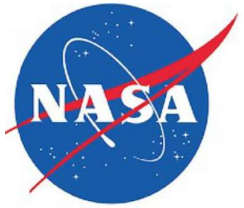
¿Qué es AWS?

- AWS es un proveedor de Cloud
- **Te proporcionan servidores y servicios que puedes utilizar bajo demanda y escalar fácilmente**

- AWS ha revolucionado la IT a lo largo del tiempo
- AWS impulsa algunos de los mayores sitios web del mundo
  - Netflix, Airbnb, NASA, Twitch, Samsung, Slack…









© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Sobre el instructor

Joan Amengual

- Ingeniero Full Stack en una empresa tecnológica en Silicon Valley, USA
- He trabajado con AWS, Azure y Google Cloud varios años en diversas empresas para la migración y el escalado de servicios en el Cloud
- Premiado como Joven Talento en Ingeniería

- Puedes encontrarme en:
  - LinkedIn: https://www.linkedin.com/in/joanamengual7
  - Frecuentemente hago publicaciones interesantes sobre el Cloud



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Comunidad - Cloud Computing



LinkedIn



Discord



Telegram

https://blockstellart.com/comunidades

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

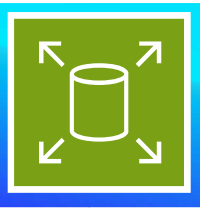


Almacenamiento de EC2

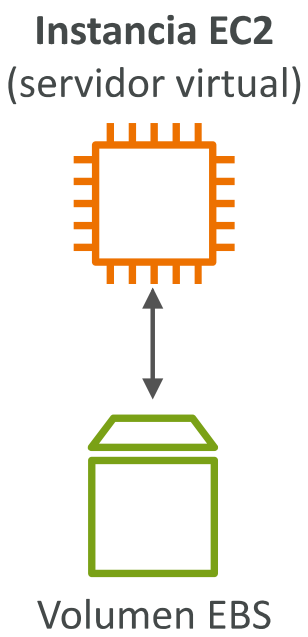
www.blockstellart.com

Todos los derechos reservados © Joan Amengual www.blockstellart.com

¿Qué es un volumen EBS?



- Un volumen EBS (Elastic Block Store) es una **unidad de red que puede adjuntar a las instancias** mientras se ejecutan
- Permite que las instancias persistan los datos, incluso después de su finalización
- Están vinculados **a una zona de disponibilidad específica**
- Piensa en ellos como una "memoria USB de red"

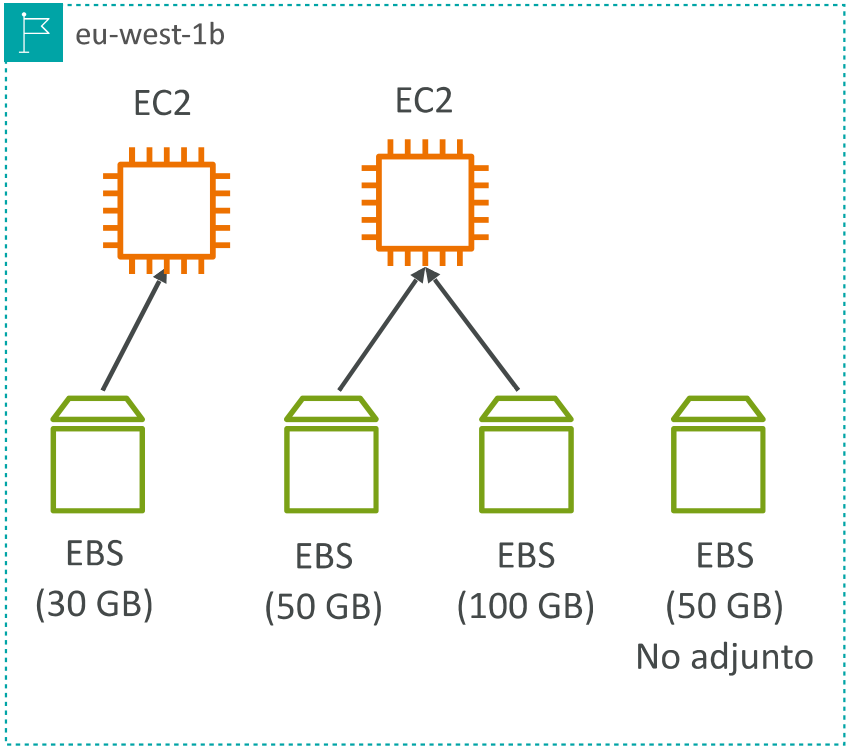


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Detalles importantes de los volúmenes EBS

- Es una unidad de red (es decir, no es una unidad física)
  - Utiliza la red para comunicar la instancia, lo que significa que puede haber un poco de latencia
  - Se puede separar de una instancia EC2 y conectarla a otra rápidamente
- Está bloqueado en una Zona de Disponibilidad (AZ)
  - Un volumen EBS en eu-west-1a no puede adjuntarse a eu-west-1b
  - Para trasladar un volumen, primero hay que hacer un snapshot del mismo
- Tener una capacidad provisionada (tamaño en GBs, e IOPS)
  - Se facturará toda la capacidad aprovisionada
  - Puede aumentar la capacidad de la unidad con el tiempo

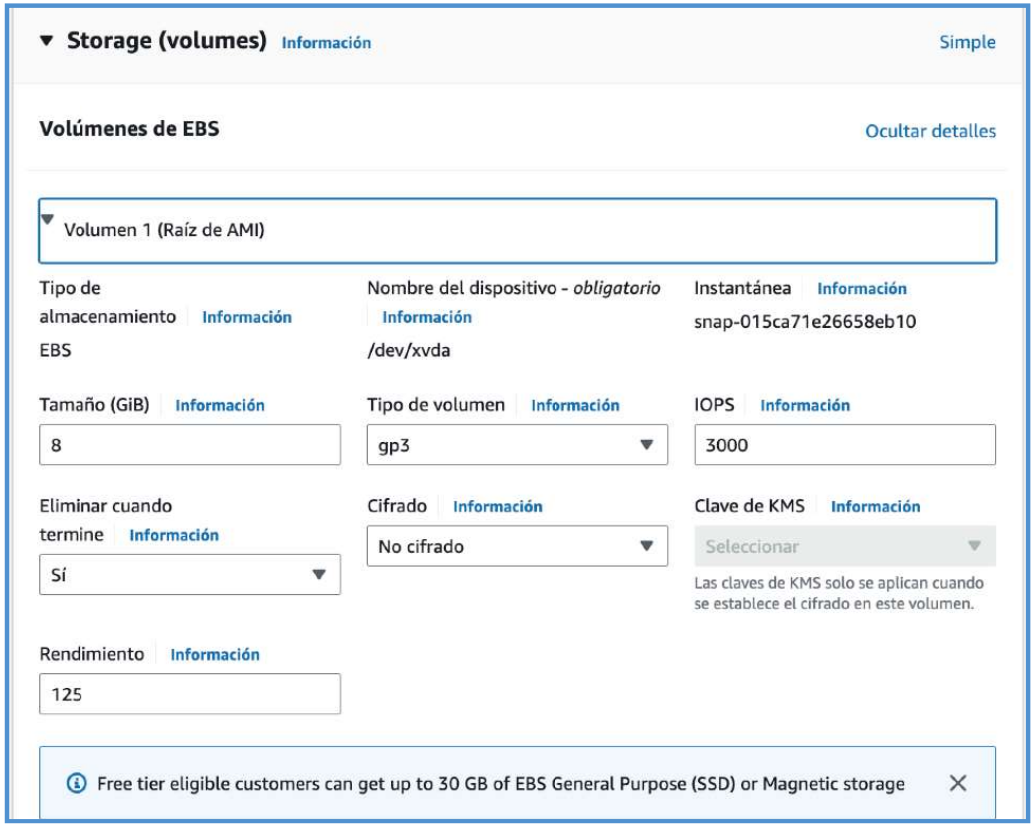


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Configuración de volúmenes EBS

- El volumen EBS se puede configurar **durante la creación de la instancia EC2**
- Los parámetros a configurar son:
  - Tamaño (GiB), tipo de volumen, IOPS, Eliminar volumen al terminar instancia EC2, cifrado, clave KMS, rendimiento
- Por defecto, se elimina el volumen EBS root / raíz (atributo habilitado)

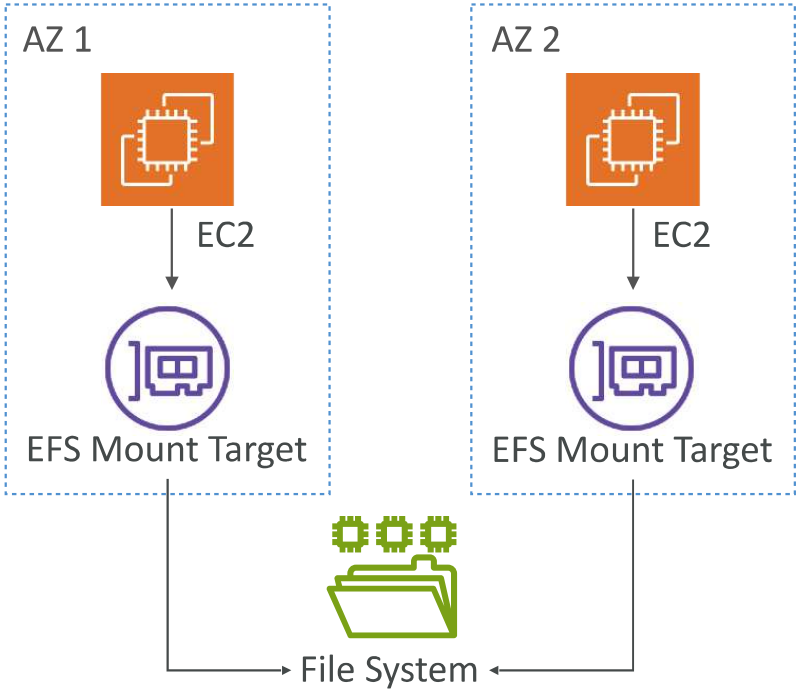


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon EFS - Elastic File System

- NFS gestionado (sistema de archivos de red) que puede montarse en muchas EC2
- EFS funciona con instancias EC2 en multi-AZ
- Alta disponibilidad, escalable, más caro, pago por uso
- Amazon EFS está **diseñado para ampliarse a petabytes** según la demanda sin interrumpir las aplicaciones



© Joan Amengual

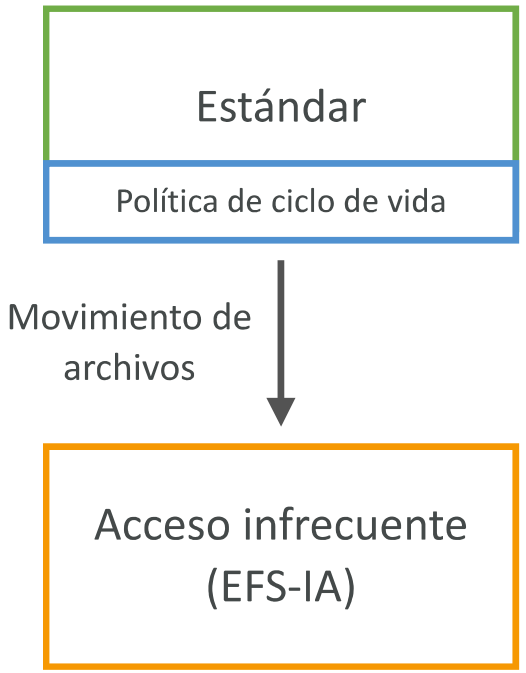
Todos los derechos reservados © Joan Amengual www.blockstellart.com

Clases de almacenamiento de Amazon EFS

- **Niveles de almacenamiento (función de gestión del ciclo de vida: mover el archivo después de N días)**
  - Estándar: para archivos de acceso frecuente
  - Acceso infrecuente (EFS-IA): coste de recuperación de los archivos, menor precio de almacenamiento. Habilita EFS-IA con una política de ciclo de vida

- **Disponibilidad y durabilidad**
  - Estándar: Multi-AZ, ideal para prod
  - Una zona: Una AZ, genial para dev, copia de seguridad activada por defecto, compatible con IA (EFS One Zone-IA)

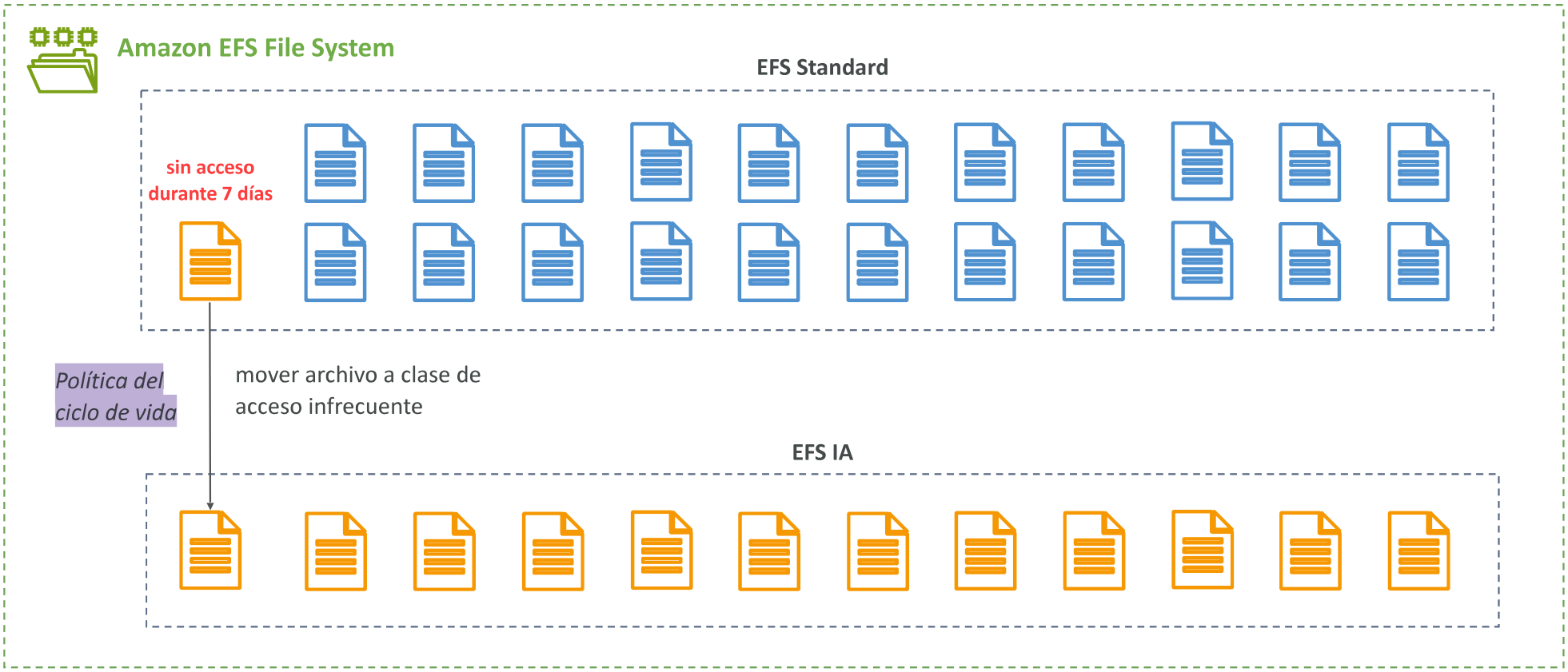
- Más del 90% de ahorro de costes



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Clases de almacenamiento de Amazon EFS



© Joan Amengual

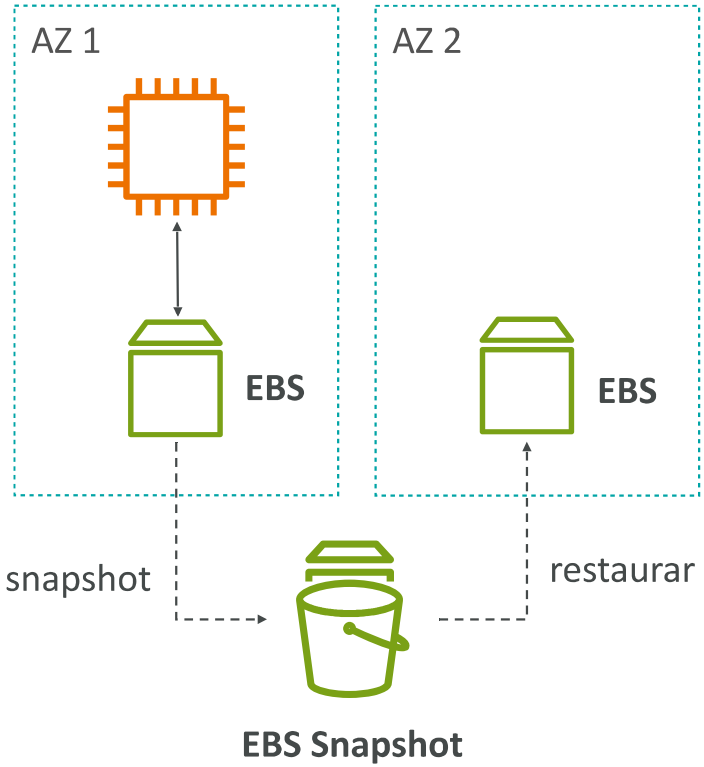
Todos los derechos reservados © Joan Amengual www.blockstellart.com

