio, los usuarios de YouTube cargan 72 horas de video y ven más de 91,000 horas de video.

Google Cloud Storage escalará a los volúmenes requeridos y proyectados por ti. Google Cloud Storage no requiere que se "pre-asigne" o "pre-compre" el almacenamiento. Simplemente se irá subiendo más datos y Cloud Storage escalará automáticamente.



Tipos:

**Multiregional:**

Almacenamiento con redundancia geográfica, así como con el máximo nivel de disponibilidad y rendimiento. Ideal para el suministro de contenido de baja latencia con gran número de consultas por segundo a usuarios distribuidos en distintas regiones geográficas.

**Regional:**

El máximo nivel de disponibilidad y rendimiento dentro de una misma región. Ideal para las cargas de trabajo informáticas, de aprendizaje automático y análisis en una región específica.

**Nearline:**

Almacenamiento rápido, asequible y muy duradero para los datos a los que se accede menos de una vez al mes.

**Coldline:**

Almacenamiento rápido, asequible y muy duradero para los datos a los que se accede menos de una vez al año.

|                 | Regional   | Multi-Regional  | Nearline   | Coldline   |
|-----------------|--|---|--|--|
| Design Patterns | Data that is used in one region or needs to remain in region | Data that is used globally and has no regional restrictions | Backups<br>Data that is accessed no more than once a month | Archival or Disaster Recovery (DR) data that is accessed once a year or less often |
| Feature         | Regional   | Geo-redundant   | Backup   | Archived or DR   |
| Availability    | 99.9%  | 99.95%  | 99.0%  | 99.0%  |
| Durability      | 99.9999999999%   | 99.9999999999%  | 99.9999999999%   | 99.9999999999%   |
| Duration        | Hot data   | Hot data  | 30 day minimum   | 90 day minimum   |
| Retrieval cost  | none   | none  | \$   | \$\$   |

## Opciones de Almacenamiento en VMs

De forma predeterminada, cada instancia de Compute Engine tiene un solo disco persistente raíz que contiene el sistema operativo. Cuando alguna aplicación utilizada por ti requiera espacio de almacenamiento adicional, se puede agregar una o más opciones (discos) de almacenamiento adicionales a su instancia.

Compute Engine cifra automáticamente sus datos antes de que viajen fuera de su instancia a un espacio de almacenamiento en disco persistente. No necesita cifrar archivos antes de escribirlos en discos persistentes. Usando las claves de cifrado proporcionadas por ti, es posible controlar las claves de cifrado que se utilizan para encriptar sus datos.

## Discos Persistentes

Los discos persistentes son dispositivos de almacenamiento duradero que funcionan de manera similar a los discos físicos en un escritorio o un servidor. Compute Engine administra el hardware de estos dispositivos para garantizar la redundancia de datos y optimizar el rendimiento para TELEFONICA. Los discos permanentes son independientes de las instancias de su máquina virtual, por lo que puede separar o mover los discos persistentes para mantener sus datos incluso después de eliminar las instancias asociadas. El rendimiento del disco persistente se amplía automáticamente con el tamaño, por lo que puede cambiar el tamaño de sus discos persistentes existentes o agregar más discos persistentes a una instancia para cumplir con los requisitos de rendimiento y capacidad.

## SSD y Local SSDs

Los SSD locales están conectados físicamente al servidor que aloja su instancia de máquina virtual. Los SSD locales tienen un mayor rendimiento y una menor latencia que los discos persistentes estándar o discos persistentes SSD. Los datos que almacena en un SSD local persisten solo hasta que detiene o elimina la instancia. Los datos almacenados en discos SSD persistentes sin embargo persisten independientemente de la instancia.

Cada SSD local tiene un tamaño de 375 GB, pero puede conectar hasta ocho dispositivos SSD locales por instancia para 3 TB de espacio de almacenamiento SSD local total. Los SSD locales se aprovechan cuando se necesita un disco o memoria caché rápida y no se quiere usar memoria de instancia. También se recomienda usar SSD locales cuando la carga de trabajo se replique en varias instancias.

- **Google Cloud BigQuery.**

- **Data WareHouse PaaS:**

- Google Cloud BigQuery

### **Datawarehouse empresarial en la nube**

BigQuery es el servicio datawarehouse empresarial de Google de bajo coste, totalmente administrado y apto para analizar petabytes de datos. BigQuery no requiere servidor. Como no hay que administrar ninguna infraestructura ni se necesita un administrador de bases de datos, cada uno puede centrarse en analizar los datos para obtener información importante mediante el conocido lenguaje SQL.

BigQuery permite consultas SQL súper rápidas utilizando la potencia de procesamiento de la infraestructura de Google.

Dedicaremos un documento y presentación específicos sobre Google Cloud BigQuery más adelante, por lo que simplemente lo dejamos definido aquí.

- Google Cloud DataFlow.

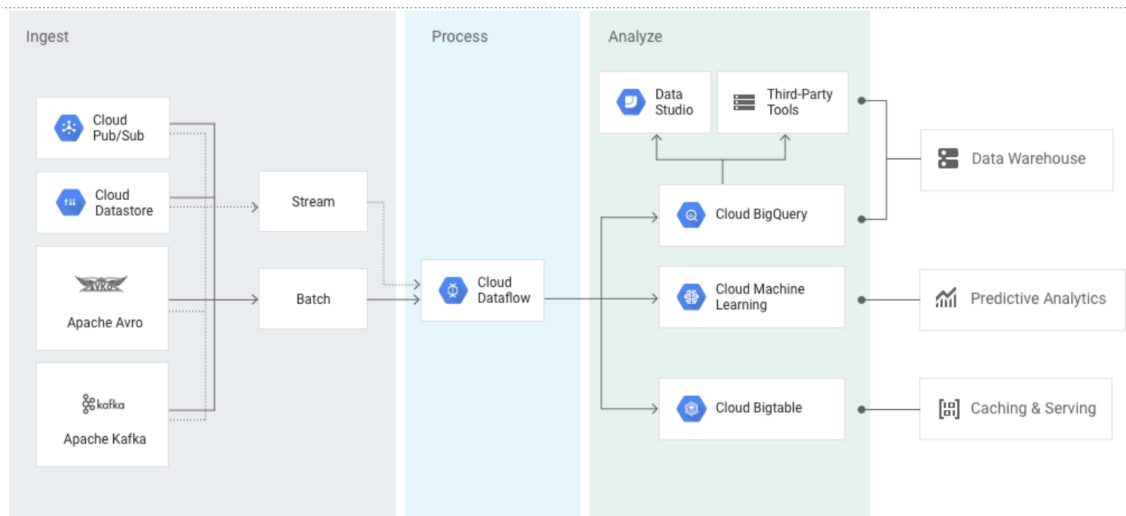
- ETL PaaS:
- Google Cloud DataFlow.

## *Cloud DataFlow*

Cloud Dataflow ofrece un modelo de programación unificada para ejecutar una amplia gama de patrones de procesamiento de datos que incluyen “streaming analytics”, ETL y procesamiento por lotes, en una arquitectura conocida como Kappa, y que sigue el modelo de la iniciativa de software libre Apache Beam, siendo compatible con esta.

Cloud Dataflow es un servicio administrado que maneja de forma transparente la vida útil de los recursos y puede aprovisionarse de forma dinámica para mantener el rendimiento, reduciendo drásticamente la planificación de capacidad y gestión de recursos. Dataflow proporciona primitivas de programación que se pueden aplicar tanto en fuentes de datos basadas en lotes como en streaming, eliminando la necesidad de mantener dos modelos de tratamiento de la información independientes.

