

# Gestión de Datos con Amazon Web Services

Germán Moltó

Departamento de Sistemas Informáticos y Computación -  
Instituto de Instrumentación para Imagen Molecular

[gmolto@dsic.upv.es](mailto:gmolto@dsic.upv.es)

<https://www.grycap.upv.es/gmolto>

# Resultados de Aprendizaje

- Se espera que tras esta presentación seas capaz de:
  - Comprender los diferentes mecanismos para la gestión de datos en la nube.
  - Recordar las características de Amazon S3 y Glacier y CloudFront para el almacenamiento y distribución de ficheros.
  - Utilizar Amazon RDS para desplegar instancias de SGBD relacionales en la nube.
  - Conocer algunas opciones de bases de datos NoSQL, ejemplificando sobre Amazon SimpleDB
  - Entender las principales ventajas de Amazon DynamoDB así como sus conceptos fundamentales para la consulta eficiente de datos NoSQL.

# Introducción

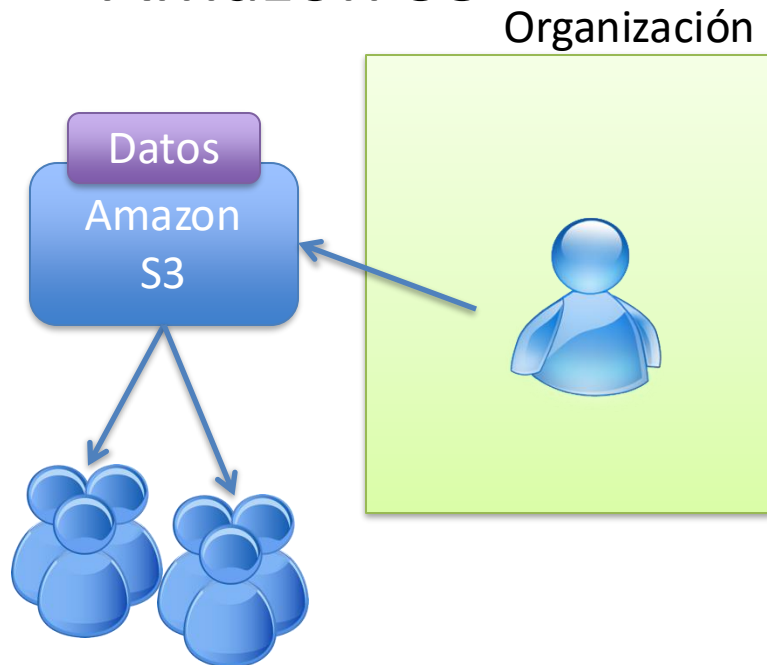
- La gestión de datos en la nube presenta ventajas:
  - Delegación del almacenamiento (replicado), custodia y accesibilidad de los datos a una entidad externa especializada.
  - Externalización de la descarga de datos a un tercer proveedor (ancho de banda, capacidad de almacenamiento).
- Y también inconvenientes:
  - Datos sensibles fuera de los límites de la organización.

# Aproximaciones a la Gestión de Datos en la Nube

- Opciones para la gestión de datos en la nube
  - Ficheros
    - Por ejemplo, mediante Amazon S3
  - Volúmenes orientados a bloques
    - Por ejemplo, mediante EBS (en Amazon EC2)
  - Bases de datos relacionales
    - Por ejemplo, mediante Amazon RDS
  - Bases de datos no relacionales
    - Por ejemplo, mediante Amazon DynamoDB.
  - Otros mecanismos.

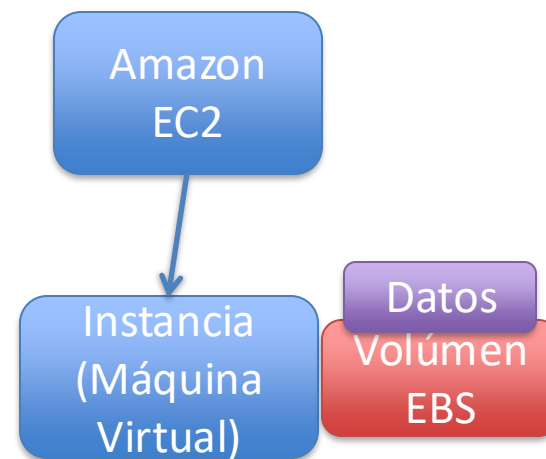
# Gestión de Datos Mediante Ficheros y Volúmenes Orientados a Bloques

- Ficheros mediante Amazon S3



- Acceso mediante HTTP (y otros)

- Volúmenes mediante EBS (en Amazon EC2)



- Acceso desde la instancia.

# Bases de Datos Relacionales en la Nube

- Despliegue de un SGBDR (Sistema Gestor de Bases de Datos Relacionales) sobre recursos aprovisionados de un Cloud IaaS.
  - Configuración, actualización, *snapshots* y alta disponibilidad delegada en el proveedor.
- Ejemplos:
  - Amazon RDS (MySQL, SQL Server, Oracle, PostgreSQL)
  - Rackspace Cloud Databases (MySQL), SQL Azure Database, cleardb.com, etc.

# Bases de Datos No Relacionales (NoSQL)

- Bases de datos del estilo clave-valor, orientadas a documento o basadas en grafos, apropiadas para escalado horizontal.
  - Sin JOINS, sin soporte completo (en general) a ACID (Atomicidad, Coherencia, Aislamiento, Durabilidad).

- Ejemplos:

- Amazon SimpleDB, DynamoDB y muchas más.



**Cassandra**

**mongoDB**



**membase**



# Otras Aproximaciones de Almacenamiento de Datos en Nube

- Backend as a Service
  - Externalización del almacenamiento de datos para aplicaciones móviles.
    - Datos en JSON, librerías REST
  - Ejemplos:
    - Firebase, Pusher, Firehose.io, Backendless, etc.

Cloud Backend  
as a Service







(Amazon RDS)

# Amazon Relational Database Service (I)

- Servicio web que permite operar una base de datos relacional en la nube, de bajo coste y capacidad redimensionable.
  - Características y funcionalidad de entornos familiares (MySQL, Oracle o SQL Server).
    - Facilita la migración al Cloud de aplicaciones que usan estos SGBD como back-end.
  - Copias de seguridad y actualización de software automáticas.
  - Alta disponibilidad.
  - Almacenamiento y capacidad de cómputo escalable.

# Usando Amazon RDS

- Desde la AWS Console es la opción más cómoda

The screenshot displays the Amazon RDS console interface. On the left is a navigation sidebar with a hamburger menu icon at the top, followed by the 'Amazon RDS' header and a back arrow. The sidebar lists several categories: 'Panel' (with links to Bases de datos, Editor de consultas, Información sobre rendimiento, Instantáneas de, Exportaciones en Amazon S3, Copias de seguridad automatizadas, Instancias reservadas, and Proxies), 'Grupos de subredes', 'Grupos de parámetros', 'Grupos de opciones', 'Versiones de motor personalizadas', 'Integraciones sin extracción, transformación y carga (ETL)', 'Nuevo' (with a link to 'Nuevo'), 'Eventos', 'Suscripciones a eventos', 'Recomendaciones' (with a badge showing '0'), and 'Actualización del certificado'. The main content area is divided into three sections. The top section, 'Recursos', includes an 'Actualizar' button and text stating the user is using resources in the US East (N. Virginia) region. It lists various resource counts: 'Instancias de base de datos (0/40)' (with a link to 'Almacenamiento asignado (0 TB/100 TB)' and a note about Neptune and DocumentDB), 'Clústeres de base de datos (0/40)', 'Instancias reservadas (0/40)', 'Instantáneas de (9)' (with a 'Manual' link), 'Automatizado' (with links to 'Clúster de base de datos (0)' and 'Instancia de base de datos (0)'), 'Eventos recientes (0)', and 'Suscripciones a eventos (0/20)'. It also lists 'Grupos de parámetros (52)' (with links to 'Predeterminado (51)' and 'Personalizada (1/100)'), 'Grupos de opciones (50)' (with links to 'Predeterminado (50)' and 'Personalizada (0/20)'), 'Grupos de subredes (2/50)', and 'Plataformas compatibles' (with a link to 'VPC' and a note 'Red predeterminada none'). The middle section, 'Crear base de datos', features a 'Crear base de datos' button and a 'Restaurar desde S3' button. It explains that Amazon RDS facilitates configuration, operation, and scaling of relational databases in the cloud, and notes that a security copy from Amazon S3 can be used for restoration. A note specifies that instances will be created in the US East (N. Virginia) region. The bottom section, 'Estado de los servicios', includes a 'Ver panel de estado del servicio' button and shows the 'Estado actual' as 'Amazon Relational Database Service (N. Virginia)' with a green checkmark, indicating the service is functioning normally.

**Amazon RDS**

**Panel**

- Bases de datos
- Editor de consultas
- Información sobre rendimiento
- Instantáneas de
- Exportaciones en Amazon S3
- Copias de seguridad automatizadas
- Instancias reservadas
- Proxies

Grupos de subredes

Grupos de parámetros

Grupos de opciones

Versiones de motor personalizadas

Integraciones sin extracción, transformación y carga (ETL)

[Nuevo](#)

Eventos

Suscripciones a eventos

Recomendaciones **0**

Actualización del certificado

### Recursos

Actualizar

Está usando los siguientes recursos de Amazon RDS en la región US East (N. Virginia) (usados/cuota),

- [Instancias de base de datos \(0/40\)](#)
  - Almacenamiento asignado (0 TB/100 TB)
  - Las instancias y el almacenamiento incluyen Neptune y DocumentDB. [Aumentar el límite de instancias de base de datos](#)
- [Clústeres de base de datos \(0/40\)](#)
- [Instancias reservadas \(0/40\)](#)
- [Instantáneas de \(9\)](#)
  - Manual
    - [Clúster de base de datos \(0/100\)](#)
    - [Instancia de base de datos \(9/100\)](#)
  - Automatizado
    - [Clúster de base de datos \(0\)](#)
    - [Instancia de base de datos \(0\)](#)
- [Eventos recientes \(0\)](#)
- [Suscripciones a eventos \(0/20\)](#)

- [Grupos de parámetros \(52\)](#)
  - [Predeterminado \(51\)](#)
  - [Personalizada \(1/100\)](#)
- [Grupos de opciones \(50\)](#)
  - [Predeterminado \(50\)](#)
  - [Personalizada \(0/20\)](#)
- [Grupos de subredes \(2/50\)](#)
- [Plataformas compatibles](#) [VPC](#)
  - Red predeterminada none

### Crear base de datos

Amazon Relational Database Service (RDS) facilita la configuración, el funcionamiento y el escalado de una base de datos relacional en la nube.

Puede utilizar una copia de seguridad de Amazon S3 para restaurar y crear una nueva base de datos Aurora MySQL y MySQL.

[Restaurar desde S3](#)

[Crear base de datos](#)

Nota: las instancias de base de datos se iniciarán en la región **US East (N. Virginia)**

### Estado de los servicios

[Ver panel de estado del servicio](#)

Estado actual | Detalles

[Amazon Relational Database Service \(N. Virginia\)](#) El servicio funciona con normalidad.

## Grupo de Seguridad (I)

- El grupo de seguridad (*Security Group*) permite determinar desde qué máquina es posible conectarse a la instancia de base de datos.
  - Igual que para EC2/VPC
- Opciones posibles (un mismo grupo de seguridad de base de datos puede incluir varias de ellas):
  - Autorizar a un rango de direcciones (CIDR)
    - 158.42.104.90/32
  - Autorizar a un determinado grupo de seguridad de EC2.
  - Desde aquellas instancias de EC2 desplegadas con ese grupo de seguridad de EC2 se podrá acceder a la instancia de base de datos.

# Usando RDS: Selección de SGBD

- Diferentes SGBD disponibles:

- Oracle
- MySQL
- SQL Server
- PostgreSQL
- MariaDB
- Amazon Aurora

- BYOL- Bring Your Own License vs License Included

RDS > Crear base de datos

## Crear base de datos


Elegir un método de creación de base de datos [Información](#)


☒ **Creación estándar**  
Puede definir todas las opciones de configuración, incluidas las de disponibilidad, seguridad, copias de seguridad y mantenimiento.


☐ **Creación sencilla**  
Utilice las configuraciones recomendadas. Algunas opciones de configuración se pueden cambiar después de crear la base de datos.


### Opciones del motor


Tipo de motor [Información](#)


☐ Aurora (MySQL Compatible)  



☐ Aurora (PostgreSQL Compatible)  


☐ MySQL  


☐ MariaDB  


☒ PostgreSQL  


☐ Oracle  


☐ Microsoft SQL Server  


# Usando RDS: Despliegue de la Instancia del SGBD

- Selección de la configuración de la instancia de RDS.
  - Prestaciones
  - Tipo de almacenamiento
  - Espacio inicial asignado.

## Settings

DB Instance Identifier*	<input type="text"/>
Master Username*	<input type="text"/>
Master Password*	<input type="password"/>
Confirm Password*	<input type="password"/>

## Configuración de la instancia

Las opciones de configuración de la instancia de base de datos que aparecen a continuación están limitadas a las que admite el motor que ha seleccionado anteriormente.