EBS vs EFS – Elastic Block Storage

- Los volúmenes EBS sólo pueden adjuntarse a una instancia a la vez o usar EBS Multi-Attach y están bloqueados a nivel de AZ
  - **Multi-Attach es compatible exclusivamente en volúmenes de SSD de IOPS aprovisionadas (io1 y io2 Block Express)**

- Para migrar un volumen EBS a través de la AZ:
  - Haz una snapshot
  - Restaura la snapshot en otra AZ

- Los volúmenes EBS root de las instancias se terminan por defecto si la instancia EC2 se termina (puedes desactivarlo)

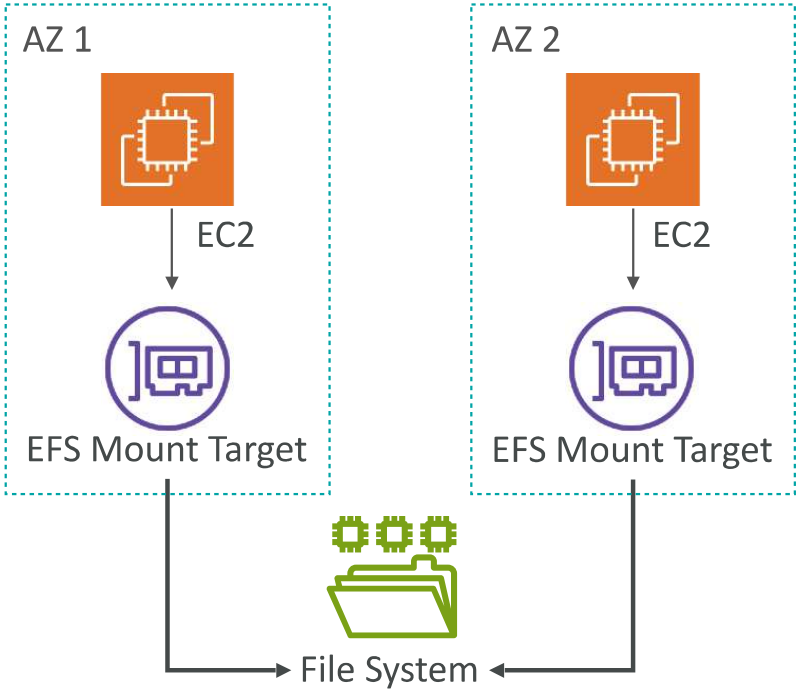


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

EBS vs EFS – Elastic File System

- EFS está diseñado para ser altamente disponible y duradero
- Permite el acceso concurrente a miles de instancias de EC2
- El almacenamiento en EFS **escala automáticamente hacia arriba o hacia abajo** a medida que se agregan o eliminan archivos
- EFS elimina la necesidad de administrar la infraestructura de almacenamiento subyacente y es más fácil de configurar y administrar en comparación con EBS
- Sólo para instancias Linux (POSIX)
- **EFS tiene un precio más elevado que EBS**



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com



Computación

www.blockstellart.com

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Instancias EC2 en el Big Data

- EC2 permite escalar recursos de computación de manera elástica según la demanda de procesamiento de Big Data
  - **Instancias spot**: Ideales para tareas que pueden tolerar interrupciones. Ofrecen bajo costo y son adecuadas para procesos con puntos de control frecuentes, como en aprendizaje automático (ML)
  - **Instancias reservadas**: Recomendadas para clusters y bases de datos de largo plazo (más de un año) debido a su estabilidad y costo predecible
  - **Instancias bajo demanda**: Utilizadas para las cargas de trabajo que no encajan en las categorías anteriores, proporcionando flexibilidad sin compromisos a largo plazo
- EC2 se integra sin problemas con otros servicios de AWS como Amazon S3 para almacenamiento, Amazon RDS y DynamoDB para bases de datos, y Amazon EMR para procesamiento de datos distribuidos

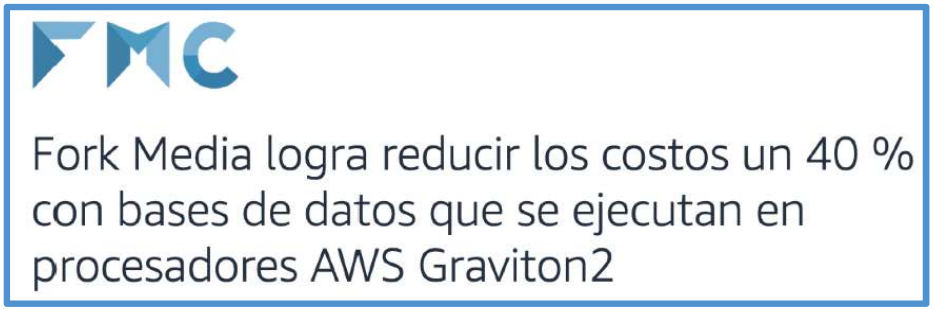


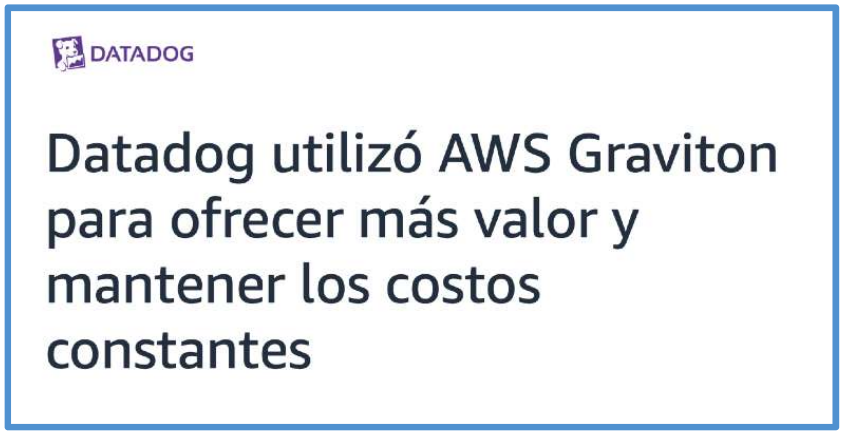
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Procesadores AWS Graviton

- AWS Graviton es una **familia de procesadores** diseñada para ofrecer el mejor precio y rendimiento para sus cargas de trabajo en la nube que se ejecutan en Amazon EC2
  - Propósito general: M7, T4
  - Optimizado para cómputo: C7, C6
  - Optimizado para memoria: R7, X2
  - Optimizado para almacenamiento: Im4, Is4
  - Cómputo acelerado (inferencia ML): G5
- **Mejor rendimiento por mejor precio**
- Opción para muchos servicios de ingeniería de datos:
  - MSK, RDS, MemoryDB, ElastiCache, OpenSearch, EMR, Lambda, Fargate





© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

BLOCKSTELLART

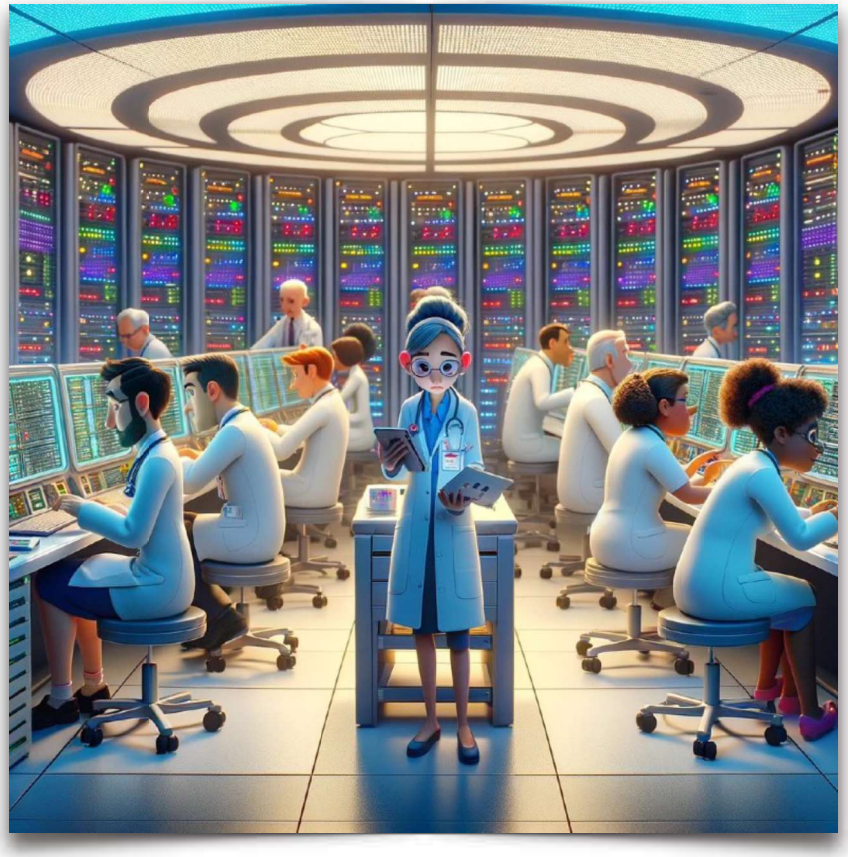
Gestión y gobierno

www.blockstellart.com

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Por qué es importante la monitorización

- Nuestras aplicaciones se despliegan, y a nuestros usuarios no les importa cómo lo hicimos...
- ¡A nuestros usuarios sólo les importa que la aplicación funcione!
  - Latencia de la aplicación: ¿aumentará con el tiempo?
  - Caídas de la aplicación: la experiencia del cliente no debe degradarse
  - Que los usuarios se pongan en contacto con el departamento de IT o se quejen no es un buen resultado
- Supervisión interna:
  - ¿Podemos prevenir los problemas antes de que ocurran?
  - Rendimiento y coste
  - Tendencias (patrones de escalado)
  - Aprendizaje y mejora



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Monitorización en AWS

- **AWS CloudWatch:**
  - Métricas: Recopila y realiza un seguimiento de las métricas clave
  - Logs: Recopila, monitoriza, analiza y almacena logs
  - Eventos: Envía notificaciones cuando ocurran determinados eventos
  - Alarmas: Reacciona en tiempo real ante métricas / eventos
- **AWS X-Ray:**
  - Solución de problemas de rendimiento y errores de la aplicación
  - Rastreo distribuido de microservicios
- **AWS CloudTrail:**
  - Monitorización interna de las llamadas a la API que se realizan
  - Auditoría de los cambios realizados por tus usuarios







© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon CloudWatch

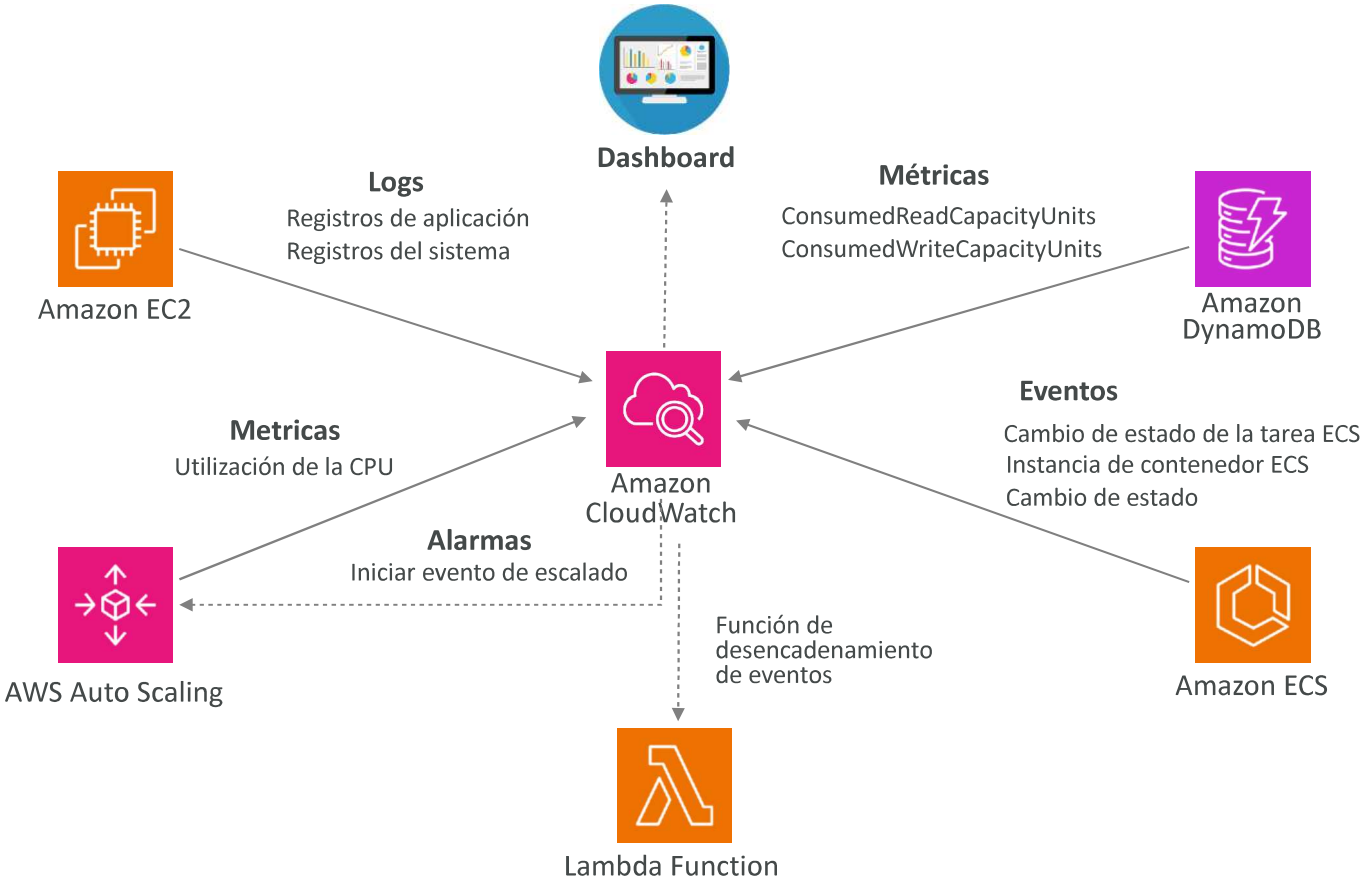
- **CloudWatch proporciona monitoreo en tiempo real de recursos y aplicaciones de AWS**
- CloudWatch proporciona métricas para todos los servicios de AWS
- Algunas métricas destacables de CloudWatch son:
  - CPUUtilization
  - NetworkIn/NetworkOut
  - DiskReadOps/DiskWriteOps
  - Latency
- Las métricas tienen **marcas de tiempo**
- Puedes crear **dashboards de CloudWatch** con las métricas procedentes de CloudWatch



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Visión general de Amazon CloudWatch



© Joan Amengual

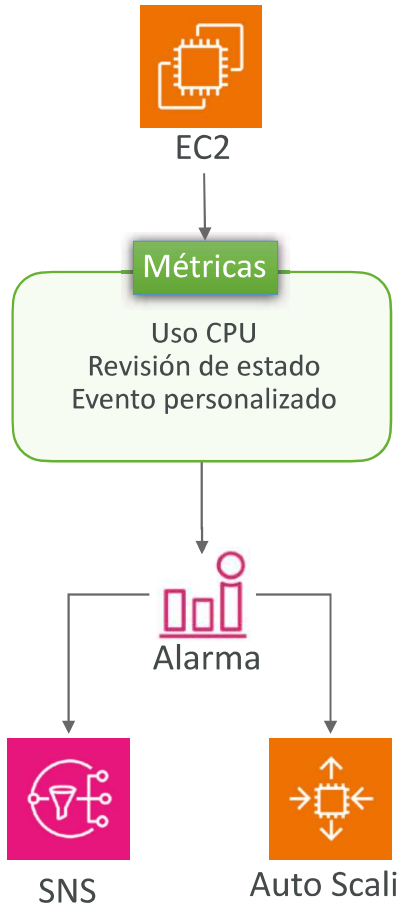
Todos los derechos reservados © Joan Amengual www.blockstellart.com

Monitorización detallada de EC2

- Las métricas de las instancias EC2 tienen métricas "cada 5 minutos"
- Con la monitorización detallada (por un coste), obtienes datos "cada 1 minuto"
- ¡Utiliza la monitorización detallada si quieres escalar más rápido tu ASG!