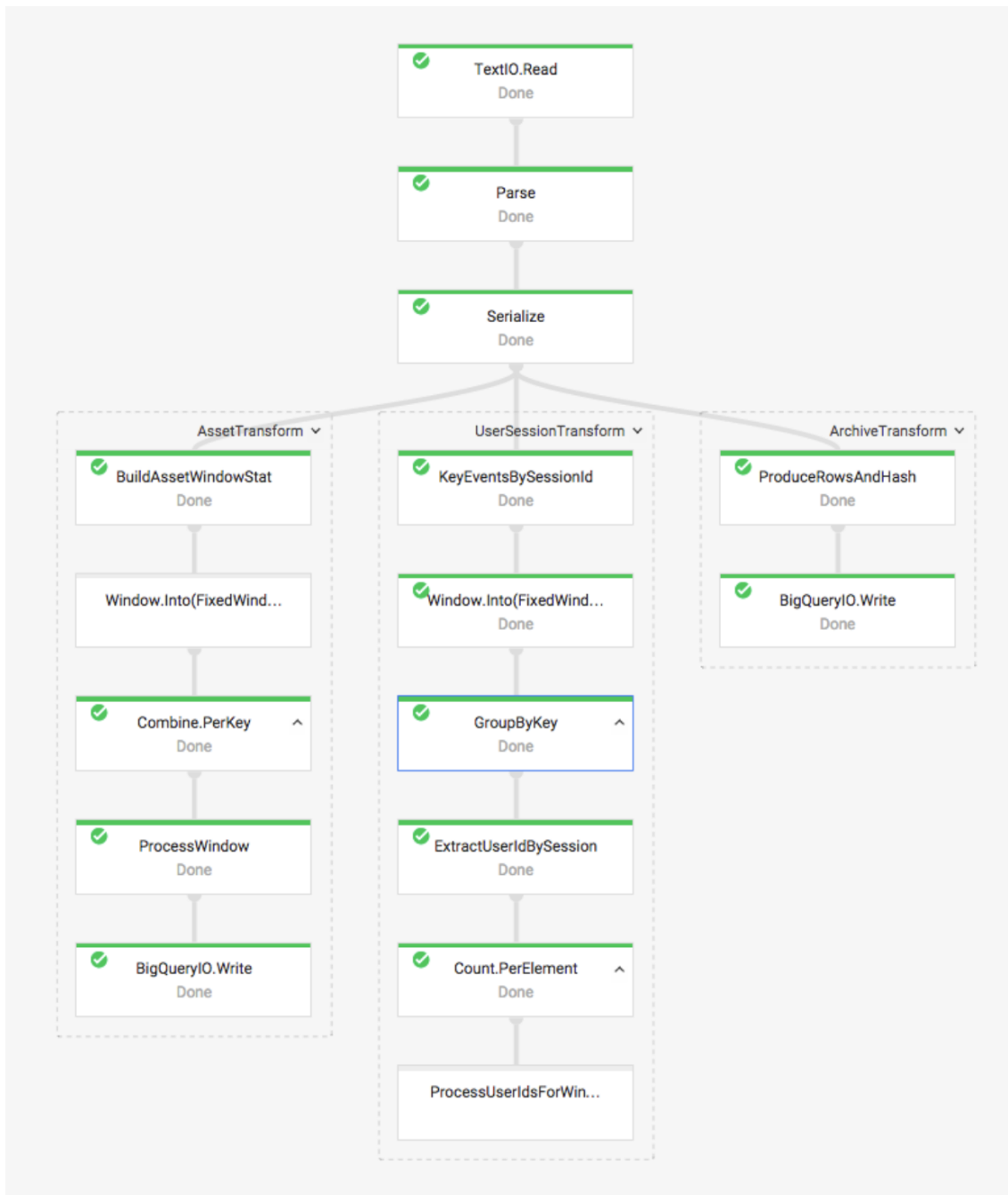
Los SDK de Cloud Dataflow están disponibles para Java y Python, permitiendo a los desarrolladores implementar extensiones personalizadas. Así mismo, se integra con otros servicios de Google Cloud como Cloud Storage, Pub / Sub, Cloud Datastore, Bigtable y BigQuery, y herramientas de terceros, tal y como se puede ver en el siguiente diagrama:



El modelo de programación de Dataflow está diseñado para simplificar la mecánica del procesamiento de datos a gran escala. Cuando programamos con un SDK de Dataflow, básicamente estamos creando un trabajo de procesamiento de datos para ser ejecutado por uno de los servicios de Google Cloud Dataflow. Este modelo permite concentrarse en la composición lógica de su trabajo de procesamiento de datos, en lugar de la orquestación física del procesamiento paralelo. Con Google Cloud DataFlow te puedes centrar en lo que necesita que haga su trabajo en lugar de exactamente cómo se ejecuta ese trabajo.

El modelo de Dataflow proporciona varias abstracciones que lo aíslan de los detalles de bajo nivel del procesamiento distribuido, como la coordinación de trabajadores individuales, la fragmentación de conjuntos de datos y otras tareas similares.

Estos detalles de bajo nivel están completamente administrados por los servicios Google Cloud Dataflow.





Cuando se piense en el procesamiento de datos con Dataflow, puede pensar en términos de cuatro conceptos principales:

- Pipelines
- PCollections
- Transforms
- I/O Sources and Sinks

### ***Pipelines***

Una Pipeline encapsula una serie completa de cálculos que acepta algunos datos de entrada de fuentes externas, transforma esos datos para proporcionar cierta inteligencia útil y produce algunos datos de salida. Los datos de salida a menudo se escriben en un receptor de datos externo. El receptor de origen y de salida de entrada puede ser el mismo o pueden ser de diferentes tipos, lo que le permite convertir fácilmente datos de un formato a otro.

Cada Pipeline representa un único trabajo potencialmente repetible, de principio a fin, en el servicio de Dataflow.

### ***PCollections***

Una PCollection representa un conjunto de datos en su pipeline. Las clases de PCollection son clases de contenedor especializadas que pueden representar conjuntos de datos de tamaño prácticamente ilimitado. Una PCollection puede contener un conjunto de datos de un tamaño fijo (como datos de un archivo de texto o una tabla BigQuery) o un conjunto de datos ilimitado de una fuente de datos que se actualiza continuamente (como una suscripción de Google Cloud Pub / Sub).

PCollections son las entradas y salidas para cada paso en su pipeline.

## ***Transformaciones***

Una transformación es una operación de procesamiento de datos, o un paso, en su pipeline. Una transformación toma una o más PCollections como entrada, realiza una función de procesamiento que usted proporciona en los elementos de esa PCollection, y produce una salida PCollection.

Tus transformaciones no necesitan estar en una secuencia lineal estricta dentro de su pipeline. Puedes usar condicionales, bucles y otras estructuras de programación comunes para crear una pipeline de derivación o una pipeline con estructuras repetidas. Puedes pensar en su pipeline como un gráfico dirigido de pasos, en lugar de una secuencia lineal.

## ***I/O Sources and Sinks***

Los SDK de Dataflow proporcionan fuentes de datos y API de receptores de datos para las Entrada / Salida de los pipelines. Utiliza las API de origen para leer datos en tu pipeline, y las API de receptor para escribir datos de salida de tu pipeline. Estas operaciones I/O representan las raíces y los puntos finales de tu pipeline.

Las API de I/O de Dataflow permiten que tu pipeline funcione con datos de diferentes formatos de almacenamiento de datos, como archivos en Google Cloud Storage, tablas BigQuery y más. También puedes usar una fuente de datos personalizada (o receptor) enseñando a Dataflow cómo leer desde (o escribir) en paralelo.

## ***Supervisión***

Cloud Dataflow se encuentra integrado en la consola de Google Cloud Platform para ofrecer estadísticas como el rendimiento y el retraso de las canalizaciones, así como para proporcionar una inspección unificada del registro de pipelines. Y todo, casi en tiempo real.

### ***Escalabilidad automática***

La escalabilidad automática horizontal de los recursos, cumple los requisitos óptimos de rendimiento y resulta en una mejor relación precio-rendimiento

### ***Procesamiento uniforme y fiable***

Cloud Dataflow es compatible con una ejecución tolerante a fallos que siempre es uniforme y correcta, sea cual sea el tamaño de los datos, el tamaño del clúster, el patrón de procesamiento o la complejidad de las pipelines.

### ***Código abierto***

Los desarrolladores que quieran ampliar el modelo de programación de Dataflow pueden bifurcar o enviar solicitudes de extracción en los SDK de Apache Beam. Las pipelines de Dataflow se pueden ejecutar también en tiempos alternativos como Spark y Flink.

- **Google Cloud DataProc.**

- **Hadoop PaaS:**
- Google Cloud DataProc.

### **Hadoop y Spark administrados**

Google Cloud Dataproc es un servicio de Apache Hadoop, Apache Spark, Apache Pig y Apache Hive para procesar sin esfuerzo grandes conjuntos de datos a bajo coste.

Puedes crear clústeres administrados de cualquier tamaño y desactivarlos cuando acabes para controlar los costes. Cloud Dataproc se integra en todos los productos de Google Cloud Platform y ofrece una plataforma de procesamiento de datos potente y completa.

Dedicaremos un documento y presentación específicos sobre Google Cloud DataProc mas adelante, por lo que simplemente lo dejamos definido aquí.

- **Google Cloud Machine Learning.**

- **Machine Learning PaaS:**
- Google Cloud Machine Learning

Descripción general del motor Cloud ML

Cloud Machine Learning Engine combina la infraestructura administrada de Google Cloud Platform con la potencia y flexibilidad de TensorFlow. Puedes usarlo para entrenar tus modelos de aprendizaje automático a escala y para alojar modelos capacitados para hacer predicciones sobre nuevos datos en la nube.

Dedicaremos un documento y presentación específicos sobre Google Cloud Machine Learning más adelante, por lo que simplemente lo dejamos definido aquí.

## ● Google Cloud DataLab.

- **Data Science PaaS:**
- Google Cloud DataLab.

### **Documentación de Google Cloud Datalab**

Utilice Cloud Datalab para explorar, visualizar, analizar y transformar datos de manera sencilla mediante el uso de lenguajes familiares, como Python y SQL, de forma interactiva. Los notebooks introductorios, de muestra y tutoriales preinstalados de Jupyter le muestran cómo:

- Acceda, analice, monitoree y visualice datos
- Utilice notebooks con las API de Python, TensorFlow Machine Learning, Google Analytics y Google BigQuery

### **Conceptos y componentes clave**

Vamos a ver os conceptos clave y los detalles de los componentes para Cloud Datalab.

