
Bases de datos en AWS

Relational Database Service

Características de Bases de Datos RDS.

Relational Database Service (RDS)

☐ Amazon Aurora

Amazon
Aurora

☒ MySQL



☐ MariaDB



☐ PostgreSQL



☐ Oracle

ORACLE

☐ Microsoft SQL Server



Características

Backups Automáticos



1 a 35 días → DD:HH:MM:SS

Backup retention period [Info](#)

Select the number of days that Amazon RDS should retain automatic backups of this DB instance.

7 days



Backups Manuales



Creados en cualquier momento y al eliminar la BD por defecto crea uno.

Create final snapshot?

Determines whether a final DB Snapshot is created before the DB instance is deleted.

Yes



Storage



General Purpose (SSD).


Provisioned (SSD) → Uso intensivo E/S.

Seguridad



Cifrado de datos en reposo.

Encryption

☒ Enable encryption [Learn more](#) 

Select to encrypt the given instance. Master key ids and aliases appear in the list after they have been created using the Key Management Service(KMS) console.

☐ Disable encryption

Actualizaciones



Actualizaciones del motor de la BD.

Auto minor version upgrade

Specifies if the DB instance should receive automatic engine version upgrades when they are available.

☒ Yes

☐ No

IAM

● Usa tokens para conexión a la BD.
10 a 20 conexiones por segundo.

Monitoreo

● Enhanced monitoring → Tiempo Real.
No disponible para instancias small.

Precio

● Tipo y tamaño de instancia.
Monitoreo y Configuración.



1BD - Instancia
BYOL - Incluida.
Multi AZ.



Ilimitadas BD.
SW Adicional.
Multi AZ.



Ilimitadas BD.
Optimización
Queries.
Multi AZ.

Demo - Creando nuestra primer Base de Datos en RDS.

Demo - Conectándonos a nuestra Base de Datos RDS - Forma Gráfica.

Demo - Conectándonos a nuestra Base de Datos RDS - SSH.

Backups, Performance y HA en RDS.

Relational Database Service

Backups en RDS

Manuales

Son Incrementales y se mantienen al borrar la BD.

Puede copiarse a AZ y regiones.

Automáticos

Se hacen a diario.

E/S pueden quedar suspendidos (sg).

Precio

Depende del storage de la BD.

Aumenta con la retención.

Demo - Restauración de un backup automático en RDS.

Performance en RDS

Monitoreo

Lectura y Escritura, CPU, DD y Memoria.
Conexiones.

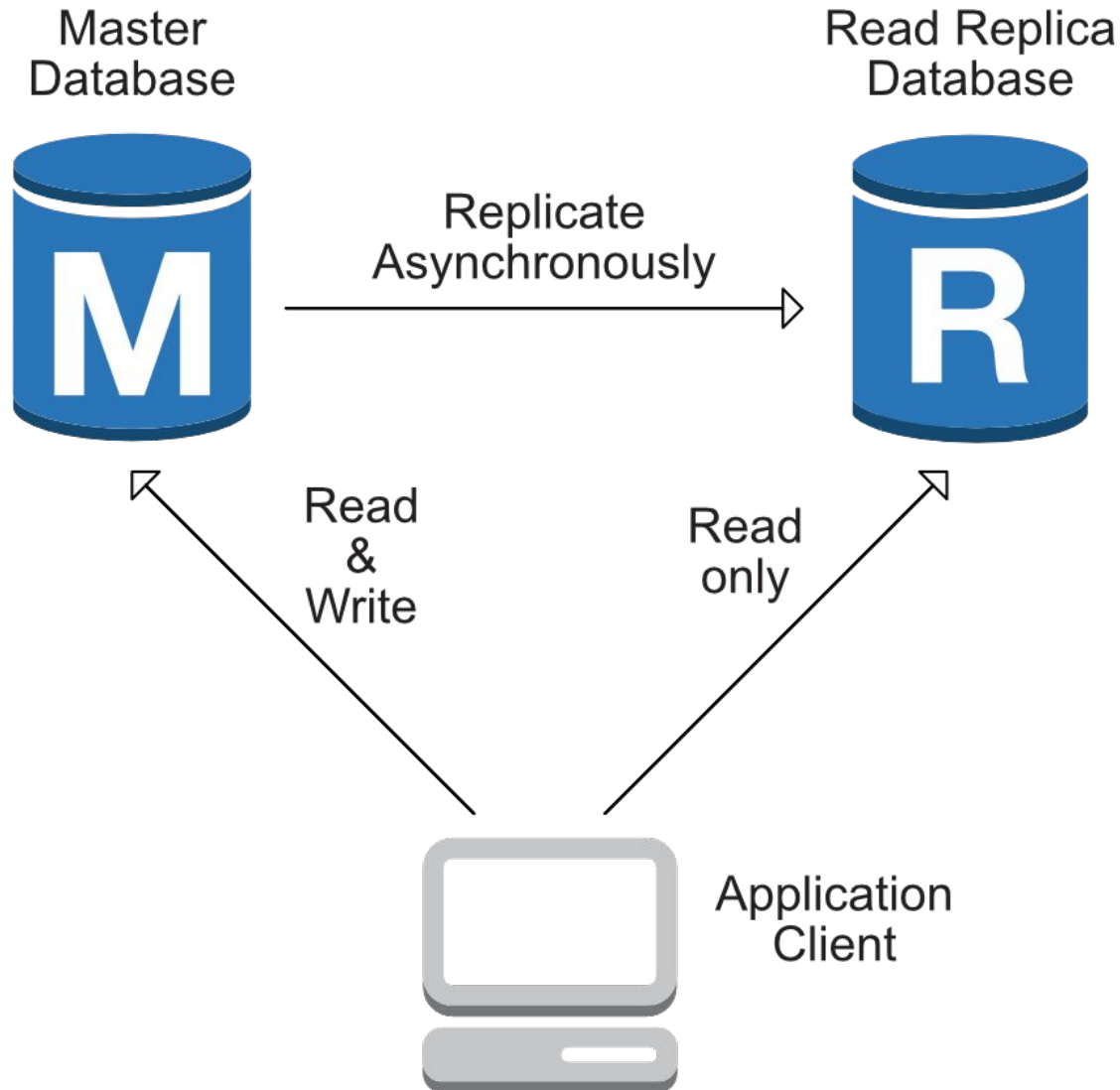
Replicas Lectura

Mejorar desempeño de la BD.
No disponibles para Oracle y SQL Server.

Provisioned

Diseñado para transacciones OLTP.
1.000 a 40.000 IOPS.

Réplicas de Lectura en RDS.



Alternativas

BD en memoria (ElastiCache).
Dividir la BD en más pequeñas.

Performance Insights

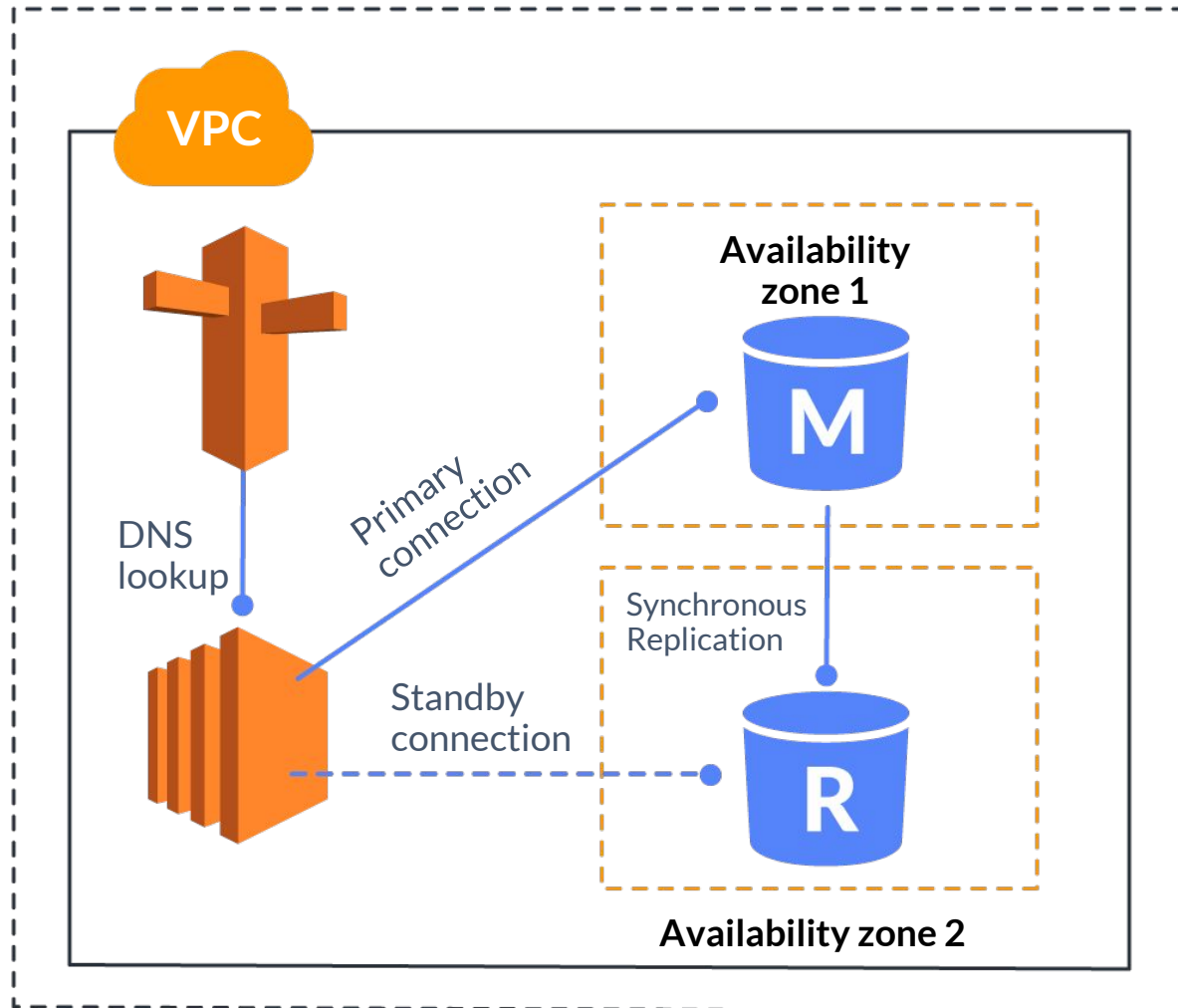
Monitoreo de performance.
Cuándo y qué acciones tomar.

Adicionales

Estrategias dependen del motor.
AWS recomienda Aurora.

Multi AZ en RDS.

AWS REGION



1

Incrementan la disponibilidad de la BD.
Recomendadas para producción.
Se compone de una Master-Standby.

2

Replicación síncrona.
Failover automático.
Conmutación por error.

3

Replica entre AZ.
El pricing es como tener 2 BD.
Backups se hacen de BD Standby.

Migración a RDS.

Relational Database Service

Adaptar los recursos a la carga de trabajo.

1

Solo se paga por los recursos mientras se hace el trabajo.

2

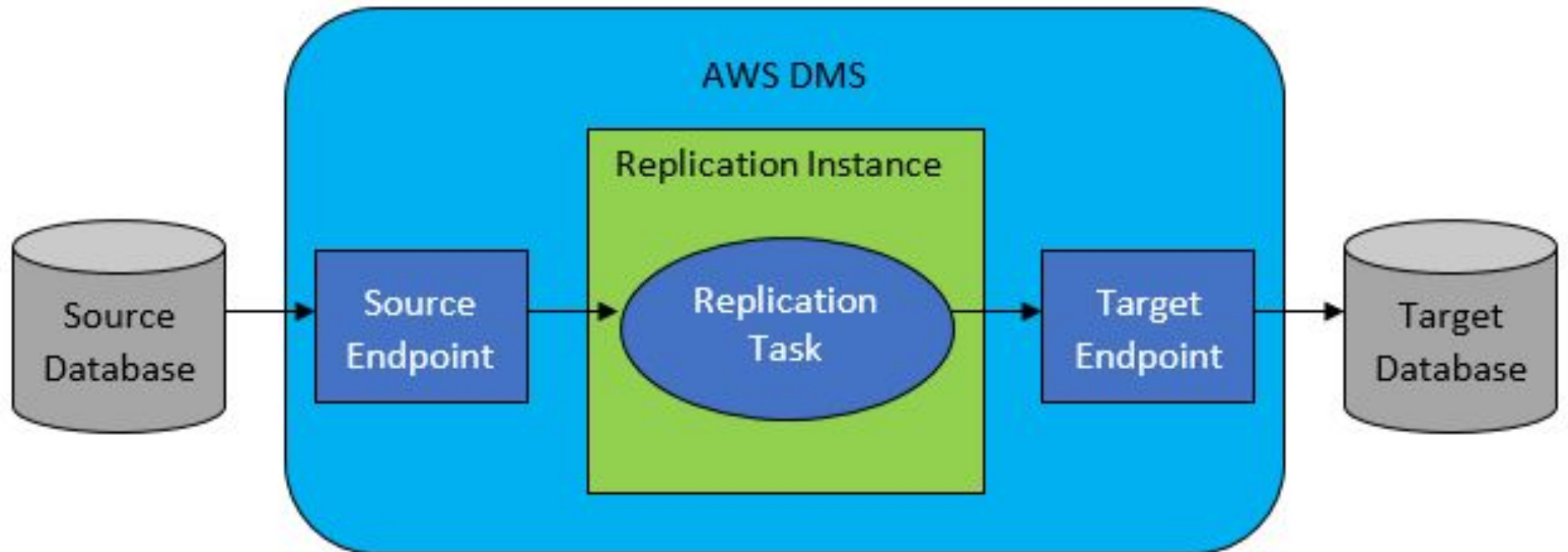
Administra la infraestructura necesaria para el trabajo de migración: Hardware, Software y parches.

3

Conmutación por error automática.

Los datos en reposo se cifran con KMS y la migración es usando conexión segura SSL.

Funcionamiento DMS



Migración Homogénea



Migración Heterogénea

STEP 1:



STEP 2:

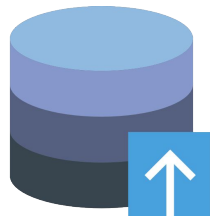


Proceso



Origen y Destino - Compatibles.