- La capa gratuita de AWS nos permite disponer de 10 métricas de monitorización detallada

- Nota: el uso de memoria de EC2 no se envía por defecto (debe enviarse desde dentro de la instancia como métrica personalizada)

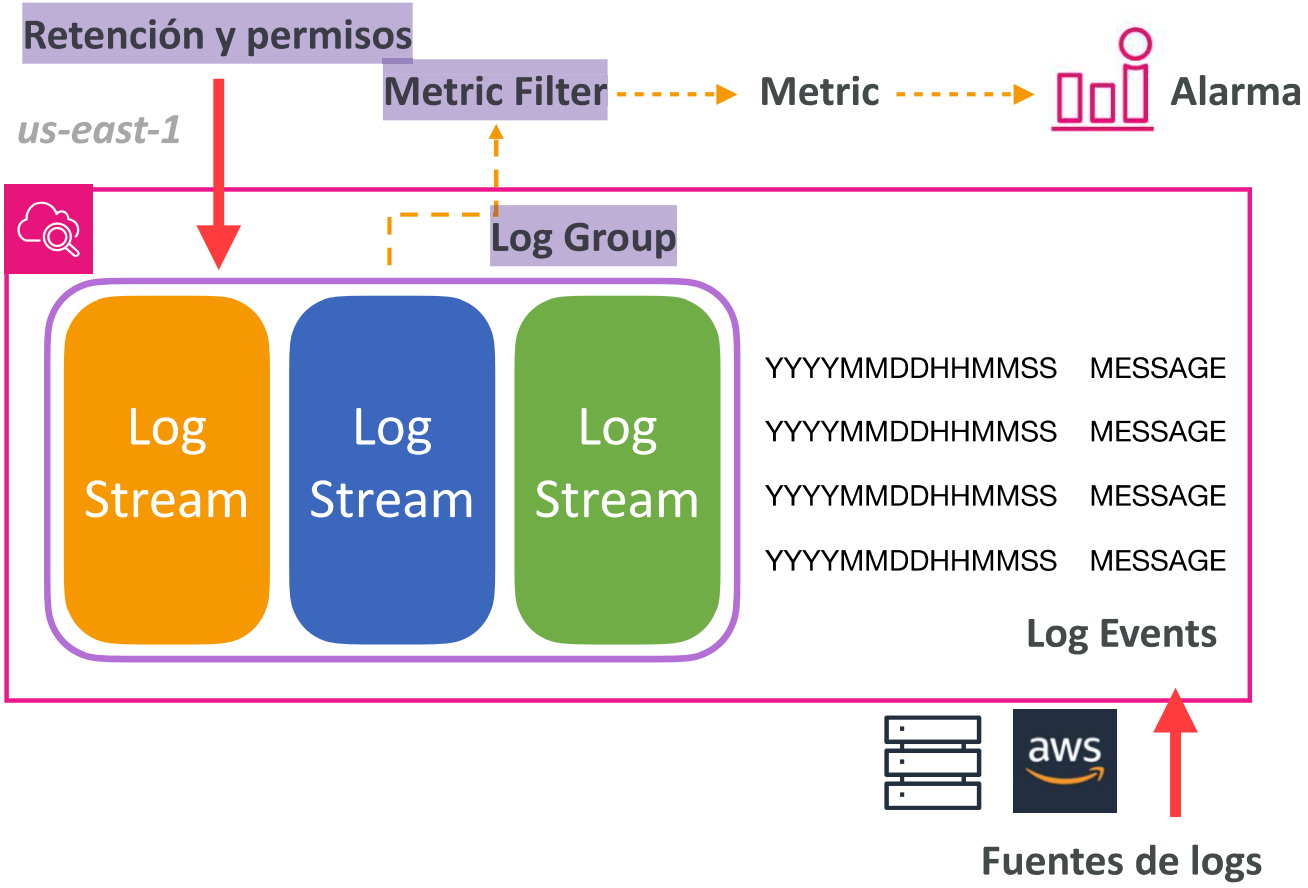


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon CloudWatch Logs

- CloudWatch Logs permite **centralizar los logs de aplicaciones y servicios en AWS**, facilitando su gestión y análisis en un único lugar.
- **Monitorización de logs en tiempo real**, permitiendo identificar y responder rápidamente a los problemas
- CloudWatch Logs puede recoger logs de aplicaciones, contenedores, funciones, instancias, consultas DNS…
- Retención de logs de CloudWatch ajustable



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Registros de CloudWatch (Logs)

- Puede definir políticas de expiración de logs (nunca expiran, 30 días, etc..)
- **CloudWatch Logs puede enviar logs a destinos como:**
  - Amazon S3 (exportaciones)
  - Flujos de datos de Kinesis
  - Kinesis Data Firehose
  - AWS Lambda
  - Etc…



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

CloudWatch Logs - Fuentes

- SDK, agente de CloudWatch Logs, agente unificado de CloudWatch



- Elastic Beanstalk: recogida de logs desde la aplicación



- ECS: recopilación desde contenedores



- AWS Lambda: recopilación de registros de funciones



- Registros de flujo de VPC



- API Gateway



- CloudTrail basado en filtro



- Route 53 (DNS)



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Filtro de métricas e información de CloudWatch Logs

- CloudWatch Logs puede utilizar **expresiones de filtro**
  - Por ejemplo, encontrar una IP específica dentro de un archivo de logs
  - O contar ocurrencias de "ERROR" en los logs
- Los filtros de métricas pueden utilizarse para activar alarmas de CloudWatch

- CloudWatch Logs Insights puede utilizarse para consultar registros y añadir consultas a CloudWatch Dashboards

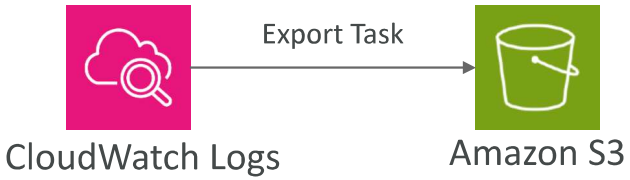


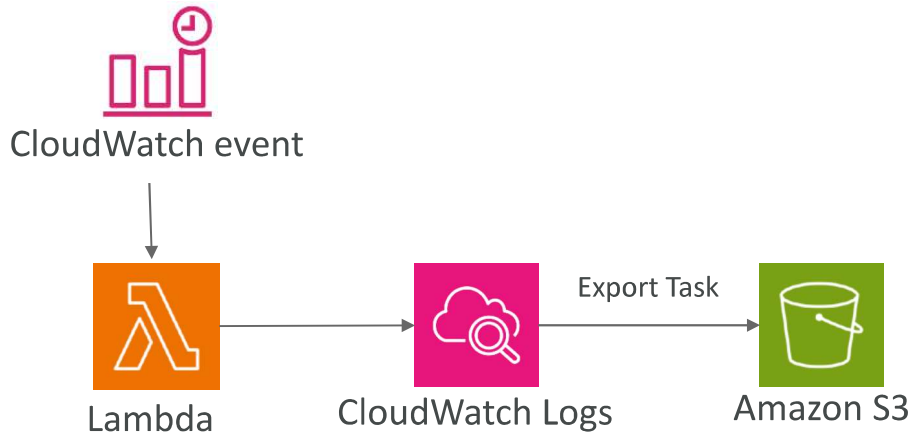
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

CloudWatch Logs - Exportación a S3

- Los datos de registro pueden tardar **hasta 12 horas** en estar disponibles para la exportación
- La llamada a la API es **CreateExportTask**
- No en tiempo casi real ni en tiempo real... utiliza suscripciones a registros en su lugar

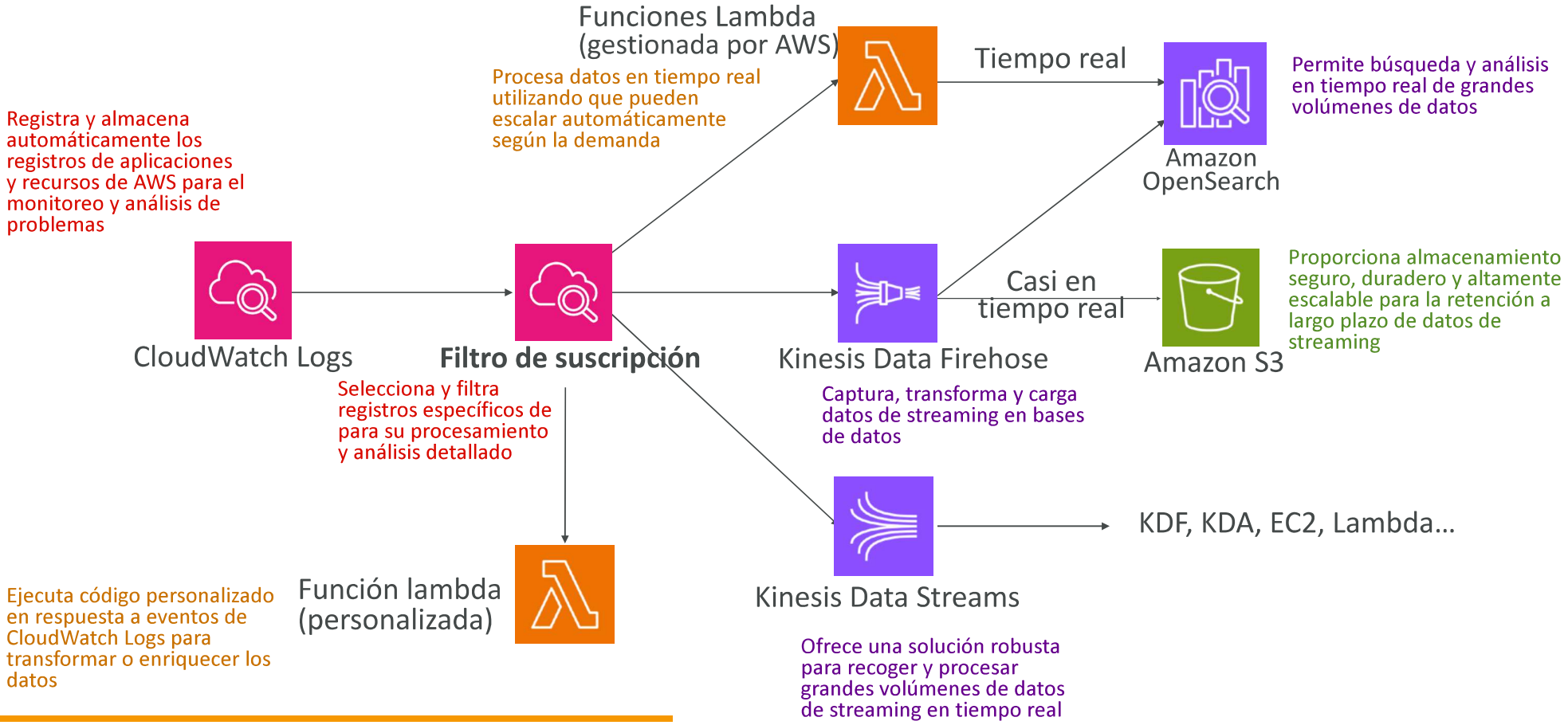




© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

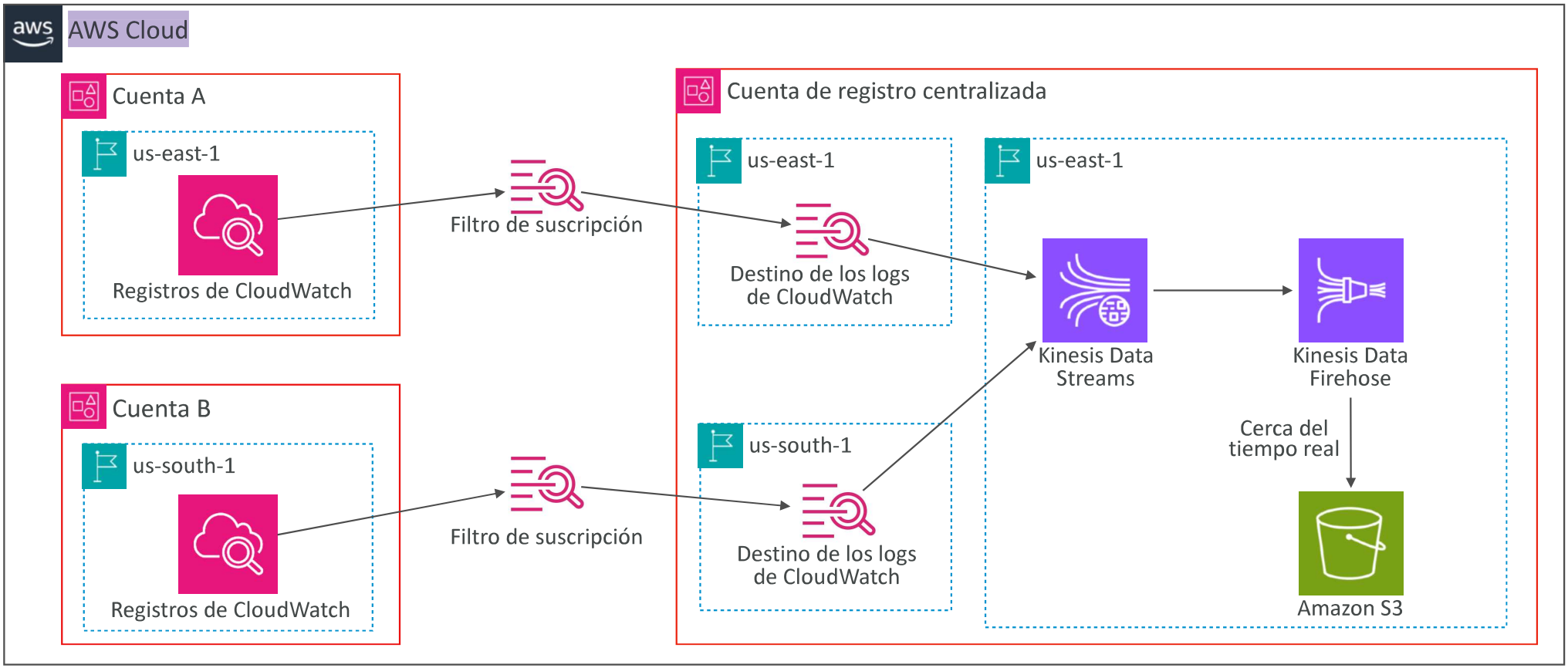
Suscripciones a CloudWatch Logs



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Agregación de CloudWatch Logs Multi-Cuenta y Multi-Región

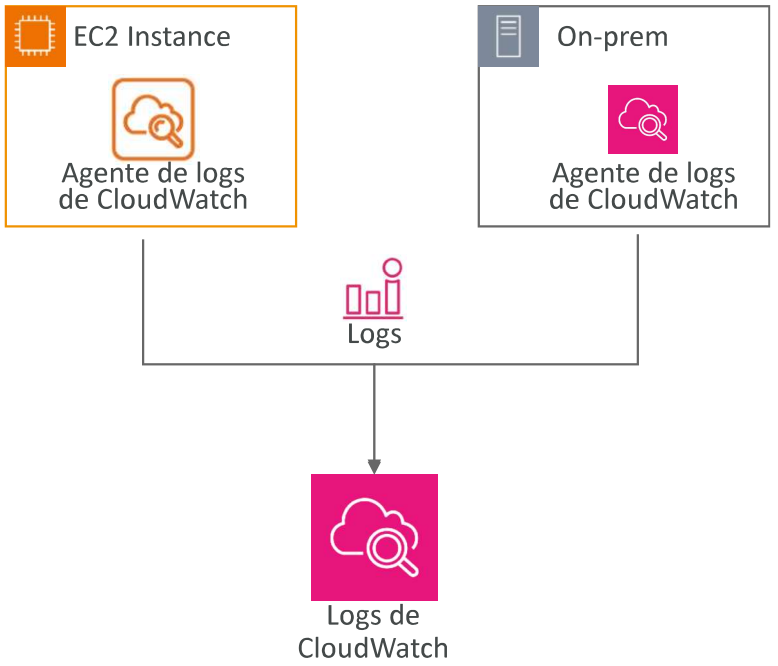


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

CloudWatch Logs para EC2

- Por defecto, los logs de tu máquina EC2 no irán a CloudWatch
- Necesitas ejecutar un **agente de CloudWatch en EC2** para enviar los logs que desees.
- Asegúrate de que los permisos IAM son correctos
- El agente de logs de CloudWatch también puede configurarse en local



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Agente CloudWatch Logs y Agente Unificado

- Para servidores virtuales (instancias EC2, servidores locales...)
- **Agente de logs de CloudWatch**
  - Versión antigua del agente
  - Sólo puede enviar a CloudWatch Logs

- **Agente Unificado CloudWatch**
  - Recoge métricas adicionales a nivel de sistema, como RAM, procesos, etc...
  - Recoge logs para enviarlos a CloudWatch Logs



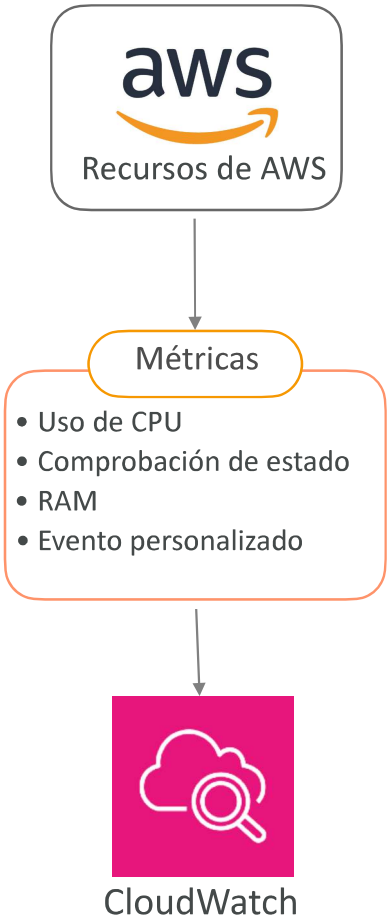
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Agente Unificado CloudWatch - Métricas

- Recopilados directamente en tu servidor Linux / instancia EC2

- **CPU** (activa, huésped, inactiva, sistema, usuario)
- **Métricas de disco** (libre, usado, total), IO de disco (escrituras, lecturas, bytes, IOPS)
- **RAM** (gratis, inactiva, usada, total, en caché)
- **Netstat** (número de conexiones TCP y UDP, paquetes netos, bytes)
- **Procesos** (totales, muertos, bloqueados, inactivos, en ejecución, en reposo)
- **Espacio de intercambio** (gratis, usado, % usado)