#### **Cloud Datalab y notebooks**

Cloud Datalab se empaqueta como un contenedor y se ejecuta en una instancia de máquina virtual (VM). El inicio rápido explica la creación de máquinas virtuales, ejecuta el contenedor en esa máquina virtual y establece una conexión desde su navegador al contenedor de Cloud Datalab, que le permite abrir los notebooks existentes de Cloud Datalab y crear nuevos notebooks.

Lea los cuadernos introductorios en el directorio / docs / intro para tener una idea de cómo se organiza y ejecuta un cuaderno.

Cloud Datalab usa notebooks en lugar de los archivos de texto que contienen el código. Los notebooks juntan el código, la documentación y los resultados de la ejecución del código, ya sea como texto, imagen o HTML / JavaScript. Al igual que un editor de código o IDE, los notebooks le ayudan a escribir código: le permiten ejecutar código de manera interactiva e iterativa, representando los resultados junto con el código. Además, cuando comparte un notebooks con miembros del equipo, puede incluir código, documentación y resultados que incluyen gráficos interactivos, para proporcionarles un contexto que va más allá de lo que solo los archivos de código SQL o Python pueden proporcionar.

Los notebooks de Cloud Datalab pueden almacenarse en Google Cloud Source Repository, un repositorio git. Este repositorio de git se clona en un disco persistente conectado a la máquina virtual. Este clon forma su espacio de trabajo, donde puede agregar, eliminar y modificar archivos. Para compartir su trabajo con otros usuarios del repositorio, usted compromete sus cambios usando el cliente git para enviar sus cambios desde este espacio de trabajo local al repositorio. Los notebooks se guardan automáticamente en un disco persistente periódicamente, y puede guardarlos cuando lo desee. Tenga en cuenta que si elimina

el disco persistente, se perderán los notebooks que no se envíen explícitamente al repositorio de git. Por lo tanto, recomendamos encarecidamente que NO elimine el disco persistente.

Cuando abre un notebook, se inicia un proceso de "kernel" de fondo para administrar las variables definidas durante la sesión y ejecutar el código de su notebooks. Cuando el código ejecutado accede a los servicios de Google Cloud, como BigQuery o Google Machine Learning Engine, utiliza la cuenta de servicio disponible en la VM. Por lo tanto, la cuenta del servicio debe estar autorizada para acceder a los datos o solicitar el servicio. Para mostrar los nombres de las cuentas de servicios y proyectos en la nube, haga clic en el ícono de usuario del ícono del usuario en la esquina superior derecha de la lista de notebooks o cuadernos de Cloud Datalab en su navegador (es posible que necesite cambiar el tamaño de la ventana del navegador). La máquina virtual utilizada para ejecutar Cloud Datalab es un recurso compartido al que pueden acceder todos los miembros del proyecto asociado de la nube. Por lo tanto, no se recomienda utilizar las credenciales de nube personal de una persona para acceder a los datos.

A medida que ejecuta código en el notebook, el estado del proceso que ejecuta el código cambia. Si asigna o reasigna una variable, su valor se usa para cálculos posteriores como un efecto secundario. Cada notebook en ejecución se muestra como una sesión en Cloud Datalab. Puede hacer clic en el ícono de sesión del ícono de sesión en la página de lista de notebook de Cloud Datalab para listar y detener sesiones. Mientras se ejecuta una sesión, el proceso subyacente consume recursos de memoria. Si detiene una sesión, el proceso subyacente desaparece junto con su estado en memoria y se libera la memoria utilizada por la sesión. Los resultados guardados en el notebook permanecen en formato persistente en el disco.

### **Escenarios de uso de Cloud Datalab**

Cloud Datalab es un entorno interactivo de análisis de datos y aprendizaje automático diseñado para Google Cloud Platform. Puede usarlo para explorar, analizar, transformar y visualizar sus datos de forma interactiva y para construir modelos de Machine Learning a partir de sus datos. En la carpeta Cloud Datalab / docs, encontrará una serie de tutoriales y muestras que ilustran algunas de las tareas que puede realizar. Cloud Datalab incluye un conjunto de bibliotecas de Python de código abierto utilizadas comúnmente para el análisis de datos, la visualización y el Machine Learning. También agrega bibliotecas para acceder a servicios clave de Google Cloud Platform, como Google BigQuery, Google Machine Learning Engine, Google Dataflow y Google Cloud Storage.

Aquí hay algunas ideas para comenzar:

- Escriba algunas consultas SQL para explorar los datos en BigQuery. Coloque los resultados en un Dataframe y visualícelos como un histograma o un gráfico de líneas.
- Lea los datos de un archivo CSV en Google Cloud Storage y colóquelos en un Dataframe para calcular medidas estadísticas como la media, la desviación estándar y los cuantiles utilizando Python.
- Pruebe un modelo TensorFlow o 1scikit-learn1 para predecir resultados o clasificar datos.

### Bibliotecas incluidas

La siguiente es una lista de bibliotecas incluidas y disponibles para usted en los notebooks de Cloud Datalab (la lista de la biblioteca y la información de la versión están sujetas a cambios):

```
argparse at version 1.2.1
bs4 at version 0.0.1
crcmod at version 1.7
future at version 0.16.0
futures at version 3.0.5
ggplot at version 0.6.8
google-api-python-client at version 1.5.1
google-cloud at version 0.19.0
httplib2 at version 0.9.2
ipykernel at version 4.4.1
ipywidgets at version 5.2.2
jinja2 at version 2.8
jsonschema at version 2.6.0
matplotlib at version 1.5.3
mock at version 2.0.0
nltk at version 3.2.1
notebook at version 4.2.3
numpy at version 1.11.2
oauth2client at version 2.2.0
pandas at version 0.19.1
pandas-profiling at version at least 1.0.0a2
pandocfilters at version 1.3.0
pillow at version 3.4.1
plotly at version 1.12.5
psutil at version 4.3.0
pygments at version 2.1.3
python-dateutil at version 2.5.0
pytz at version 2016.7
PyYAML at version 3.11
pyzmq at version 16.0.2
requests at version 2.9.1
```



scikit-learn at version 0.17.1

scipy at version 0.18.0

seaborn at version 0.7.0

six at version 1.10.0

statsmodels at version 0.6.1

sympy at version 0.7.6.1

tornado at version 4.4.2

## ● Google Cloud DataStore.

- **NoSQL PaaS:**
- Google Cloud DataStore.

### **Descripción general de Google Cloud Datastore**

Google Cloud Datastore es una base de datos de documentos NoSQL creada para escalar de forma automática, alto rendimiento y facilidad de desarrollo de aplicaciones.

Las características de Cloud Datastore incluyen:

- Transacciones atómicas: Cloud Datastore puede ejecutar un conjunto de operaciones donde todas tienen éxito o ninguna.
- Alta disponibilidad de lecturas y escrituras. Cloud Datastore se ejecuta en los centros de datos de Google, que utilizan la redundancia para minimizar el impacto de los puntos de falla.
- Escalabilidad masiva con alto rendimiento. Cloud Datastore utiliza una arquitectura distribuida para administrar automáticamente la escala. Cloud Datastore usa una combinación de índices y restricciones de consulta para que sus consultas se escalen con el tamaño de su conjunto de resultados, no el tamaño de su conjunto de datos.
- Almacenamiento flexible y consulta de datos. Cloud Datastore se asigna de forma natural a lenguajes orientados a objetos y scripting, y se expone a aplicaciones a través de múltiples clientes. También proporciona un lenguaje de consulta similar a SQL.
- Equilibrio de consistencia fuerte y eventual. Cloud Datastore garantiza que las búsquedas de entidades por clave y las consultas de ancestros siempre reciban datos consistentes. Todas las demás consultas son finalmente consistentes. Los modelos de coherencia permiten que su aplicación proporcione una excelente experiencia de usuario al manejar grandes cantidades de datos y usuarios.
- Cifrado en reposo. Cloud Datastore encripta automáticamente todos los datos antes de que se escriban en el disco y descifra automáticamente los datos cuando los lee un usuario autorizado. Para obtener más información, vea Cifrado del lado del servidor.
- Totalmente administrado sin tiempo de inactividad planeado. Google maneja la administración del servicio Cloud Datastore para que pueda enfocarse en su aplicación. Su aplicación aún puede usar Cloud Datastore cuando el servicio recibe una actualización planificada.