Escrituras optimizadas para Amazon RDS: **nuevo** [Información](#)

☐ Mostrar las clases de instancia que admiten las escrituras optimizadas de Amazon RDS

Clase de instancia de base de datos [Información](#)

- ☐ Clases estándar (incluye clases m)
- ☐ Clases optimizadas para memoria (incluye clases r y x)
- ☒ Clases con ráfagas (incluye clases t)

db.t3.micro

2 vCPUs 1 GiB RAM Red: 2085 Mbps

☐ Incluir clases de generación anterior

## Almacenamiento

Tipo de almacenamiento [Información](#)

SSD de IOPS provisionadas (io1)

Flexibilidad en el aprovisionamiento de E/S

Almacenamiento asignado [Información](#)

400

GiB

El valor mínimo es 100 GiB y el valor máximo es 6144 GiB



Después de modificar el almacenamiento de una instancia de base de datos, el estado de la instancia de base de datos pasará a ser de optimización del almacenamiento. La instancia permanecerá disponible a medida que se complete la operación de optimización del almacenamiento. [Más información](#)

IOPS provisionadas [Información](#)

3000

IOPS

# Usando RDS: Configuración Adicional

## Conectividad [Información](#)



### Recurso de computación

Seleccione si desea configurar una conexión a un recurso de computación para esta base de datos. Al establecer una conexión, se cambiará automáticamente la configuración de conectividad para que el recurso de computación se pueda conectar a esta base de datos.

☒ **No se conecte a un recurso informático EC2**  
No configure una conexión a un recurso informático para esta base de datos. Puede configurar manualmente una conexión a un recurso informático más adelante.

☐ **Conectarse a un recurso informático de EC2**  
Configure una conexión a un recurso informático EC2 para esta base de datos.

### Nube privada virtual (VPC) [Información](#)

Elija la VPC. La VPC define el entorno de red virtual para esta instancia de DB.

default (vpc-83a213fb)  
9 Subredes, 6 Zonas de disponibilidad

Solo se muestran las VPC con grupos de subredes de base de datos correspondientes.

Después de crear una base de datos, no puede cambiar su VPC.

### Grupo de subredes de la base de datos [Información](#)

Elija el grupo de subred de DB. El grupo de subred de DB define las subredes e intervalos de IP que puede usar la instancia de la VPC seleccionada.

db-subnet-group-default-public  
5 Subredes, 5 Zonas de disponibilidad

### Acceso público [Información](#)

☒ **Sí**  
RDS asigna una dirección IP pública a la base de datos. Las instancias de Amazon EC2 y otros recursos fuera de la VPC pueden conectarse a la base de datos. Los recursos de la VPC también pueden conectarse a la base de datos. Elija uno o varios grupos de seguridad de VPC que especifiquen qué recursos pueden conectarse a la base de datos.

☐ **No**  
RDS no asigna una dirección IP pública a la base de datos. Solo las instancias de Amazon EC2 y otros recursos dentro de la VPC pueden conectarse a la base de datos. Elija uno o varios grupos de seguridad de VPC que especifiquen qué recursos pueden conectarse a la base de datos.

### Grupo de seguridad de VPC (firewall) [Información](#)

Elija uno o varios grupos de seguridad de VPC para permitir el acceso a su base de datos. Asegúrese de que las reglas del grupo de seguridad permitan el tráfico entrante adecuado.

☒ **Elegir existente**  
Elegir grupos de seguridad de VPC existentes

☐ **Crear nuevo**  
Crear un grupo de seguridad nuevo de VPC

### Grupos de seguridad de VPC existentes

Elegir una o más opciones

default ✕

## Estimated Monthly costs

Instancia de base de datos	12.41 USD
Almacenamiento	50.00 USD
IOPS provisionadas	300.00 USD
<b>Total</b>	<b>362.41 USD</b>

Esta estimación de facturación se basa en el uso bajo demanda, tal como se describe en [Precios de Amazon RDS](#). La estimación no incluye los costos de almacenamiento de copias de seguridad, operaciones de E/S (si proceden) ni transferencia de datos.


Realice una estimación de sus costos mensuales de la instancia de base de datos mediante la [Calculadora costo mensual AWS](#).

# Usando RDS: Opciones de Mantenimiento y Copia de Seguridad

- Definición de periodo de retención de copias de seguridad y la ventana de copia de seguridad.
- Actualización automática de versiones menores.

## Copia de seguridad

☒ Habilitar las copias de seguridad automatizadas.  
Crea una instantánea de un momento dado de su base de datos

 Tenga en cuenta que, actualmente, las copias de seguridad automáticas son solo compatibles con el motor de almacenamiento InnoDB. Si está usando MyISAM, consulte los detalles [aquí](#).

### Periodo de retención de copia de seguridad [Información](#)

El número de días (1 a 35) durante los que se conservan las copias de seguridad automáticas.

7 días

### Periodo de copia de seguridad [Información](#)

El intervalo de tiempo diario (en UTC) durante el cual RDS realiza copias de seguridad automatizadas.

- ☒ Elegir una ventana  
☐ Sin preferencia

Hora de inicio

00 : 00 UTC

Duración

0.5 horas

☒ Copiar las etiquetas en las instantáneas

### Replicación de copias de seguridad [Información](#)

☐ Habilitar la replicación en otra región de AWS  
Cuando se habilita la replicación, se crean de forma automática copias de seguridad de la instancia de base de datos para la recuperación de desastres en la región seleccionada, además de la región actual.



# Actualizaciones y Snapshots

- Los cambios menores de versión se aplican automáticamente durante la ventana de mantenimiento.
- Los cambios mayores de versión no ocurren automáticamente y deben ser iniciados por el usuario.
- Para probar un cambio mayor de versión:
  1. Hacer un snapshot de una instancia de RDS
  2. Desplegar una instancia de RDS con la nueva versión del SGBDR
  3. Testear la instancia.

# RDS en la AWS Console

- El *endpoint* determina “la máquina” en la que se ejecuta la instancia de MySQL en la nube.

**Creating database dbinstance00.**  
Your database might take a few minutes to launch. [View credential details](#)

RDS > Databases > dbinstance00

**dbinstance00** [Modify](#) [Actions](#)

**Summary**

DB identifier dbinstance00	CPU -	Status Creating	Class db.t3.small
Role Instance	Current activity	Engine MySQL Community	Region & AZ us-east-1e

**Connectivity & security** | Monitoring | Logs & events | Configuration | Maintenance & backups | Tags

**Connectivity & security**

<b>Endpoint &amp; port</b>	<b>Networking</b>	<b>Security</b>
Endpoint -	Availability zone us-east-1e	VPC security groups gs-default-mysql (sg-43524731) (active)
Port -	VPC default (vpc-83a213fb)	Public accessibility Yes
	Subnet group db-subnet-group-default-public	Certificate authority rds-ca-2019
	Subnets subnet-c2f25afd subnet-c0c33fef subnet-13c9384e subnet-2bfb6c4f subnet-432a9408	Certificate authority date Aug 22nd, 2024

Aug 22nd, 2024

# ¿Cómo Acceder a la Base de Datos?

## (I)

- Amazon no proporciona herramientas cliente para conectarse a las instancias de bases de datos.
  - Utilizar las herramientas clientes de cada SGBD.
- Ejemplo para MySQL:
  - Usar la herramienta cliente 'mysql':
    - `mysql -h mydbinstance.cg5vg1c1zci.us-east-1.rds.amazonaws.com -P 3306 -u awsuser -p`
  - Usar cualquier otro cliente que soporte MySQL.

# ¿Cómo Acceder a la Base de Datos?

## (II)

```
gmolto@jonsu: ~  
gmolto@jonsu:~$ mysql -h mydbinstance.cgg5vg1c1zci.us-east-1.rds.amazonaws.com -P 3306 -u awsuser -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 206  
Server version: 5.5.27 Source distribution  
  
Copyright (c) 2000, 2012, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> use mydb1;  
Database changed  
mysql> show tables;  
Empty set (0.10 sec)  
  
mysql> 
```

# ¿Cómo Trabajar con la Base de Datos?

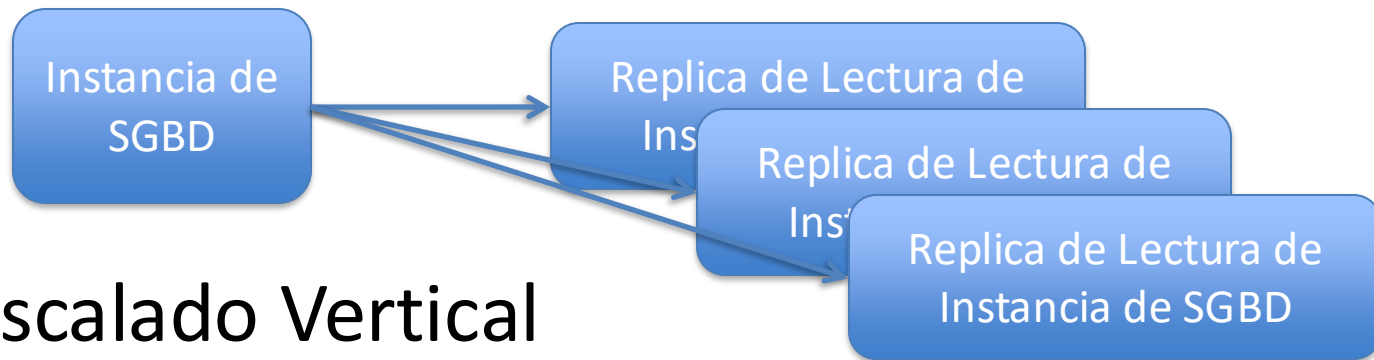
- Amazon RDS proporciona el despliegue de la instancia del SGBD en la nube.
- El usuario debe usar las herramientas cliente del SGBD para hacer operaciones CRUD sobre la BBDD.
- Ejemplo de importar BBDD a la nube:
  - A partir de un fichero datos.sql
    - `mysql -h mydbinstance.cgg5vglc1zci.us-east-1.rds.amazonaws.com -P 3306 -u awsuser -p < datos.sql`

# Configuración de la Instancia de SGBD

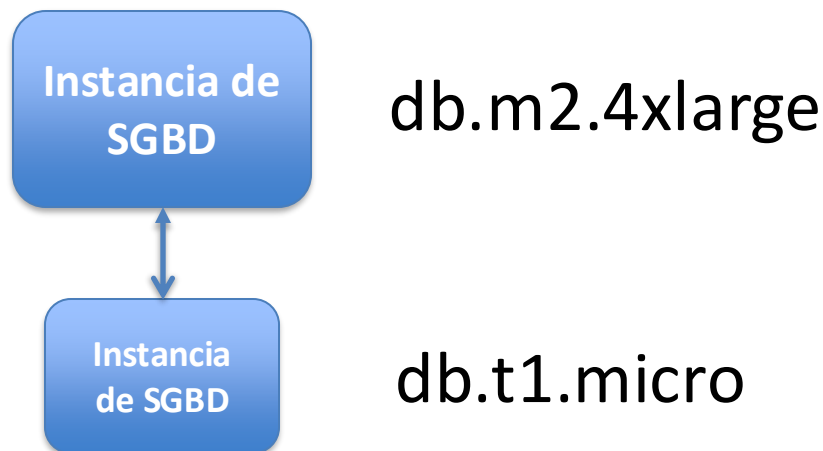
- Es posible modificar en caliente\* la configuración de la instancia de SGBD:
  - Estrategia de copias de seguridad y mantenimiento.
  - Espacio asignado
  - Nombre de la instancia y contraseña
  - Tamaño de la instancia
- La configuración se puede aplicar inmediatamente (disponible en (pocos) minutos) o diferir los cambios a la próxima ventana de mantenimiento.
- El cambio de tamaño provoca una breve desconexión de la instancia.

# ¿Como Escalar la instancia de SGBD?

- Escalado Horizontal (Réplicas de Lectura)



- Escalado Vertical



# Escalado Horizontal: Replicas de Lectura

- Uso de las capacidades de replicación de MySQL, PostgreSQL y MariaDB para crear (hasta 5) Réplicas de Lectura (*in-region* y *cross-region*).
  - Deben estar habilitadas las copias de seguridad para poder crear Réplicas de Lectura.
  - Las actualizaciones realizadas en la instancia fuente se propagan **asíncronamente** a las Réplicas de Lectura.
  - Distribuir tráfico de lectura entre diferentes Réplicas de Lectura en diferentes AZs.
  - Se recomienda que tengan como mínimo las mismas capacidades que la instancia maestra (reducir el *replication lag*).
- Ejemplos de uso:
  - Dirigir picos de carga de lectura a las Réplicas de Lectura.
  - Informes o data mining sobre alguna Réplica para no afectar a las prestaciones de la instancia fuente.