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon CloudWatch Alarms



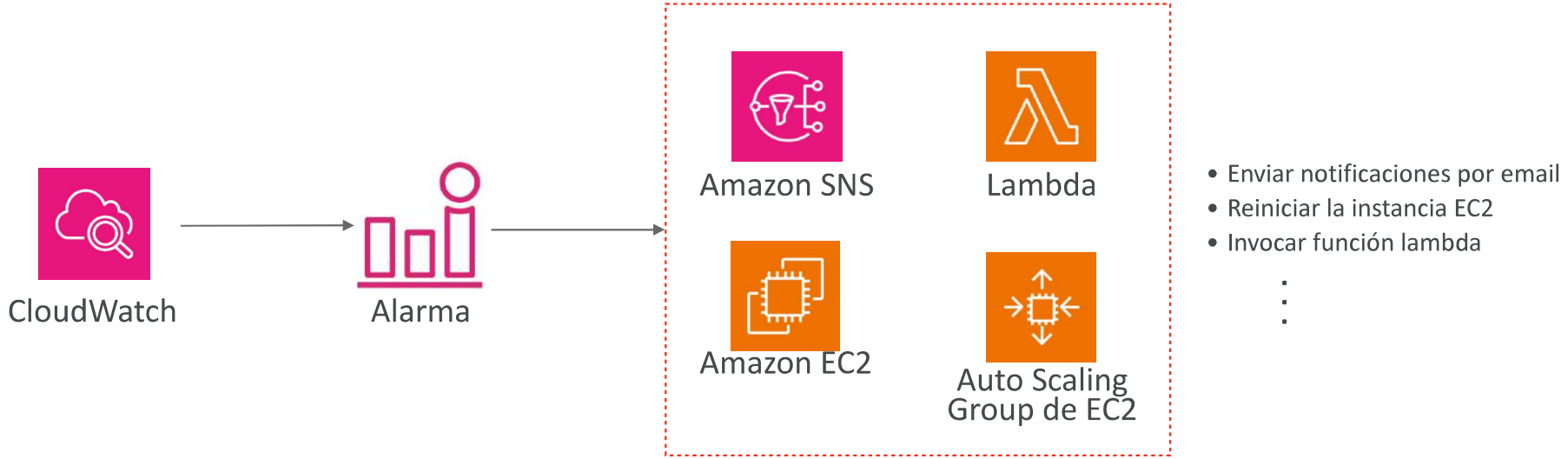
- Configura alertas para **detectar anomalías en el comportamiento de tus recursos AWS**, permitiendo una respuesta rápida
- Automatiza respuestas para resolver problemas o escalar recursos en función del estado de las alarmas
- Establece **umbrales** específicos para las métricas monitorizadas, adecuados a las necesidades y patrones de uso de tus aplicaciones (ej: 70%)
- Ejemplo: crear **una alarma de facturación** en la métrica de facturación de CloudWatch
- Estados de la alarma:
  - OK, INSUFFICIENT_DATA, ALARM

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Objetivos de alarma de CloudWatch

- Detener, terminar, reiniciar o recuperar una instancia EC2
- Activar la acción de autoescalado
- Enviar notificación a SNS (desde donde puedes hacer prácticamente cualquier cosa)



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon EventBridge



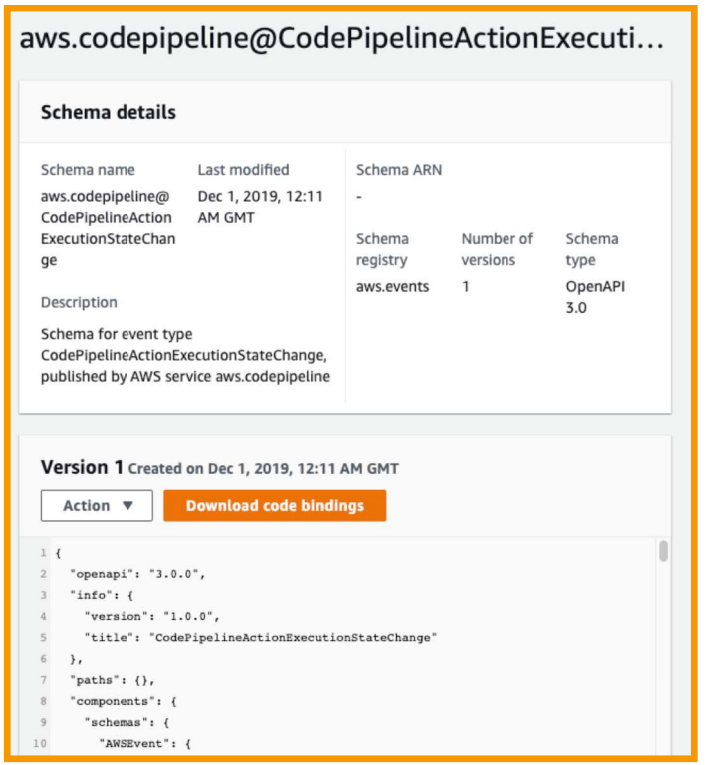
- **EventBridge facilita la conexión de aplicaciones con datos provenientes de diversas fuentes de AWS y aplicaciones de software como servicio (SaaS)** externas, permitiendo una integración sin problemas y la automatización de flujos de trabajo
- Los usuarios pueden definir y enviar sus propios eventos a través de buses de eventos de EventBridge, lo que permite crear aplicaciones altamente personalizadas y reactivas basadas en eventos específicos de negocio.
  - **Bus de eventos predeterminado** - generado por servicios de AWS (CloudWatch Events)
  - **Bus de eventos de socios** - recibe eventos de servicios SaaS o aplicaciones (Zendesk, DataDog, Segment, Auth0...)
  - **Buses de eventos personalizados** - para tus propias aplicaciones

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon EventBridge - Registro de esquemas

- EventBridge puede analizar los eventos de tu bus e inferir el **esquema**
  - **Los esquemas son estructuras de datos que definen la forma y el formato de los eventos*
- El **registro de esquemas** te permite generar código para tu aplicación, que sabrá de antemano cómo se estructuran los datos en el bus de eventos
- El esquema puede versionarse



© Joan Amengual

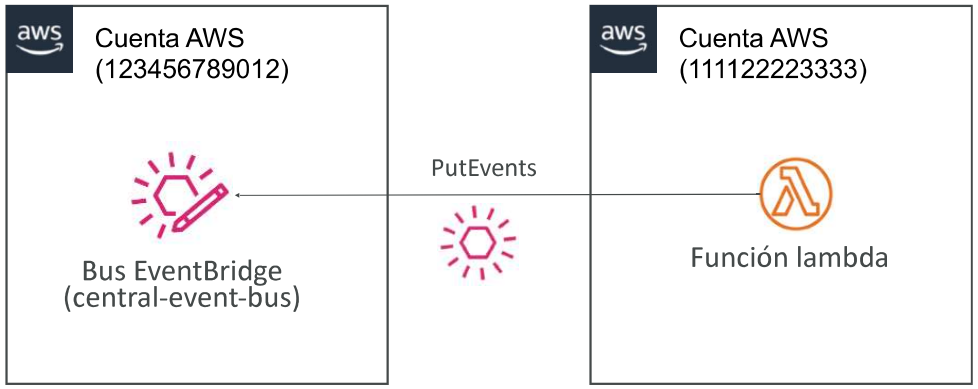
Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon EventBridge - Política basada en recursos

- Gestionar permisos para un bus de eventos específico
- Ejemplo: permitir/denegar eventos de otra cuenta AWS o región AWS
- Caso práctico: agregar todos los eventos de tu organización de AWS en una única cuenta AWS o región de AWS

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
          "Effect": "Allow"
          "Action": "events:PutEvents"
          "Principal": {"AWS": "1112223333" }
          "Resource": "arn:aws:events:us-
east-1_123456789012:event-bus/central-event-bus"
      }
   ]
}
```

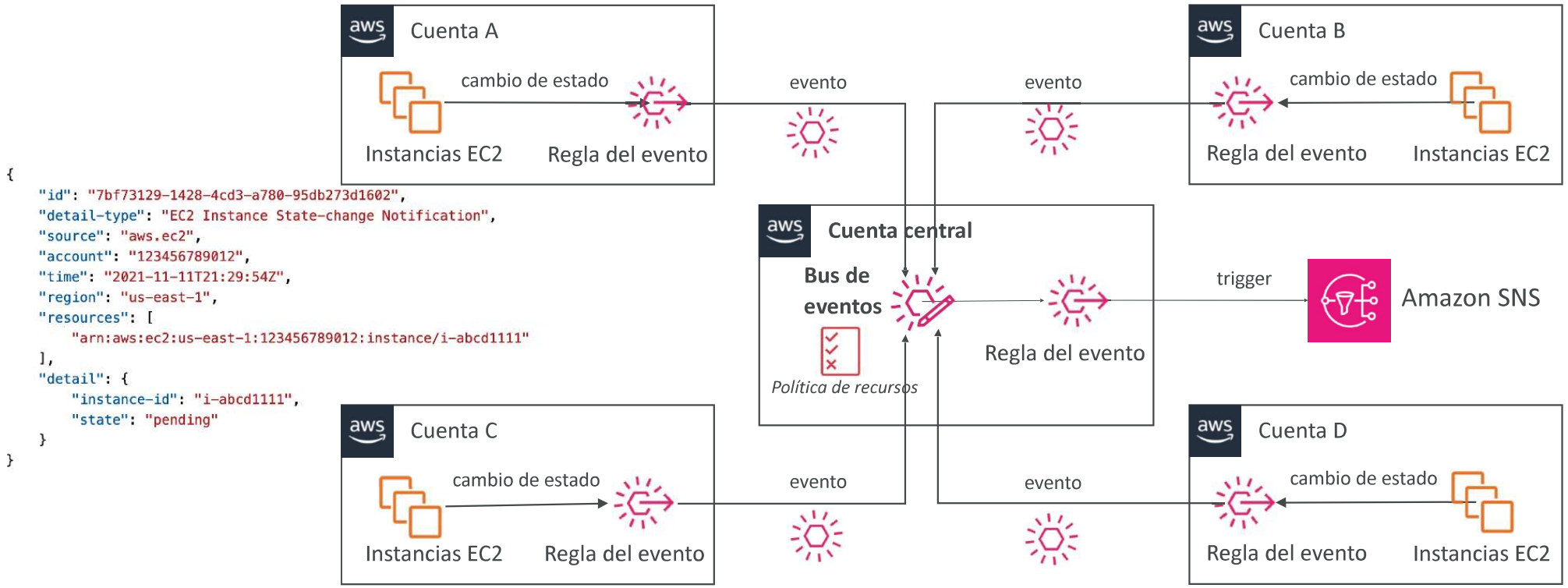
Permitir **eventos** desde otra cuenta AWS



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

EventBridge - Agregación multicuenta