Base de datos de
origen



Tarea en la instancia
de replicación



Base de datos de
destino

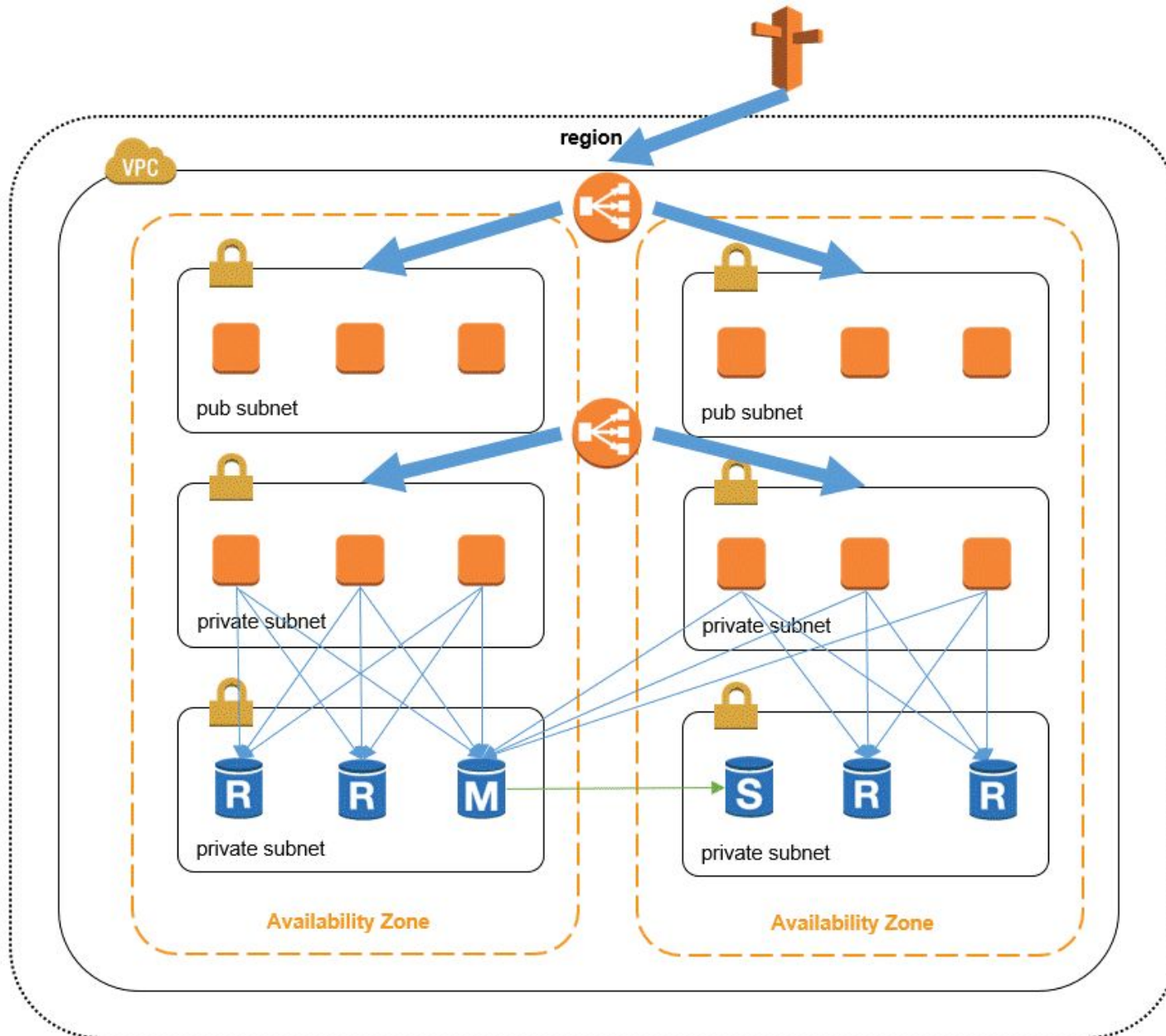


Bases de Datos

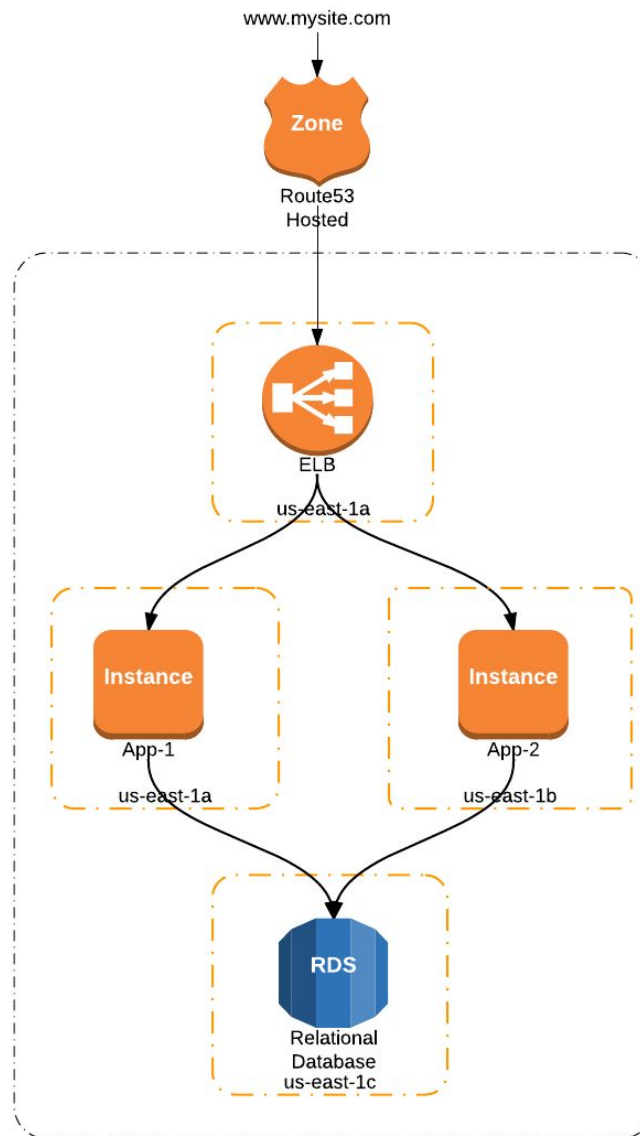


- Oracle 10.2.
- SQL Server 2005.
- MySQL 5.5.
- DB2 9.7.
- SAP ASE 12.5.
- DynamoDB.
- Aurora DB.
- Maria DB.
- PostgreSQL 9.4.
- Mongo DB 2.6.x.
- BD Azure.
- Redshift.
- S3.

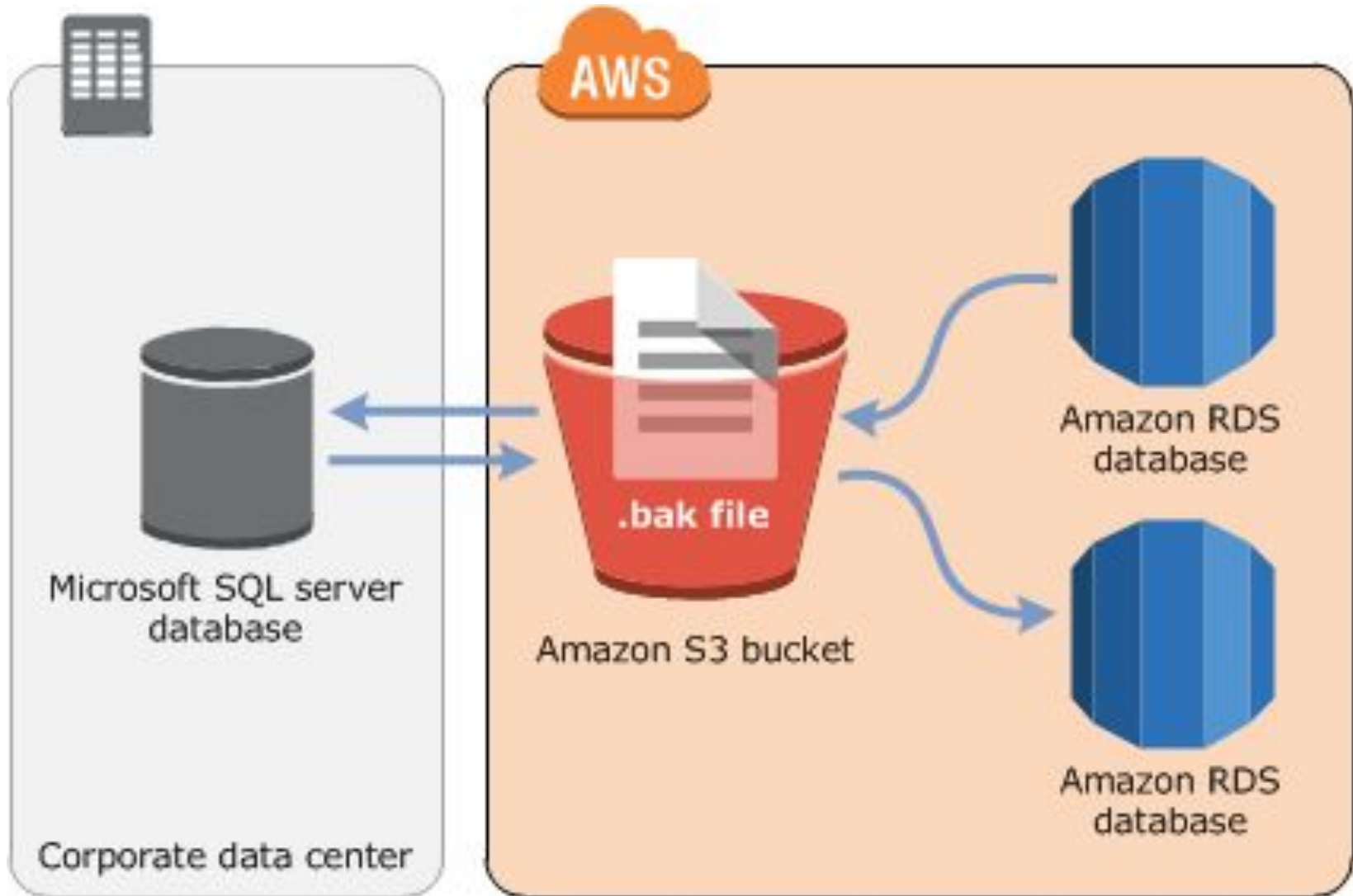
RDS - Casos de Uso



RDS - Casos de Uso



RDS - Casos de Uso



Aurora - RDS

Relational Database Service

Performance



5 veces superior a MySQL.

3 veces superior a PostgreSQL.

Use case

Do you plan to use this database for production purposes?

Use case

- ☒ **Production - Amazon Aurora** Recommended
MySQL-compatible, enterprise-class database at 1/10th the cost of commercial databases.
- ☐ **Production - MySQL**
Use [Multi-AZ Deployment](#) and [Provisioned IOPS Storage](#) as defaults for high availability and fast, consistent performance.
- ☐ **Dev/Test - MySQL**
This instance is intended for use outside of production or under the [RDS Free Usage Tier](#).

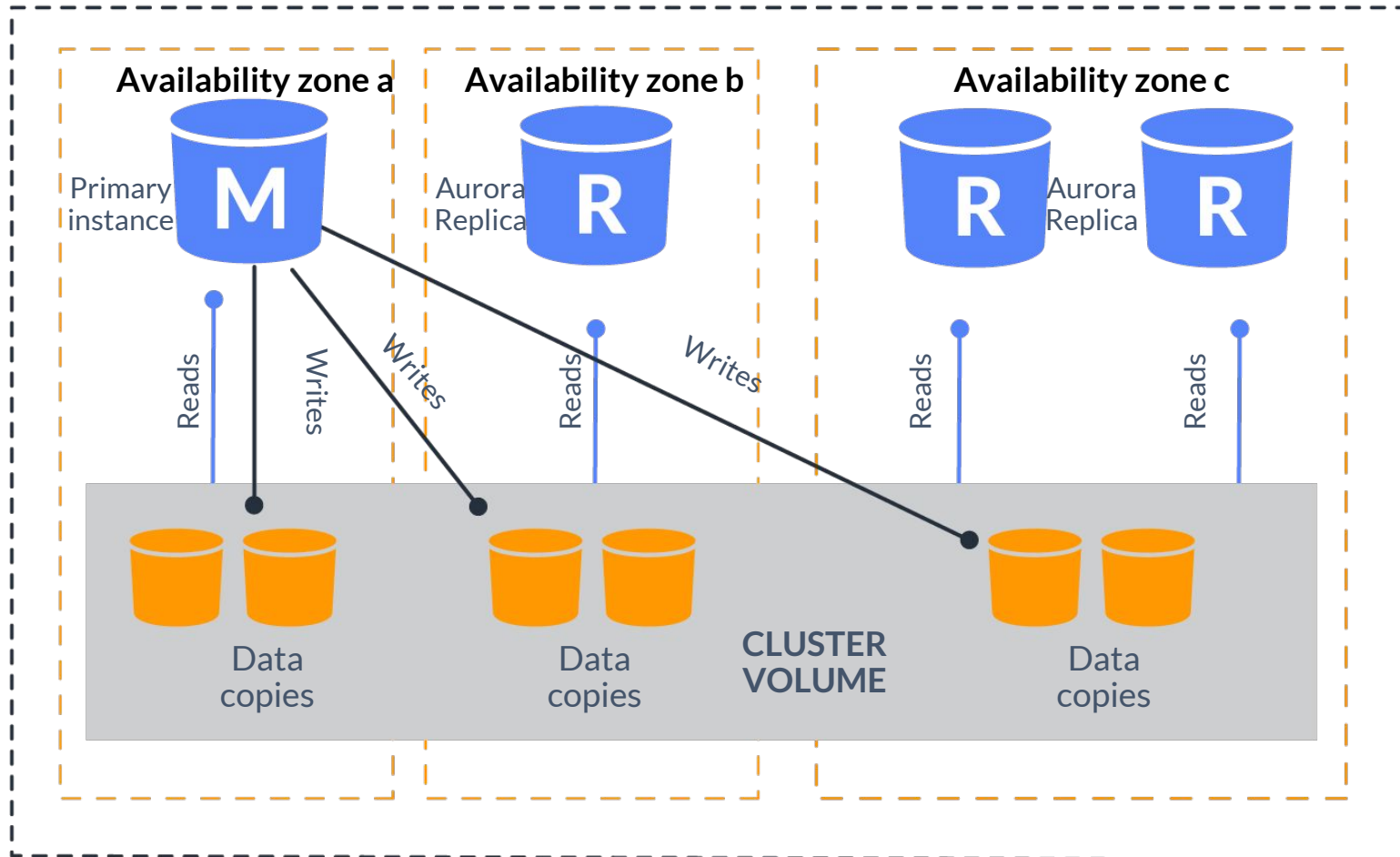
64TB Storage.