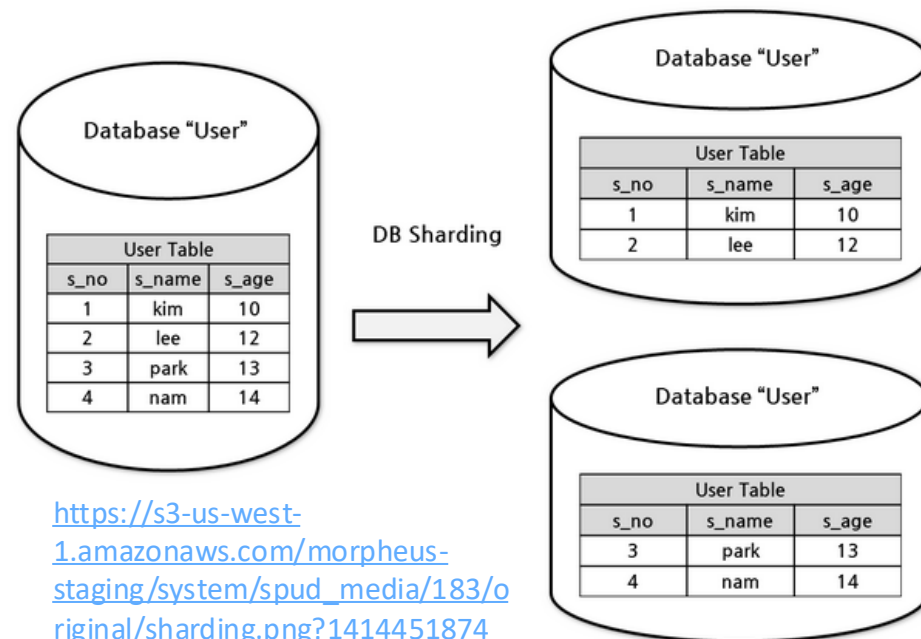


# Promoción de Réplicas de Lectura

- Las Réplicas de Lectura no pueden ejecutar operaciones de inserción/actualización (UPDATE FROM, INSERT INTO).
- En caso de fallo de la instancia maestra:
  - Se termina la replicación de la instancia maestra.
  - La Réplica de Lectura se convierte en una instancia independiente (como la maestra) en el caso de PostgreSQL. En el caso de MySQL y MariaDB se queda como una instancia de lectura independiente.
- Es posible promocionar manualmente una réplica de lectura a una instancia maestra.
- Utilidades:
  - Modificar la estructura de la base de datos (operación DDL – Data Definition Language) tras promocionar y luego redirigir tráfico a ella.
  - Sharding: Crear RLs para cada shard, promocionarlas, distribuir el tráfico entre shards y en cada shard, eliminar las filas innecesarias.

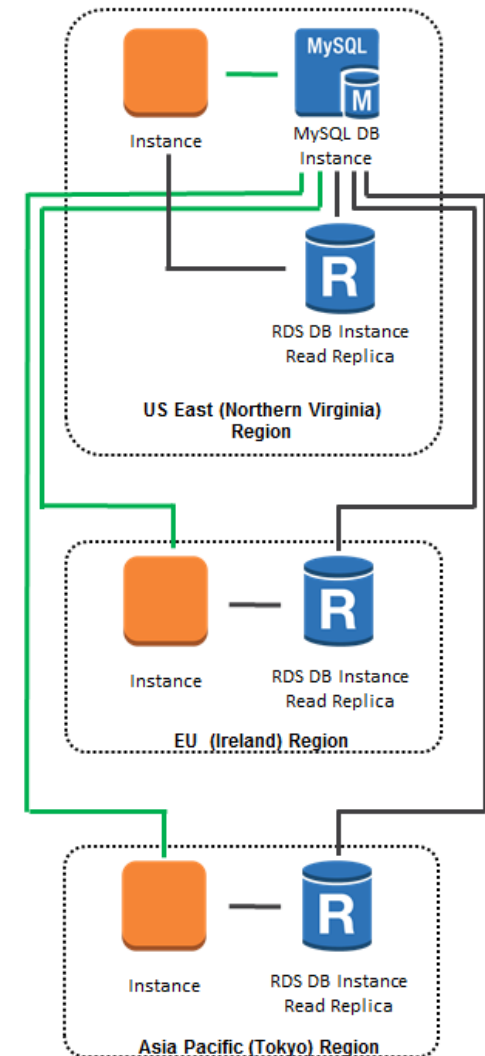
# Sobre el *Sharding*

- Un *shard* de base de datos consiste en una partición horizontal de datos en una base de datos. Cada *shard* se almacena en una instancia de base de datos con el objetivo de repartir la carga de transacciones y el espacio de almacenamiento.
- Ventajas
  - Permite reducir el tamaño de los índices, acelerando las búsquedas.
- Desventajas
  - Incremento de latencia si hay que buscar en varios shards.
  - Consistencia entre shards en caso de fallos.



# Arquitectura de Aplicación con Cross-Region Read Replicas

- Instancia maestra en la región us-east-1
- 3 Réplicas de Lectura en diferentes regiones.
- Líneas verdes: Actualizaciones/Inserciones, dirigidas a la instancia maestra.
- Líneas negras: Consultas, dirigidas a la réplica más cercana.

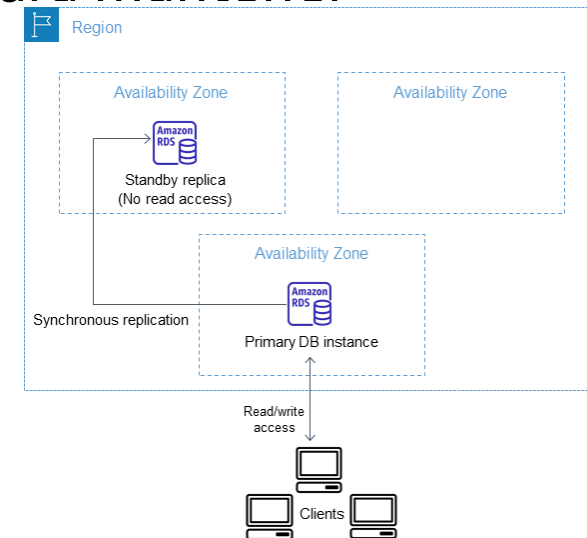
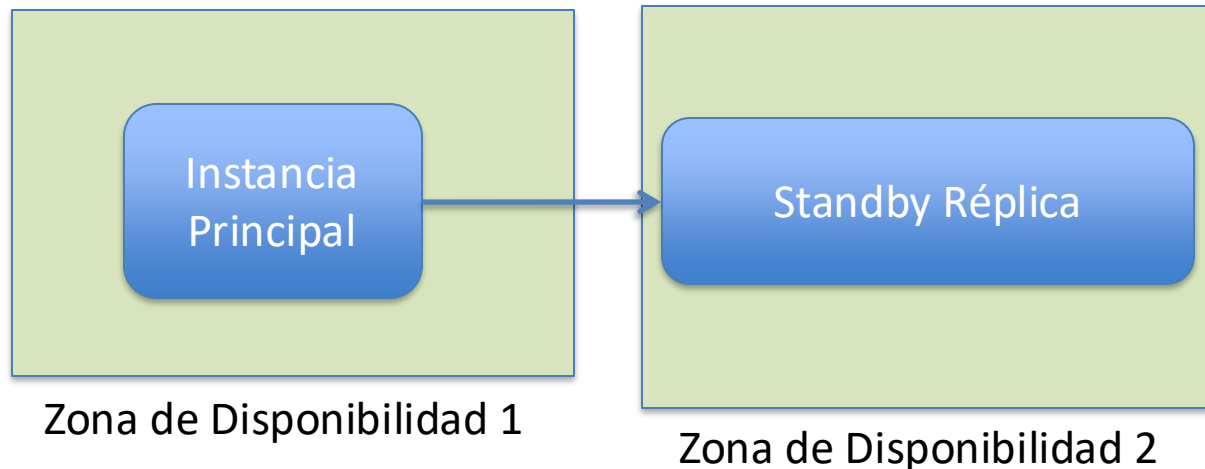


# Escalado Vertical: Aumento de Prestaciones

- Modificar el tipo de instancia de cómputo sobre la que se aloja la instancia de base de datos.
  - db.t1.micro hasta db.m2.4xlarge
- Esto permite aumentar las capacidades de la máquina para absorber picos de transacciones en la BBDD.
  - Útil para escrituras masivas en las que no se puede delegar en réplicas de lectura.

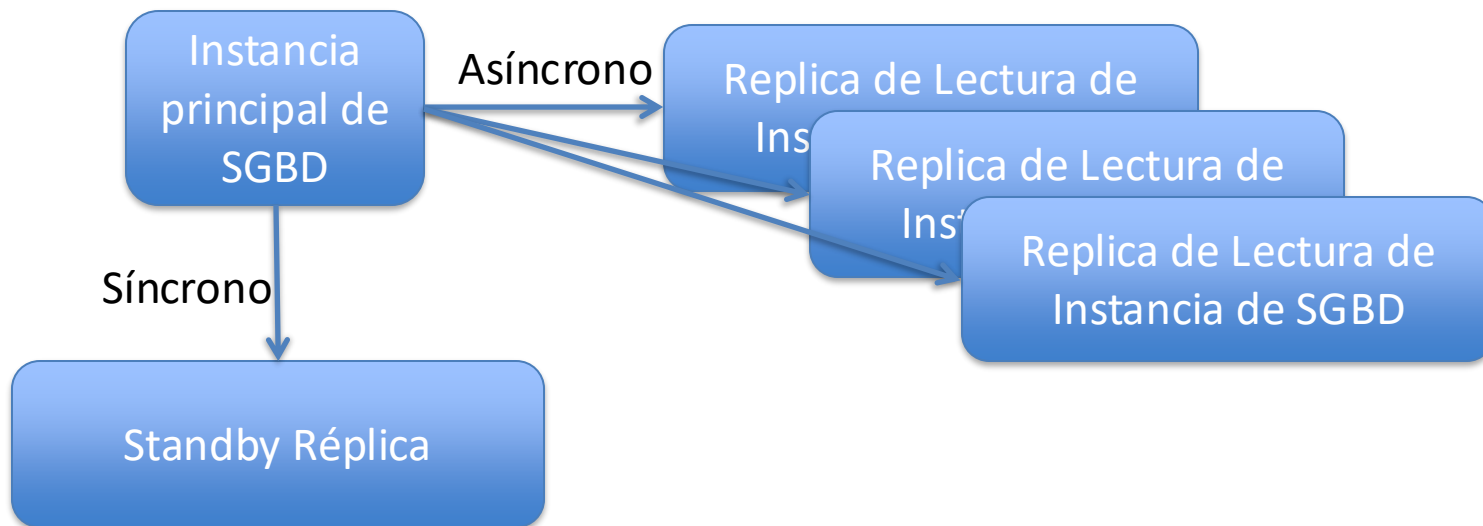
# Despliegues Multi-AZ

- Aprovisiona una réplica en espera (*standby replica*), sincronizada con la instancia principal, en una zona de disponibilidad independiente.
  - Si falla la instancia principal, automáticamente se conmuta a la réplica en espera (que pasa a ser el nuevo máster y se despliega una nueva réplica en espera para mantener Multi-AZ).



# Despliegues Multi-AZ y Réplicas de Lectura

- Uso combinado de Multi-AZ y Réplicas de Lectura.



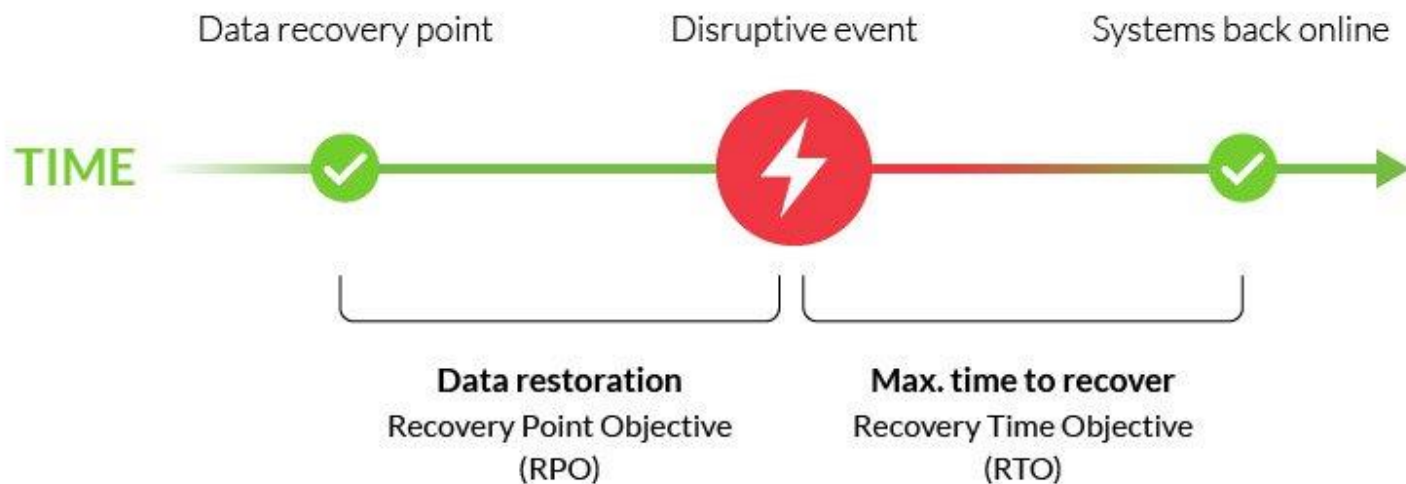
- En caso de failover, cuando la réplica en espera toma el control se reanuda la replicación asíncrona a las réplicas de lectura.