```
{
    "id": "7bf73129-1428-4cd3-a780-95db273d1602",
    "detail-type": "EC2 Instance State-change Notification",
    "source": "aws.ec2",
    "account": "123456789012",
    "time": "2021-11-11T21:29:54Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1111"
    ],
    "detail": {
        "instance-id": "i-abcd1111",
        "state": "pending"
    }
}
```

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Caso práctico real de Amazon EventBridge

- **Caso Práctico: Monitoreo de salud en tiempo real**
  - Una empresa de tecnología utiliza AWS para hospedar su infraestructura de aplicaciones. Quieren monitorear la salud de sus servicios en tiempo real para responder rápidamente a cualquier incidente
- **Implementación con EventBridge**:
  - **Detección de incidentes**
    - Los servicios como Amazon EC2 y Amazon RDS envían automáticamente alertas de estado a EventBridge cuando detectan problemas como sobrecarga de CPU o fallos de conexión
  - **Automatización de respuestas**
    - EventBridge recibe estas alertas y dispara funciones AWS Lambda preconfiguradas para realizar acciones inmediatas, como escalar recursos o reiniciar instancias
  - **Notificación a equipos**
    - Además, EventBridge puede enviar notificaciones a un canal de Slack o Microsoft Teams para informar al equipo de TI sobre el incidente y las acciones tomadas

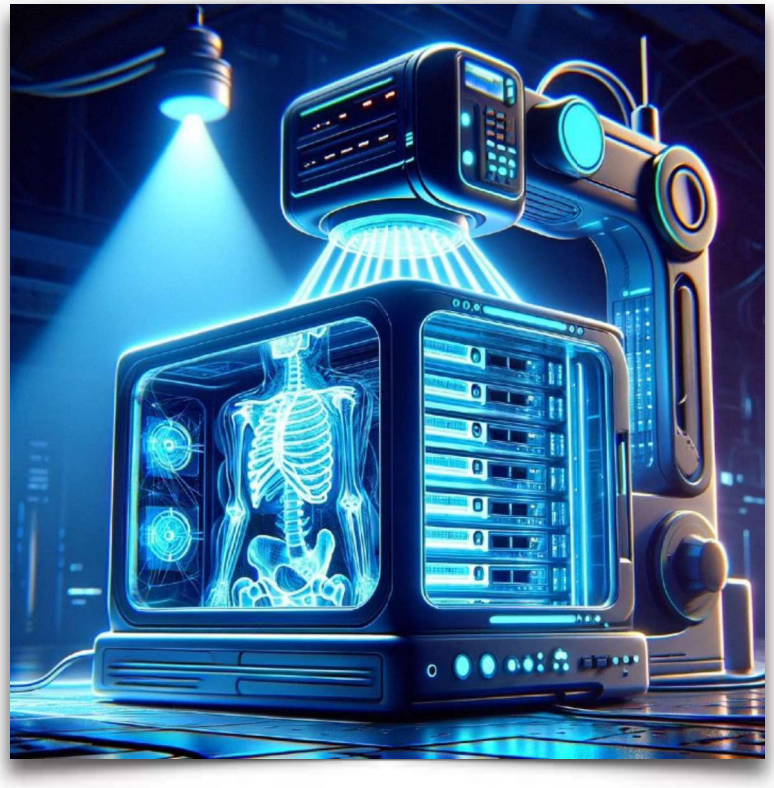
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

AWS X-Ray

- Depurar en producción, a la antigua:
  - Test localmente
  - Añade logs por todas partes
  - Vuelve a desplegar en producción
- Los formatos de logs difieren entre las aplicaciones que utilizan CloudWatch y el análisis es difícil
- Depuración: monolito "fácil", servicios distribuidos "difícil"
- No hay vistas comunes de toda tu arquitectura

- Entra... ¡AWS X-Ray!

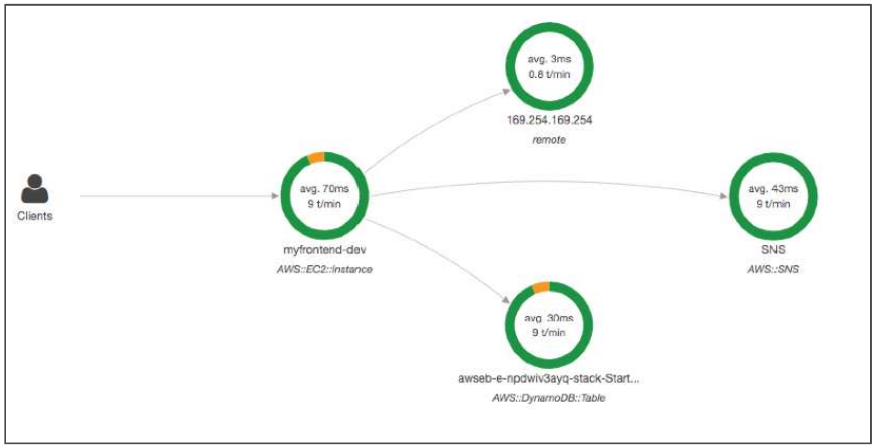


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

La magia de X-Ray

- El servicio X-Ray recoge datos de todos los diferentes servicios
- El mapa de servicios se calcula a partir de todos los segmentos y trazas
- X-Ray es gráfico, por lo que incluso personas no técnicas pueden ayudar a solucionar problemas

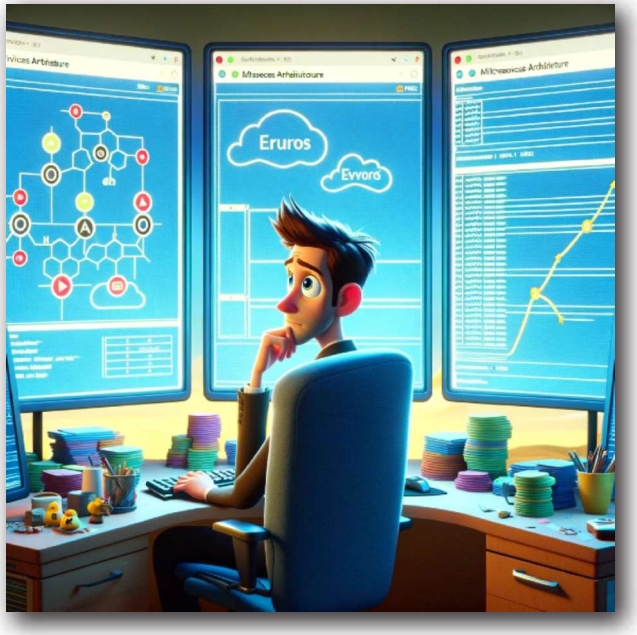


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Ventajas de AWS X-Ray

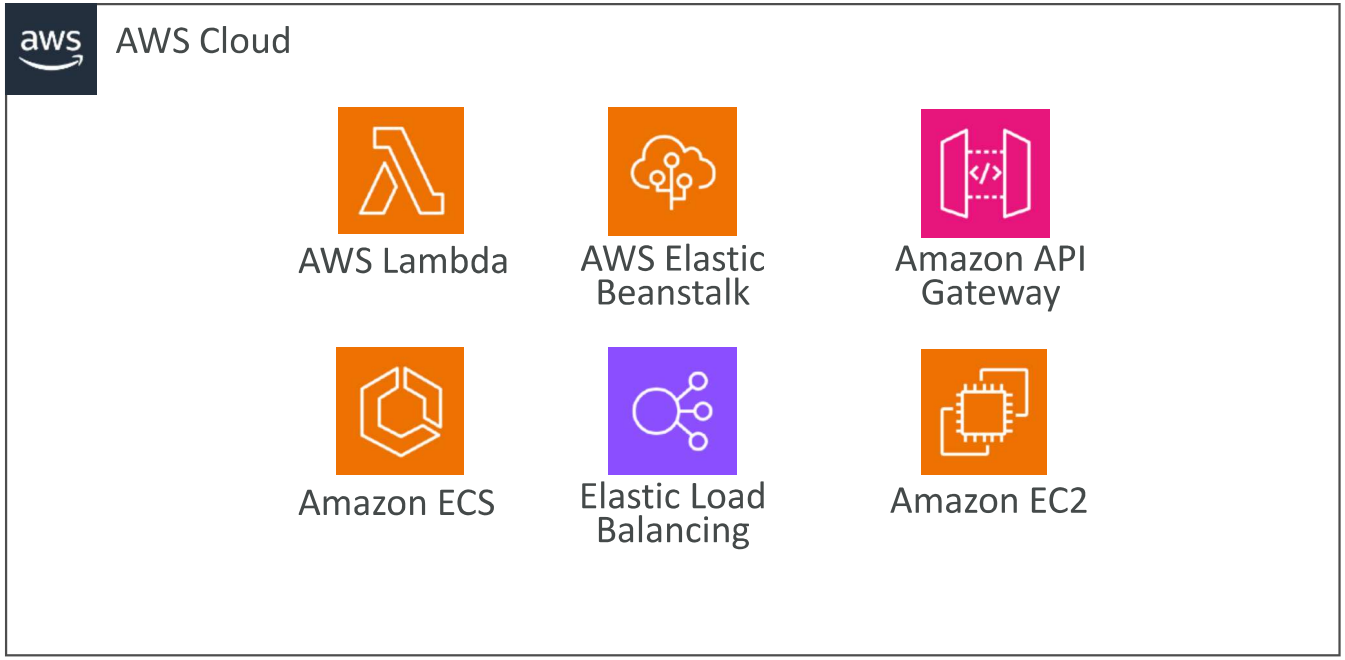
- Resolución de problemas de rendimiento (cuellos de botella)
- Comprender las dependencias en una arquitectura de microservicios
- Localizar problemas de servicio
- Revisar el comportamiento de las peticiones
- Encontrar errores y excepciones
- ¿Cumplimos el SLA de tiempo?
- ¿Dónde estoy estrangulado?
- Identifica a los usuarios afectados



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Compatibilidad con X-Ray



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

AWS X-Ray aprovecha el rastreo

- El rastreo es una forma integral de seguir una "petición"
- Cada componente que se ocupa de la petición añade su propio "seguimiento"
- El seguimiento se compone de segmentos (+ subsegmentos)
- Se pueden añadir anotaciones a las trazas para proporcionar información adicional
- Posibilidad de rastrear:
  - Cada petición
  - Muestra de petición (en %, por ejemplo, o en tasa por minuto)
- Seguridad X-Ray:
  - IAM para autorización
  - KMS para cifrado en reposo

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

AWS X-Ray
¿Cómo activarlo?

I) Tu código (Java, Python, Go, Node.js, .NET) debe importar el SDK de AWS X-Ray

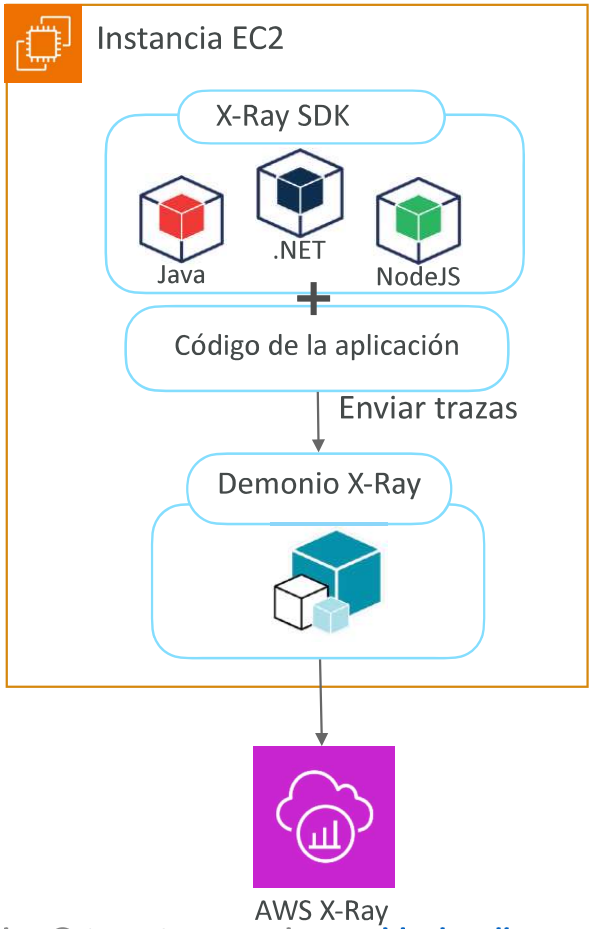
- Se necesita muy poca modificación del código
- A continuación, el SDK de la aplicación capturará
  - Llamadas a servicios de AWS
  - Peticiones HTTP / HTTPS
  - Llamadas a bases de datos (MySQL, PostgreSQL, DynamoDB)
  - Llamadas a colas (SQS)

2) Instala el demonio (daemon) X-Ray o activa la integración de X-Ray en AWS

El demonio X-Ray funciona como un interceptor de paquetes UDP de bajo nivel (Linux / Windows / Mac...)

AWS Lambda / otros servicios de AWS ya ejecutan el demonio X-Ray por ti

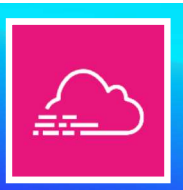
Cada aplicación debe tener los derechos IAM para escribir datos en X-Ray



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

AWS CloudTrail

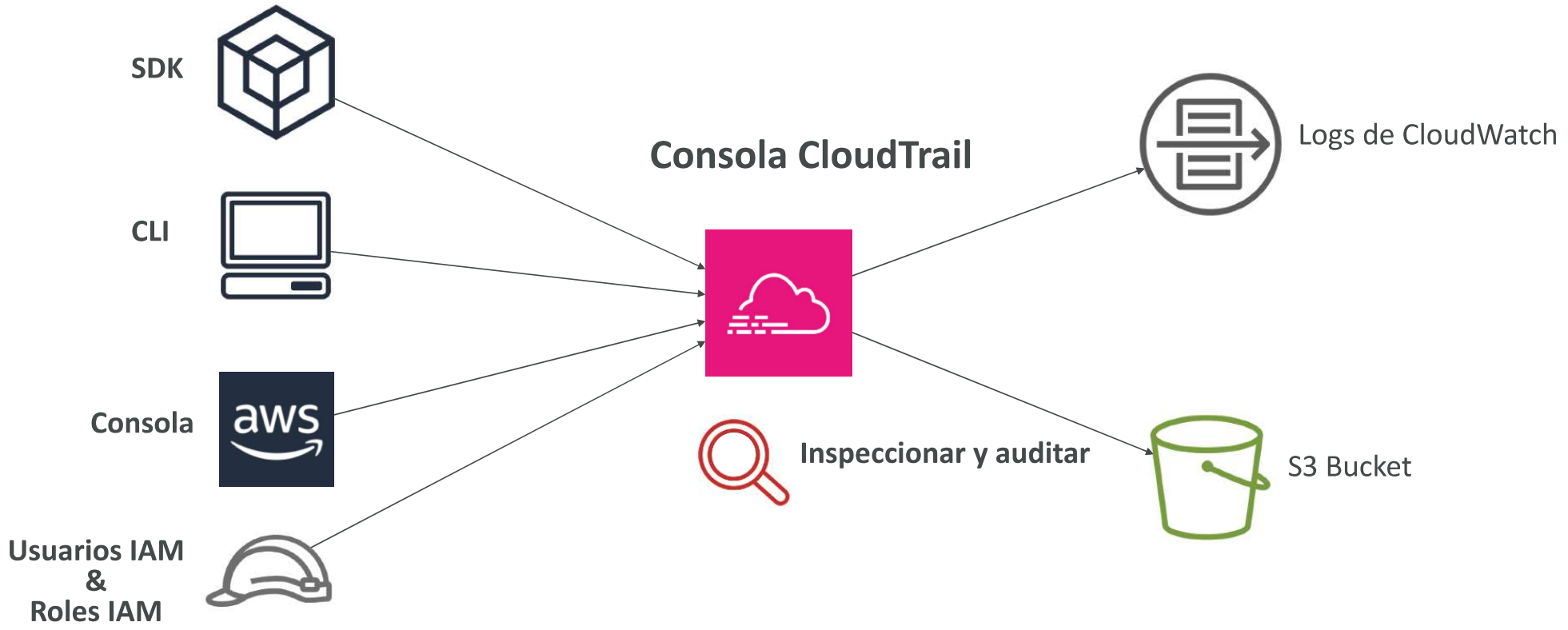


- **Proporciona gobernanza, normativa y auditoría para tu cuenta de AWS**
- CloudTrail está activado por defecto
- Obtén **un historial de eventos / llamadas a la API realizadas dentro de tu Cuenta de AWS** por:
  - Consola
  - SDK
  - CLI
  - Servicios de AWS
- Puedes poner logs de CloudTrail en CloudWatch Logs o S3
- **Un rastro (trail) puede aplicarse a todas las regiones (por defecto) o a una sola región**
- Si se elimina un recurso en AWS, ¡investiga primero en CloudTrail!

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Diagrama de CloudTrail

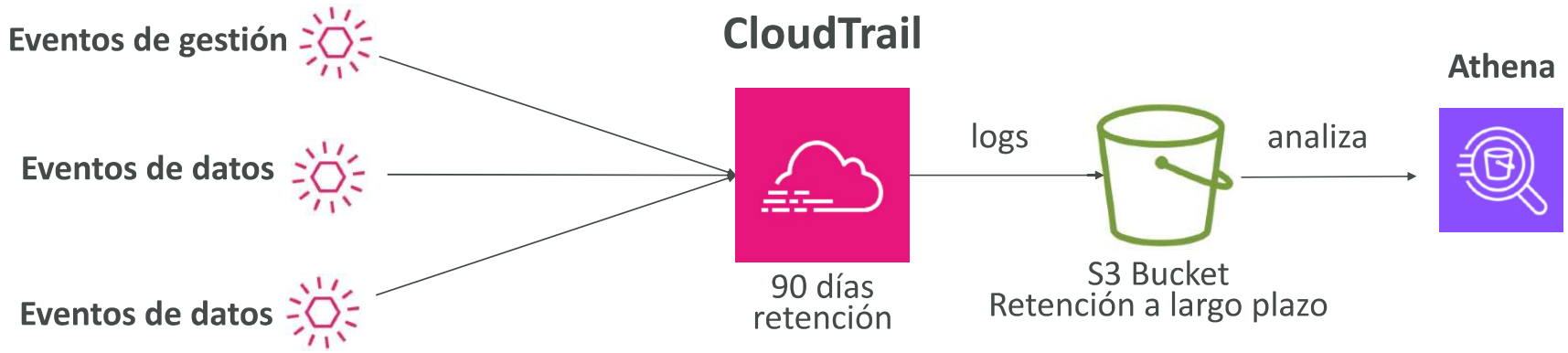


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Retención de Eventos CloudTrail

- Los eventos se almacenan durante 90 días en CloudTrail
- Para conservar los eventos más allá de este periodo, regístralos en S3 y utiliza Athena



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

CloudTrail vs CloudWatch vs X-Ray

- CloudTrail:
  - Auditoría de llamadas a la API realizadas por usuarios / servicios / consola de AWS
  - Útil para detectar llamadas no autorizadas o la causa raíz de los cambios
- CloudWatch:
  - CloudWatch Metrics a lo largo del tiempo para monitorización
  - CloudWatch Logs para almacenar logs de aplicaciones
  - CloudWatch Alarms para enviar notificaciones en caso de métricas inesperadas
- X-Ray:
  - Análisis de trazas automatizado y visualización de mapas de servicios
  - Análisis de latencia, errores y fallos
  - Seguimiento de peticiones en sistemas distribuidos

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com



S3 (Simple Storage Service)

www.blockstellart.com

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Visión general de Amazon S3



- Amazon S3 es una **plataforma global de almacenamiento** - **regional**
- Servicio **público**, con capacidad de almacenar infinita información
  - Películas, Audio, Fotos, Texto, Datasets muy grandes, …
- S3 se define como uno de los principales bloques de construcción de AWS

- Muchos sitios web utilizan Amazon S3 como columna vertebral
- Muchos servicios de AWS utilizan Amazon S3 como una integración también

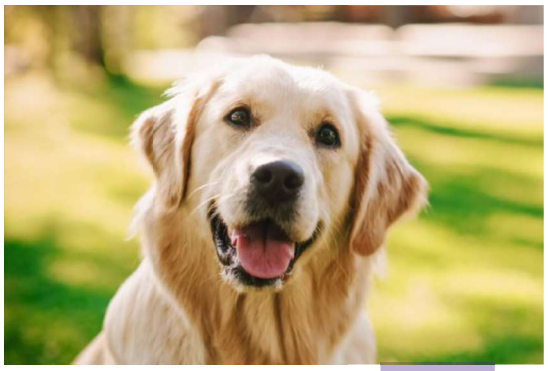
- Componentes a destacar de S3: **Buckets** y **Objetos**

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Objetos de Amazon S3





Cero Bytes - 5 TB

Key
ID de versión
Valor
Metadatos
Subrecursos
Información de control de acceso

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Visión general de Amazon S3 - Buckets

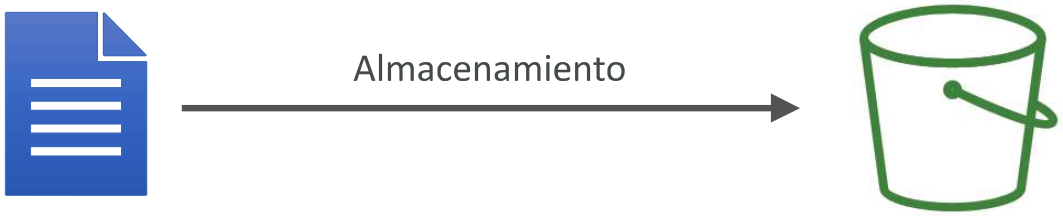
- Amazon S3 permite almacenar objetos (archivos) en "buckets" (directorios)
- Los buckets deben tener un **nombre único a nivel global (en todas las regiones, todas las cuentas)**
- S3 parece un servicio global pero los buckets se crean en una región



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon S3 - Objetos (1/2)



Objeto: mi-archivo.txt **Bucket: mi-bucket**

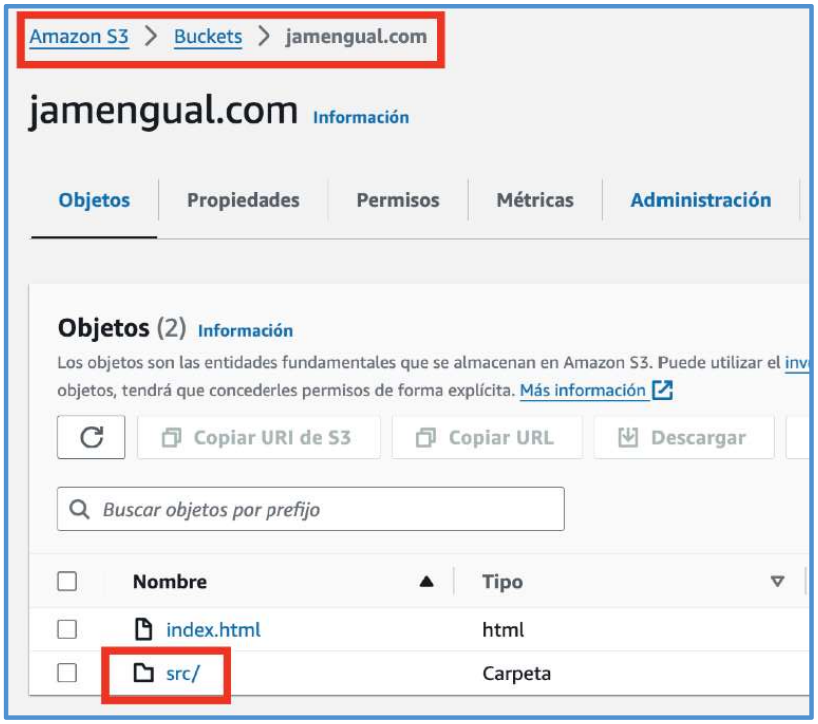
- Los objetos (archivos) tienen una clave (key)
- La **clave** es la ruta **COMPLETA**:
  - s3://mi-bucket/**mi-archivo.txt**
  - s3://mi-bucket/**mi_carpeta1/otra_carpeta/mi_archivo.txt**

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon S3 - Objetos (2/2)

- **No existe el concepto de "directorios"** dentro de los buckets (aunque la interfaz de usuario te hará pensar lo contrario)
  - Sólo claves con nombres muy largos que contienen barras inclinadas (“/“)
- **El tamaño máximo del objeto es de 5TB (5000GB)**
- Si se suben más de 5GB, se debe usar "carga de varias partes”



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Ejemplo - Claves de objetos



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Casos de uso de Amazon S3

- Copia de seguridad y almacenamiento
- Recuperación de desastres
- Gestión de datos para aplicaciones IoT
- Almacenamiento en el Cloud híbrido
- Alojamiento de aplicaciones
- Alojamiento de medios
- Uso para análisis big data
- Entrega de software
- Sitio web estático





J.P.Morgan

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Políticas de bucket S3

- **Resource**: el bucket, el objeto, el punto de acceso o el trabajo de Amazon S3 al que se aplica la política
- **Effect**: el efecto obtenido cuando el usuario solicita la acción específica, que puede ser *permitir* o *denegar*
- **Actions**: conjunto de operaciones para cada recurso
- **Entidad principal**: la cuenta o el usuario con permiso de acceso a las acciones
- **Condition**: condiciones para cuando surte efecto una política