15 Read Réplicas - <10 ms Replica Lag.

Monitoring y Failover < 30 sg

Aurora Clúster

AMAZON AURORA DB CLUSTER



Endpoint - Aurora

Primary DB



Se cuenta con un endpoint para la instancia Principal.

`mydbcluster.cluster-123456789012.us-east-1.rds.amazonaws.com:3306`

Storage



Puede crecer hasta 64TB.
Data replicada en múltiples Az.

Endpoint - Aurora

Instancia



Provee control sobre cargas de trabajo sobre la Instancia Primaria y las Replicas.

`mydbinstance.123456789012.us-east-1.rds.amazonaws.com:3306`

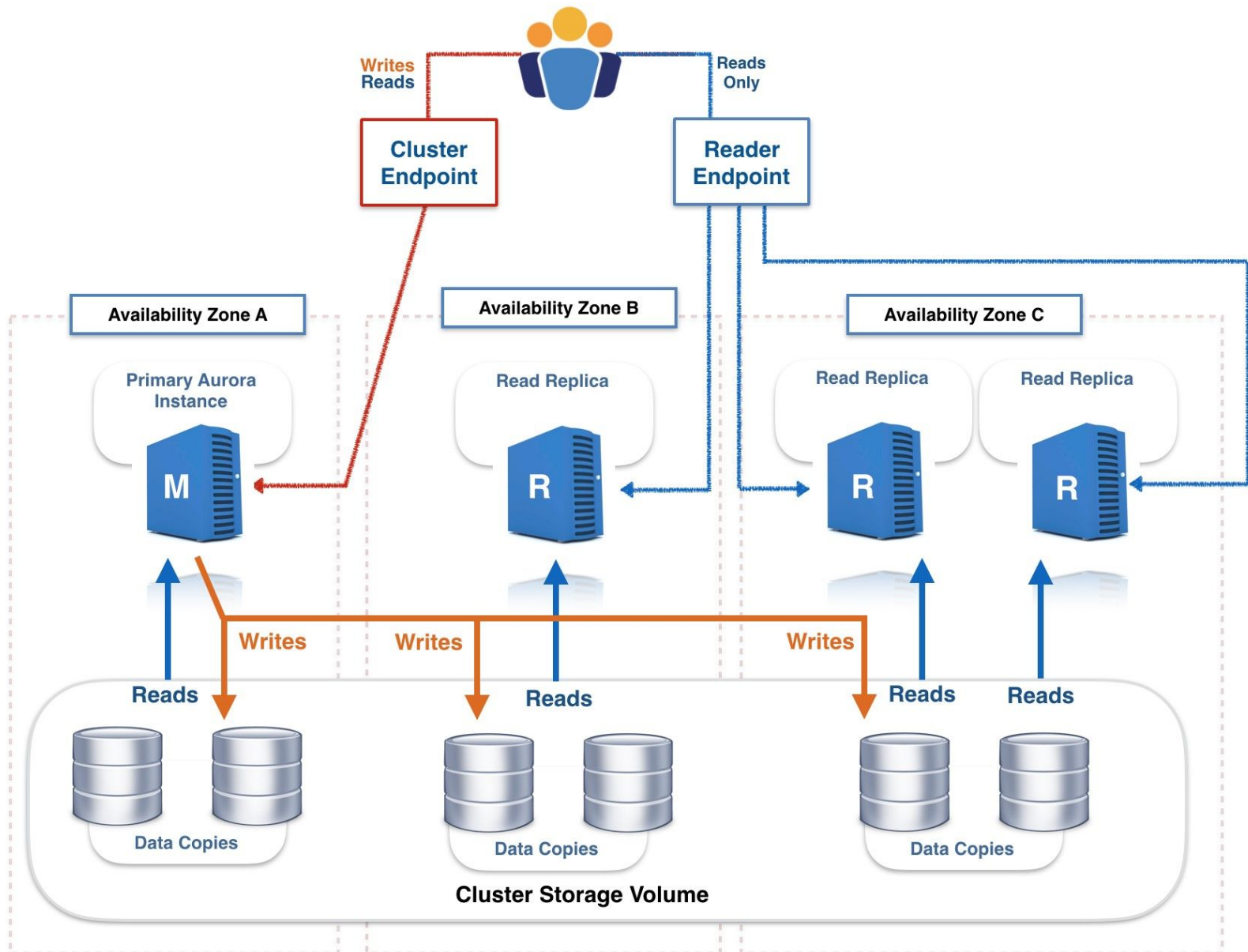
Replicas de Lectura



Conecta a una de las réplicas de lectura de Aurora.

`mydbcluster.cluster-ro-123456789012.us-east-1.rds.amazonaws.com:3306`

Aurora - Endpoints



Auto-reparación

Repara fallas en los discos automáticamente.
Al momento de reparar guarda la data en otros discos.

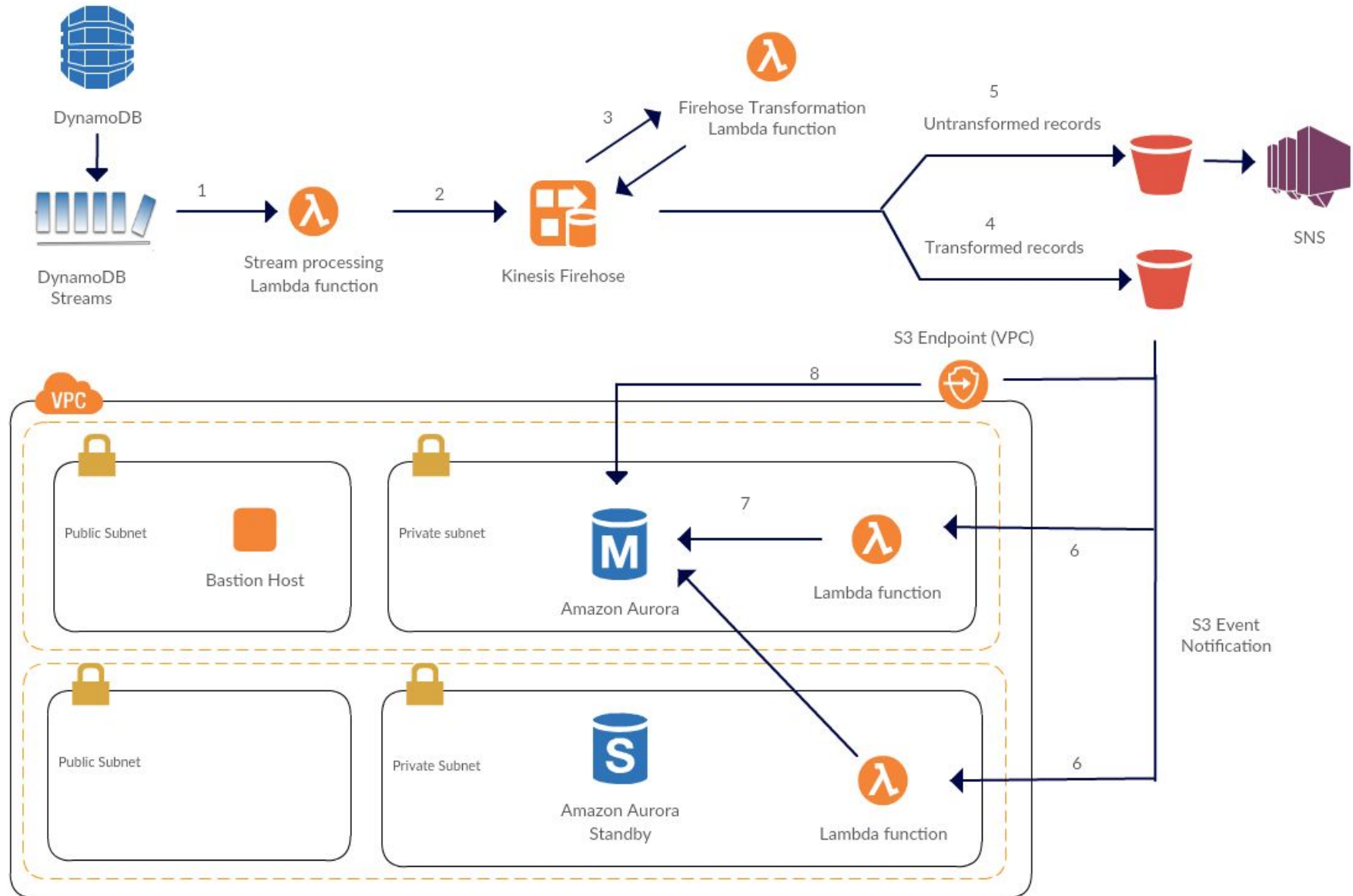
Cache - Warm

Hace un pre-calentamiento del caché al iniciar con las consultas más comunes y sus resultados.

Recuperación de Accidentes

Si falla la instancia principal Aurora promueve una réplica de lectura o crea una instancia primaria nuevamente.

Aurora - Caso de Uso



Serverless

- Seleccionar mínimo y máximo de instancia.

Minimum Aurora capacity unit [Info](#)

2
4GB RAM ▼

Maximum Aurora capacity unit [Info](#)

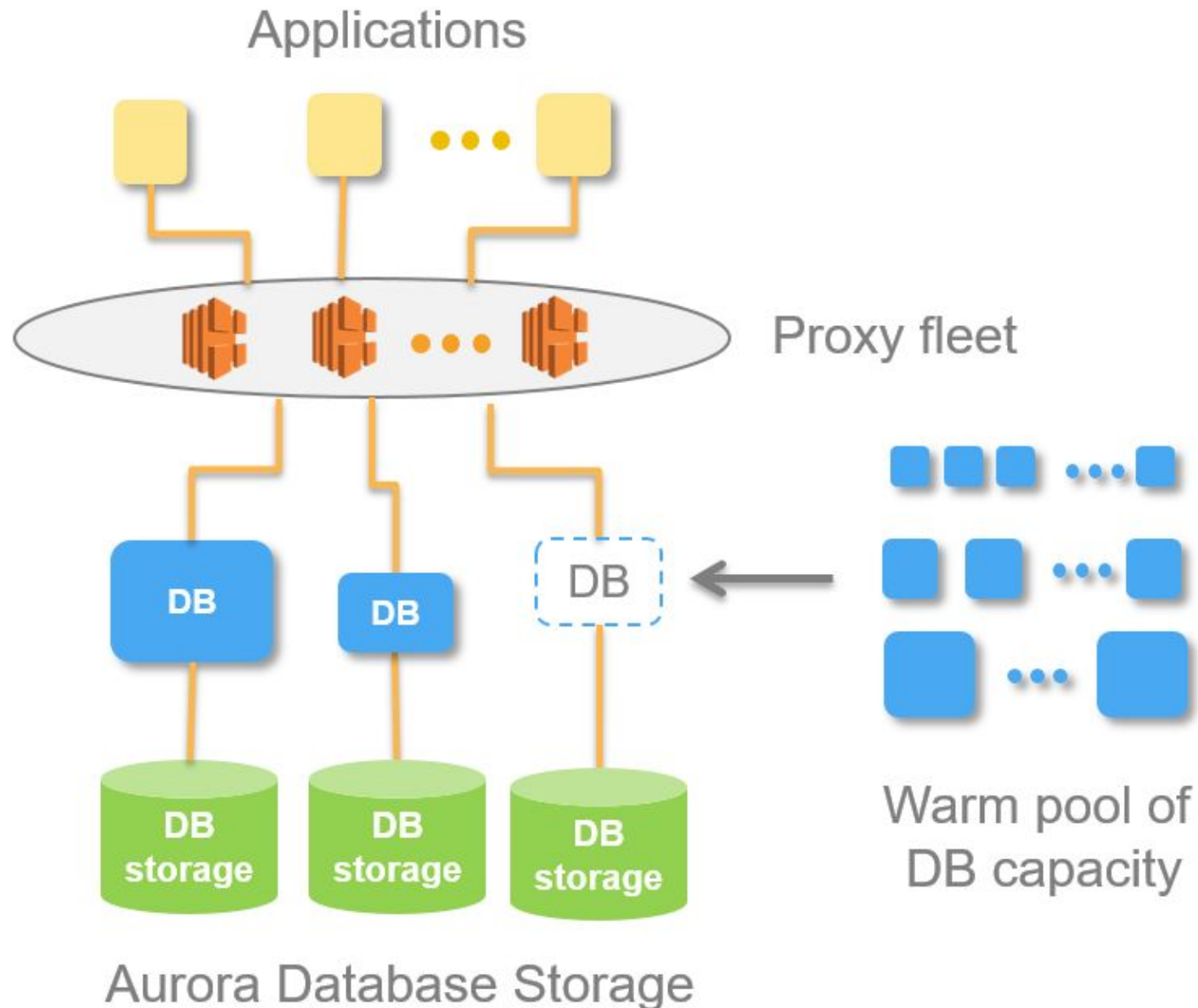
256
488GB RAM ▼

- Compatible solo con MySQL 5.6

☐ Pause compute capacity after consecutive minutes of inactivity [Info](#)
You are only charged for database storage while the compute capacity is paused

- ☒ MySQL 5.6-compatible
Aurora Serverless and Parallel Query capacities are only available with this edition.
- ☐ MySQL 5.7-compatible
- ☐ PostgreSQL-compatible

Aurora Serverless



Demo - Creación de una base de datos AURORA.

DynamoDB

No SQL



No relacional.

Completamente administrada.



Compuesta de varios Nodos.


Distribuida en varias regiones.

Baja latencia.



Almacenamiento en Caché.

Completamente escalable.



Unidades de lectura (RCU) 4KB Bloques seg.
Unidades de Escritura (WCU) 1KB Bloques seg.



La BD está replicada en diferentes locaciones.



Al crear la tabla se deben especificar las capacidades de lectura y escritura.