# Opciones de Copia de Seguridad

- Automated Backups
  - Copia de seguridad completa diaria y almacena registros de transacciones.
  - Permite *point-in-time recovery* (PITR) a cualquier segundo dentro del periodo de retención hasta los últimos 5 minutos.
    - Parte de la copia diaria y aplica las transacciones posteriores hasta el punto de restauración (almacenadas cada 5 minutos transparentemente en Amazon S3).
  - Periodo de retención de hasta 35 días.
- Database (DB) snapshots
  - Copias de seguridad de toda la base de datos iniciadas por el usuario y almacenadas en RDS hasta que son borradas de forma explícita.
  - Pueden ser copiadas entre regiones y permite desplegar una nueva instancia de RDS a partir de ellas.

# RTO vs RPO

- RPO: Tiempo máximo de datos en riesgo de pérdida (datos introducidos desde el último backup).
- RTO (*Recovery Time Objective*): Tiempo máximo necesario para restaurar las funciones críticas de un servicio.



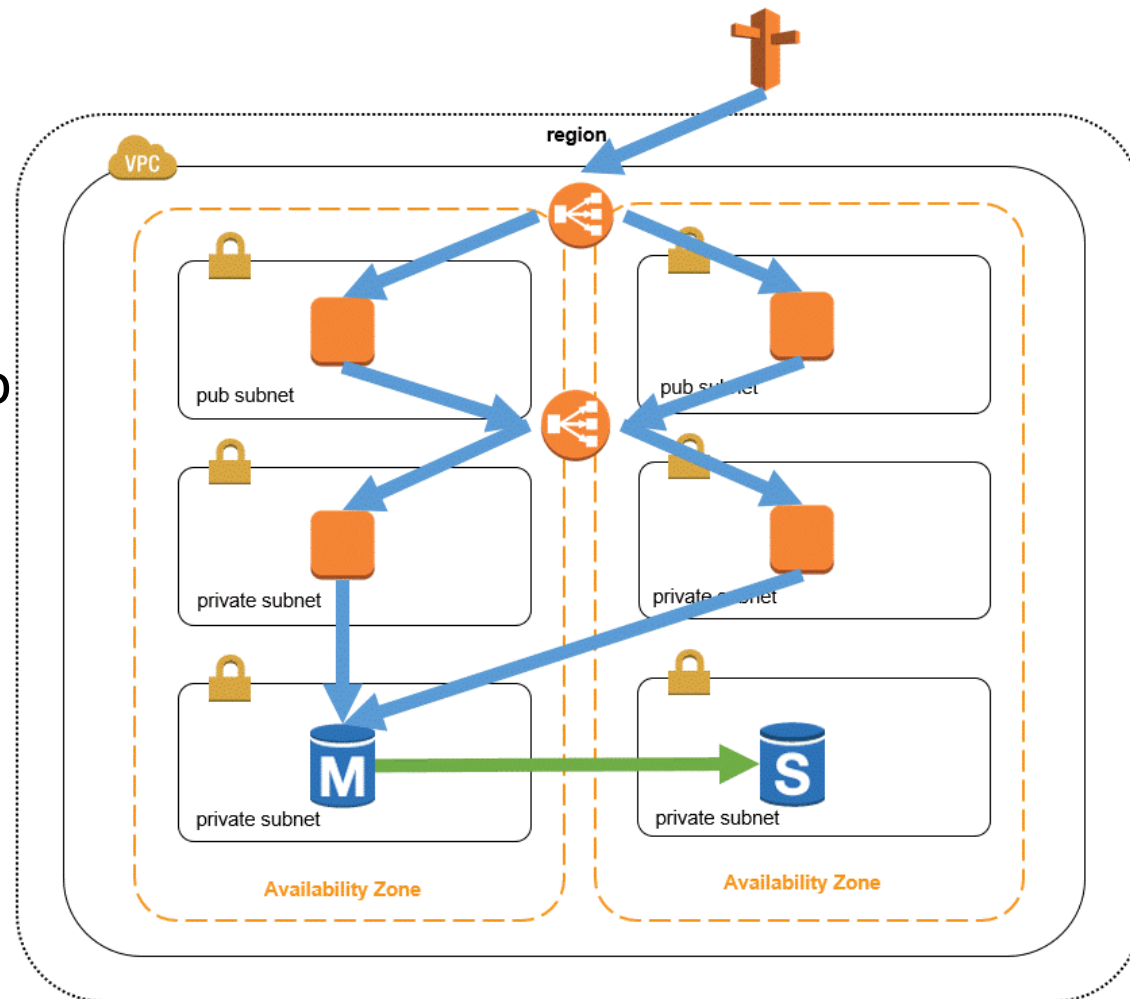


# Consideraciones Adicionales

- Las operaciones de E/S se suspenden mientras se realiza una copia de seguridad (pocos minutos).
  - Uso de despliegues Multi-AZ para evitar estos *downtimes*, dado que la copia se realiza de la instancia en *standby* (en general).
- Existen instancias reservadas (RI) de RDS para reducir el coste.
  - RI sin pago inicial (descuento hasta 30%)
  - RI de pago inicial parcial (descuento hasta 60%)
  - RI de pago total inicial (descuento hasta 63%)
- Se puede incrementar el tamaño de almacenamiento asociado a la instancia pero no reducirlo
  - En el caso de SQL Server, no es posible aumentar el tamaño por lo que hay que dimensionar adecuadamente el espacio asignado inicialmente.

# Ejemplo de Arquitecturas con Bases de Datos Relacionales (I)

- Ventajas:
  - Despliegue Multi-AZ para tener tolerancia a fallos.
  - *Failover* automático (a nivel de DNS).
- Desventaja:
  - Una única instancia de base de datos procesa todas las transacciones.



# Precios de RDS

- Precios superiores a los de las instancias EC2 subyacentes.
  - db.t2.micro con MySQL en us-east-1 (\$0.017)
  - t2.micro en EC2 en us-east-1 (\$0.0116)
- El coste de Multi-AZ es el doble.
- Coste por almacenamiento en función del tipo de almacenamiento.
- Coste por transferencias salientes de datos.

# Conclusiones

- Amazon RDS (Relational Database Service) permite desplegar instancias de Sistemas Gestores de Bases de Datos Relacionales en la nube.
  - MySQL, Oracle, Microsoft SQL Server
- Gestiona automáticamente las copias de seguridad y las actualizaciones.
- Permite escalado horizontal y vertical para lidiar con aumentos del número de transacciones.

# Gestión de Datos con Amazon Web Services



## El Servicio Amazon SimpleDB

Germán Moltó

Instituto de Instrumentación para  
Imagen Molecular

[gmolto@dsic.upv.es](mailto:gmolto@dsic.upv.es)

<https://www.grycap.upv.es/cursocloudaws>



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica



etsinf



# Resultados de Aprendizaje

- Se espera que una vez acabes esta presentación seas capaz de:
  - Conocer la funcionalidad principal del servicio Amazon SimpleDB y las principales diferencias con respecto a un RDBMS.
  - Comprender el concepto de dominio y su estructuración en ítems y atributos.
  - Comprender el procedimiento de acceso a un dominio creado con Amazon SimpleDB.
  - Conocer el lenguaje pseudo-SQL utilizado para consultar un dominio.
  - Conocer las opciones de coherencia de lectura disponibles.

# Amazon SimpleDB

- Permite crear BBDD NoSQL (sin modelo relacional, sin esquema) con conjuntos de datos inferiores a 10 Gb.
  - Acceso a datos mediante peticiones a un servicio web.
  - Creación automática de replicas geográficamente distribuidas para ofrecer alta disponibilidad.
  - Actualmente no es posible acceder a Amazon SimpleDB desde la AWS Management Console.

# Diferencias entre un RDBMS y SimpleDB

BBDD Relacional	Amazon SimpleDB
Bases de datos que contienen tablas	Dominios ligados a una cuenta de AWS
Esquemas que definen la estructura de las tablas	Sin estructura predefinida. Se pueden añadir atributos en cualquier momento.
Tablas, registros y columnas	Dominios, Items y Atributos
Las columnas únicamente tienen un valor	Los atributos pueden tener múltiples valores
Los índices se definen manualmente	Los atributos se indexan automáticamente
Se utilizan JOINS para agregar información entre tablas	No existen JOINS, o bien se duplica los datos o se realizan múltiples consultas a nivel de aplicación
Uso de transacciones para garantizar consistencia de datos	Consistencia eventual, lectura consistente o put/delete condicionales

Inspirado en la Tabla 4-1 del libro “Programming Amazon EC2”

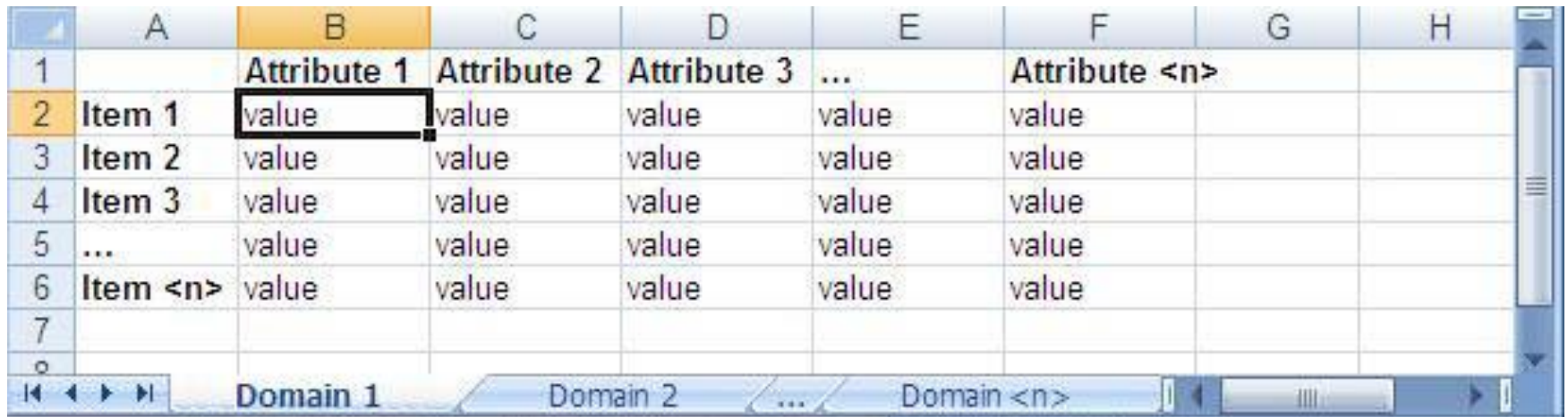


# Resumen de Amazon SimpleDB



- Un usuario de AWS puede crear hasta 250 Dominios de 10 GB cada uno (incrementables).
- Los dominios contienen Items, que representan las entidades de datos almacenados.
- Los Items constan de Atributos (no más de 256), que almacenan uno o más valores. Pares (nombre, valor)

# Analogía con una Hoja Excel



The screenshot shows an Excel spreadsheet with columns A through H. Row 1 contains headers: 'Attribute 1' in B, 'Attribute 2' in C, 'Attribute 3' in D, '...' in E, and 'Attribute <n>' in F. Rows 2 through 6 represent items: 'Item 1' in A2, 'Item 2' in A3, 'Item 3' in A4, '...' in A5, and 'Item <n>' in A6. Each item row contains a 'value' in columns B through F. The cell B2 is highlighted with a black border. At the bottom, the 'Domain' tabs are visible: 'Domain 1' (selected), 'Domain 2', '...', and 'Domain <n>'.

	A	B	C	D	E	F	G	H
1		Attribute 1	Attribute 2	Attribute 3	...	Attribute <n>		
2	Item 1	value	value	value	value	value		
3	Item 2	value	value	value	value	value		
4	Item 3	value	value	value	value	value		
5	...	value	value	value	value	value		
6	Item <n>	value	value	value	value	value		
7								

- Los dominios constan de items, que a su vez constan de atributos, los cuales almacenan valores (pueden almacenar más de un valor).
- El API de Amazon SimpleDB restringe las consultas a un solo dominio.

# Ejemplo de Dominio (I)

ID	Género	Autor	Título	Encuadernación	Tamaño	Año
Libro_01	Divulgación	Jane Bowling	La vida de las ardillas	dura, blanda	estándar, bolsillo	2001
Libro_02	Histórico	Arthur Clarke	Historia de la ciencia	dura	bolsillo	2008
Libro_03	Divulgación	Alfred Miller	La vida de los caballos	blanda	estándar	2008

- El Item Libro\_01 consta de los siguientes atributos:
  - (Género,Divulgación), (Autor,Jane Bowling), (Título,La vida de las ardillas), (Encuadernación,dura), (Encuadernación,blanda), (Tamaño,estándar), (Tamaño,bolsillo), (Año,2001)
- Flexibilidad. Se puede añadir un nuevo atributo para un ítem concreto, sin necesidad de especificar valor para el resto de items. No hay esquema prefijado en los datos.