```
{
    "Version": "2012-10-17",
    "Id": "ExamplePolicy01",
    "Statement": [
        {
            "Sid": "ExampleStatement01",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Dave"
            },
            "Action": [
                "s3:GetObject",
                "s3:GetBucketLocation",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::awsexamplebucket1/*",
                "arn:aws:s3:::awsexamplebucket1"
            ]
        }
    ]
}
```

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Bloquear el acceso público al almacenamiento de Amazon S3

- De forma predeterminada, los buckets, puntos de acceso y objetos nuevos **no permiten el acceso público**
- Estos ajustes se crearon para evitar la filtración de datos de la empresa
- Sin embargo, los usuarios pueden modificar las políticas de bucket, las políticas de punto de acceso o los permisos de objeto para **permitir el acceso público**



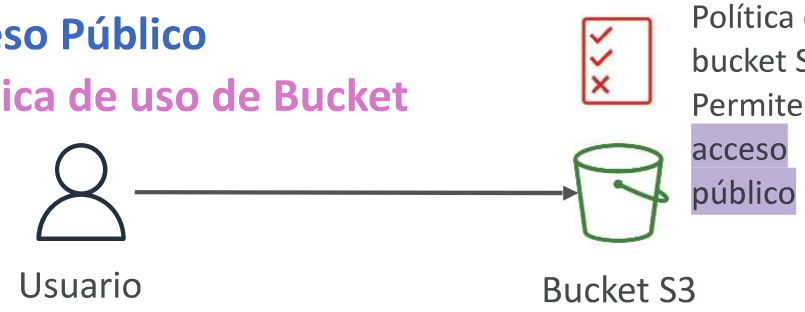
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Ejemplos de seguridad en Amazon S3

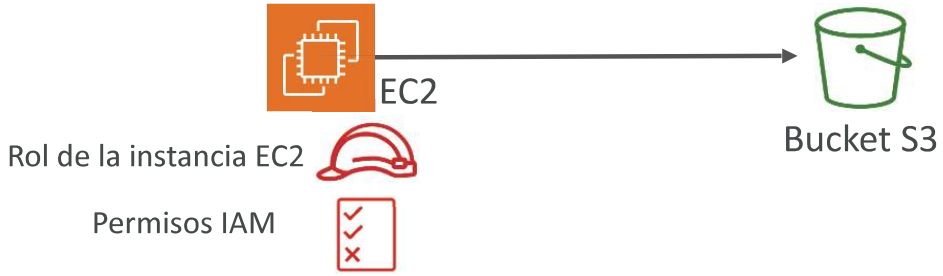
Acceso Público

Política de uso de Bucket



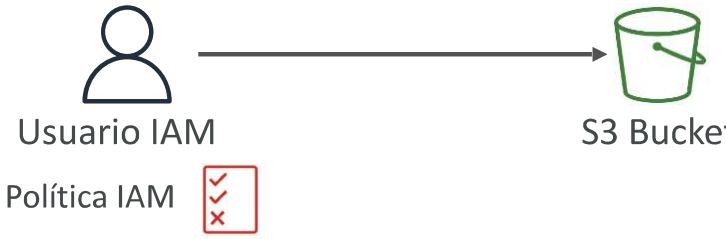
Acceso a instancias EC2

Usar roles IAM



Acceso del usuario a S3

Permisos IAM



Acceso entre cuentas

Política de Bucket

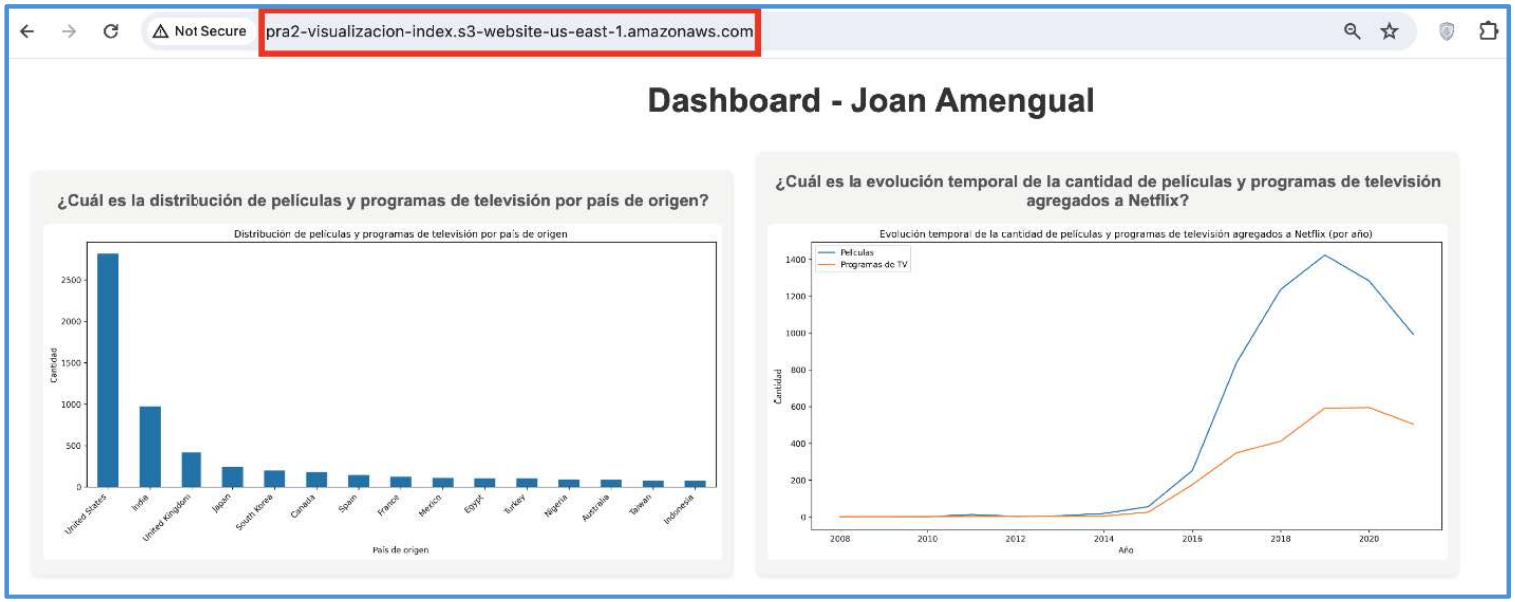


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Alojamiento de sitios web estáticos en S3

- **S3 puede alojar sitios web estáticos y tenerlos accesibles en Internet**
- La URL del sitio web cambiará según la región

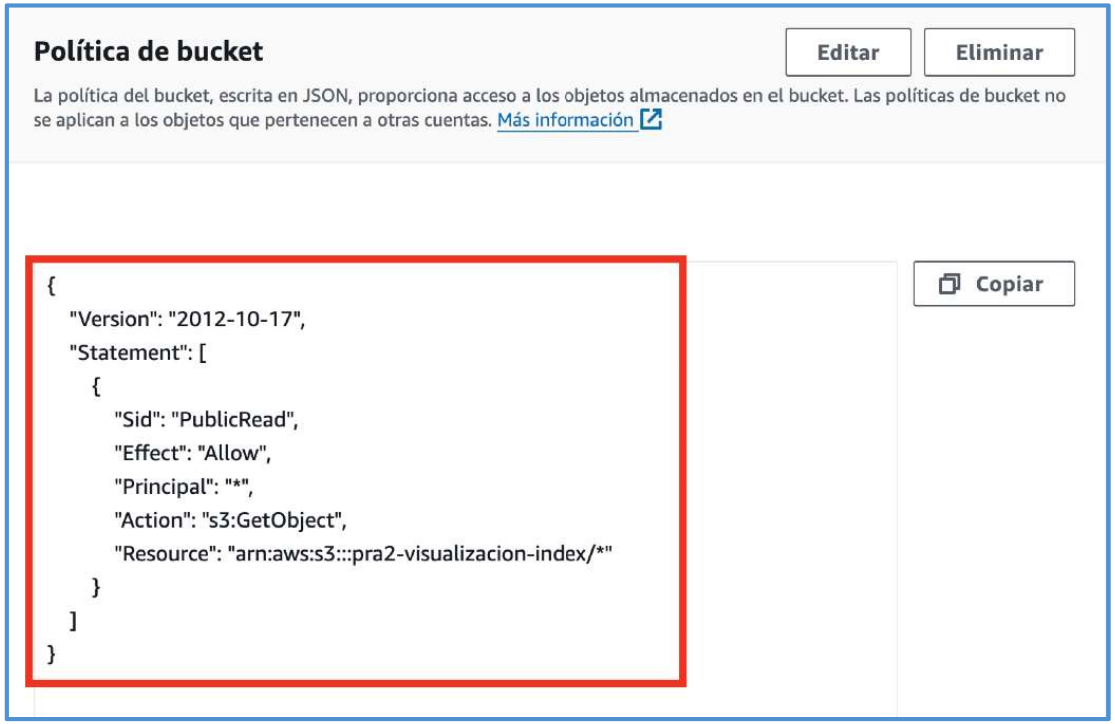


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

¡No te olvides de la política de bucket!

Si obtienes un **error 403 Prohibido**, debes asegurarte de que la política del bucket permite las lecturas públicas

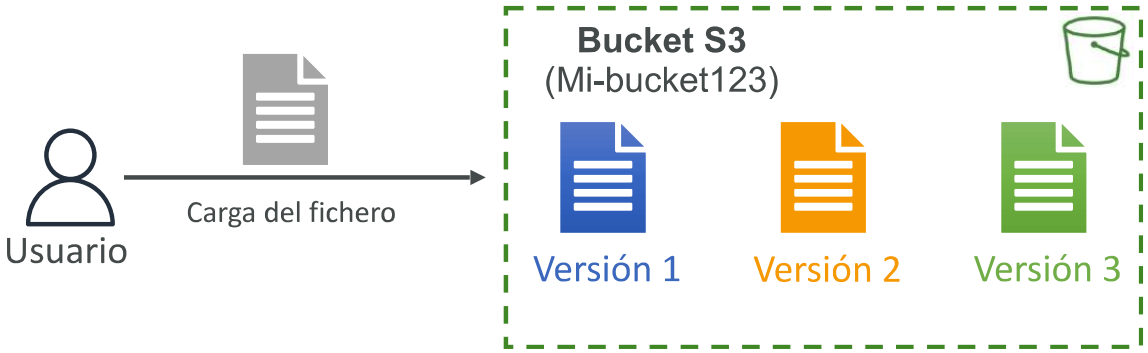


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon S3 - Versionado

- Amazon S3 permite versionar nuestros archivos
- Esta característica se activa a nivel de bucket



- Cualquier archivo que no esté versionado antes de activar el versionado tendrá la versión "nula"
- Suspender el versionado no elimina las versiones anteriores

¿Por qué es una buena práctica versionar tus buckets?

- Protege contra los borrados involuntarios
- Facilidad para volver a la versión anterior

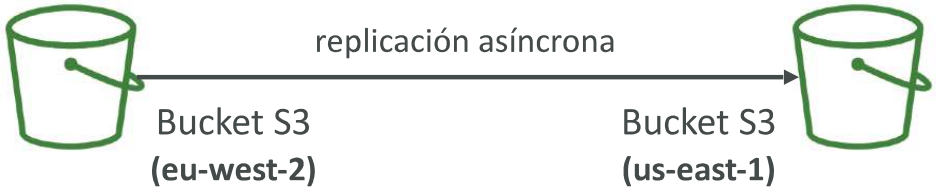


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Replicación de información en buckets S3

- **Podemos replicar la información en diferentes buckets S3**
- ⚠ **IMPORTANTE**: Debes activar el control de versiones en los buckets de origen y destino
- Los buckets pueden estar en diferentes cuentas de AWS
- La copia es **asíncrona**
- Debes dar los **permisos IAM** adecuados a S3
- Después de activar la replicación, **sólo se replican los objetos nuevos**
- Existen diferentes tipos de replicación:
  - **Replicación entre regiones (CRR)**: normativa, acceso de menor latencia, replicación entre cuentas
  - **Replicación en la misma región (SRR)**: agregación de logs, replicación en vivo entre cuentas de producción y de prueba



No hay encadenamiento de la replicación

- Si el bucket 1 tiene replicación en el bucket 2, que tiene replicación en el bucket 3
- Entonces los objetos creados en el bucket 1 **no** se replican en el bucket 3

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Clases de almacenamiento de Amazon S3

- Amazon S3 ofrece una **variedad de clases de almacenamiento** entre las cuales puedes elegir en función de:
  - Requisitos de rendimiento
  - Acceso a los datos
  - Resiliencia
  - Costos de sus cargas de trabajo

- Las clases de almacenamiento de S3 se crearon específicamente para brindar el **menor costo posible de almacenamiento para los diferentes patrones de acceso**

- Se puede pasar de una clase a otra **manualmente** o utilizando las **configuraciones del ciclo de vida de S3**

S3 Standard

S3 Express One Zone

S3 Standard-IA

S3 One Zone-IA

S3 Glacier Instant Retrieval

S3 Glacier Flexible Retrieval

S3 Glacier Deep Archive

S3 Intelligent Tiering

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Clases de almacenamiento (1/3)



S3 Standard

- Datos activos, frecuentemente accedidos
- Acceso en milisegundos
- > 3 AZ



S3 Express One Zone

- Acceso ultrarrápido a datos
- Selección precisa de AZ
- 1 AZ
- Para cargas de trabajo intensivas en datos



S3 Standard-IA

- Datos accedidos infrecuentemente
- Acceso en milisegundos
- > 3 AZ
- Tarifa por recuperación por GB
- Duración mínima de almacenamiento
- Tamaño mínimo de objeto
- Mín. duración de almacenamiento: 30 días



S3 One Zone-IA

- Datos recreables, menos accedidos
- Acceso en milisegundos
- 1 AZ
- Tarifa por recuperación por GB
- Duración mínima de almacenamiento
- Tamaño mínimo de objeto
- Mín. duración de almacenamiento: 30 días

**Precios de Norte de Virginia*

0,023 USD por GB	0,16 USD por GB	0,0125 USD por GB	0,01 USD por GB

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Clases de almacenamiento (2/3)



S3 Glacier Instant Retrieval

- Almacenamiento de archivos de bajo costo
- Recuperación en milisegundos para datos raramente accesibles
- Mín. duración de almacenamiento: 90 días

0,004 USD por GB



S3 Glacier Flexible Retrieval

- Recuperación de datos asincrónica
- > 3 AZ
- Tarifa por recuperación por GB
- Duración mínima de almacenamiento
- Mín. duración de almacenamiento: 90 días

0,0036 USD por GB



S3 Glacier Deep Archive

- Datos archivados a largo plazo
- > 3 AZ
- Tarifa por recuperación por GB