DynamoDB - Key Concepts

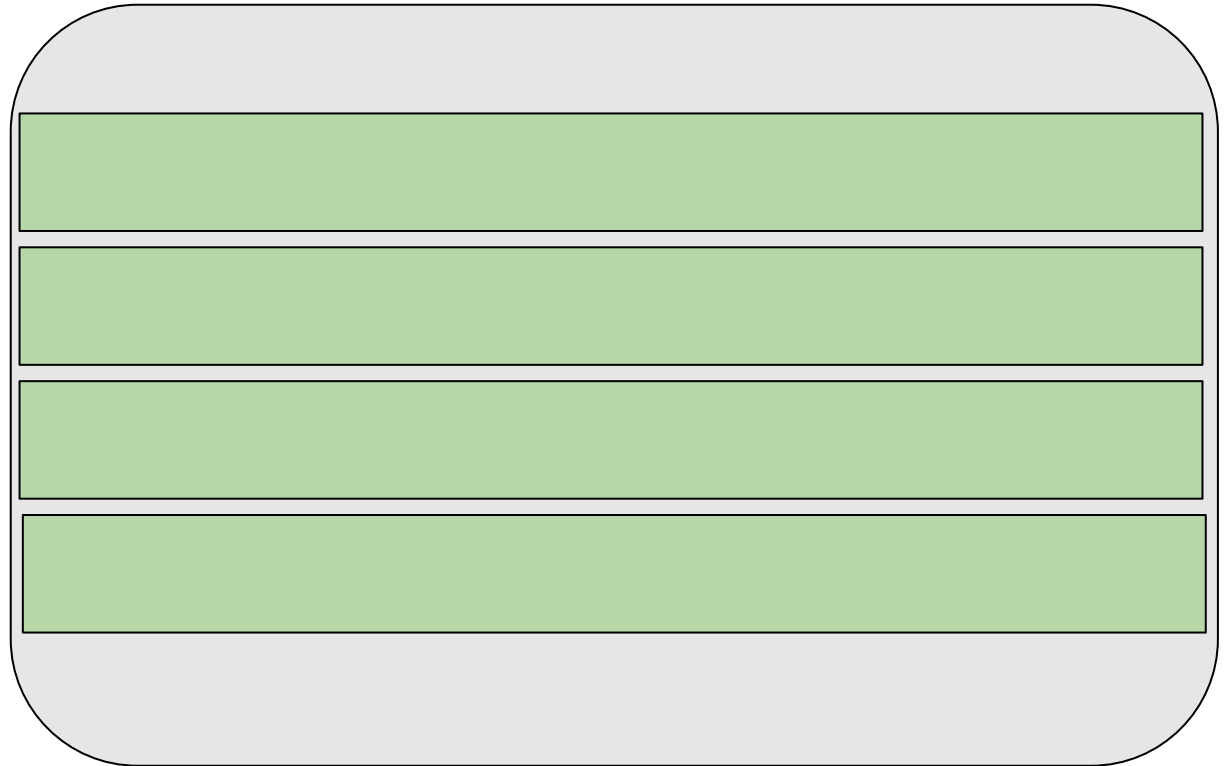
Table



DynamoDB - Key Concepts

Table

Items

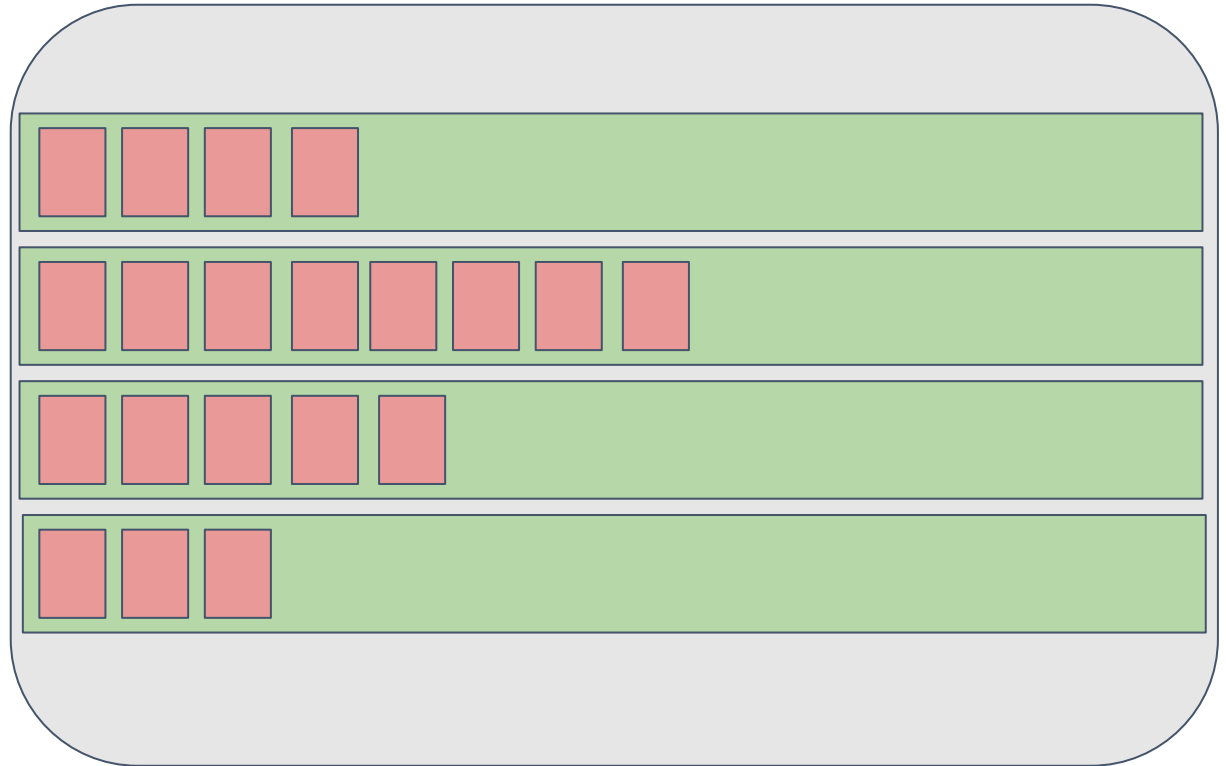


DynamoDB - Key Concepts

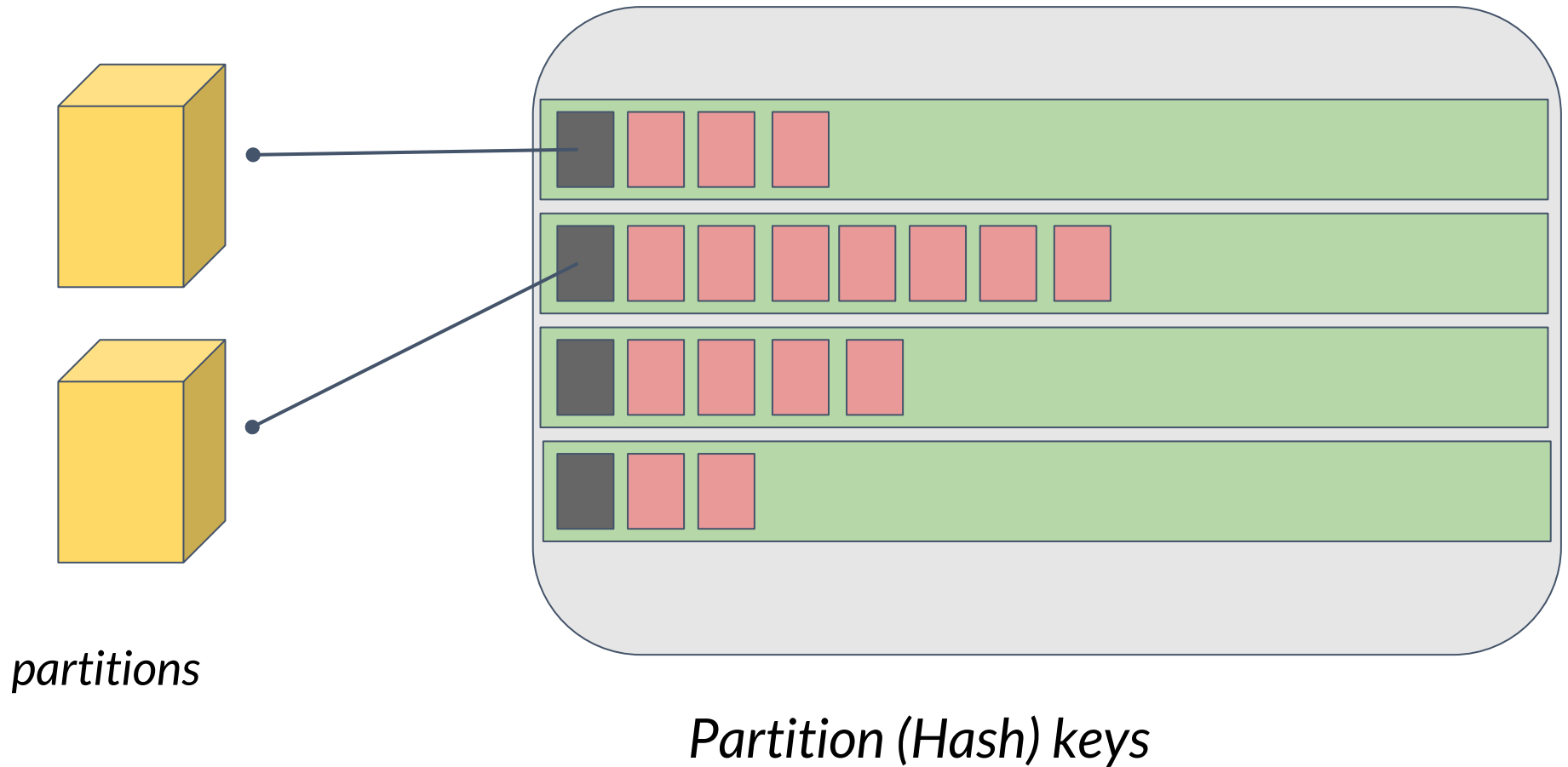
Table

Items

Attributes



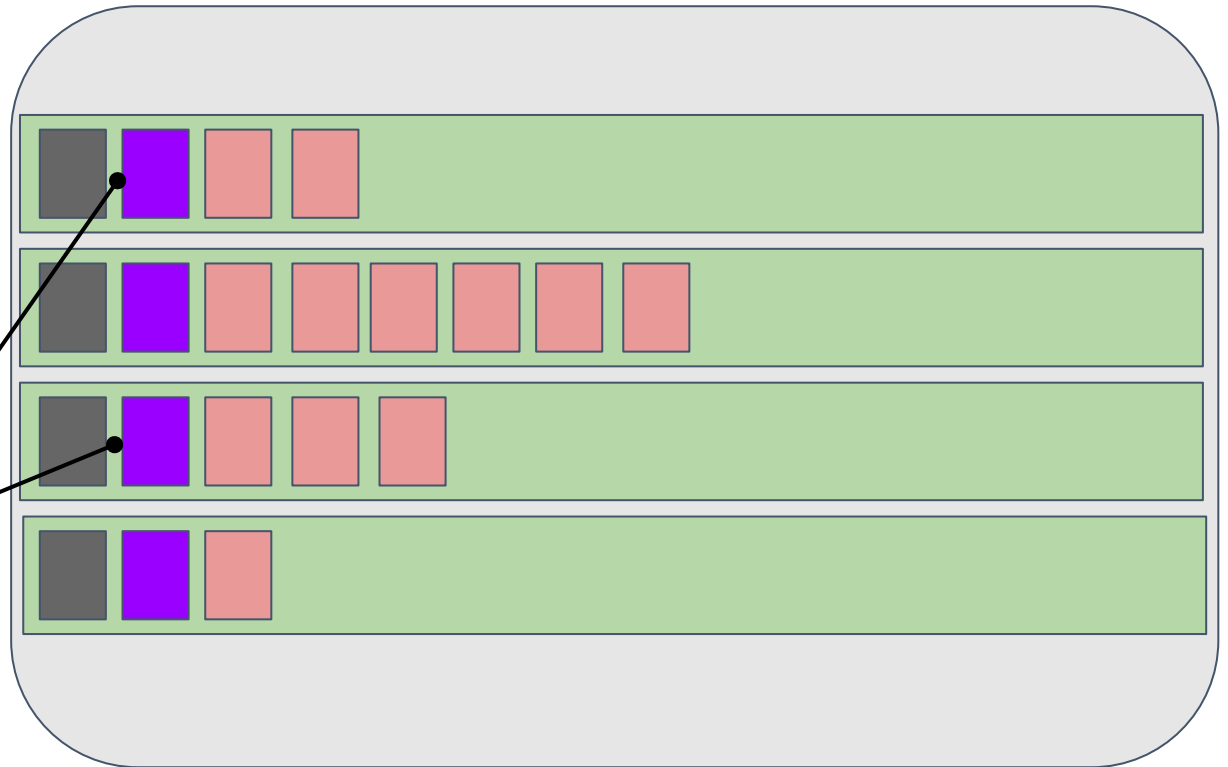
DynamoDB - Key Concepts



DynamoDB - Key Concepts

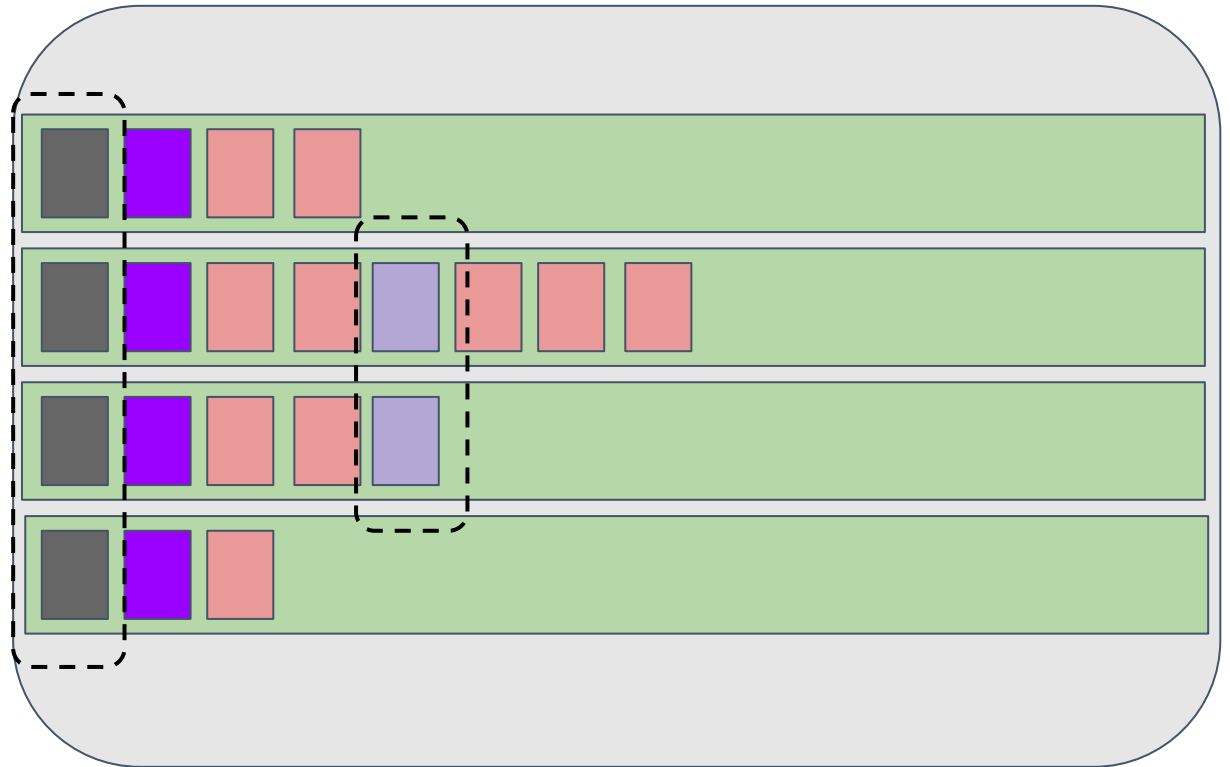
Partition (Hash) keys

Sort (Range) keys



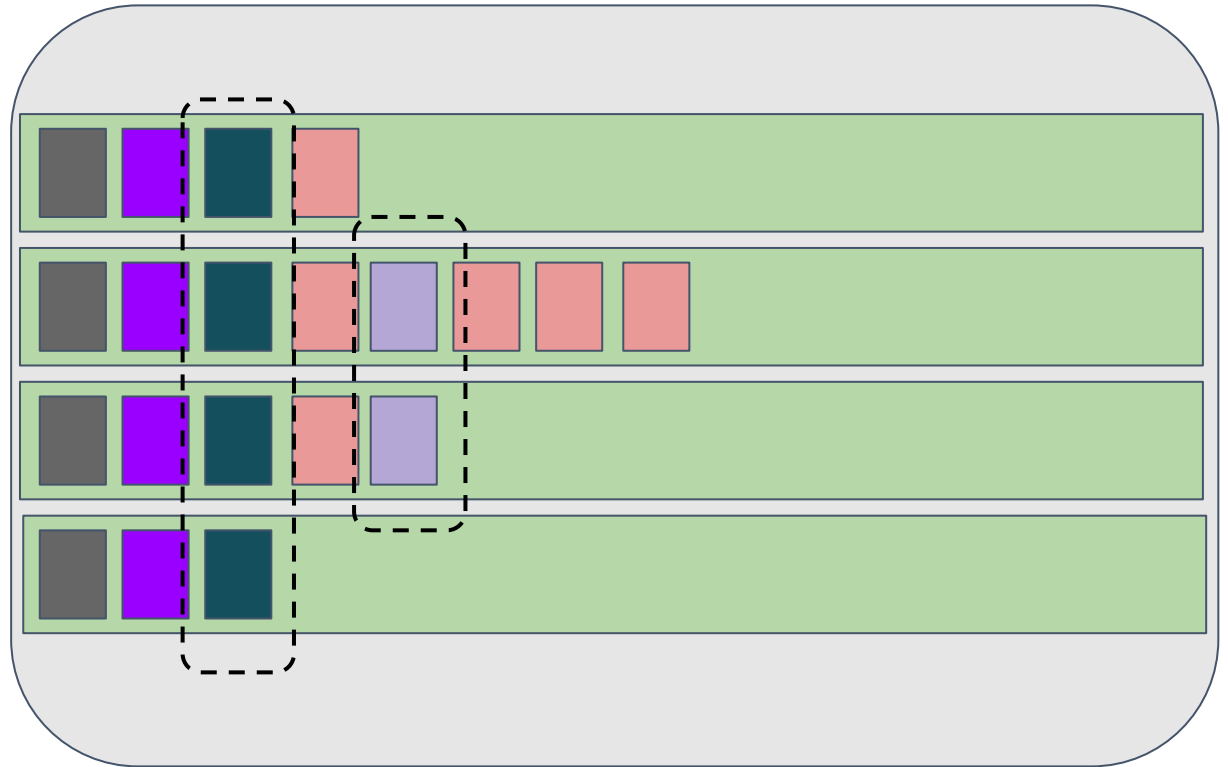
DynamoDB - Key Concepts

*Local Secondary
Index*



DynamoDB - Key Concepts

Global Secondary Index



Dynamo - Tabla

```
{  
  "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"  
}
```

Consistencia Eventual de Lectura

Funcionamiento

Cuando consultamos una tabla la respuesta puede no mostrar los resultados de una tarea de escritura reciente.

Consumo

Consume 4KB de bloques por segundo.

Consistencia Fuerte de Lectura

Funcionamiento

Cuando consultamos una tabla la respuesta es la más reciente que se encuentre en la tabla.

Consumo

Consume el doble que la consistencia eventual.

Demo - Creación de una base de datos DynamoDB.

DYNAMODB

Casos de Uso



Mobile.

IoT

Web.

Gaming.

Manejo de sesiones.

RealTime.

Particiones e Índices- DynamoDB

DynamoDB

Definición

Los datos se almacenan en particiones.