### **Comparación con bases de datos tradicionales**

Si bien la interfaz de Cloud Datastore tiene muchas de las mismas características que las bases de datos tradicionales, como base de datos NoSQL difiere de ellas en la forma en que describe las relaciones entre los objetos de datos. Aquí hay una comparación de alto nivel de Cloud Datastore y conceptos de bases de datos relacionales:

| Concept                       | Cloud Datastore | Relational database |
|-------------------------------|-----------------|---------------------|
| Category of object            | Kind            | Table               |
| One object                    | Entity          | Row                 |
| Individual data for an object | Property        | Field               |
| Unique ID for an object       | Key             | Primary key         |

A diferencia de las filas en una tabla de base de datos relacional, las entidades de Cloud Datastore del mismo tipo pueden tener diferentes propiedades, y diferentes entidades pueden tener propiedades con el mismo nombre pero diferentes tipos de valores. Estas características únicas implican una forma diferente de diseñar y administrar datos para aprovechar la capacidad de escalar automáticamente. En particular, Cloud Datastore difiere de una base de datos relacional tradicional en las siguientes formas importantes:

- Cloud Datastore está diseñado para escalar automáticamente a conjuntos de datos muy grandes, lo que permite que las aplicaciones mantengan un alto rendimiento a medida que reciben más tráfico:
- Cloud Datastore escribe y escala distribuyendo automáticamente datos según sea necesario.
- Cloud Datastore lee la escala porque las únicas consultas admitidas son aquellas cuyo rendimiento se escala con el tamaño del conjunto de resultados (a diferencia del conjunto de datos). Esto significa que una consulta cuyo conjunto de resultados contiene 100 entidades realiza el mismo si busca más de cien entidades o un millón. Esta propiedad es la razón clave por la que algunos tipos de consultas no son compatibles.
- Debido a que todas las consultas son atendidas por índices previamente construidos, los tipos de consultas que se pueden ejecutar son más restrictivas que las permitidas en una base de datos relacional con SQL. En particular, Cloud Datastore no incluye soporte para operaciones de combinación, filtrado de desigualdad en múltiples propiedades o filtrado de datos en función de los resultados de una subconsulta.
- A diferencia de las bases de datos relacionales tradicionales que imponen un esquema, Cloud Datastore no requiere entidades del mismo tipo para tener un conjunto de propiedades consistente (aunque puede optar por hacer cumplir dicho requisito en su propio código de aplicación).

### Para qué es bueno

Cloud Datastore es ideal para aplicaciones que dependen de datos estructurados de alta disponibilidad a escala.

Puedes usar Cloud Datastore para almacenar y consultar todos los tipos de datos siguientes:

- Catálogos de productos que proporcionan inventario en tiempo real y detalles del producto para un minorista.
- Perfiles de usuario que entregan una experiencia personalizada basada en las actividades y preferencias pasadas del usuario.
- Transacciones basadas en propiedades de ACID, por ejemplo, transfiriendo fondos de una cuenta bancaria a otra.

### Otras opciones de almacenamiento

Cloud Datastore no es ideal para cada caso de uso. Por ejemplo, Cloud Datastore no es una base de datos relacional, y no es una solución de almacenamiento efectiva para datos analíticos.

Aquí hay algunos escenarios comunes en los que probablemente debería considerar una alternativa al Cloud Datastore:

- Si necesitas una base de datos relacional con soporte SQL completo para un sistema de procesamiento de transacciones en línea (OLTP), considere Google Cloud SQL.
- Si no necesitas soporte para transacciones ACID o si tus datos no están muy estructurados, considera Cloud Bigtable.
- Si necesitas consultas interactivas en un sistema de procesamiento analítico en línea (OLAP), considera Google BigQuery.
- Si necesitas almacenar grandes manchas inmutables, como imágenes grandes o películas, considera Google Cloud Storage.

### Ubicaciones de Google Cloud Datastore

Cuando utilizas Cloud Datastore por primera vez, debes elegir una ubicación donde se almacenan los datos del proyecto. Para reducir la latencia y aumentar la disponibilidad, almacena sus datos cerca de los usuarios y servicios que lo necesitan.

#### Tipos de ubicaciones

Hay dos tipos de ubicaciones donde puedes almacenar datos usando Cloud Datastore: ubicaciones de múltiples regiones y ubicaciones regionales.

Las ubicaciones de varias regiones brindan redundancia de múltiples regiones y mayor disponibilidad. Las ubicaciones regionales ofrecen una latencia de escritura y una ubicación conjunta más bajas con otros recursos de Google Cloud Platform que tu aplicación puede usar.

#### Ubicaciones multi-región

Una ubicación multi-región es un área geográfica general, como los Estados Unidos. Las ubicaciones de varias regiones contienen ubicaciones regionales múltiples.

Las siguientes ubicaciones de múltiples regiones están disponibles:

| Multi-Region Name | Multi-Region Description |
|-------------------|--------------------------|
| europa-west       | Europe                   |
| us-central        | United States            |

#### Ubicación regional

Una ubicación regional es un lugar geográfico específico, como Carolina del Sur. Las siguientes ubicaciones regionales están disponibles:

| Region Name          | Region Description |
|----------------------|--------------------|
| North America        |                    |
| us-east1             | South Carolina     |
| us-east4             | Northern Virginia  |
| South America        |                    |
| southamerica-east1 * | São Paulo          |
| Europe               |                    |
| eu-west-2            | London             |
| eu-west-3            | Frankfurt          |
| Asia                 |                    |
| asia-northeast1      | Tokyo              |
| asia-south1          | Mumbai             |
| Australia            |                    |
| australia-southeast1 | Sydney             |

### Elegir una ubicación

Debes especificar una ubicación para su proyecto antes de usar Cloud Datastore, y después de seleccionar una ubicación para su proyecto, no puedes cambiarlo.

Después de crear un proyecto, visita la página Entidades de almacén de datos y haga clic en Probar> Crear entidad para establecer la ubicación. El menú del selector de ubicación, que se muestra a continuación, aparece solo si aún no ha seleccionado una ubicación.

1 Select a location

2 Create an entity

## Where would you like to store your data?

Select a region for your Datastore. To reduce latency, choose a region near the applications that need your data. You can't change the region for this project later.

**App Engine:** If you plan to use App Engine in this project, note that your App Engine app will also be hosted in the region you select here.



### Select a region

Some regions are restricted by this project's organization policy.

us-central

Next

### Cifrado del lado del servidor

Google Cloud Datastore cifra automáticamente todos los datos antes de que se escriban en el disco.

No se requiere configuración o configuración y no es necesario modificar la forma de acceder al servicio.

Los datos se descifran de forma automática y transparente cuando los lee un usuario autorizado.

Con el cifrado del lado del servidor, Google administra las claves criptográficas en su nombre utilizando los mismos sistemas de administración de claves reforzados que usamos para nuestros propios datos cifrados, incluidos los controles de acceso de claves estrictas y la auditoría. Los datos y metadatos de cada objeto de Cloud Datastore están encriptados según el Estándar de cifrado avanzado, y cada clave de cifrado se encripta con un conjunto de claves maestras giradas regularmente.

El cifrado del lado del servidor se puede usar en combinación con el cifrado del lado del cliente. En el cifrado del lado del cliente, usted administra sus propias claves de cifrado y encripta los datos antes de escribirlos en Cloud Datastore. En este caso, tus datos se cifran dos veces, una vez con tus claves y una vez con las claves de Google.

## **MultiTenant**

Puede admitir la propiedad multiempresa en su aplicación al proporcionar particiones de datos separadas para varias organizaciones de clientes, conocidas como tenants. Esto te permite personalizar los valores de datos para cada tenant, manteniendo el mismo esquema de datos para todos los tenant. Esto hace que el aprovisionamiento de nuevos tenant sea más eficiente porque no tiene que cambiar la estructura de datos cuando agrega un tenant.

### **Beneficios de multitenancy**

Google Cloud Datastore permite que una aplicación multiusuario use silos separados de datos para cada tenant mientras sigue usando:

- un solo proyecto
- una sola estructura lógica para los tipos
- un único conjunto de definiciones de índice, porque los tipos son los mismos lógicamente para cada tenant
- Cloud Datastore habilita multitenancy proporcionando espacios de nombres. Multitenancy también funciona para otras API de Google App Engine habilitadas para espacios de nombres (Go, Java, Python).

### **Multiempresa y datos particionados**

Cloud Datastore usa particiones para datos del silo para cada tenant. La combinación de una ID de proyecto y una ID de espacio de nombres forma una ID de partición, que identifica cada partición. Una entidad pertenece a una única partición, y las consultas tienen un ámbito en una sola partición.

Los espacios de nombres no son un mecanismo de seguridad en Cloud Datastore. Un usuario con acceso a una partición en un proyecto tiene acceso a todas las particiones en el proyecto. Los espacios de nombres proporcionan una forma de organizar sus entidades dentro de un proyecto.

### **Especificando un espacio de nombre para una entidad**

Especifica el espacio de nombres cuando crea la entidad: después de crear la entidad, no puede cambiar el espacio de nombres. Si no especifica explícitamente un espacio de nombre para una entidad, se asigna automáticamente al espacio de nombre predeterminado, que no tiene identificador de cadena.

### Usar espacios de nombres con entidades padre

Una entidad y todos sus antepasados pertenecen a un solo espacio de nombres. Esto significa que cuando crea una entidad con otra entidad designada como principal, la entidad hijo está en el mismo espacio de nombres que su elemento primario: no puede especificar otro espacio de nombres.