## Ejemplo de Dominio (II)

ID	Category	Subcategory	Name	Color	Size	Model	Year
Item_01	Education	Preschool	Crayola	Multicolor	Big, Small		2009,2010
Item_02	Technology	Photography	Canon	Silver		IXUS 230	2011
Item_03	Cartridge		HP		Small	Laserjet 300	2008
Item_04	Education	Higher Education	LaTiza	White	Small		2009

- Cada Ítem consta de unos atributos que no tienen porqué coincidir con los atributos de otro Ítem dentro del mismo dominio.
- Los atributos del producto Item\_01 son:
  - (Category,Education), (Subcategory,Preschool), (Name,Crayola),(Color,Multicolor), (Size,Big), (Size,Small), (Year,2009), (Year,2010)

# Ejemplo de Inserción de Datos en un Dominio

ID	Category	Subcategory	Name	Color	Size	Model	Year
Item_01	Education	Preschool	Crayola	Multicolor	Big, Small		2009,2010

- Ejemplo usando la herramienta 'aws':

```
aws create-domain Products
```

```
aws patt Products -i Item_01 -n Category -v Education
```

```
aws patt Products -i Item_01 -n SubCategory -v Preschool
```

```
aws patt Products -i Item_01 -n Name -v Crayola
```

```
aws patt Products -i Item_01 -n Color -v Multicolor
```

```
aws patt Products -i Item_01 -n Size -v Big
```

```
aws patt Products -i Item_01 -n Size -v Small
```

```
aws patt Products -i Item_01 -n Year -v 2009
```

```
aws patt Products -i Item_01 -n Year -v 2010
```

...

Usar comillas dobles  
si el valor es una  
palabra compuesta

Diferentes valores  
para un atributo  
implican diferentes  
instrucciones

# Ejemplos de Acceso a Datos del Dominio (I)

- Es posible obtener información usando consultas muy parecidas a SQL
  - Pero sin *joins* entre dominios, etc.
- Obtener un listado de todos los items y sus atributos (pares nombre,valor) de un dominio concreto:
  - `aws --simple select 'select * from Products'`

Item_01	Category	Education
Item_01	SubCategory	Preschool
Item_01	Name	Crayola
Item_01	Color	Multicolor
Item_01	Size	Big
Item_01	Size	Small
Item_01	Year	2009
Item_01	Year	2010

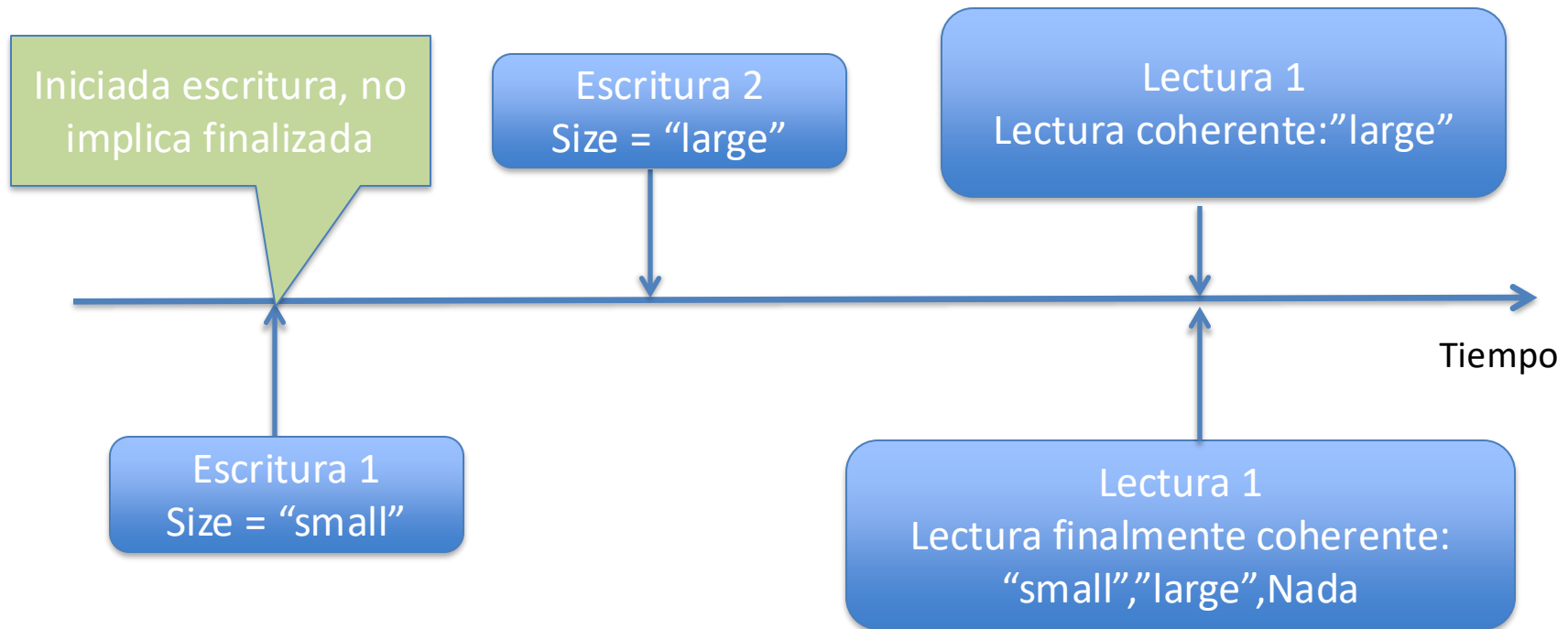
...

# Ejemplos de Acceso a Datos del Dominio (II)

- Todos los productos de la categoría “Education”
  - `aws --simple select 'select * from Products where Category = "Education"'`
- Todos los productos que tengan algún año superior a 2009
  - `aws --simple select 'select * from Products where Year > "2009"'`
- El producto cuyo identificador de Item sea “Item\_03”
  - `aws --simple select 'select * from Products where itemName()="Item_03"'`
- El nombre de los productos que únicamente tengan un atributo año y sea 2009.
  - `aws --simple select 'select Name from Products where every(Year) = "2009"'`

# Sobre la Coherencia

- Los problemas de coherencia surgen en aplicaciones concurrentes, donde pueden haber múltiples escrituras.





# Dos Opciones de Coherencia

- Lectura finalmente coherentes (o consistencia eventual) (*eventually consistent read*)
  - Puede no reflejar los resultados de una escritura reciente. La consistencia entre las réplicas de los datos generalmente se alcanza en 1 segundo.
  - Necesaria lectura diferida para obtener resultado actualizado (pero menor latencia y mayor rendimiento de lectura). Esta es la opción por defecto.
- Lectura coherente (*consistent read*)
  - Garantía de obtener el resultado actualizado tras una escritura previa (pero mayor latencia y menor rendimiento de lectura)

# ¿Qué Esquema de Coherencia Utilizar?

- Se recomienda usar lecturas finalmente coherentes en:
  - Aplicación que prefiera rendimiento de lectura a coherencia.
    - Muchas lecturas, pocas escrituras
  - Coherencia percibida por el usuario.
- Se recomienda usar lecturas coherentes en:
  - Generación de estadísticas o informes sobre datos que reciben múltiples grabaciones simultáneas.
    - Garantizar la lectura de datos actualizados
- Regla: La coherencia se obtiene en  $\approx 1$  s.

# Destrucción del Dominio

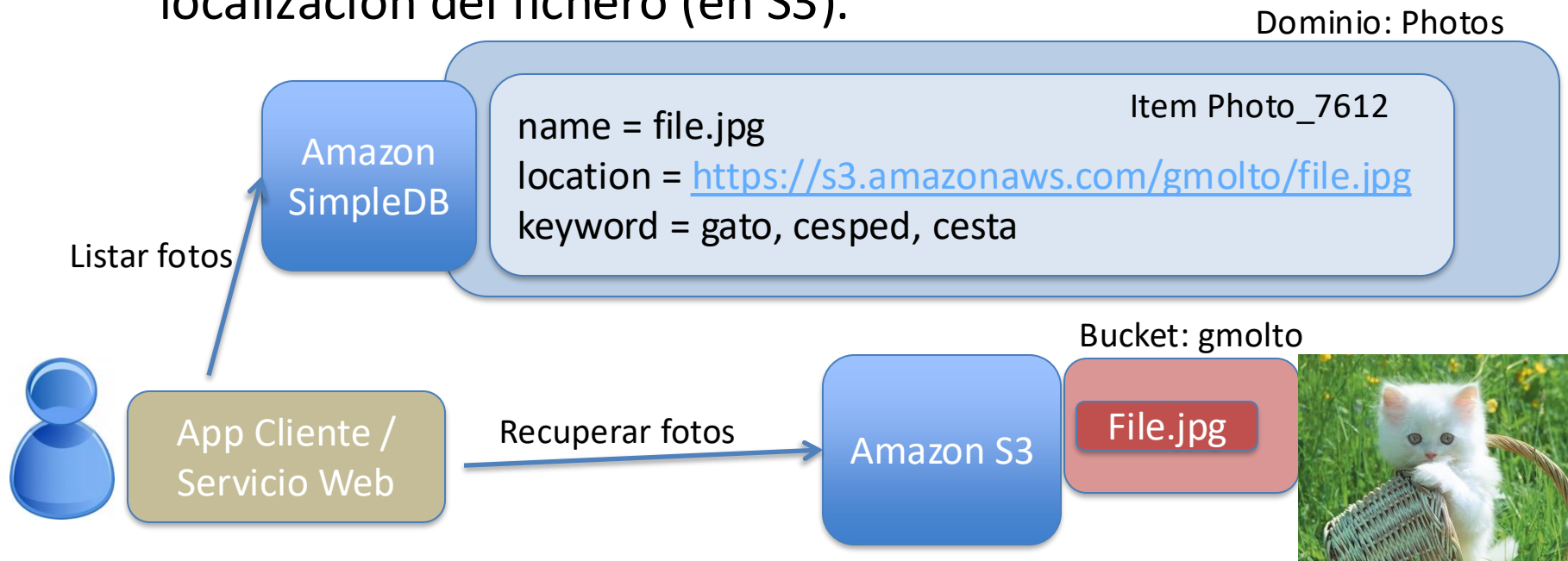
- La destrucción del dominio implica la eliminación automática de todos los Items almacenados en él.
- `aws delete-domain Products`

# Recomendaciones de Uso de Amazon SimpleDB

- Se recomienda particionar los datos en dominios de acuerdo a una partición natural:
  - Catálogo de productos particionado en los dominios:
    - Libros, Películas, Ropa, etc.
- Las consultas se restringen a un solo dominio.
  - A nivel de aplicación, lanzar consultas en paralelo a diferentes dominios para abordar grandes conjuntos de datos.
- Distribución arbitraria de datos entre dominios para superar límites y mayor *throughput*
  - Últimos n bits del hash (MD5) del nombre del Item para indicar el dominio donde se almacena. 2 bits: 4 dominios, 3 bits: 8 dominios, etc.

# Casos de Uso de Amazon SimpleDB (I)

- Indexación de ficheros almacenados en S3, con información de metadatos.
  - Subir ficheros a S3 y registrar objeto como un Item en un Dominio de Amazon SimpleDB que incluya metadatos y localización del fichero (en S3).



# Gestión de Metadatos con Amazon SimpleDB (II)

- Fotos etiquetadas con palabras clave

1. Subir fichero a S3 y hacerlo público.

1. `aws put --public gmolto/file.jpg file.jpg`

Usar UUIDs  
para evitar  
duplicados

2. Dar de alta el Item en un Dominio llamado Photos de SimpleDB <https://www.timkay.com/aws/>

1. `aws create-domain Photos`

2. `aws patt Photos -i Photo_7612 -n name -v file.jpg`

3. `aws patt Photos -i Photo_7612 -n location -v`  
<http://s3.amazonaws.com/gmolto/file.jpg>

4. `aws patt Photos -i Photo_7612 -n keyword -v gato`

5. `aws patt Photos -i Photo_7612 -n keyword -v cesped`

6. `aws patt Photos -i Photo_7612 -n keyword -v cesta`

Usar UUIDs  
para evitar  
duplicados

# Otros Posibles Usos de Amazon SimpleDB

- Almacenamiento de datos de juegos online
  - Puntuaciones, configuraciones de estado de las partidas, información sobre elementos generados por los usuarios, etc.
- Sistemas débilmente acoplados
  - Comparten información vía SimpleDB, sin necesidad de desplegar un componente adicional.
- Clientes gruesos (*Fat clients*)
  - App móvil con consulta de datos de SimpleDB mediante servicios web y procesado en local.
- Información personal
  - Listas de la compra, listas de tareas, etc.

# Conclusiones

- Amazon SimpleDB permite la creación de bases de datos NoSQL en la nube
  - Sistema de tipo clave/valor
- Posee un lenguaje de consulta parecido a SQL pero sin tanta expresividad/funcionalidad.
- Útil para almacenamiento de datos estructurados sin necesidad de soporte a transacciones, JOINS, modelo relacional, etc.



# Gestión de Datos con Amazon Web Services



## El Servicio Amazon DynamoDB

Germán Moltó

Instituto de Instrumentación para  
Imagen Molecular

[gmolto@dsic.upv.es](mailto:gmolto@dsic.upv.es)

<https://www.grycap.upv.es/cursocloudaws>

# Resultados de Aprendizaje

- Se espera que una vez acabes esta presentación seas capaz de:
  - Conocer la funcionalidad principal del servicio Amazon DynamoDB y los principales conceptos como tabla, índice, capacidad de lectura, etc.
  - Aprender a diseñar tablas básicas para almacenar datos semi-estructurados en DynamoDB.
  - Entender las ventajas y limitaciones de los índices para optimizar consultas a una tabla de DynamoDB.