Es una asignación de almacenamiento.

Asignación

Al crear una tabla DynamoDB se encarga de asignar las particiones para que la tabla pueda satisfacer el desempeño aprovisionado.

Aumento

En ciertas situaciones DynamoDB asigna particiones adicionales a una tabla: Aumento desempeño o partición llena.

Límites

Partición puede ser de hasta **10GB**.
3.000 RCU y 1.000 WCU.

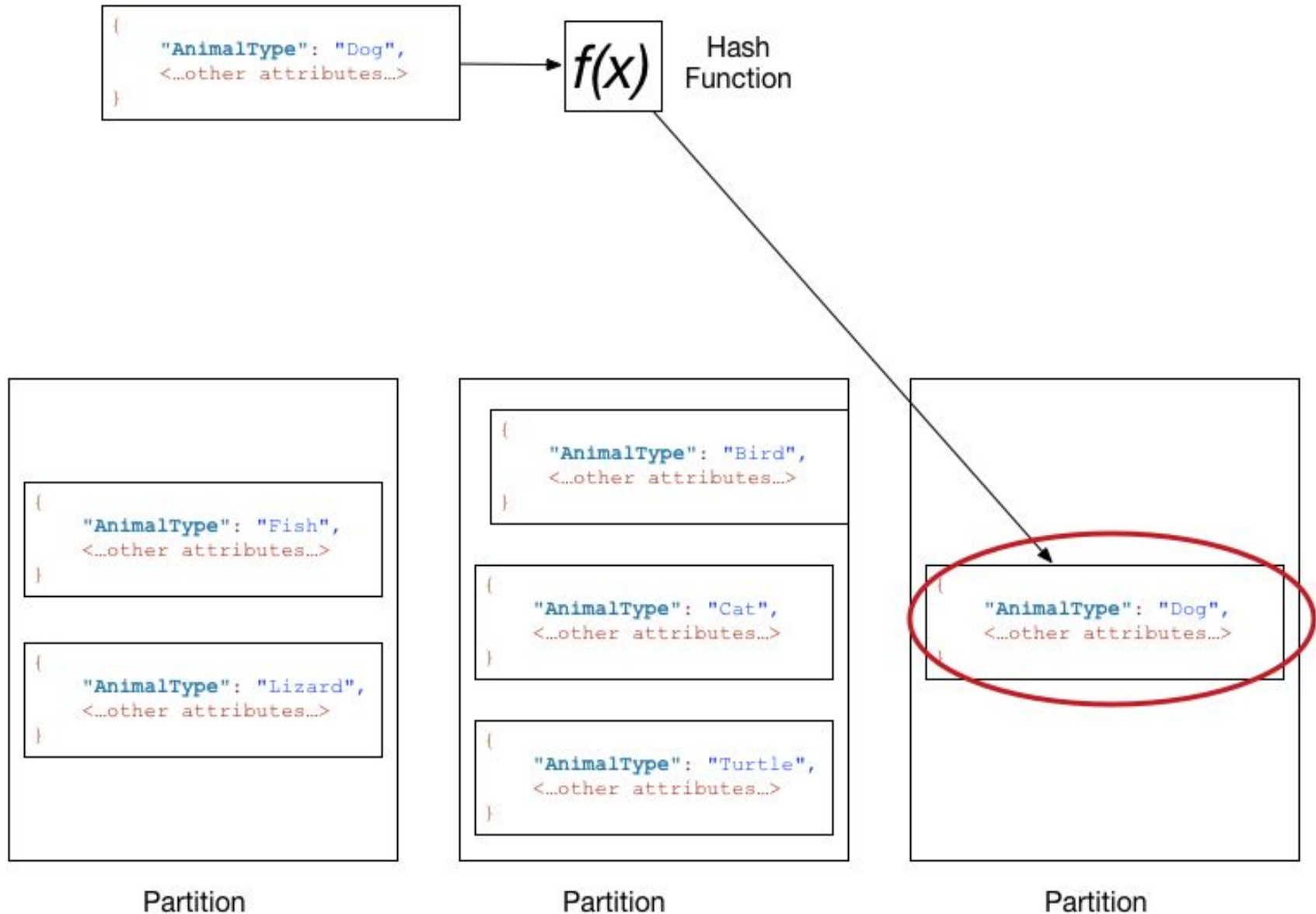
Asignación

Si una tabla tiene una clave principal simple **DynamoDB** almacenará y recuperará cada elemento basándose en esa clave.

Hash

DynamoDB utiliza el **valor de la clave de partición** como información para una función hash interna.

DynamoDB - Particiones



“

DynamoDB está optimizado para distribuir los elementos uniformemente entre las particiones de una tabla, con independencia del número de particiones que haya.
*Recomendamos **elegir una clave de partición que pueda tener un amplio abanico de valores distintos** en relación con el número de elementos de la tabla.*

”

Clave Compuesta

Cuando la tabla tiene una clave compuesta el elemento se almacena en la partición determinada por la clave.

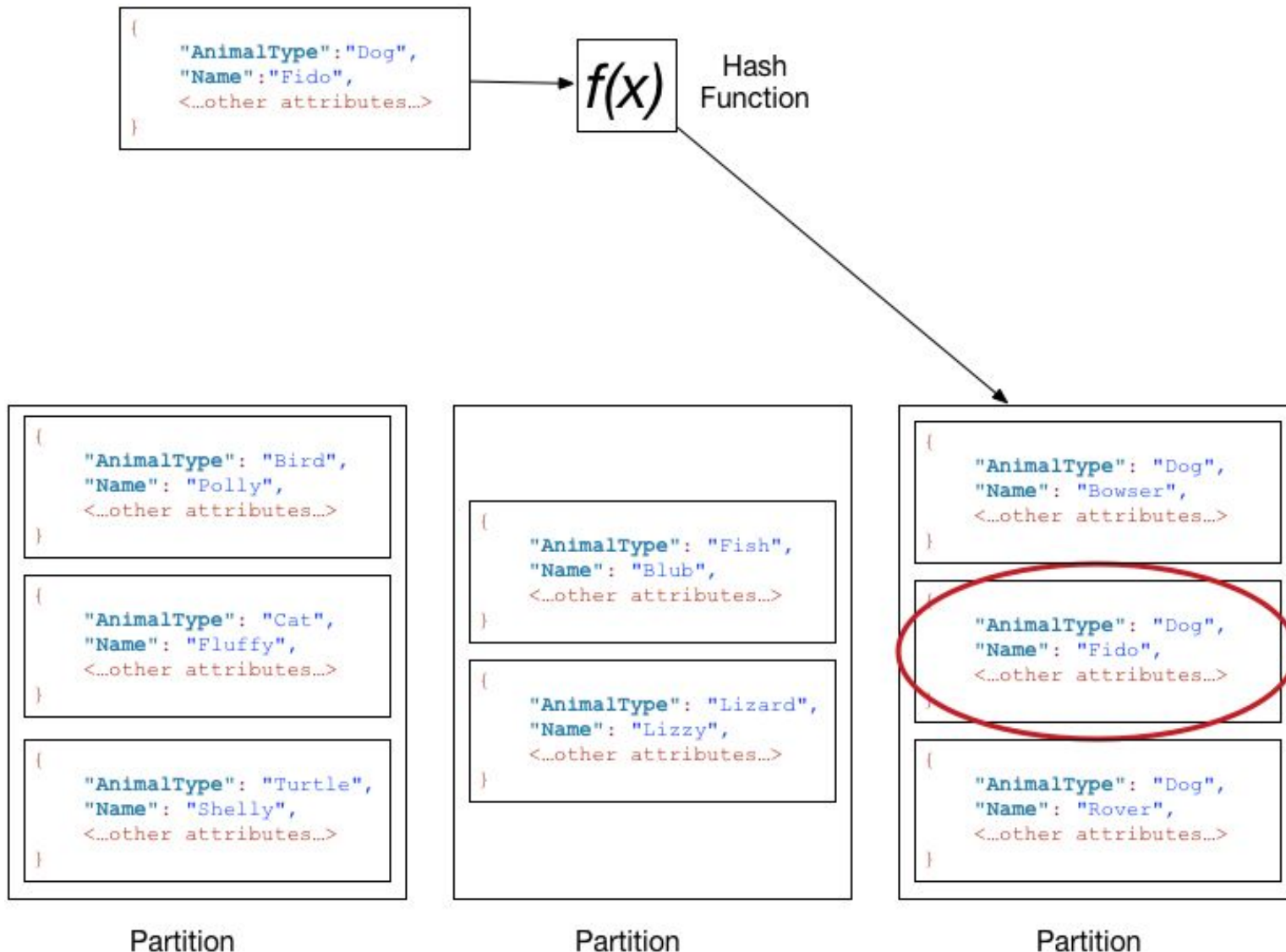
Ordenación

El elemento queda en la partición especificada anteriormente de acuerdo a la clave de ordenación.