- Duración mínima de almacenamiento
- Mín. duración de almacenamiento: 180 días

0,00099 USD por GB

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Clases de almacenamiento (3/3)

Reduce costos moviendo automáticamente datos a niveles de precio más bajo según su uso.

- **Costos de migración**: Pequeño **cargo mensual por monitoreo**; no hay costos de recuperación.
- **Ahorro**: Desde 2018, ha ahorrado a los clientes mil millones de dólares frente a S3 Standard.
- **Uso**: Ideal para **patrones de acceso impredecibles**, aplicable a cualquier carga de trabajo.
- **Sin costos extras**: No hay gastos de operación, cargos de ciclo de vida o mínimos de almacenamiento.
- **Tamaño de objeto**: Objetos **menores de 128 KB** no son elegibles para la automatización y se tarifican al nivel frecuente.



S3 Intelligent-Tiering

0,0025 USD por 1000 objetos
+ coste por clase usada

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com



S3 Avanzado

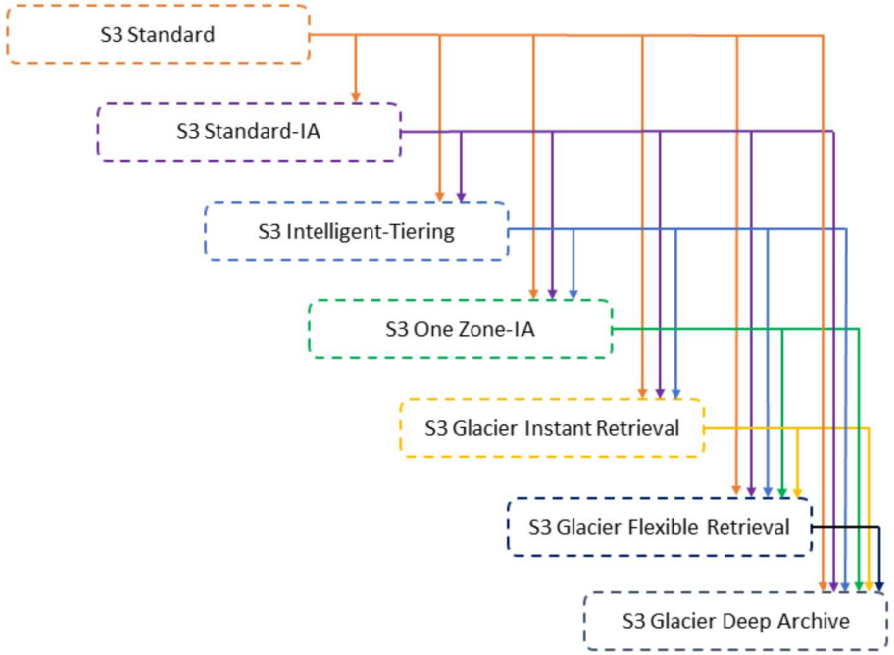
www.blockstellart.com

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Movimiento de objetos entre diferentes clases de almacenamiento de Amazon S3

- En una configuración del ciclo de vida de S3, puedes definir **reglas para pasar objetos de una clase de almacenamiento a otra** para ahorrar costos de almacenamiento

- Para los objetos a los que se accede con poca frecuencia, muévelos a S3 Standard-IA
- Para los objetos a los que no necesitas acceder rápidamente, muévelos a S3 Glacier
- Amazon S3 admite un **modelo en cascada** para las transiciones entre clases de almacenamiento



https://docs.aws.amazon.com/es_es/AmazonS3/latest/userguide/lifecycle-transition-general-considerations.html

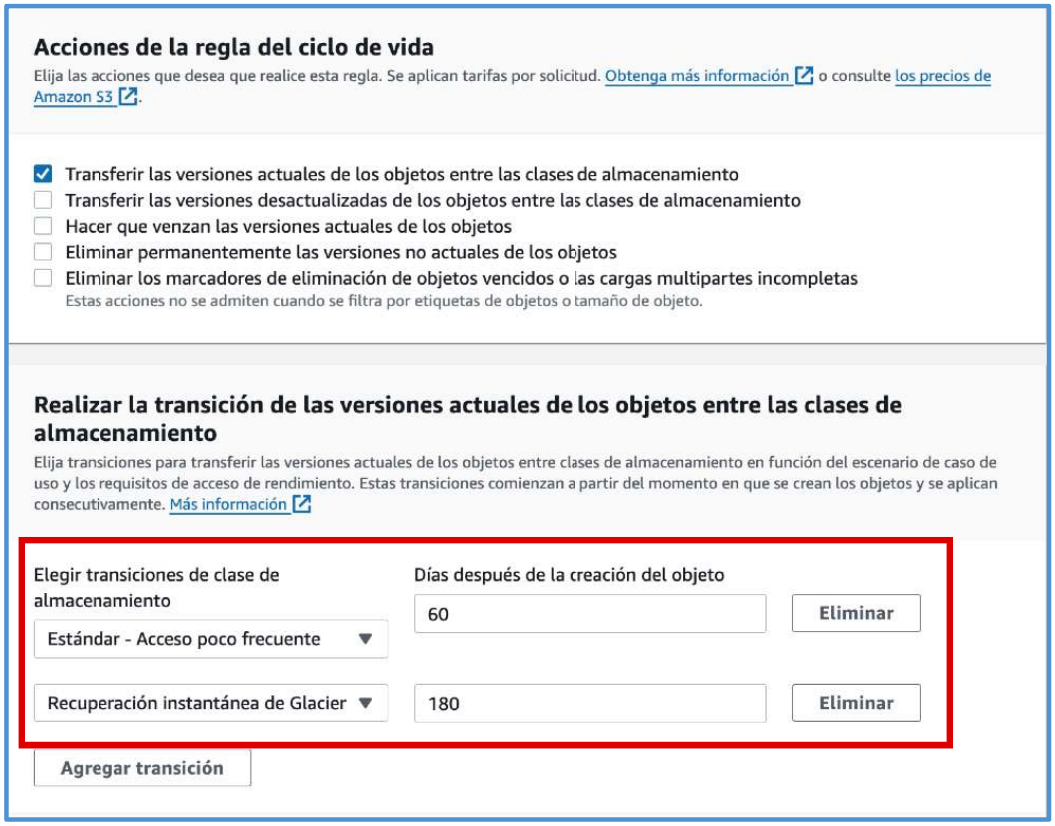
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Reglas del ciclo de vida de Amazon S3

- Movimiento de los objetos entre diferentes clases de almacenamiento
  - Mover los objetos a la clase S3 Standard-IA 60 días después de su creación
  - Mover a S3 Glacier Flexible Retrieval para archivar después de 180 días

- Configura los objetos para que se eliminen después de X tiempo
  - **Se puede utilizar para eliminar versiones antiguas de archivos (si el versionado está activado)**
- Se pueden crear reglas…
  - para un determinado prefijo (ejemplo: *s3://mibucket/fotos/**)
  - para determinados objetos con etiquetas (ejemplo: *Fase:Test*)



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Ejercicio 1 - Reglas del ciclo de vida

- **Imagina que tienes una aplicación en AWS que procesa y almacena vídeos de capacitación en S3. Cada vídeo se transcodifica en varios formatos para diferentes dispositivos y solo se necesitan durante 90 días, ya que después se actualizan con nuevas versiones. Los vídeos originales deben estar disponibles de inmediato durante esos 90 días, pero después pueden almacenarse de forma más económica si su acceso es menos frecuente. ¿Cuál sería tu enfoque de diseño?**

- Los vídeos originales podrían almacenarse en S3 con la clase de almacenamiento Estándar, y se usaría una política de ciclo de vida para moverlos a S3 Glacier o Glacier Deep Archive tras 90 días.
- Los formatos transcodificados, que son recreables, se almacenarían en una clase de almacenamiento S3 IA (Infrequent Access) con una política de ciclo de vida para eliminarlos automáticamente después de los 90 días.

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Ejemplo 2 - Reglas del ciclo de vida

- **Supongamos que estás diseñando un sistema de copias de seguridad para una base de datos en AWS, donde los snapshots se crean diariamente y deben ser accesibles inmediatamente por un período de 45 días. Sin embargo, después de ese periodo, estos pueden ser archivados, aunque con la posibilidad de recuperación dentro de un día en situaciones de emergencia que podrían suceder ocasionalmente. Los snapshots después de 45 días no necesitarían ser recuperados más que una vez al año. ¿Cómo estructurarías esta solución?**

- Implementa el versionado en S3 para los snapshots de la base de datos, asegurando que se mantengan accesibles durante el periodo de 45 días en la clase de almacenamiento Estándar.
- Después de 45 días, mueve los snapshots a S3 Standard-IA, y tras un año, a S3 Glacier o Glacier Deep Archive, con políticas de ciclo de vida para la transición y recuperación dentro de 24 horas en caso de ser necesarios.

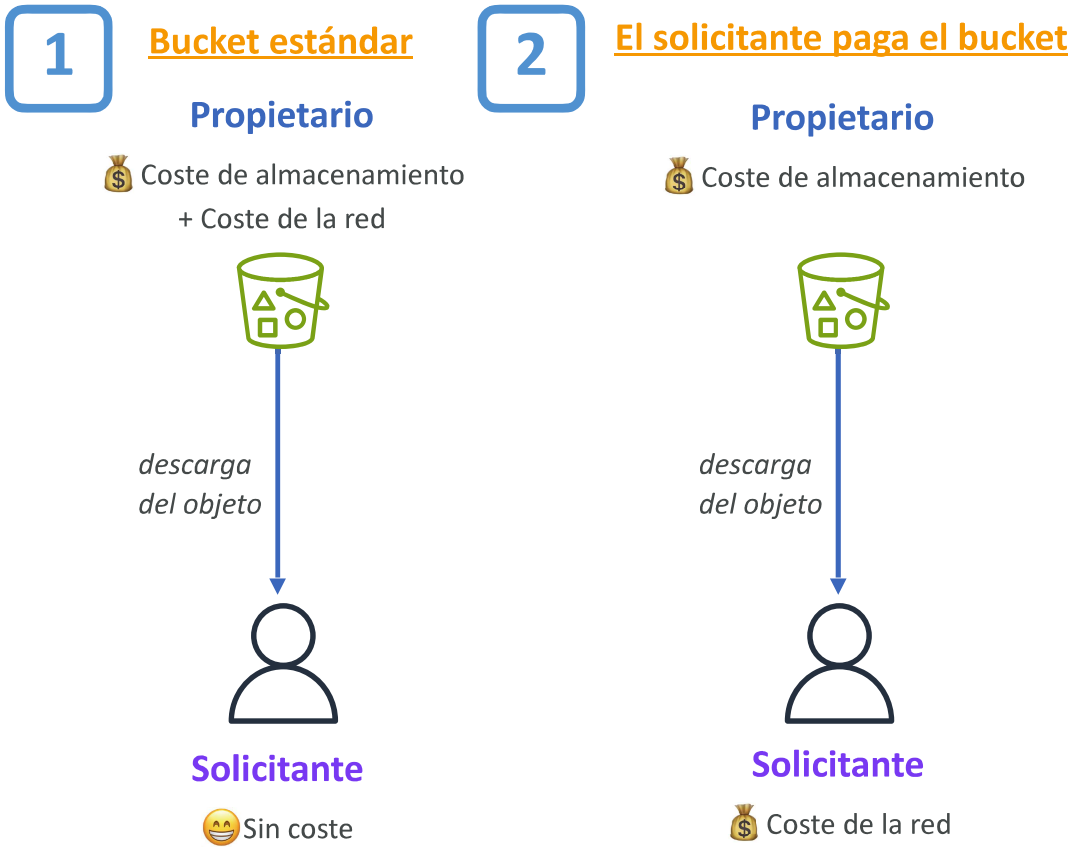
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Buckets S3 de pago por solicitante

- Normalmente, los propietarios de los buckets S3 pagan todos los costes de almacenamiento y transferencia de datos de Amazon S3 asociados a su bucket
- No obstante, hay una opción donde **los solicitantes realizan el pago del coste de petición y descarga de datos**
- Es útil cuando quieres compartir grandes conjuntos de datos con otras cuentas

⚠️ Si habilitas los pagos por solicitante en un bucket, no se permite el acceso anónimo a ese bucket.



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Contextualización del uso de CORS

Sac - The Cloud Dog

HTML

Hola, me llamo Sac y soy un Cloud Dog certificado por AWS. 😉



Vídeo

Bucket S3:**mi-bucket-page**
*(Sitio web estático **activado**)*

Bucket S3: **mi-bucket-videos**
*(Sitio web estático **activado**)*





© Joan Amengual

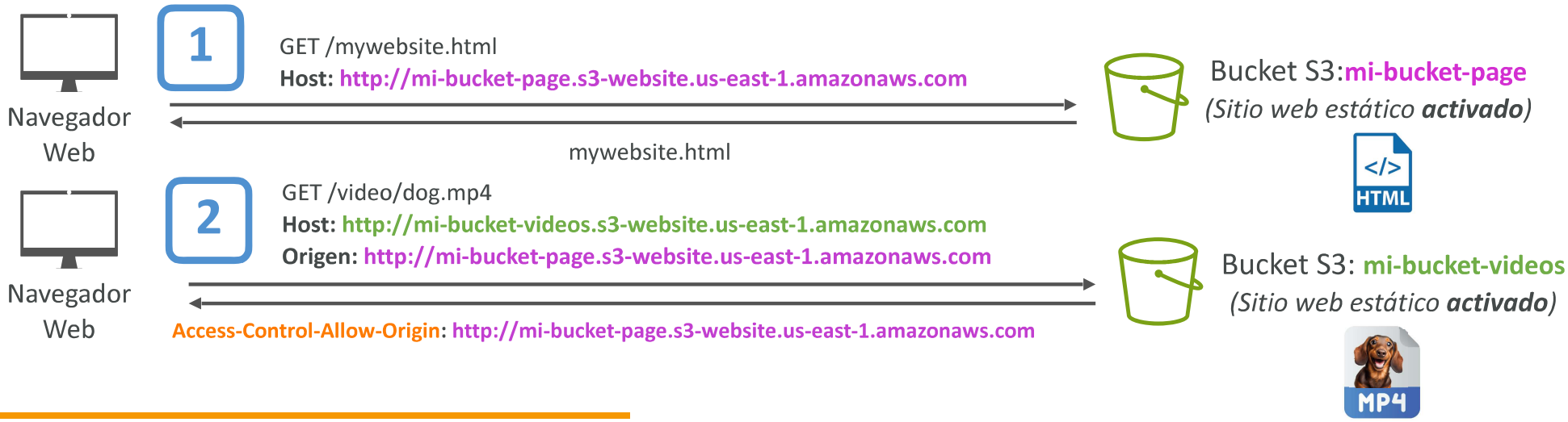
Todos los derechos reservados © Joan Amengual www.blockstellart.com

Uso compartido de recursos entre orígenes (CORS)

- Mecanismo **basado en el navegador web** para permitir peticiones a otros orígenes mientras se visita el origen principal