# Motivación

- DynamoDB es el sistema de gestión de base de datos que soporta el proceso de ventas de Amazon.

- 105.2M req./s en Amazon Prime Day 2022

(<https://aws.amazon.com/blogs/aws/amazon-prime-day-2022-aws-for-the-win/>)

Some key serverless service stats from @jeffbarr during Amazon Prime Day this year:

- CloudFront: 290m reqs/min at peak
- SQS: 47.7m msgs/sec at peak
- DynamoDB: 89.2m reqs/sec at peak

And more! Read his post at [aws.amazon.com/blogs/aws/prim...](https://aws.amazon.com/blogs/aws/prime-day-2022-aws-for-the-win/)

#aws #serverless

Traducir Tweet




Prime Day 2021 – Two Chart-Topping Days | Amazon Web Services

In what has now become an annual tradition (check out my 2016, 2017, 2019, and 2020 posts for a look back), I am happy to share some of the metrics from...

aws.amazon.com



 Amazon Prime Day 2020, powered by AWS. Over the course of the 66-hour Prime Day, there were 16.4 trillion calls to the Amazon DynamoDB API, peaking at 80.1M requests per second! 🔍 Read this blog post to learn how AWS supported Prime Day 2020!

[go.aws/2GjqJa8](https://aws.amazon.com/blogs/aws/amazon-prime-day-2022-aws-for-the-win/)

Traducir Tweet



8:20 p. m. · 30 oct. 2020 · Sprinklr

68 Retweets 28 Tweets citados 407 Me gusta

<https://twitter.com/awscloud/status/1322257181467619331>

<https://twitter.com/jbesw/status/1410314776518041601>

# Esquema Relacional vs NoSQL

- En un RDBMS (e.g. MySQL) hay flexibilidad en la consulta de datos pero éstas no escalan bien en situaciones de alta carga.
  - Diseño basado en normalización, sin pensar en detalles de implementación o prestaciones. La optimización de consultas no afecta al diseño del esquema.
- En una BD NoSQL (e.g. DynamoDB) los datos pueden ser consultados eficientemente de pocas formas. De lo contrario, las consultas son caras y lentas.
  - El diseño del esquema se realiza para que las consultas más frecuentes se realicen lo más rápido y barato posible. Adaptar estructuras de datos a casos de uso.

# ¿Qué es DynamoDB?

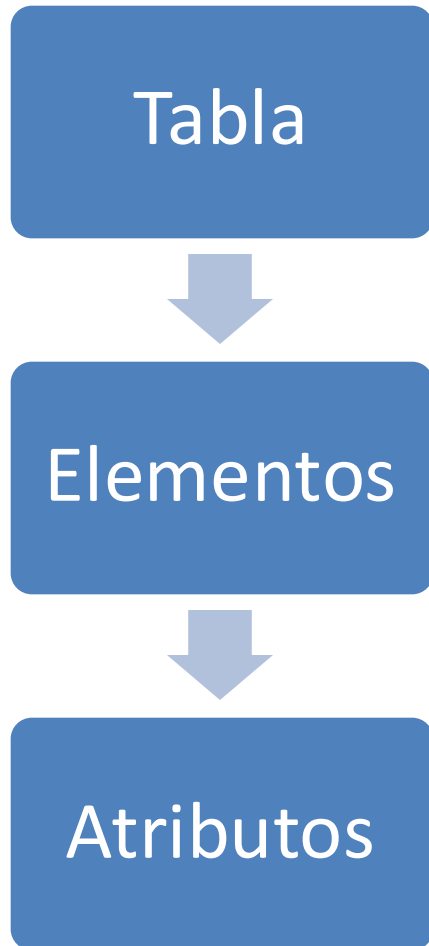
- DynamoDB es un servicio de bases de datos NoSQL totalmente administrado que ofrece un desempeño rápido y previsible, así como una escalabilidad óptima.
  - Replicación automática de datos (en SSD) y ofrece latencia en búsquedas por debajo de 10 ms.
  - Soporta cifrado en reposo
  - Replicación automática en múltiples AZs
  - Backup bajo demanda.
  - *Provisioned Throughput* (Read / Write)



# Antes de Utilizar DynamoDB

- No se debe comenzar a definir el esquema de datos hasta que no estén claras las consultas que se necesitan responder.
  - El esquema debe optimizar (tiempo y coste) las consultas más frecuentes.
- Mantener el mínimo número de tablas posibles (idealmente una sola).
  - Una sola tabla con índices puede ser usada para obtener complejas estructuras de datos.

# Conceptos de DynamoDB



- Tabla: Colección de datos (sin límite de tamaño)
- Elemento (*Item*): Grupo de atributos que puede identificarse de forma exclusiva (mediante una clave primaria única).
  - Límite de tamaño de 400 KB
- Atributos: Unidad indivisible de información.
- No existen relaciones ni JOINS, sin esquema prefijado.

# Ejemplo de Tabla de DynamoDB (I)

People

Tabla

- Clave principal.

Elemento

Clave principal

Atributo  
anidado

```
{  
  "PersonID": 101,  
  "LastName": "Smith",  
  "FirstName": "Fred",  
  "Phone": "555-4321"  
}
```

```
{  
  "PersonID": 102,  
  "LastName": "Jones",  
  "FirstName": "Mary",  
  "Address": {  
    "Street": "123 Main",  
    "City": "Anytown",  
    "State": "OH",  
    "ZIPCode": 12345  
  }  
}
```

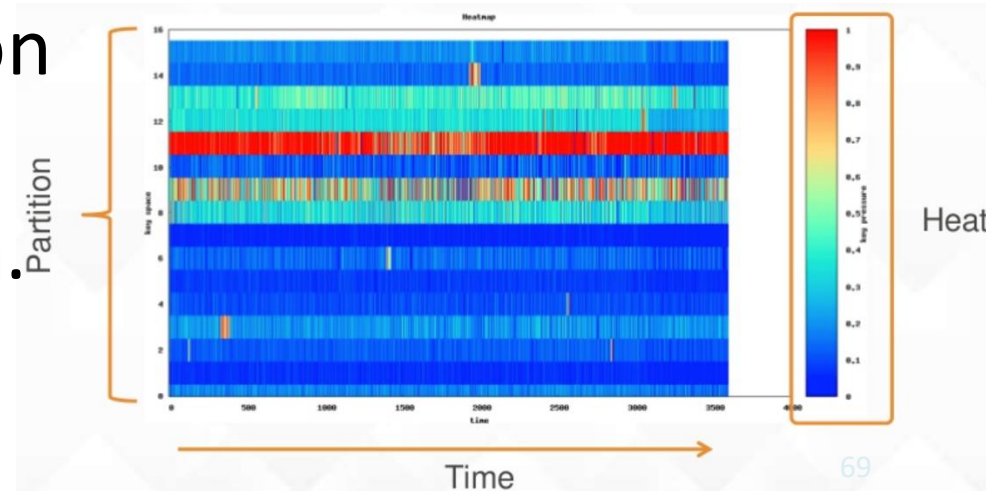
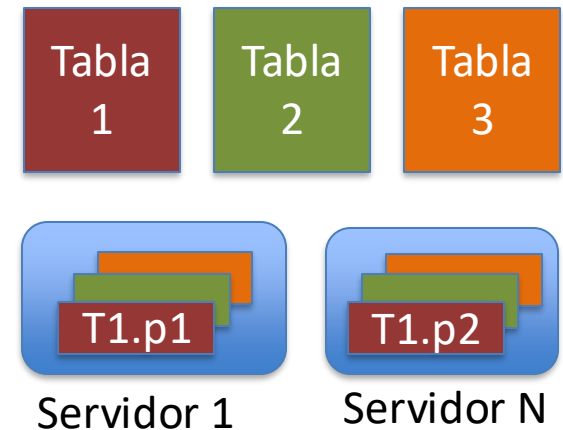
```
{  
  "PersonID": 103,  
  "LastName": "Stephens",  
  "FirstName": "Howard",  
  "Address": {  
    "Street": "123 Main",  
    "City": "London",  
    "PostalCode": "ER3 5K8"  
  },  
  "FavoriteColor": "Blue"  
}
```

- También conocida como clave de partición (*partition key* o *hash key*).
- Debe ser única para los elementos de una misma tabla.
- Determina las particiones lógicas de datos y dado que las prestaciones son por particiones, debe tener un amplio espectro de valores.



# Particiones en DynamoDB (I)

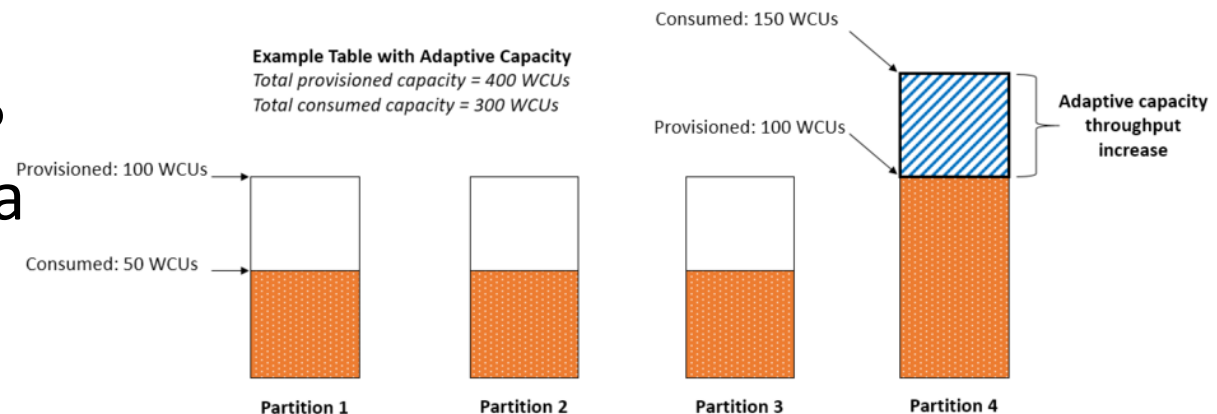
- DynamoDB hace *sharding* para distribuir horizontalmente tablas entre particiones almacenadas en diferentes servidores.
- Usar claves de partición que favorezcan la distribución equitativa. Evitar particiones calientes.



# Particiones en DynamoDB (II)

- Antiguamente DynamoDB asignaba capacidad de lectura y escritura de forma equitativa entre particiones.
  - Una partición caliente podía provocar *ProvisionedThroughputExceededException*
  - *El SDK reintentaba las operaciones de forma transparente.*

- Ya no: DynamoDB Instant Capacity (a cambio de precio por petición)



# Ejemplo de Tabla de DynamoDB (II)

Music

<pre>{   "Artist": "No One You Know",   "SongTitle": "My Dog Spot",   "AlbumTitle": "Hey Now",   "Price": 1.98,   "Genre": "Country",   "CriticRating": 8.4 }</pre>
<pre>{   "Artist": "No One You Know",   "SongTitle": "Somewhere Down The Road",   "AlbumTitle": "Somewhat Famous",   "Genre": "Country",   "CriticRating": 8.4,   "Year": 1984 }</pre>
<pre>{   "Artist": "The Acme Band",   "SongTitle": "Still in Love",   "AlbumTitle": "The Buck Starts Here",   "Price": 2.47,   "Genre": "Rock",   "PromotionInfo": {     "RadioStationsPlaying": [       "KHCR",       "KQBX",       "WTNR",       "WJJH"     ],     "TourDates": {       "Seattle": "20150625",       "Cleveland": "20150630"     },     "Rotation": "Heavy"   } }</pre>
<pre>{   "Artist": "The Acme Band",   "SongTitle": "Look Out, World",   "AlbumTitle": "The Buck Starts Here",   "Price": 0.99,   "Genre": "Rock" }</pre>

Clave principal  
compuesta por  
dos atributos:

- Artist: Clave  
de partición
- SongTitle:  
Clave de  
ordenación.

- Clave principal:
  - Simple (1 atributo):  
*Clave de partición  
(Hash key)*
  - Compuesta (2  
atributos): *Clave de  
partición y clave de  
ordenación (range  
attribute o sort  
key).*

# Tablas en DynamoDB

Title	Year	Length	IMDB-Rating
The Hobbit	2012	182	pg-13
Die Hard	1988	132	r
Piano, The	1993	121	r
Weird Science	1985	94	pg-13
Mad Max: Fury Road	2015	120	r

←→  
Clave de partición

Title	Member	Role
The Hobbit	Martin Freeman	Bilbo Bolson
The Hobbit	Ian McKellen	Gandalf

←→      ←→  
Clave de partición      Clave de ordenación

Este elemento consta de los atributos:

(Title = "The Hobbit"; Member = "Martin Freeman"; Role = "Bilbo Bolson")

# Operaciones Sobre una Tabla

- SCAN
  - Escanea **todos** los elementos de una tabla.
  - No utiliza índices.
  - Se trata de una operación lenta.
- QUERY
  - Permite hacer búsquedas por la clave de partición.
  - Para claves compuestas, permite buscar por clave de partición y luego aplicar una operación ( $=$ ,  $<$ ,  $<=$ ,  $>$ ,  $>=$ , *Between*, *Begins with*) por la clave de ordenación.
  - Permite aplicar la búsqueda en índices (LSIs y GSIs).