### Caso de uso de Ejemplo

Un beneficio clave de multitenancy es tener la misma aplicación para múltiples organizaciones de clientes. Para lograr este beneficio, para un tipo determinado, su aplicación debe comportarse igual independientemente del espacio de nombres. Por ejemplo, desde la perspectiva de la aplicación, una entidad de tipo Tarea en un espacio de nombre lógicamente debería ser la misma que una entidad de tipo Tarea en todos los demás espacios de nombres. Su aplicación podría utilizar un único conjunto de definiciones de índice para admitir consultas de tareas, independientemente de qué espacios de nombres contengan entidades de Tarea.

Por ejemplo, considere una aplicación de la Lista de tareas que silos datos por usuario. La aplicación podría definir espacios de nombres basados en el nombre de usuario, lo que da como resultado las siguientes particiones:

```
Partition ID: project:"my_project_id"/namespace:"Joe"
```

```
Partition ID: project:"my_project_id"/namespace:"Alice"
```

```
Partition ID: project:"my_project_id"/namespace:"Charlie"
```

La aplicación podría definir una estructura lógica de un tipo de tarea de la siguiente manera, para usar en todos los espacios de nombres:

```
kind: Task
```

```
properties:
```

- "done", Boolean
- "created", DateTime
- "description", String, excluded from index

Cuando un usuario crea una entidad de tipo Tarea, la entidad se almacena en la propia partición del usuario, lo que da como resultado datos en silos. La aplicación procesa las entidades de tareas de forma consistente a través de espacios de nombres porque solo se utiliza un esquema para el tipo de tarea. Una aplicación con datos en silos y un comportamiento consistente sería multiusuario.

Si la estructura lógica de un tipo de tarea difiere según el espacio de nombres, la aplicación no sería multiusuario porque procesa las entidades de tareas de manera diferente en los espacios de nombres. Por ejemplo, considere los tipos de tareas que tienen un esquema diferente basado en el espacio de nombres:



Las entidades de tareas en el espacio de nombres Joe excluyen la propiedad de descripción del índice

Las entidades de tarea en el índice Alice incluyen la propiedad de descripción del índice

La aplicación podría consultar en la propiedad de descripción para las entidades de Tarea de Alicia, pero no podría consultar en la propiedad de descripción para las entidades de Tarea de Joe, por lo que la aplicación no sería multiusuario.

## Recomendaciones con DataStore

### General

Utiliza siempre caracteres UTF-8 para nombres de espacios de nombres, nombres de tipo, nombres de propiedades y nombres de teclas personalizadas. Los caracteres que no son UTF-8 utilizados en estos nombres pueden interferir con la funcionalidad de Cloud Datastore. Por ejemplo, un carácter que no sea UTF-8 en un nombre de propiedad puede evitar la creación de un índice que use la propiedad.

No use una barra inclinada (/) en nombres bonitos o nombres de teclas personalizadas. Las barras inclinadas hacia adelante en estos nombres podrían interferir con la funcionalidad futura.

Evite almacenar información confidencial en una ID de Cloud Project. Un ID de Proyecto en la Nube podría ser retenido más allá de la duración de su proyecto.

### Llamadas al API

- Usa las operaciones por lotes para sus lecturas, escrituras y eliminaciones en lugar de operaciones únicas. Las operaciones por lotes son más eficientes porque realizan múltiples operaciones con la misma sobrecarga que una sola operación.
- Si una transacción falla, asegúrese de intentar deshacer la transacción. La reversión minimiza la latencia de reintentos para una solicitud diferente que compita por el mismo recurso (s) en una transacción. Tenga en cuenta que una reversión en sí misma puede fallar, por lo que la reversión debe ser solo un intento de mejor esfuerzo.
- Usa llamadas asíncronas donde estén disponibles en lugar de llamadas sincrónicas. Las llamadas asíncronas minimizan el impacto de latencia. Por ejemplo, considera una aplicación que necesita el resultado de una búsqueda () y los resultados de una consulta antes de que pueda generar una respuesta. Si la búsqueda () y la consulta no tienen una dependencia de datos, no hay necesidad de esperar sincrónicamente hasta que la búsqueda () finalice antes de iniciar la consulta.

### Entidades

- Agrupa datos altamente relacionados en grupos de entidades. Los grupos de entidades habilitan las consultas de ancestros, que devuelven resultados muy consistentes. Las consultas de ancestros también escanean rápidamente un grupo de entidades con E / S mínima porque las entidades en un grupo de entidades se almacenan en lugares físicamente cercanos en los servidores de Cloud Datastore.
- Evita escribir en un grupo de entidades más de una vez por segundo. Escribir a un ritmo sostenido por encima de ese límite hace que las lecturas coherentes eventualmente sean más eventuales, conduce a tiempos de espera para lecturas muy consistentes y da como resultado un rendimiento

general más lento de su aplicación. Una escritura por lotes o transaccional en un grupo de entidades cuenta como una sola escritura contra este límite.

- No incluyas la misma entidad (por clave) varias veces en la misma confirmación. Incluir la misma entidad varias veces en el mismo compromiso podría afectar la latencia de Cloud Datastore.

### Claves

- Para una clave que usa un nombre personalizado, siempre use caracteres UTF-8 excepto una barra inclinada (/). Los caracteres que no son UTF-8 interfieren con varios procesos, como la importación de una copia de seguridad de Cloud Datastore en Google BigQuery. Una barra inclinada podría interferir con la funcionalidad futura.
- Para una clave que usa una ID numérica:
  - No uses un número negativo para la ID. Una identificación negativa podría interferir con la clasificación.
  - No uses el valor 0 (cero) para la ID. Si lo haces, obtendrás una identificación asignada automáticamente.
  - Si deseas asignar sus propios ID numéricos manualmente a las entidades que crea, haz que tu aplicación obtenga un bloque de ID con el método `allocateIds()`. Esto evitará que Cloud Datastore asigne uno de tus ID numéricos manuales a otra entidad.
- Si asigna su propia identificación numérica manual o nombre personalizado a las entidades que crea, no use valores monótonamente crecientes, tales como:

1, 2, 3, ...,

"Customer1", "Customer2", "Customer3", ....

"Product 1", "Product 2", "Product 3", ....

Si una aplicación genera un gran tráfico, dicha numeración secuencial podría generar hotspots que afecten la latencia de Cloud Datastore. Para evitar el problema de los ID numéricos secuenciales, obtenga ID numéricos del método `allocateIds()`. El método `allocateIds()` genera secuencias bien distribuidas de ID numéricos.

### Índices

- Si una propiedad nunca será necesaria para una consulta, excluya la propiedad de los índices. Indizar innecesariamente una propiedad podría resultar en una mayor latencia para lograr consistencia y mayores costos de almacenamiento de las entradas de índice.
- Evita tener demasiados índices compuestos. El uso excesivo de índices compuestos podría resultar en una mayor latencia para lograr consistencia y mayores costos de almacenamiento de las entradas de índice. Si necesita ejecutar consultas ad hoc en grandes conjuntos de datos sin índices previamente definidos, use Google BigQuery.

- No indexes propiedades con valores monótonamente crecientes (como una marca de tiempo NOW ()). El mantenimiento de dicho índice podría generar hotspots que afectan la latencia de Cloud Datastore para aplicaciones con altas tasas de lectura y escritura.

### Consultas

- Si necesitas acceder solo a la clave desde los resultados de la consulta, use una consulta de solo claves. Una consulta de solo claves devuelve resultados a menor latencia y costo que recuperar entidades enteras.
- Si necesitas acceder solo a propiedades específicas de una entidad, use una consulta de proyección. Una consulta de proyección arroja resultados a menor latencia y costo que recuperar entidades enteras.
- Del mismo modo, si necesitas acceder solo a las propiedades que se incluyen en el filtro de consulta (por ejemplo, las enumeradas en una cláusula order by), use una consulta de proyección.
- No uses offsets. En su lugar usa cursores. El uso de un desplazamiento solo evita devolver las entidades omitidas a su aplicación, pero estas entidades aún se recuperan internamente. Las entidades omitidas afectan la latencia de la consulta y su aplicación recibe una factura por las operaciones de lectura requeridas para recuperarlas.
- Si necesitas una coherencia sólida para sus consultas, use una consulta ancestro. (Para utilizar las consultas antecesoras, primero necesita estructurar sus datos para lograr una mayor consistencia). Una consulta antecesora arroja resultados muy consistentes. Tenga en cuenta que una consulta de solo claves no ancestrales seguida de una búsqueda () no arroja resultados sólidos, porque la consulta de solo claves no ancestrales podría obtener resultados de un índice que no es consistente en el momento de la consulta.

### Diseñando para escalar

Actualizaciones a un solo grupo de entidades

Un solo grupo de entidades en Cloud Datastore no debe actualizarse demasiado rápido.