Aumento

Para leer el elemento se deben especificar los valores de su clave compuesta.

DynamoDB - Particiones



Operaciones Scan

- Son menos eficientes.

- Examina la tabla o índice secundario en su totalidad.

- Para tablas grandes evite usar Scan.

- Examina cada elemento para comprobar si presenta los valores solicitados.

Lectura

Utiliza bastantes unidades de lectura.

Consumo

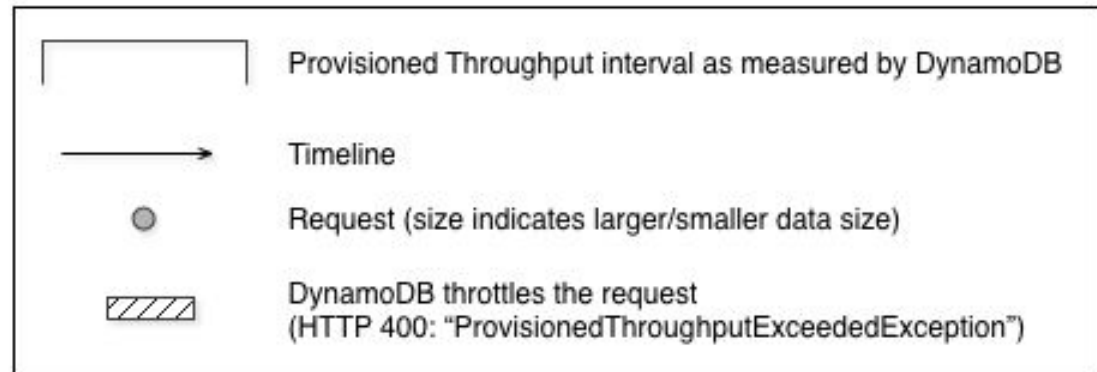
Realiza lecturas consistentes finales de forma predeterminada y puede devolver hasta 1 MB (una página) de datos

Costo

Al consumir las unidades de lectura provoca aumento de costos.

DynamoDB

- Prácticas consultas



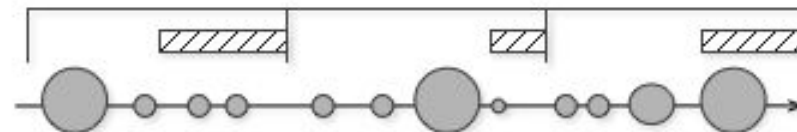
1. Good: Even distribution of requests and size



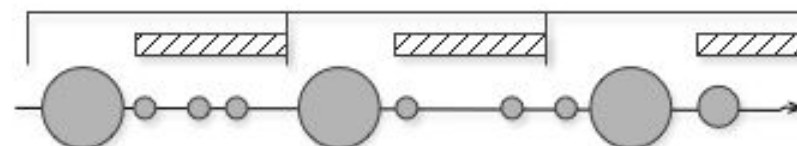
2. Not as Good: Frequent requests in bursts



3. Bad: A few random large requests



4. Bad: Large scan operations



Optimización en Scan

- Reducir el tamaño de la página usando el parámetro LIMIT en el request.

- Se pueden duplicar tablas para hacer operaciones Scan sobre tablas no principales.

DynamoDB - Scan

Examen ▼

[Tabla] ▼

^

Filtro

Escribir atributo

Cadena ▼

= ▼

Introducir un valor

×

Y

Escribir atributo

Cadena ▼

= ▼

Introducir un valor

×

+ Añadir filtro

Iniciar búsqueda

Cancelar cambios

Operaciones Consulta

- Busca elementos según los valores de la clave principal.

- Puede consultar cualquier tabla o índice secundario que cuenta con una clave principal compuesta (una clave de partición y una clave de ordenación).

Query

Para especificar los criterios de búsqueda, se utiliza una expresión de condición de clave.

Condición de clave

Se trata de una cadena que determina los elementos que se van a leer en la tabla o el índice.

Condiciones

Debe especificar el nombre y valor de la clave de partición como una condición de igualdad.

DynamoDB - Consulta

- $a = b$: es true si el atributo a es igual al valor b .
- $a < b$: es true si a es menor que b .
- $a \leq b$: es true si a es menor o igual que b .
- $a > b$: es true si a es mayor que b .
- $a \geq b$: es true si a es mayor o igual que b .
- a BETWEEN b AND c : es true si a es mayor o igual que b y menor o igual que c .

<i>Si Query se aplica...</i>	<i>DynamoDB consume unidades de capacidad de lectura..</i>
Tabla	La capacidad de lectura provisionada de la tabla
Global secondary index	La capacidad de lectura provisionada del índice
Índice secundario local	La capacidad de lectura provisionada de la tabla base

Límites

Limitar el número de elementos que devuelve en el resultado. Para ello, establezca el parámetro Limit.

Consumo

De forma predeterminada, una operación Query no devuelve datos sobre la cantidad de capacidad de lectura que consume.

Consumo

Puede especificar el parámetro ReturnConsumedCapacity en una solicitud Query para obtener información de consumo.

DynamoDB - Consulta

Consulta ▾

[Tabla] ▾

⬆

Clave de partición

Cadena

=

Introducir un valor

Filtro

Escribir atributo

Cadena ▾

= ▾

Introducir un valor

✕

Y

Escribir atributo

Cadena ▾

= ▾

Introducir un valor

✕

⊕ Añadir filtro

Orden

☐ Ascendente

☒ Descendente

Atributos

☐ Todo

☒ Proyectado

Iniciar búsqueda

Cancelar cambios

Demo - Scan y Query en una tabla DynamoDB.

Streams y Replicación- DynamoDB

DynamoDB

Streams

- Proporciona una **secuencia ordenada** por tiempo de cambios de nivel de elemento en cualquier tabla.

- Permite ampliar el poder de DynamoDB con replicación entre regiones, análisis continuo con integración con **Redshift**, **notificaciones de cambios** y muchos otros escenarios

Funcionamiento

Captura una secuencia en orden cronológico de las modificaciones de los elementos en una tabla y almacena esta información - 24 horas.

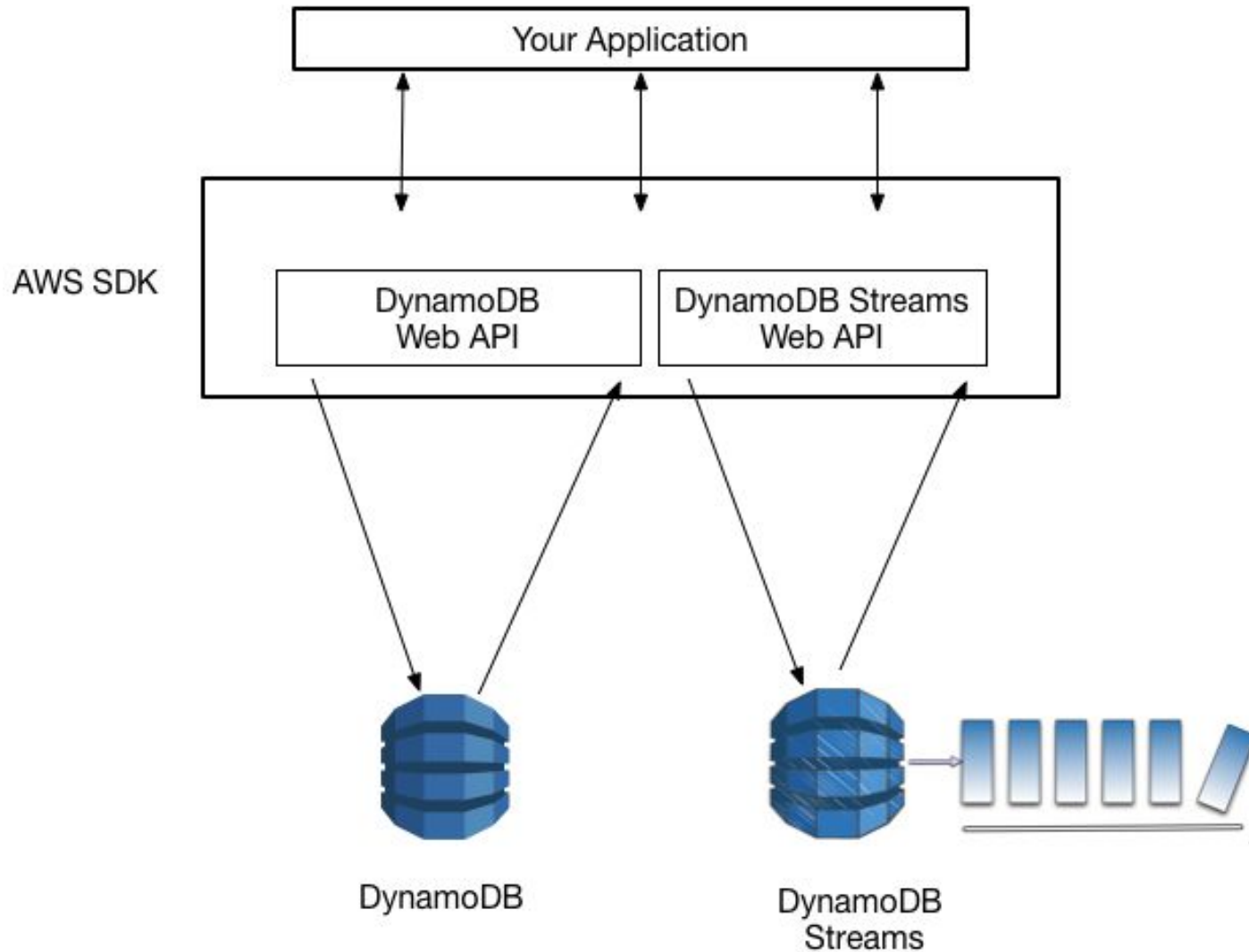
Aplicaciones

Las aplicaciones pueden obtener acceso a este registro y ver los elementos de datos tal y como se encontraban antes y después.

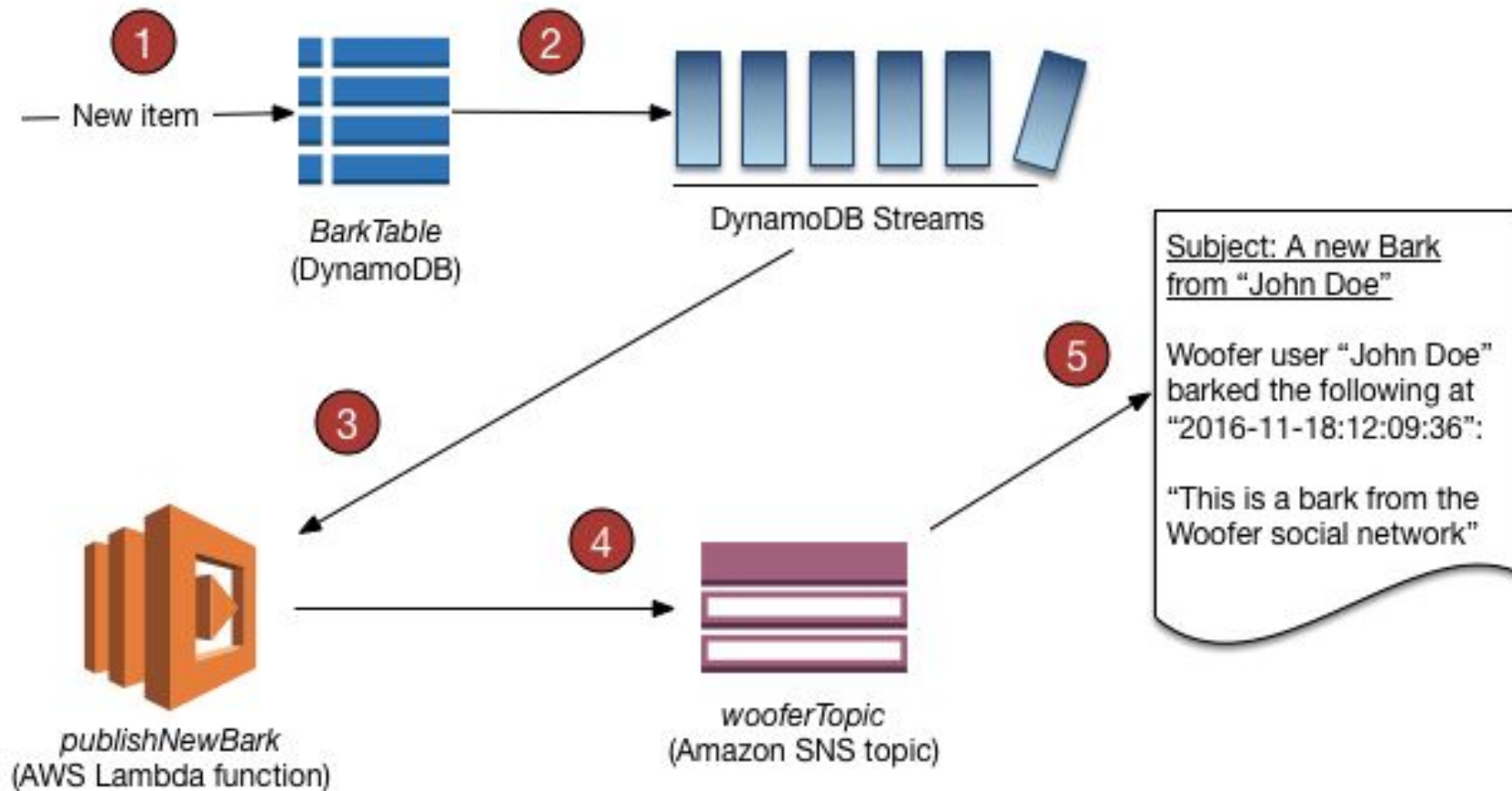
Registro

Un registro de secuencia contiene información sobre una modificación de los datos de un solo elemento de una tabla.