# Índices en DynamoDB

- Los índices permiten realizar consultas en la tabla por medio de atributos diferentes a la clave principal y/o clave de ordenación:
- Tipos de índices:
  - Índice Secundario Local (LSI)
    - Misma clave de partición, otra clave de ordenación
  - Índice Secundario Global (GSI)
    - Diferente clave de partición, otra clave de ordenación.

# Índices Secundario Local (LSI) (I)

- Índice Secundario Local
  - Reutiliza la clave de partición pero especifica otra clave de ordenación (por ejemplo para ordenar por “role” en la tabla anterior).

Title	Member	Role
The Hobbit	Martin Freeman	Bilbo Bolson
The Hobbit	Ian McKellen	Gandalf



- Encontrar todas las películas de *Title* “The Hobbit”, ordenadas por *Role*.

# Índice Secundario Local (LSI) (II)

- Índice Secundario Local (LSI)
  - La partición no puede exceder de 10 GB.
  - Comparte las prestaciones de la tabla donde se ha definido.
  - Máximo de 5 LSIs por tabla.
  - Pueden soportar consistencia fuerte en las lecturas.



# Índice Secundario Global (GSI) (I)

- Índice Secundario Global (GSI)
  - Permite reutilizar los datos de la tabla pero especificando una nueva clave de partición, para realizar consultas con dicha clave.

Title	Year	Length	Rating
The Hobbit	2012	182	pg-13
Die Hard	1988	132	r
Piano, The	1993	121	r
Weird Science	1985	94	pg-13
Mad Max: Fury Road	2015	120	r

Clave de ordenación    Clave de partición

- Encuentra todas las películas del año 2005, ordenadas por título (o con un determinado título)

# Índice Secundario Global (GSI) (II)

- Los cambios en un tabla se propagan con *consistencia eventual* al GSI.
  - No hay forma de hacer consultas con consistencia estricta a un GSI (*p99 latency of 10 ms\**)
- Se pueden crear múltiples GSIs:
  - Almacenamiento: Compartido con la tabla.
  - Prestaciones: Independientes de las de la tabla
- Sin límite de tamaño y las consultas pueden abarcar toda la tabla.
  - Máximo de 20 GSIs por tabla.

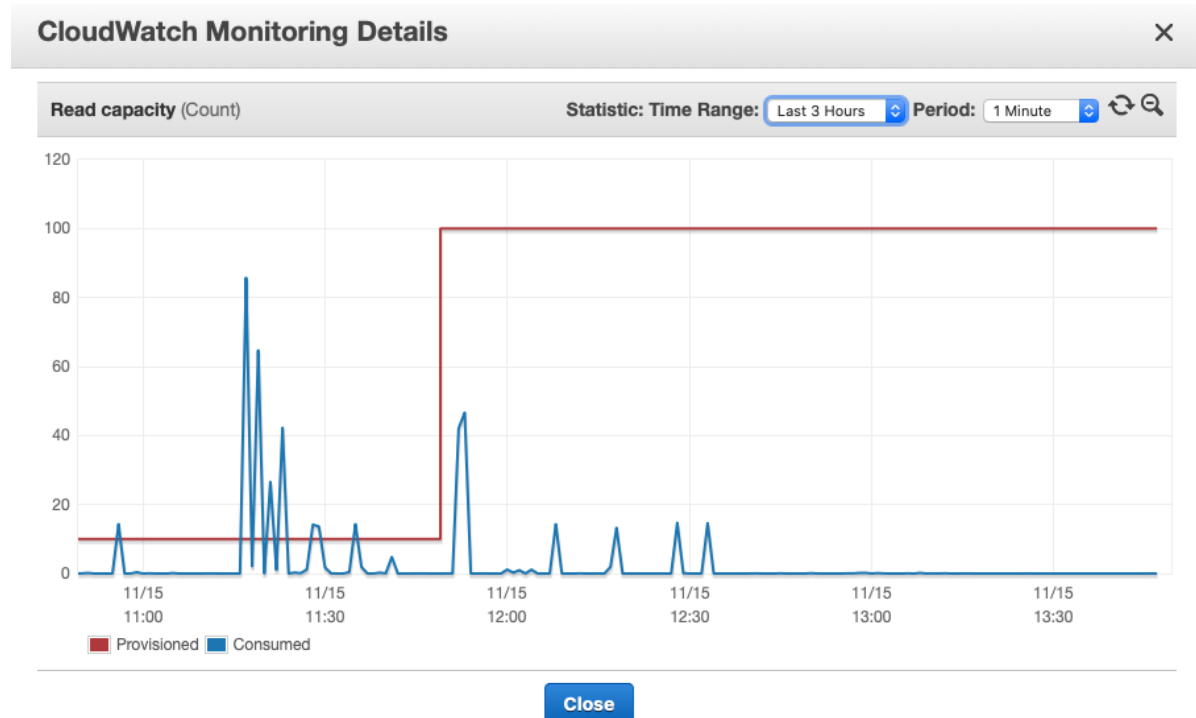
\* <https://www.youtube.com/watch?v=6yqfmXiZTIM>

# Asignando Prestaciones a una Tabla en DynamoDB (I)

- Modo Aprovisionado (*Free-tier eligible*)
  - Se asignan *unidades de capacidad* de READ y de WRITE, modificables en cualquier momento (RCU, WCU).
  - Útil si se conoce el número de lecturas y escrituras por segundo que necesita la aplicación (tabla de DynamoDB monitorizable con CloudWatch).
  - Una unidad de capacidad equivale a:
    - Una lectura de consistencia alta por segundo o dos lecturas con consistencia eventual por segundo para un elemento con tamaño hasta 4 KB (proporcionalidad para tamaños mayores).
    - Una escritura por segundo de un elemento de hasta 1 KB (proporcionalidad para tamaños mayores).
  - Existe auto-escalado para poder modificar automáticamente dichos valores (con límites)

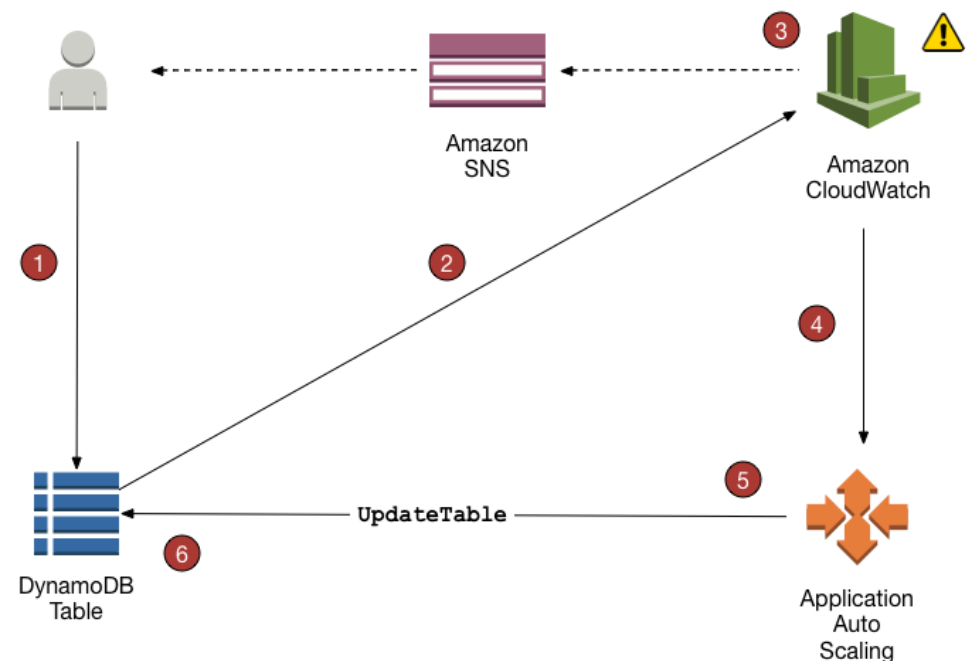
# Caso Práctico: Modo Aprovisionado

- Superación de Read Capacity por operación de *data mining* sobre tabla de DynamoDB. Posterior repetición con mayor capacidad aprovisionada para evitar reintentos desde el cliente (transparentes por el SDK)



# Auto-Escalado en DynamoDB (I)

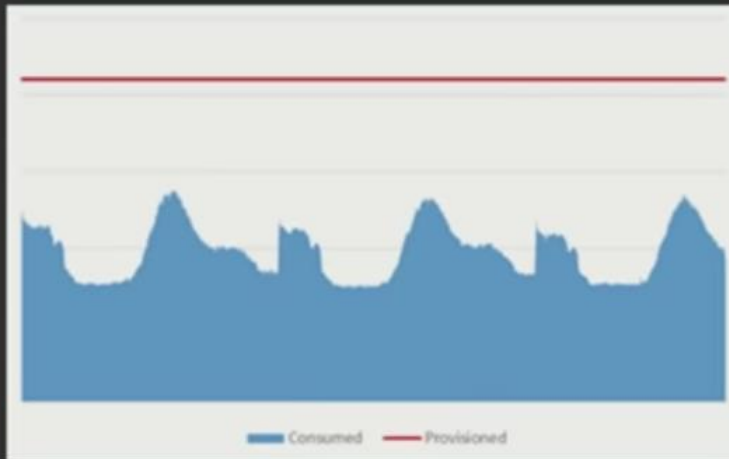
- Aplicable a la tabla o a un GSI.
- Puede ser aplicado a capacidad de lectura y/o de escritura.
- Aplica para patrones de uso predecibles y generalmente periódicos.
- Para lecturas intensivas: DAX



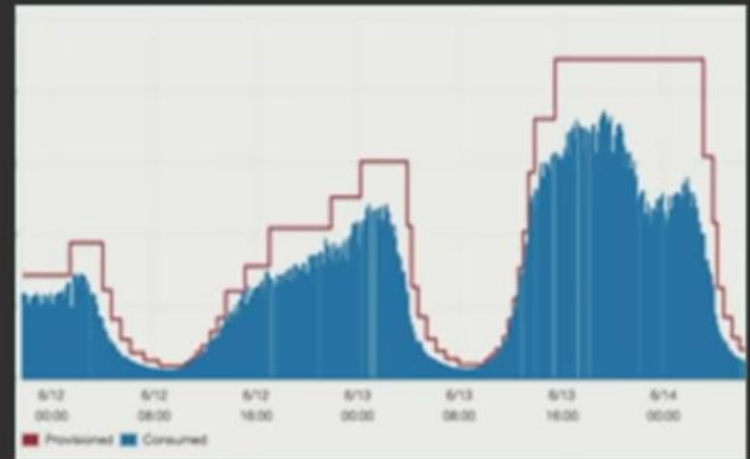
# Auto-Escalado en DynamoDB (II)

Auto scaling

Throughput automatically adapts to your actual traffic



Without auto scaling



With auto scaling

AWS re:Invent 2019: [REPEAT 1] Amazon DynamoDB deep dive: Advanced design patterns (DAT403-R1) <https://www.youtube.com/watch?v=6yqfmXiZTIM>

# Asignando Prestaciones a una Tabla en DynamoDB (I)

- Modo Bajo Demanda
  - Permite absorber *de forma instantánea* incrementos de peticiones de lectura y escritura sin tener que planificar la capacidad asignada.
  - Permite olvidarse de definir cotas superiores al auto-escalado, a cambio de un posible incremento de coste.
    - *DynamoDB On-Demand pricing is about 6.94x the cost of provisioned capacity*
      - <https://serverless.com/blog/dynamodb-on-demand-serverless/>
  - Se puede cambiar el tipo de prestaciones de la tabla una vez cada 24 horas.

# Buenas Prácticas de Uso de DynamoDB: Claves (I)

- *“Para obtener las mejores prestaciones de DynamoDB, crea tablas donde la Clave de Partición tenga un gran número de valores distintos, y estos valores sean solicitados de forma relativamente uniforme, tan aleatoriamente como sea posible”*
  - *DynamoDB Developer Guide*
- Espacio: El acceso está repartido de forma uniforme sobre el espacio de claves
- Tiempo: Las peticiones están equiespaciadas a lo largo del tiempo.



# Buenas Prácticas de Uso de DynamoDB: Claves (II)

- Sobre la clave de partición:

Clave de Partición	Uniformidad
Identificador de usuario, donde la aplicación tiene múltiples usuarios	BUENA
Código de estado, donde hay unos pocos valores posibles	MALA
Fecha de creación del ítem, redondeada al periodo de tiempo más cercano (días, horas, minuto)	MALA
Identificador de dispositivo, donde cada dispositivo consulta datos a intervalos similares	BUENA
Identificador de dispositivo, donde existen muchos dispositivos pero solo uno de ellos es consultado de forma mucho más habitual	MALA

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-partition-key-uniform-load.html>

- Sobre la clave de ordenación:
  - Pueden representar información jerárquica si están bien diseñadas. Por ejemplo, para información geográfica:  
[country]#[region]#[state]#[county]#[city]#[neighborhood]

# Buenas Prácticas de Uso de DynamoDB: Índices

- Se puede definir un GSI con misma clave de partición y de ordenación para disponer de una réplica *eventualmente consistente* de una tabla.
  - Aliviar de carga de lectura a la tabla de DynamoDB, pues el GSI tiene asignadas sus propias capacidades.