Si está utilizando Cloud Datastore, Google recomienda que diseñe su aplicación para que no necesite actualizar un grupo de entidades más de una vez por segundo. Recuerde que una entidad sin padres y sin hijos es su propio grupo de entidades. Si actualiza un grupo de entidades con demasiada rapidez, sus grabaciones de Cloud Datastore tendrán una mayor latencia, tiempos de espera y otros tipos de errores. Esto se conoce como contención.

Las tasas de escritura de Cloud Datastore en un solo grupo de entidades a veces pueden exceder el límite de una por segundo, por lo que las pruebas de carga pueden no mostrar este problema. Algunas sugerencias para diseñar su aplicación para reducir las tasas de escritura en grupos de entidades se encuentran en el artículo de tienda Cloud Datastore.

### Altas tasas de lectura / escritura en un rango de Claves pequeño

Evite las altas tasas de lectura o escritura en las claves de Cloud Datastore que están lexicográficamente cercanas.

Cloud Datastore está construido sobre la base de datos NoSQL de Google, Bigtable, y está sujeto a las características de rendimiento de Bigtable. Bigtable escalas agrupando filas en tabletas separadas, y estas filas se ordenan lexicográficamente por clave.

Si estás utilizando Cloud Datastore, puede obtener grabaciones lentas debido a una tableta caliente si tiene un aumento repentino en la velocidad de escritura a un rango pequeño de claves que excede la capacidad de un solo servidor de tableta. Bigtable eventualmente dividirá el espacio clave para soportar una carga alta.

El límite para las lecturas suele ser mucho mayor que para las escrituras, a menos que esté leyendo desde una sola clave a una tasa alta.

En algunos casos, un hotspot de Cloud Datastore puede tener un impacto más amplio en una aplicación que la prevención de lecturas o escrituras en una pequeña gama de claves. Por ejemplo, las claves de acceso rápido pueden leerse o escribirse durante el inicio de la instancia, lo que hace que las solicitudes de carga fallen.

De forma predeterminada, Cloud Datastore asigna claves utilizando un algoritmo disperso. Por lo tanto, normalmente no encontrará hotspot en las escrituras de Cloud Datastore si crea nuevas entidades con una alta tasa de escritura utilizando la política de asignación de ID predeterminada.

Hay algunos casos donde puede resolver este problema:

- Si creas nuevas entidades a una tasa muy alta utilizando la política de asignación de ID secuencial heredada.
- Si creas entidades nuevas a un ritmo muy elevado, asignará sus propios ID que aumentarán monótonamente.
- Si creas entidades nuevas a una tasa muy alta para un tipo que anteriormente tenía muy pocas entidades existentes. Bigtable comenzará con todas las entidades en el mismo servidor de tabletas y se tomará un tiempo para dividir el rango de claves en servidores de tabletas por separado.
- También verás este problema si crea nuevas entidades a gran velocidad con una propiedad indexada que aumenta monótonamente, como una marca de tiempo, porque estas propiedades son las claves para las filas en las tablas de índice en Bigtable.
- Cloud Datastore antepone el espacio de nombres y el tipo del grupo de entidades raíz a la clave de fila de Bigtable. Puedes presionar un punto de acceso si comienzas a escribir en un nuevo espacio de nombres o tipo sin aumentar gradualmente el tráfico.

Si tienes una clave o propiedad indexada que aumenta monótonamente, puedes anteponer un hash aleatorio para asegurarse de que las claves estén fragmentadas en varias tabletas.

## Borrados

Evita eliminar grandes cantidades de entidades Cloud Datastore en una pequeña gama de claves.

Bigtable reescribe periódicamente sus tablas para eliminar las entradas eliminadas y reorganizar sus datos para que las lecturas y escrituras sean más eficientes. Este proceso se conoce como compactación.

Si eliminas una gran cantidad de entidades Cloud Datastore en una pequeña gama de claves, las consultas en esta parte del índice serán más lentas hasta que se complete la compactación. En casos extremos, tus consultas podrían expirar antes de devolver los resultados.

Es un antipatrón el usar un valor de marca de tiempo para que un campo indexado represente el tiempo de caducidad de una entidad. Para recuperar entidades caducadas, necesitarías realizar una consulta en contra de este campo indexado, que probablemente se encuentre en una parte superpuesta del espacio de claves con entradas de índice para las entidades eliminadas más recientemente.

Puedes mejorar el rendimiento con "consultas fragmentadas", que anteponen una cadena de longitud fija a la marca de tiempo de caducidad. El índice se ordena en la cadena completa, de modo que las entidades en la misma marca de tiempo se ubicaran en todo el rango de claves del índice. Ejecuta múltiples consultas en paralelo para obtener resultados de cada fragmento.

Una solución más completa para el problema de la fecha y hora de vencimiento es usar un "número de generación" que es un contador global que se actualiza periódicamente. El número de generación se antepone al sello de tiempo de expiración para que las consultas se clasifiquen por número de generación, luego fragmento y, a continuación, marca de tiempo. La eliminación de entidades antiguas ocurre en una generación anterior. Cualquier entidad no eliminada debe tener su número de generación incrementado. Una vez que se completa la eliminación, avanza a la siguiente generación. Las consultas en contra de una generación anterior tendrán un rendimiento pobre hasta que se complete la compactación. Es posible que debas esperar varias generaciones para completar antes de consultar el índice para obtener la lista de entidades para eliminar, a fin de reducir el riesgo de que falten resultados debido a la coherencia final.

## Sharding y replicación

Usa sharding o replicación para las claves de Cloud Datastore.

Puedes usar la replicación si necesita leer una porción del rango de clave a una velocidad mayor que la permitida por Bigtable. Utilizando esta estrategia, almacenaría N copias de la misma entidad, lo que permite una tasa de lectura N veces mayor que la soportada por una sola entidad.

Puedes usar sharding si necesita escribir en una porción del rango de clave a una tasa mayor que la permitida por Bigtable. Sharding divide una entidad en pedazos más pequeños. Los principios se explican en el artículo Contadores de fusión.

Algunos errores comunes cuando se fragmentan incluyen:

- Sharding usando un prefijo de tiempo. Cuando el tiempo pasa al siguiente prefijo, la nueva parte sin dividir se convierte en un punto de acceso. En su lugar, debes pasar gradualmente una porción de sus escrituras al nuevo prefijo.
- Sharding solo las entidades más populares. Si reduces una pequeña proporción del número total de entidades, es posible que no haya suficientes filas entre las entidades activas para garantizar que permanezcan en diferentes divisiones.

## ● Google Cloud BigTable.

- **NoSql PaaS:**
- Google Cloud BigTable.

### **Cloud Bigtable**

Cloud Bigtable es el servicio de base de datos NoSQL Big Data de Google.

Es la misma base de datos que impulsa muchos servicios básicos de Google, incluidos la Búsqueda, los Análisis, los Mapas y Gmail.

Cloud Bigtable puede escalar a miles de millones de filas y miles de columnas, lo que le permite almacenar terabytes o incluso petabytes de datos. Un único valor en cada fila está indexado; este valor se conoce como la clave de fila. Cloud Bigtable es ideal para almacenar grandes cantidades de datos de una sola tecla con muy baja latencia. Admite un alto rendimiento de lectura y escritura a baja latencia, y es una fuente de datos ideal para las operaciones de MapReduce.

Cloud Bigtable está expuesto a aplicaciones a través de múltiples clientes, incluida una extensión compatible con la biblioteca Java Apache HBase 1.x. Como resultado, se integra con el ecosistema Apache existente del software Big Data de código abierto.

Los servidores back-end de Cloud Bigtable ofrecen varias ventajas clave sobre una instalación HBase autogestionada:

- **Increíble escalabilidad** Cloud Bigtable se escala en proporción directa a la cantidad de máquinas en su clúster. Una instalación de HBase autogestionada tiene un cuello de botella de diseño que limita el rendimiento después de alcanzar cierto QPS. Cloud Bigtable no tiene este cuello de botella, por lo que puede ampliar su clúster para manejar más consultas al aumentar el recuento de su máquina.
- **Administración simple.** Cloud Bigtable maneja las actualizaciones y reinicia de forma transparente, y mantiene automáticamente una alta durabilidad de los datos. No más maestros de gestión, regiones, clusters o nodos; todo lo que necesita hacer es diseñar sus esquemas de tabla, y Cloud Bigtable se encargará del resto por usted.
- **Cambio de tamaño del clúster sin tiempo de inactividad.** Puede aumentar el tamaño de su clúster de Cloud Bigtable durante unas horas para manejar una carga grande y luego reducir el tamaño del clúster de nuevo, todo ello sin ningún tiempo de inactividad. Después de cambiar el tamaño de un clúster, Cloud Bigtable tarda pocos minutos en cargar para equilibrar el rendimiento en todos los nodos de su clúster.

### **Para qué es bueno BigTable**

Cloud Bigtable es ideal para aplicaciones que requieren un rendimiento y una escalabilidad muy altos para datos de clave / valor no estructurados, donde cada valor normalmente no es superior a 10 MB. Cloud Bigtable también sobresale como un motor de almacenamiento para operaciones MapReduce por lotes, procesamiento / análisis de flujo y aplicaciones de aprendizaje automático.