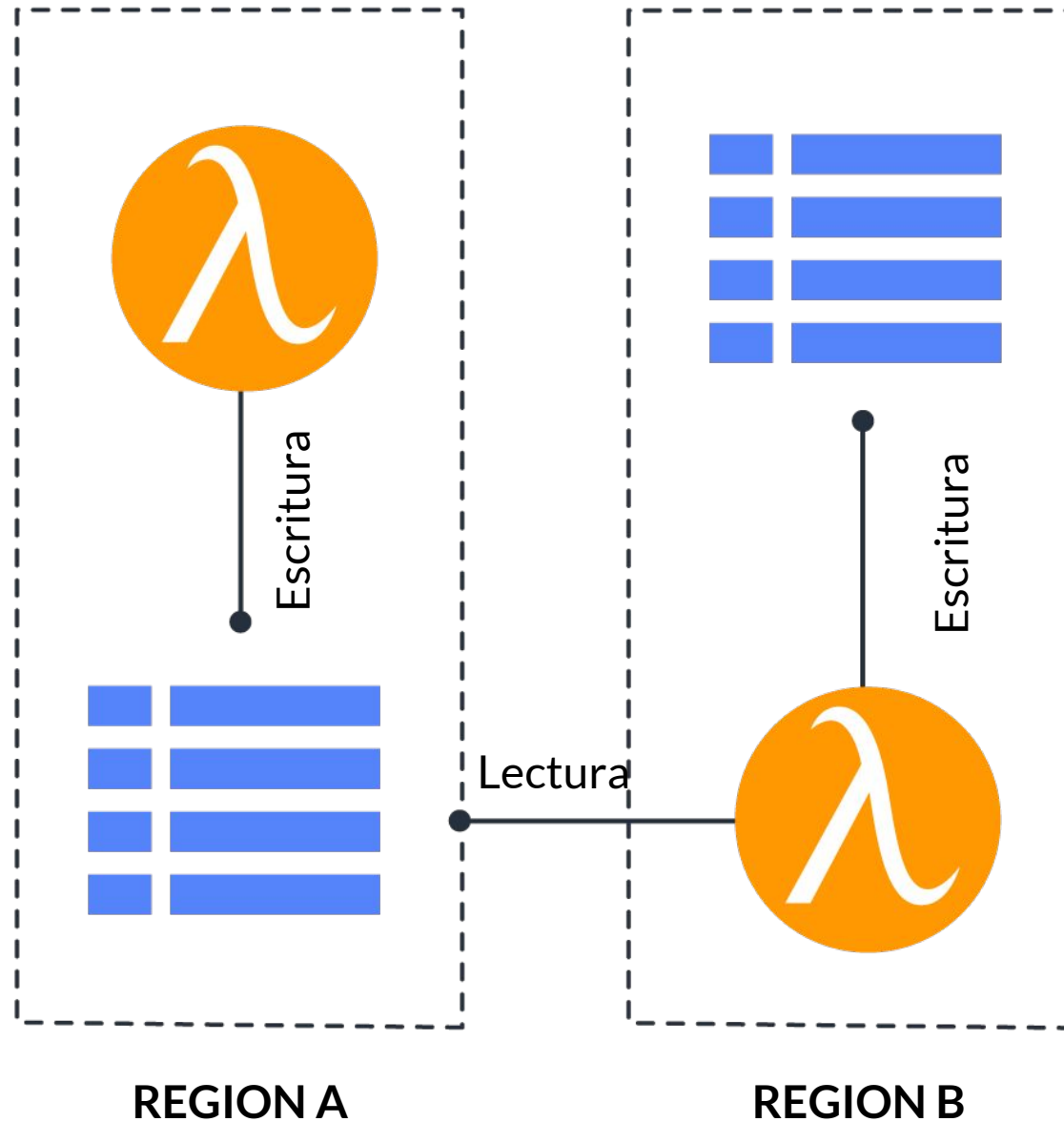
DynamoDB - Streams



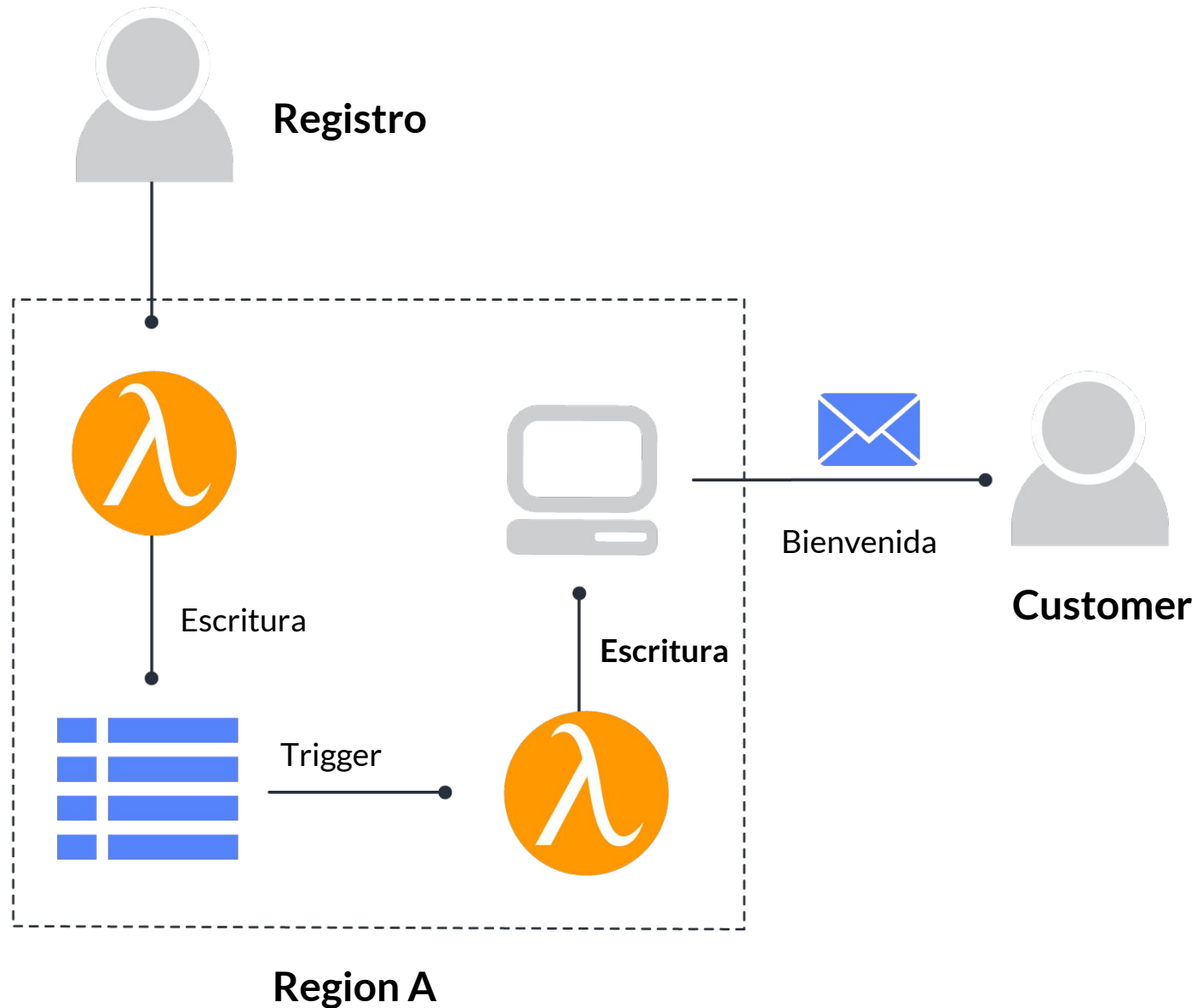
DynamoDB - Streams



DynamoDB - Streams



DynamoDB - Streams




DYNAMODB

DAX



Es un caché completamente administrado y de alta disponibilidad para DynamoDB.



Tiene un rendimiento superior hasta 10 veces, de milisegundos a microsegundos. Soporta millones de solicitudes por segundo.

Permite encriptación en reposo.

1

Permite hasta 10 nodos.

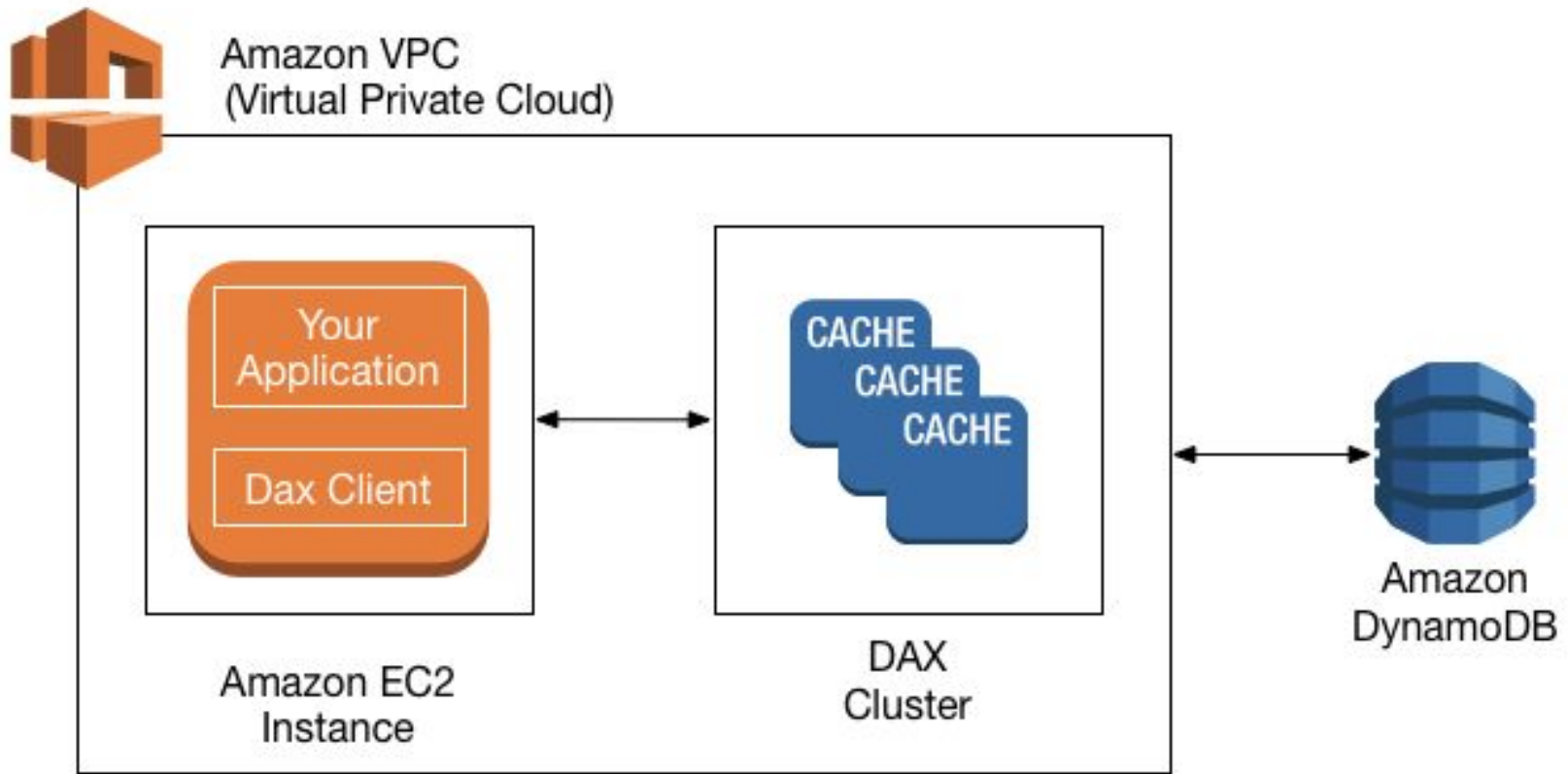
2

Permite instancias small y medium para cargas de prueba, de resto todas son tipo R (optimizadas en memoria).

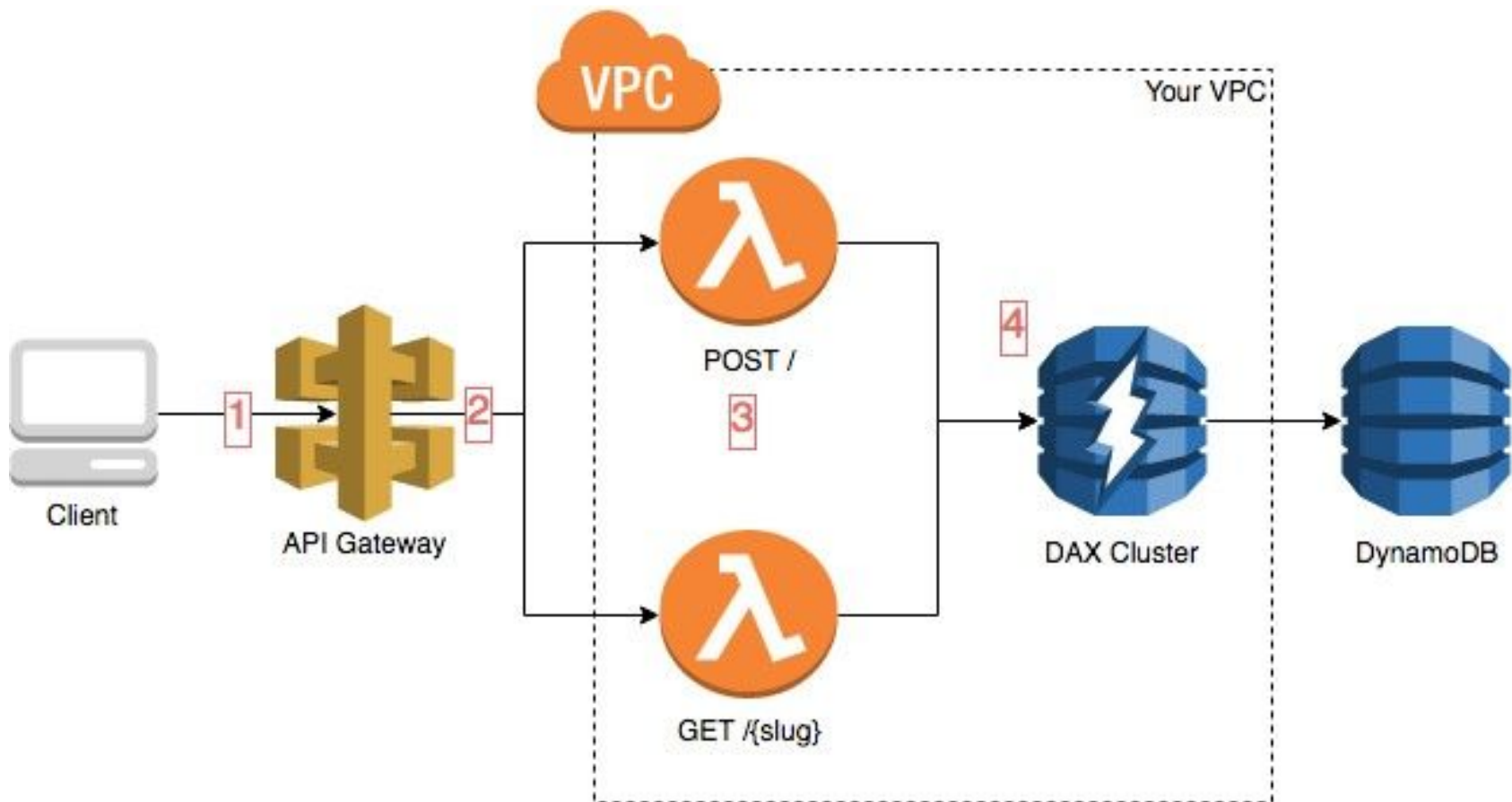
3

Se puede seleccionar la zona de disponibilidad en donde se desplegará el clúster.