# Caso de Uso: Análisis de Logs de CloudTrail (I)



- CloudTrail-Tracker:

<https://www.grycap.upv.es/cloudtrail-tracker>

- Análisis de logs de CloudTrail:

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAJ6LGCIIMKI3PKL5DC",
    "arn": "arn:aws:iam::974349055189:user/alucloud144",
    "accountId": "974349055189",
    "accessKeyId": "ASIA6FW52WTKQDEDD4VF",
    "userName": "alucloud144",
  },
  "eventTime": "2019-03-02T16:49:56Z",
  "eventSource": "ec2.amazonaws.com",
  "eventID": "77b4dce7-624c-40f3-b237-25ddd19afeaf",
  "eventName": "RunInstances",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "95.18.21.247"
  ...
}
```

- Eventos guardados en Amazon S3 en formato JSON comprimido con los eventos que ocurren en la infraestructura.

# Caso de Uso: Análisis de Logs de CloudTrail (II)

- Consultas a realizar
  - Obtener todos los eventos ocurridos en un rango temporal.
  - Obtener todos los eventos causados por un usuario.
  - Obtener todos los eventos de un determinado tipo causados por un usuario en un rango temporal.
- Estructura de la Tabla
  - Clave de partición (*Partition key*): eventId (String)
  - Clave de ordenación (*Sort key*): userIdentity\_username (String)

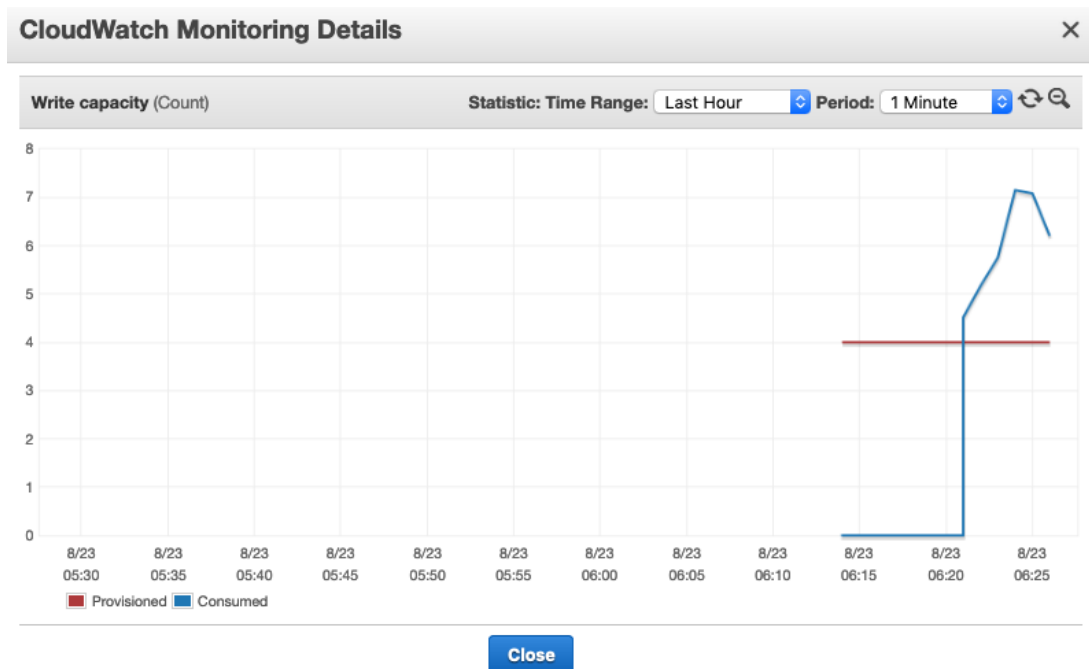
# Caso de Uso: Análisis de Logs de CloudTrail (III)

- Para filtrar los eventos en un rango temporal es necesario definir un GSI:
  - Clave de partición: `userIdentity_username` (String)
  - Clave de ordenación: `eventTime` (String)
- No es posible utilizar un LSI porque queremos cambiar la clave de partición para poder consultar eventos de un usuario concreto.

# Caso de Uso: Análisis de Logs de CloudTrail (IV)

- Pre-carga de datos históricos del bucket S3 usando AWS CLI:

— La capacidad consumida supere la aprovisionada temporalmente, pero el cliente reintenta las operaciones.



# Caso de Uso: Análisis de Logs de CloudTrail (V)

- Obtener todos los eventos realizados por alucloud144 a partir del 05 de marzo de 2019

The screenshot shows the AWS CloudTrail console interface. At the top, there's a breadcrumb 'alucloud300-ctt' and a 'Close' link. Below it are tabs for 'Overview', 'Items', 'Metrics', 'Alarms', 'Capacity', 'Indexes', 'Global Tables', 'Backups', 'Triggers', 'Access control', and 'Tags'. The 'Items' tab is selected. There's a 'Create item' button and an 'Actions' dropdown. A query is entered: '[Index] userIdentity\_userName-eventTime-index: u...'. Below the query, the 'Partition key' is set to 'useridentity\_userName' with a value of 'alucloud144'. The 'Sort key' is 'eventTime' with a value of '2019-03-05T00:00:00Z' and a greater-than-or-equal-to operator. There's an 'Add filter' button and a 'Sort' section with 'Ascending' selected. At the bottom, a table shows the first three items of the query results.

N	awsRegion	eventName	eventSource	eventTime	eventTy
1	us-east-1	ConsoleLogin	signin.amazonaws.com	2019-03-05T09:00:41Z	AwsCon
2	us-east-1	RunInstances	ec2.amazonaws.com	2019-03-05T09:01:40Z	AwsApi
3	us-east-1	CreateTags	ec2.amazonaws.com	2019-03-05T09:01:56Z	AwsApi

- Se realiza un Query sobre el GSI usando como clave de partición el identificador de alumno y aplicación una operación de ( $\geq$ ) sobre la clave de ordenación

# Caso de Uso: Análisis de Logs de CloudTrail (V)

- Obtener cuantas instancias de EC2 de tipo t2.nano ha desplegado alucloud144 el 04 de marzo de 2019

The screenshot shows the AWS CloudTrail console interface. At the top, there's a header with the account ID 'alucloud300-ctt' and a 'Close' link. Below this is a navigation bar with tabs: Overview, Items, Metrics, Alarms, Capacity, Indexes, Global Tables, Backups, Triggers, Access control, and Tags. The 'Items' tab is selected. Below the navigation bar are buttons for 'Create item' and 'Actions'. The main content area shows a query configuration for the 'useridentity\_userName-eventTime-index'. The query is '[Index] useridentity\_userName-eventTime-index: userIdentity\_userName, eventTime'. The partition key is 'userIdentity\_userName' (String) with a value of 'alucloud144'. The sort key is 'eventTime' (String) with a 'Between' filter from '1T00:00:00Z' to '03-05T00:00:00Z'. There are two filters: 'eventName' (String) with a value of 'RunInstances' and 'requestParameters\_instanceType' (String) with a value of 't2.nano'. The sort order is set to 'Ascending'. A 'Start search' button is at the bottom of the query configuration. Below the query configuration, the results are displayed in a table with columns: eventID, userIdentity\_userName, awsRegion, eventName, and eventSource. Two results are shown, both for 'alucloud144' in 'us-east-1' performing 'RunInstances' actions from 'ec2.amazonaws.com'.

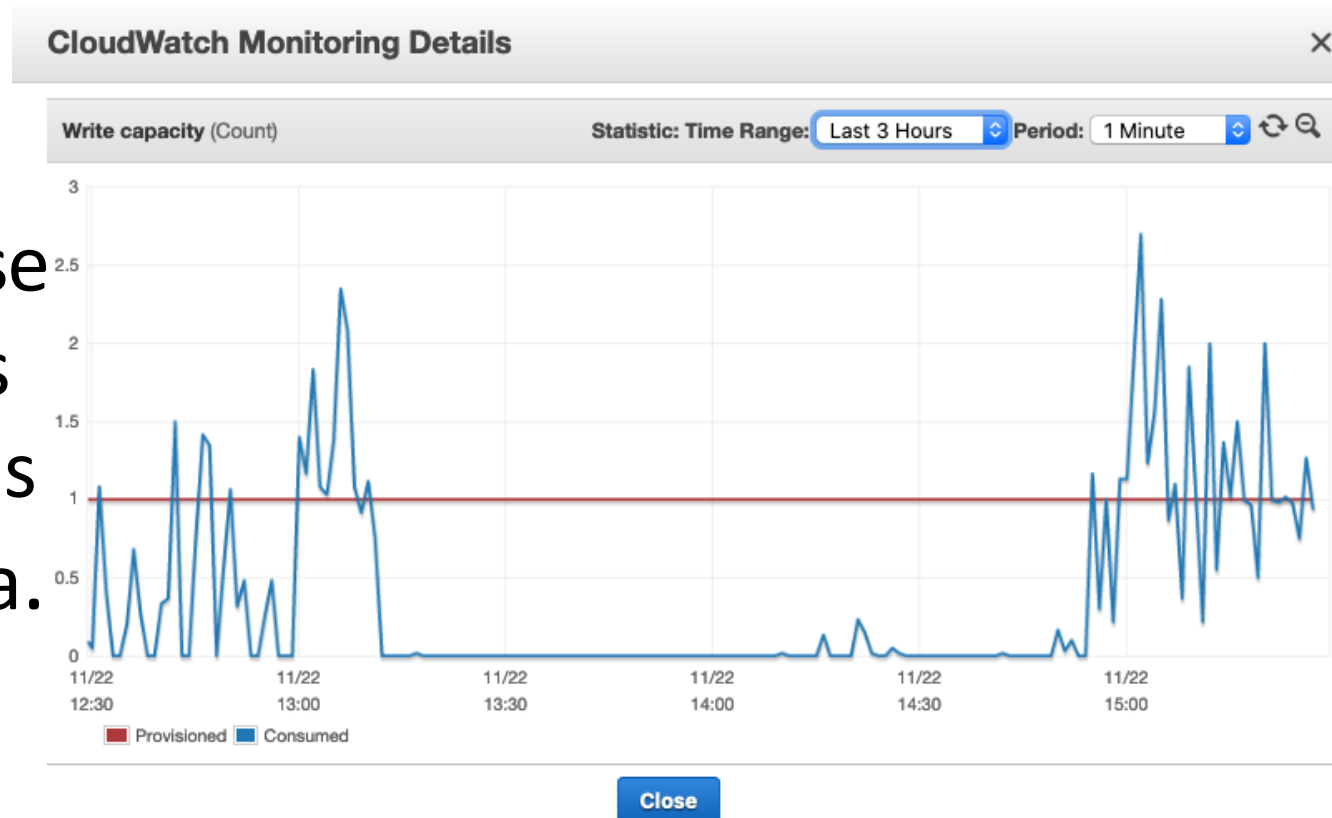
eventID	userIdentity_userName	awsRegion	eventName	eventSource
cedcc6a3-0bb8-4c71-8582-ded34a53ccaa	alucloud144	us-east-1	RunInstances	ec2.amazonaws.com
514f2b51-de14-4009-9d12-9052cf0433d0	alucloud144	us-east-1	RunInstances	ec2.amazonaws.com

- Se realiza un Query sobre el GSI usando como clave de partición el identificador de alumno y aplicación una operación de (Between) entre 2019-03-04T00:00:00Z y 2019-03-05T00:00:00Z



# Monitorizando la Carga

- Tabla en producción de CloudTrail-Tracker en curso presencial de 30 alumnos.
- En momentos puntuales se exceden las capacidades de escritura.



# Aspectos Adicionales de DynamoDB

- Lenguaje PartiQL para consultas (similar a SQL)
- DynamoDB Accelerator (DAX) mediante caché en memoria que ofrece rendimiento hasta 10 veces superior.
- *DynamoDB Standard-IA* para reducir los costes de tablas que almacenan datos que no son frecuentemente accedidos.

# Conclusiones de DynamoDB

- DynamoDB es un servicio gestionado por el proveedor para el almacenamiento de datos clave/valor sin estructura prefijada.
- Soporte diferentes esquemas de elasticidad para adaptarse a incrementos en la carga de trabajo.
- Es un bloque fundamental para realizar el almacenamiento de datos en arquitecturas de aplicaciones web en la nube modernas.

# Referencias

1. [https://aws.amazon.com/en/running\\_databases/](https://aws.amazon.com/en/running_databases/)
2. <https://aws.amazon.com/en/rds/mysql>
3. <https://aws.amazon.com/es/rds/faqs/>
4. <https://stackoverflow.com/questions/4806272/what-are-the-respective-advantages-limitations-of-amazon-rds-vs-ec2-with-mysql>
5. <https://aws.amazon.com/es/rds/#resources>
6. <https://docs.aws.amazon.com/AmazonSimpleDB/latest/DeveloperGuide/Welcome.html>
7. [https://aws.amazon.com/es/simplifiedb/faqs/#What\\_does\\_consistency\\_mean](https://aws.amazon.com/es/simplifiedb/faqs/#What_does_consistency_mean)
8. <https://www.rackspace.com/es/cloud/databases/>
9. <https://www.mongodb.com/learn/nosql>
10. <https://learning.oreilly.com/learning-paths/learning-path-aws>
11. <https://www.trek10.com/blog/dynamodb-single-table-relational-modeling>
12. <https://www.dynamodbguide.com/>
13. <https://www.alexdebrie.com/posts/dynamodb-limits>