Puede usar Cloud Bigtable para almacenar y consultar todos los tipos de datos siguientes:

- Datos de marketing, como historial de compras y preferencias del cliente.
- Datos financieros como historiales de transacciones, precios de acciones y tasas de cambio de divisas.
- Datos de Internet of Things, como informes de uso de contadores de energía y electrodomésticos.
- Datos de series de tiempo como el uso de CPU y memoria a lo largo del tiempo para múltiples servidores.

### **Modelo de almacenamiento Cloud Bigtable**

Cloud Bigtable almacena datos en tablas masivamente escalables, cada una de las cuales es un mapa clave / valor ordenado. La tabla está compuesta de filas, cada una de las cuales generalmente describe una sola entidad, y columnas, que contienen valores individuales para cada fila. Cada fila está indexada por una sola clave de fila, y las columnas que están relacionadas entre sí normalmente se agrupan en una familia de columnas. Cada columna se identifica mediante una combinación de la familia de columnas y un calificador de columna, que es un nombre único dentro de la familia de columnas.

Cada intersección de fila / columna puede contener varias celdas en diferentes marcas de tiempo, proporcionando un registro de cómo los datos almacenados se han modificado a lo largo del tiempo. Las tablas de Cloud Bigtable son dispersas; si una celda no contiene ningún dato, no ocupa espacio.

Por ejemplo, supongamos que está creando una red social para presidentes de los Estados Unidos, llámémosle Prezzy. Cada presidente puede seguir mensajes de otros presidentes. La siguiente ilustración muestra una tabla Cloud Bigtable que rastrea a cada presidente en Prezzy:



"follows" column family

|            | Follows    |        |            |           |
|------------|------------|--------|------------|-----------|
| Row Key    | gwasington | jadams | tjefferson | wmckinley |
| gwasington |            | 1      |            |           |
| jadams     | 1          |        | 1          |           |
| tjefferson | 1          | 1      |            | 1         |
| wmckinley  |            |        | 1          |           |

Multiple versions

Algunas cosas para notar en esta ilustración:

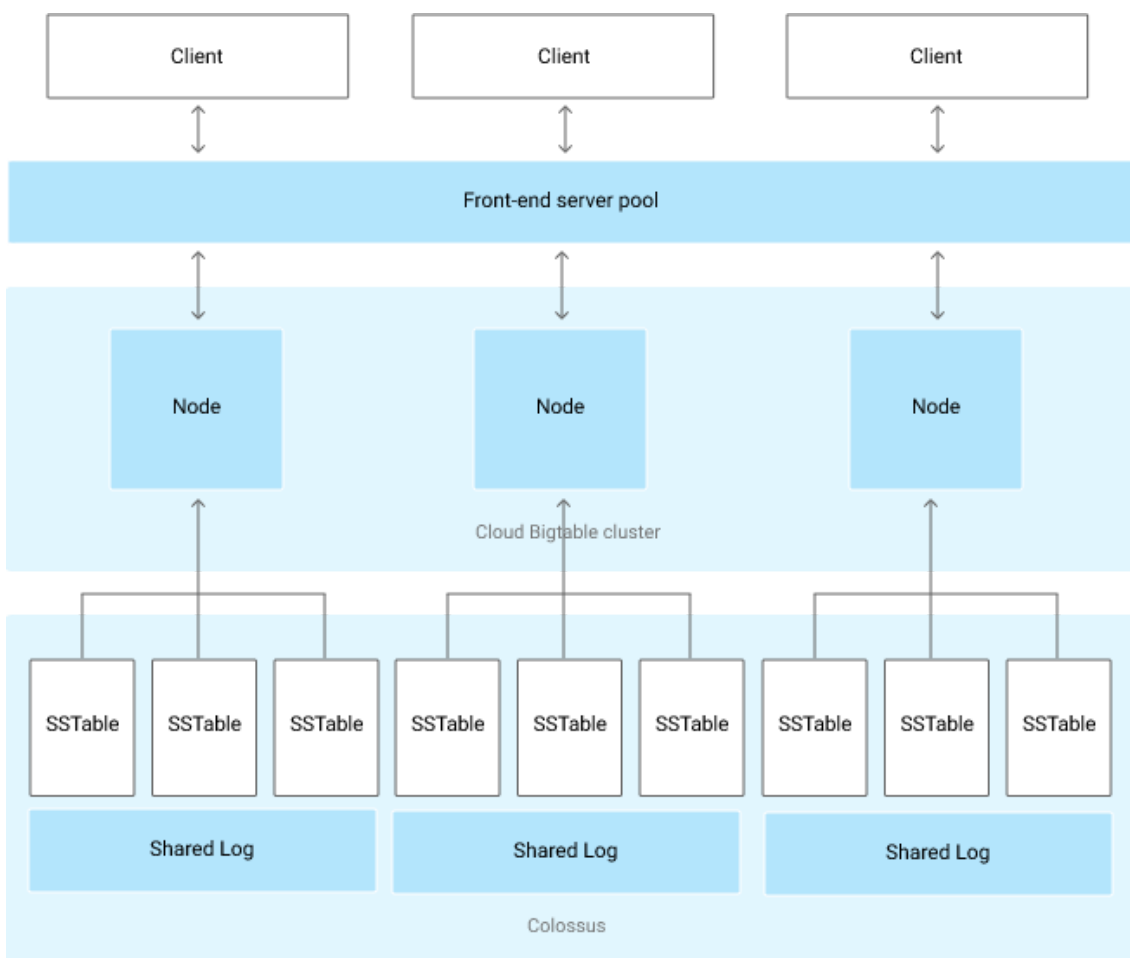
La tabla contiene una familia de columnas, la siguiente familia. Esta familia contiene calificadores de múltiples columnas.

Los calificadores de columna se usan como datos. Esta elección de diseño aprovecha la escasez de las tablas de Cloud Bigtable y el hecho de que los nuevos calificadores de columna se pueden agregar sobre la marcha.

El nombre de usuario se usa como la clave de fila. Suponiendo que los nombres de usuario se distribuyan uniformemente en el alfabeto, el acceso a los datos será razonablemente uniforme en toda la tabla. Consulte "Equilibrio de carga" para obtener más detalles sobre cómo Bigtable maneja cargas desiguales y "Elección de una clave de fila" para obtener sugerencias más avanzadas sobre cómo seleccionar una clave de fila.

### Arquitectura Cloud Bigtable

El siguiente diagrama muestra una versión simplificada de la arquitectura general de Cloud Bigtable:



Como se ilustra en el diagrama, todas las solicitudes de los clientes pasan por un servidor de aplicaciones para el usuario antes de enviarlas a un nodo Cloud Bigtable. (En el documento original de Bigtable original, estos nodos se denominan "servidores de tabletas"). Los nodos están organizados en un clúster Cloud Bigtable, que pertenece a una instancia de Cloud Bigtable, un contenedor para el clúster. Cada nodo del clúster maneja un subconjunto de las solicitudes al clúster. Al agregar nodos a un clúster, puede aumentar el número de solicitudes simultáneas que puede manejar el clúster, así como el rendimiento máximo para todo el clúster.

Una tabla Cloud Bigtable se divide en bloques de filas contiguas, llamadas tabletas, para ayudar a equilibrar la carga de trabajo de las consultas. (Las tabletas son similares a las regiones de HBase.) Las tabletas se almacenan en Colossus, el sistema de archivos de Google, en formato SSTable. Un SSTable proporciona un mapa inmutable, ordenado y persistente desde las claves hasta los valores, donde tanto las claves como los valores son cadenas de bytes arbitrarias. Cada tableta está asociada a un nodo Cloud Bigtable específico. Además de los archivos SSTable, todas las escrituras se almacenan en el registro compartido de Colossus tan pronto como son reconocidas por Cloud Bigtable, lo que proporciona una mayor durabilidad.

Es importante destacar que los datos nunca se almacenan en los nodos Cloud Bigtable; cada nodo tiene punteros a un conjunto de tabletas que se almacenan en Colossus.

Como resultado:

- Reequilibrar las tabletas de un nodo a otro es muy rápido, porque los datos reales no se copian. Cloud Bigtable simplemente actualiza los punteros para cada nodo.
- La recuperación de la falla de un nodo Cloud Bigtable es muy rápida, porque solo los metadatos deben migrarse al nodo de reemplazo.
- Cuando un nodo Cloud Bigtable falla, no se pierden datos.

### **Balanceo de carga**

Cada zona de Cloud Bigtable se gestiona mediante un proceso maestro, que equilibra la carga de trabajo y el volumen de datos dentro de los clusters. El maestro divide las tabletas más grandes / ocupadas a la mitad y combina tabletas menos accesibles / más pequeñas juntas, redistribuyéndolas entre los nodos según sea necesario. Si cierta tableta recibe un pico de tráfico, el maestro dividirá la tableta en dos, luego moverá una de las nuevas tabletas a otro nodo. Cloud Bigtable gestiona todas las divisiones, fusiones y reequilibrios de forma automática, lo que ahorra a los usuarios el esfuerzo de administrar sus tabletas manualmente. "Understanding Cloud Bigtable Performance" brinda más detalles sobre este proceso.