DynamoDB - DAX



DynamoDB - DAX



Conclusiones - RDS

DynamoDB

1

Es recomendado las bases de datos de producción RDS sean desplegadas Multi AZ.

2

Para optimizar el performance de una base de datos RDS tenemos diferentes estrategias:

- Implementar réplicas de lectura.
- Bases de datos en memoria.
- Dividir la base de datos en más pequeñas.

3

Al desplegar nuestra RDS debemos tener en cuenta el período de retención de 1 a 35 días como estrategia de backup.

4

Cuando vamos a migrar a RDS debemos tener en cuenta el tipo de migración (heterogénea u homogénea).

5

Aurora es una base de datos para grandes cargas de trabajo y es la única serverless.

Conclusiones - DynamoDB

DynamoDB

1

Si se pueden usar Querys es mucho mejor para el rendimiento de la base de datos

2

Evitar el uso de Scan para no afectar la capacidad provisionada.

3

La función de autoscaling se puede programar con lectura y escritura pero se deben tener en cuenta los costos.

4

Es clave elegir una llave principal adecuada para no afectar el performance de nuestra DynamoDB.

5

Con DynamoDB Streams podemos tener arquitecturas en tiempo real para diferentes casos de uso.

6

Cuando tenemos una tabla muy grande el particionamiento es fundamental y los límites del mismo.

DynamoDB

DynamoDB



No relacional.

Completamente administrada.



Compuesta de varios Nodos.

Distribuida en varias regiones.

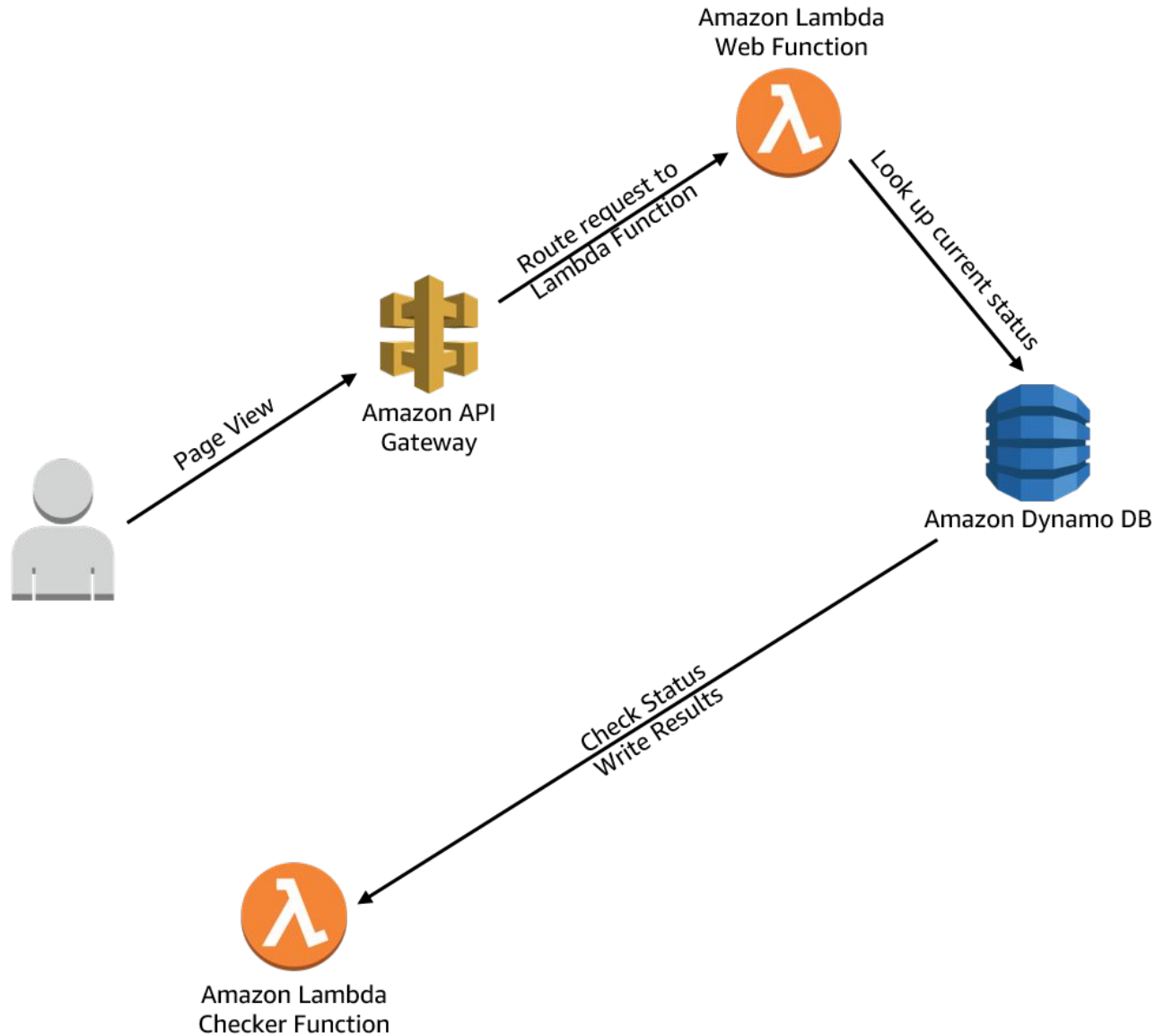
Baja latencia.



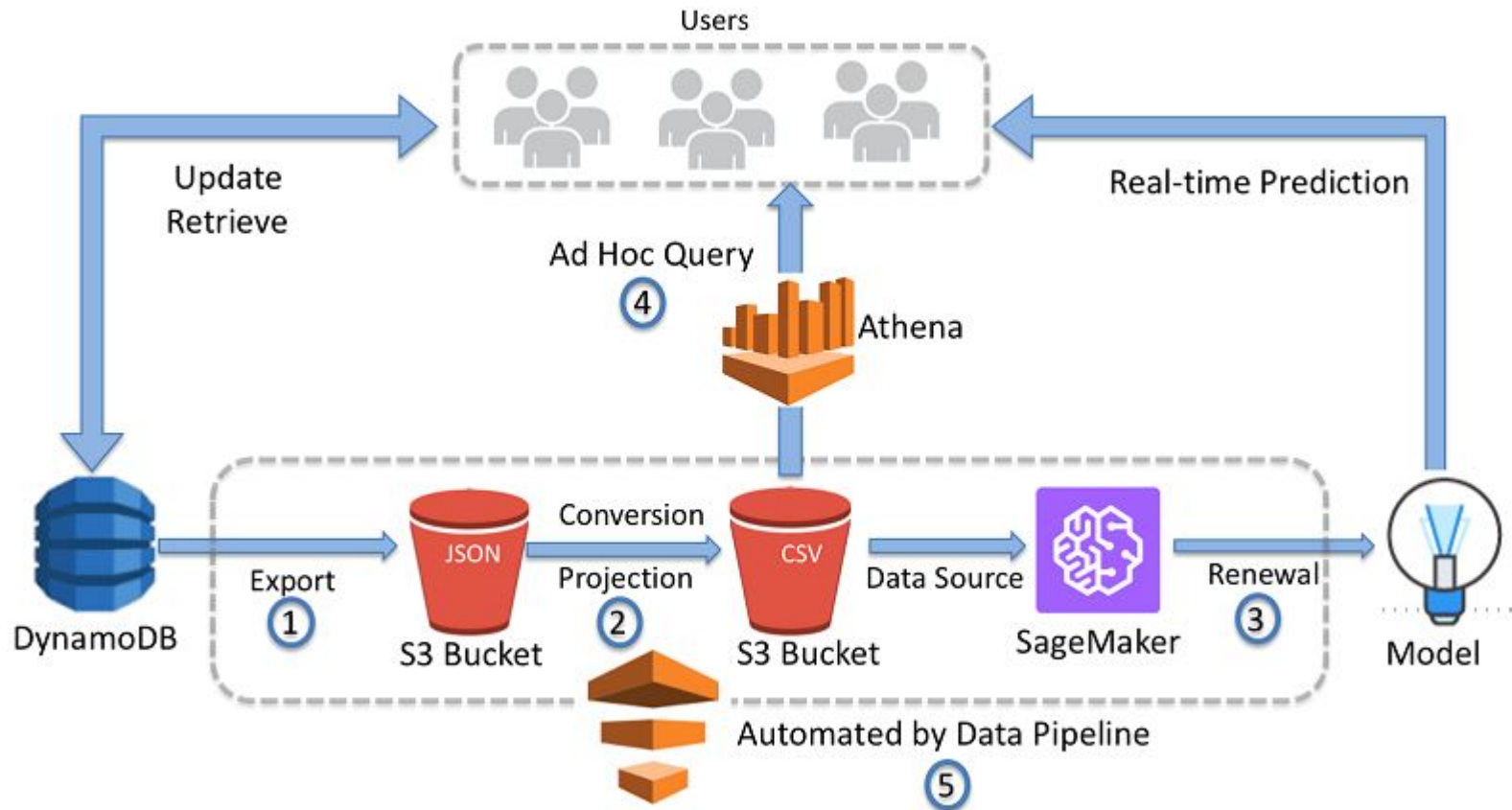
Almacenamiento en Caché.

Completamente escalable.

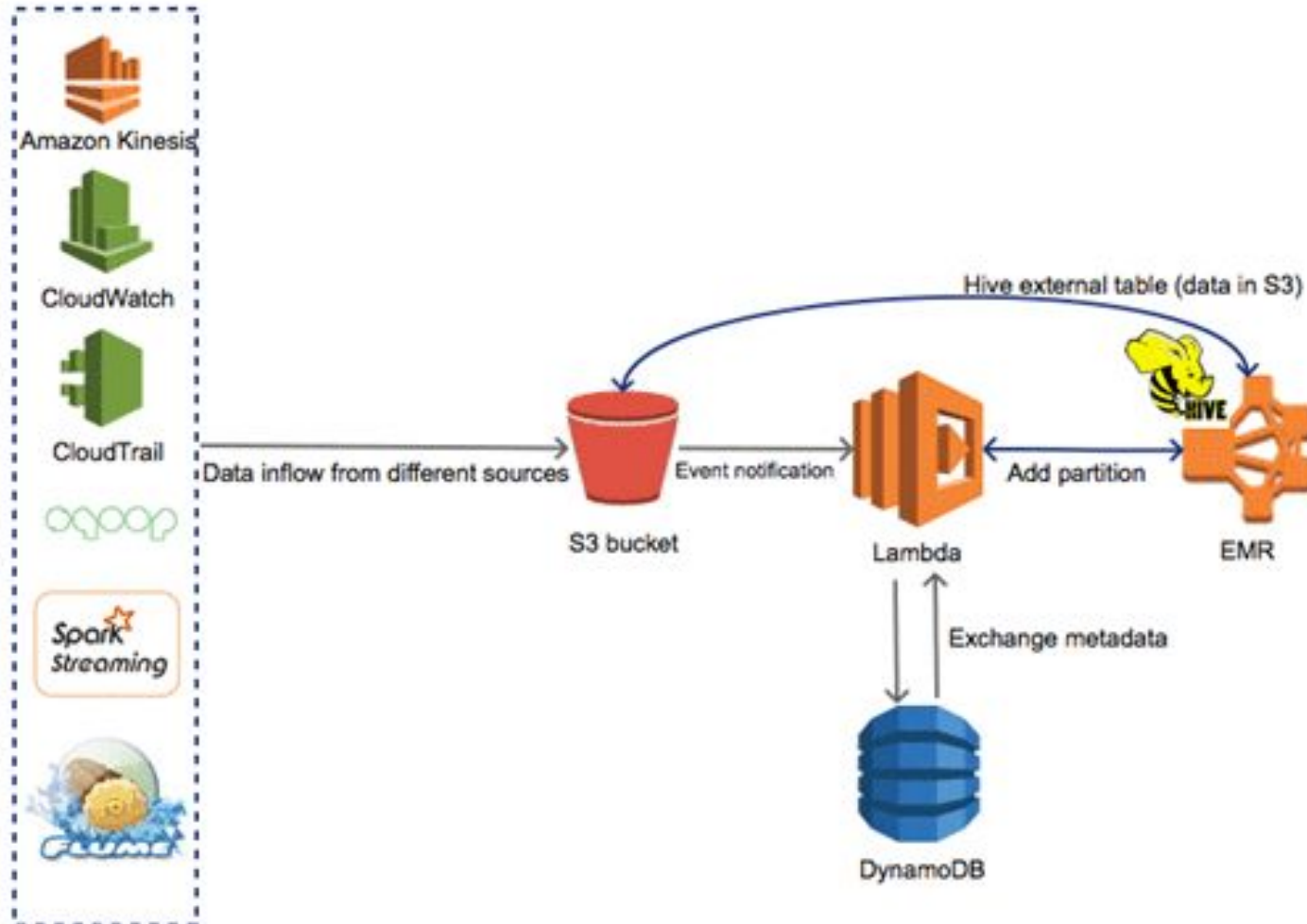
DynamoDB - Sitios Web Serverless



DynamoDB - Machine Learning



DynamoDB - BigData



DynamoDB - RealTime