Para obtener el mejor rendimiento de escritura de Cloud Bigtable, es importante distribuir las escrituras lo más uniformemente posible en los nodos. Una forma de lograr este objetivo es mediante el uso de claves de fila que no siguen un orden predecible. Por ejemplo, podría usar el hash de una cadena en lugar de la cadena real, siempre y cuando evite las colisiones hash.

Al mismo tiempo, es útil agrupar las filas relacionadas una cerca de la otra, lo que hace que sea más eficiente leer varias filas al mismo tiempo. Por ejemplo, si almacena diferentes tipos de datos meteorológicos a lo largo del tiempo, la clave de fila podría ser la ubicación donde se recopilaban los datos seguidos de una marca de tiempo (por ejemplo, WashingtonDC # 201503061617). Este tipo de clave de fila agruparía todos los datos de una ubicación. Para otras ubicaciones, la fila comenzaría con un identificador diferente; con muchas ubicaciones recolectando datos a la misma velocidad, las escrituras aún se distribuirán uniformemente entre las tabletas.

### **Tipos de datos admitidos**

Cloud Bigtable trata todos los datos como cadenas de bytes sin formato para la mayoría de los propósitos. La única vez que Cloud Bigtable intenta determinar el tipo es para operaciones de incremento, donde el objetivo debe ser un entero de 64 bits codificado como un valor de 8-byte big-endian.

### **Uso del disco y Memoria**

Las siguientes secciones describen cómo varios componentes de Cloud Bigtable afectan la memoria y el uso del disco para su instancia.

### **Celdas vacías**

Las celdas vacías en una tabla Cloud Bigtable no ocupan espacio. Cada fila es esencialmente una colección de entradas clave / valor, donde la clave es una combinación de la familia de columnas, el calificador de columna y la marca de tiempo. Si una fila no incluye un valor para una clave específica, la entrada clave / valor simplemente no está presente.

### **Calificadores de columna**

Los calificadores de columna ocupan espacio en una fila, ya que cada calificador de columna utilizado en una fila se almacena en esa fila. Como resultado, a menudo es eficiente usar calificadores de columna como datos. En el ejemplo de Prezy que se muestra arriba, los calificadores de columna en la familia siguiente son los nombres de usuario de los usuarios seguidos; la entrada de clave / valor para estas columnas es simplemente un valor de marcador de posición.

### **Compactaciones**

Cloud Bigtable reescribe periódicamente las tablas para eliminar las entradas eliminadas y reorganizar los datos para que las lecturas y escrituras sean más eficientes. Este proceso se conoce como compactación. No hay configuraciones para compactaciones: Cloud Bigtable compacta tus datos automáticamente.

### **Mutaciones y eliminaciones**

Las mutaciones, o cambios, en una fila ocupan espacio de almacenamiento adicional, porque Cloud Bigtable almacena las mutaciones secuencialmente y las compacta sólo periódicamente. Cuando Cloud Bigtable compacta una tabla, elimina los valores que ya no se necesitan. Si actualizas el valor en una celda, tanto el valor original como el nuevo se almacenarán en el disco durante cierto tiempo hasta que se compacten los datos.

Las eliminaciones también ocupan espacio de almacenamiento adicional, al menos a corto plazo, porque las eliminaciones son en realidad un tipo especializado de mutación. Hasta que la tabla se compacte, una eliminación utiliza almacenamiento adicional en lugar de liberar espacio.

### **Compresión de datos**

Cloud Bigtable comprime tus datos automáticamente utilizando un algoritmo inteligente. No puedes configurar las configuraciones de compresión para tu tabla. Sin embargo, es útil saber cómo almacenar datos para que se puedan comprimir de manera eficiente:

- Los datos aleatorios no se pueden comprimir de manera tan eficiente como los datos con patrones. Los datos modelados incluyen texto, como la página que estás leyendo en este momento.
- La compresión funciona mejor si los valores idénticos están cerca uno del otro, ya sea en la misma fila o en filas adyacentes. Si organizas las claves de fila para que las filas con fragmentos idénticos de datos estén una al lado de la otra, los datos se pueden comprimir de manera eficiente.

### **Durabilidad de los datos**

Cuando utilizas Cloud Bigtable, tus datos se almacenan en Colossus, el sistema interno de archivos de alta durabilidad de Google, que utiliza dispositivos de almacenamiento en los centros de datos de Google. No necesitas ejecutar un clúster HDFS o cualquier otro sistema de archivos para usar Cloud Bigtable.

Realmente, Google usa métodos de almacenamiento patentados para lograr la durabilidad de los datos por encima y más allá de lo que proporciona la replicación de tres vías HDFS estándar. Además, crea copias de seguridad de tus datos para proteger contra eventos catastróficos y permitir la recuperación ante desastres.

### **Seguridad**

Tu proyecto de Cloud Platform se usa para controlar el acceso a sus tablas Cloud Bigtable. Si alguien tiene acceso a tu proyecto, ya sea directamente ó a través de una cuenta de servicio, puedes acceder a cualquier tabla ubicada dentro de las instancias Cloud Bigtable de ese proyecto. Sin embargo, puedes asignar roles de Gestión de identidades y accesos (IAM) que evitan que los usuarios individuales escriban en tablas o creen instancias nuevas. Si alguien no tiene acceso a tu proyecto, no puede acceder a ninguna de sus tablas.

Puedes administrar la seguridad solo a nivel de proyecto. Cloud Bigtable no admite restricciones de seguridad a nivel de tabla, nivel de fila, nivel de columna o nivel de celda.

### **Cloud Bigtable y otras opciones de almacenamiento**

Cloud Bigtable no es una base de datos relacional; no admite consultas SQL ni combinaciones, ni admite transacciones de varias filas. Además, NO es una buena solución para menos de 1 TB de datos.

- Si necesitas soporte SQL completo para un sistema de procesamiento de transacciones en línea (OLTP), considera Google Cloud SQL.
- Si necesitas consultas interactivas en un sistema de procesamiento analítico en línea (OLAP), considera Google BigQuery.
- Si necesitas almacenar blobs inmutables de más de 10 MB, como imágenes grandes o películas, considera Google Cloud Storage.
- Si necesitas almacenar objetos altamente estructurados, o si necesitas soporte para transacciones ACID y consultas tipo SQL, considera Cloud Datastore.

## Diseñando tu esquema

Vamos a explicar cómo diseñar un esquema para una tabla Cloud Bigtable.








### Conceptos generales

Diseñar un esquema Cloud Bigtable es muy diferente a diseñar un esquema para una base de datos relacional. Al diseñar tu esquema Cloud Bigtable, ten en cuenta los siguientes conceptos:

- Cada tabla tiene solo un índice, la clave de fila. No hay índices secundarios.
- Las filas se clasifican lexicográficamente por clave de fila, desde la cadena de bytes más baja hasta la más alta. Las claves de fila se ordenan en el equivalente binario del orden alfabético.
- Todas las operaciones son atómicas en el nivel de fila. Por ejemplo, si actualizas dos filas en una tabla, es posible que una fila se actualice con éxito y la otra actualización fallará. Evita los diseños de esquema que requieran atomicidad en las filas.
- Las lecturas y escrituras idealmente deberían distribuirse uniformemente en el espacio de fila de la tabla.
- En general, guarda toda la información de una entidad en una sola fila. Una entidad que no necesita actualizaciones atómicas y lecturas puede dividirse en varias filas. Se recomienda dividir en varias filas si los datos de la entidad son grandes (cientos de MB).
- Las entidades relacionadas deben almacenarse en filas adyacentes, lo que hace que las lecturas sean más eficientes.
- Las tablas de Cloud Bigtable son dispersas. Las columnas vacías no ocupan ningún espacio. Como resultado, a menudo tiene sentido crear una gran cantidad de columnas, incluso si la mayoría de las columnas están vacías en la mayoría de las filas.

## Resumen:

Hemos visto una primera explicación de los diferentes servicios que componen un proyecto de BigData típico, desde la ingesta hasta su almacenamiento y procesado.

| Ingesta  | Almacenamiento  | Procesar y Analizar  | Explorar y Visualizar   |
|--|---|--|---|
|  Cloud Pub/Sub  |  Cloud Storage   |  Cloud Dataflow |  Cloud Console           |
|  Cloud IoT Core |  Cloud SQL       |  Cloud Dataproc |  Google Data Studio      |
|  |  Cloud Datastore |  BigQuery       |  Google Sheets           |
|  |  Cloud BigTable |  |  Cloud Datalab          |
|  |  BigQuery      |  |  BI/Analytics Partners |
|  |  Cloud Spanner |  |   |

En los siguientes módulos entraremos en más detalle alrededor de DataProc y BigQuery.