- Las peticiones no se cumplirán a menos que el otro origen permita las peticiones, utilizando **cabeceras CORS**

- Si un cliente hace una petición de origen cruzado en nuestro bucket de S3, **tenemos que habilitar las cabeceras CORS correctas**
- Puedes permitir un origen específico o todos los orígenes (*)

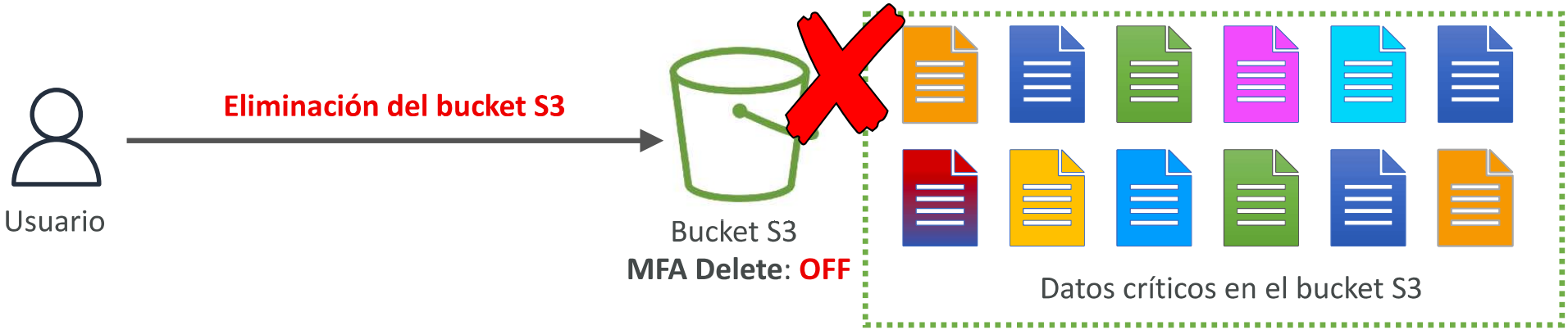




© Joan Amengual

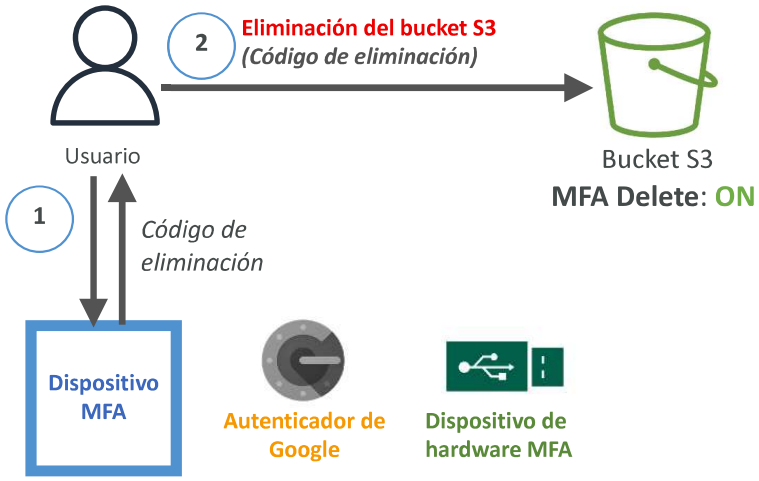
Todos los derechos reservados © Joan Amengual www.blockstellart.com

MFA Delete - Evita perder información valiosa



¿Cómo podemos evitar el borrado de información crítica en buckets S3?

- **MFA** obliga a los usuarios a generar un código en un dispositivo antes de realizar operaciones importantes en el S3
- MFA será necesario para:
  - Eliminar permanentemente una versión de un objeto
  - Suspender el control de versiones en el bucket
- Para utilizar MFA Delete, **el control de versiones debe estar activado** en el bucket S3
- **Sólo el propietario del bucket (cuenta root) puede activar/ desactivar MFA Delete**

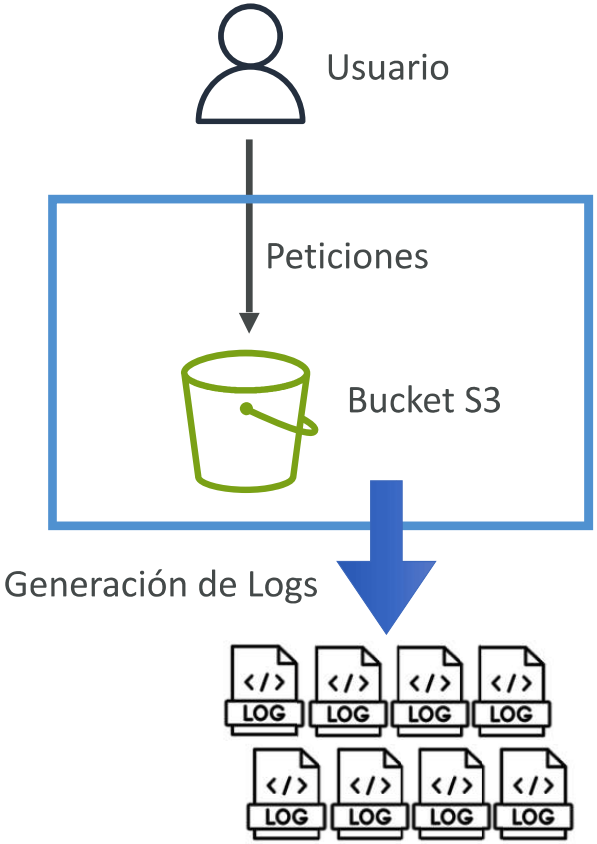


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Logs de acceso de Amazon S3

- 💡Es crucial registrar todos los **accesos a los buckets de S3** para fines de auditoría
  - Esto permitirá mantener un **seguimiento detallado de todas las operaciones** que se realizan en tus recursos
- Cualquier petición realizada a S3, desde cualquier cuenta, autorizada o denegada, se registrará en **otro** bucket de S3
- El bucket de logs de destino debe estar en la misma región de AWS

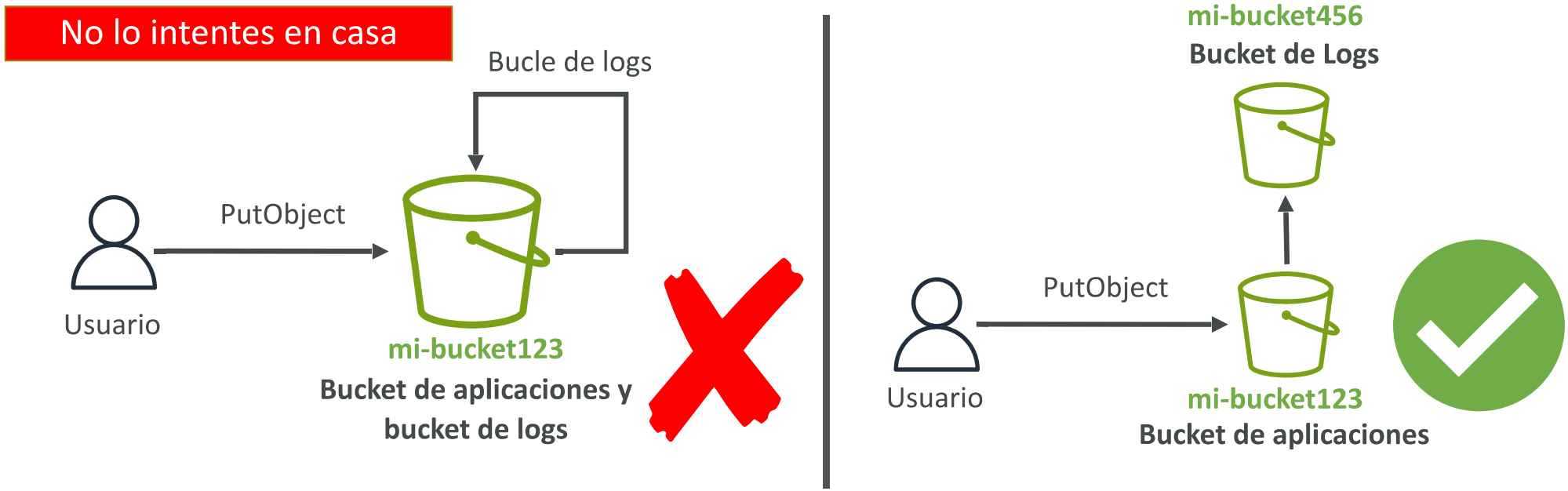


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

⚠⚠⚠ S3 Access Logs ⚠⚠⚠

- No configures tu bucket de logs para que sea el bucket monitorizado
- Se creará un bucle de logs, y **tu bucket crecerá exponencialmente**



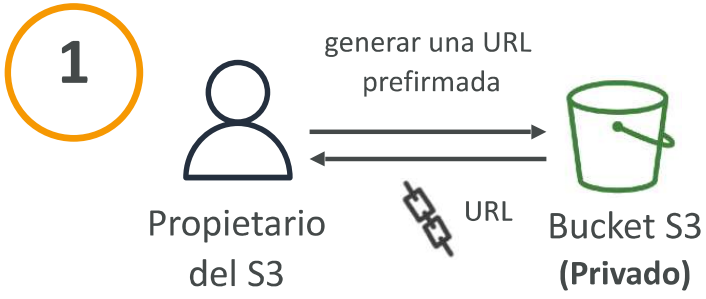
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

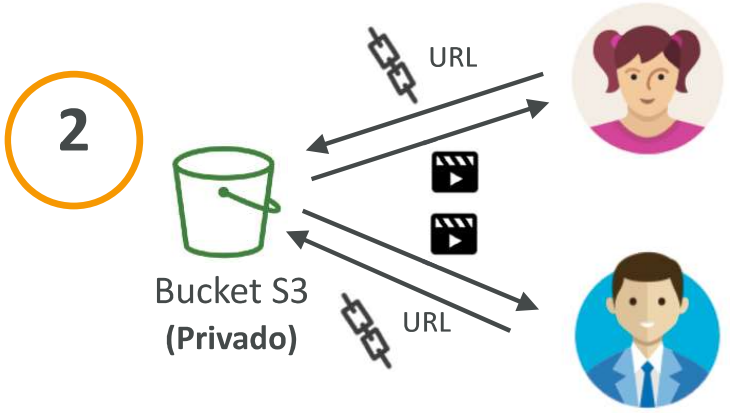
URLs prefirmadas en Amazon S3

- Otorgar **acceso limitado en el tiempo a objetos en Amazon S3** sin necesidad de actualizar su política de buckets
- Generar URLs prefirmadas usando la consola de **S3, la CLI de AWS o el SDK**

- **Tiempo de caducidad de la URL:**
  - **Consola de S3:** 1 minuto - 12 horas
  - **AWS CLI / SDK**: hasta en 7 días



- **Casos de uso de URLs prefirmadas**:
  - Permite que sólo los usuarios que han iniciado sesión descarguen un vídeo premium de un bucket de S3
  - Permitir temporalmente que un usuario suba un archivo a una ubicación precisa en tu bucket de S3



Mediante la URL prefirmada acceden al contenido premium del Bucket S3 (privado)

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Bloqueo de información con S3 Glacier Vault Lock

- Adoptar un **modelo WORM** (Write Once Read Many)
- Crea una "política de bloqueo de bóveda" (**Vault Lock Policy**)
- Bloquea la política para futuras ediciones (ya no se puede modificar ni borrar)
- Los objetos almacenados en el bucket S3 Glacier no podrán ser borrados posteriormente
- Caso de uso común: Útil para el cumplimiento de la normativa y la retención de datos



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Bloqueo de objetos con S3 Object Lock

- **Permite bloquear el borrado de una versión del objeto durante un tiempo determinado**
- El tiempo de bloqueo puede ser modificado posteriormente
- Mediante el permiso el permiso IAM **s3:PutObjectLegalHold** no es necesario aplicar un período de retención (es indefinido hasta que eliminas el permiso IAM)

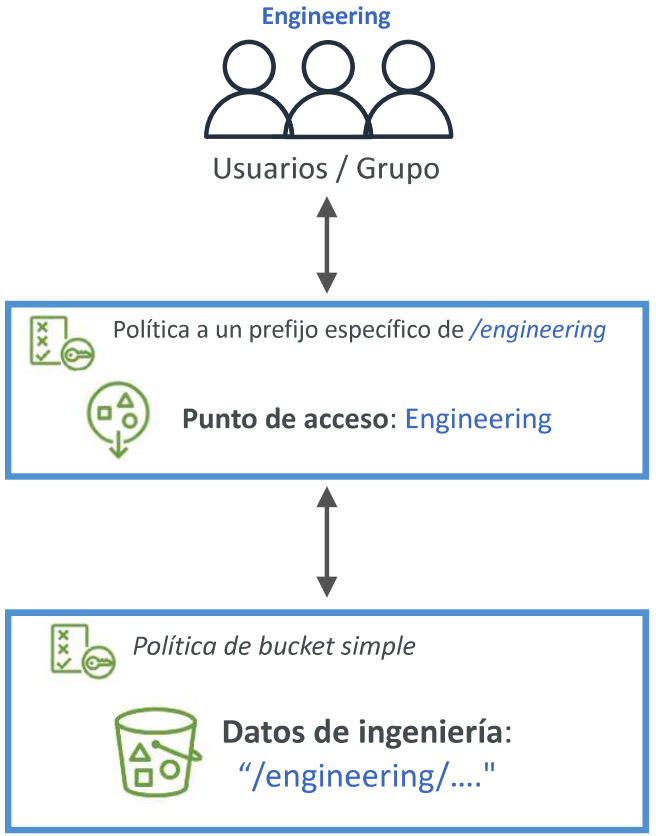


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Puntos de acceso (AP) de Amazon S3 (1/2)

- Los puntos de acceso están diseñados para **proporcionar acceso granular a los buckets de S3**
  - Permitiendo a los usuarios crear políticas de acceso específicas para diferentes aplicaciones o usuarios, mejorando la seguridad y la gestión del acceso
- Los usuarios pueden acceder a datos a través de un **ARN de punto de acceso o un alias de punto de acceso**, facilitando la integración con diferentes servicios y aplicaciones
- Cada punto de acceso está configurado con su **propia política de acceso**, adaptada a casos de uso o aplicaciones específicas, y puede admitir desde un solo usuario hasta grupos de usuarios o aplicaciones



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Puntos de acceso (AP) de Amazon S3 (2/2)



Cada punto de acceso tiene una dirección DNS única para el acceso a la red

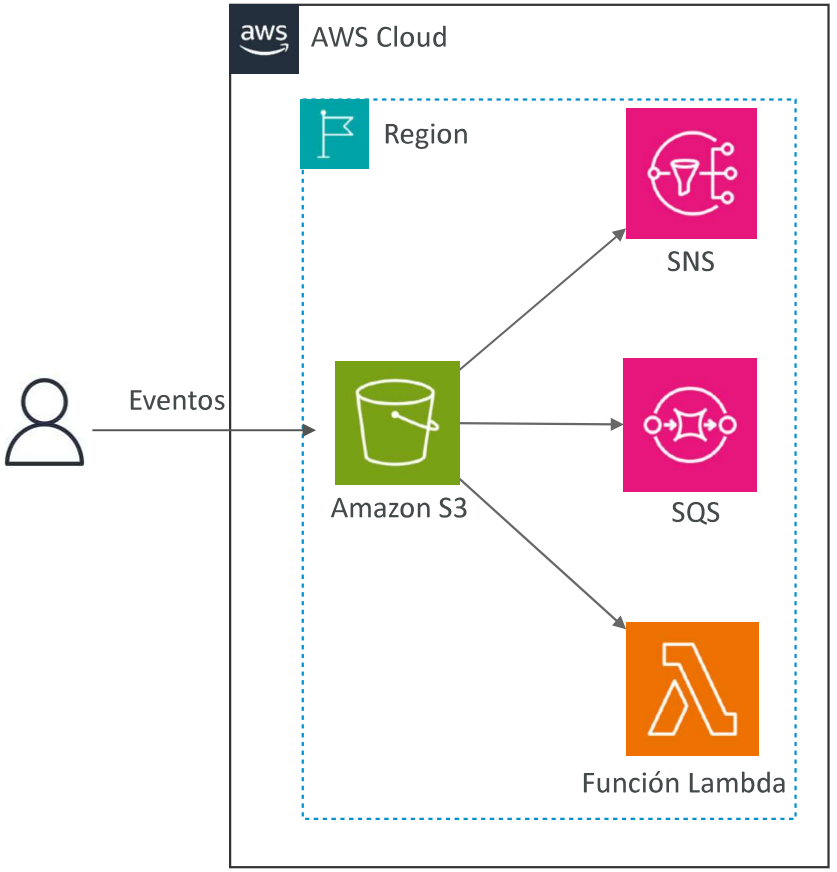
Las políticas controlan los permisos de acceso a través del punto de acceso y son funcionalmente equivalentes a una política de bucket. La política de punto de acceso puede restringir identidades a determinados prefijos, etiquetas o acciones en función de las necesidades

Los puntos de acceso pueden configurarse para el acceso a través de una VPC - requiere un endpoint VPC. El acceso a través de esta ruta se puede imponer mediante políticas de punto final

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Notificaciones de eventos S3

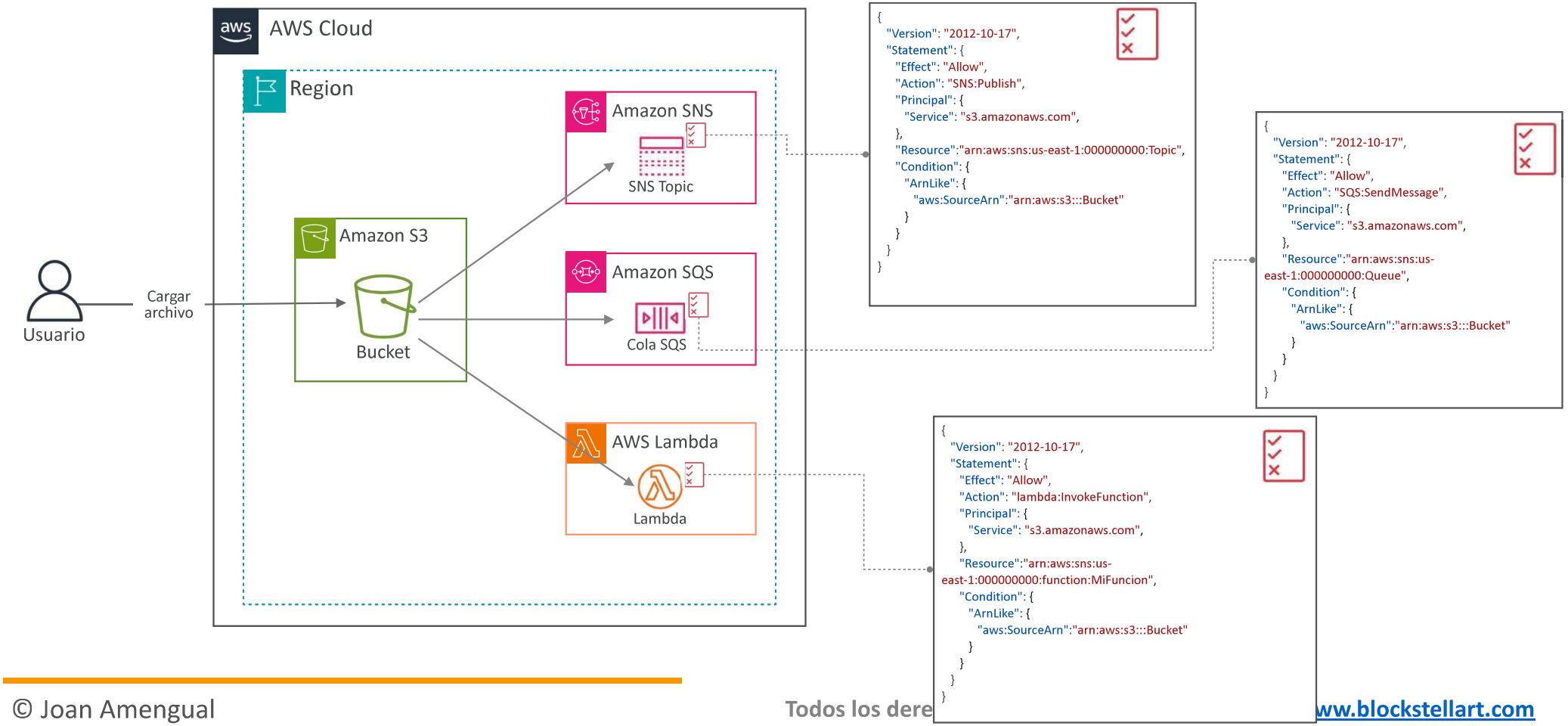
- S3 puede generar eventos como los siguientes:
  - S3:ObjectCreated, S3:ObjectRemoved, S3:ObjectRestore, S3:Replication…
- Posibilidad de filtrar el nombre del objeto (*.jpg)
- **Se pueden crear tantos "eventos S3" como se desee**
- Las notificaciones de eventos S3 suelen entregar los eventos en segundos, pero a veces pueden tardar un minuto o más



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

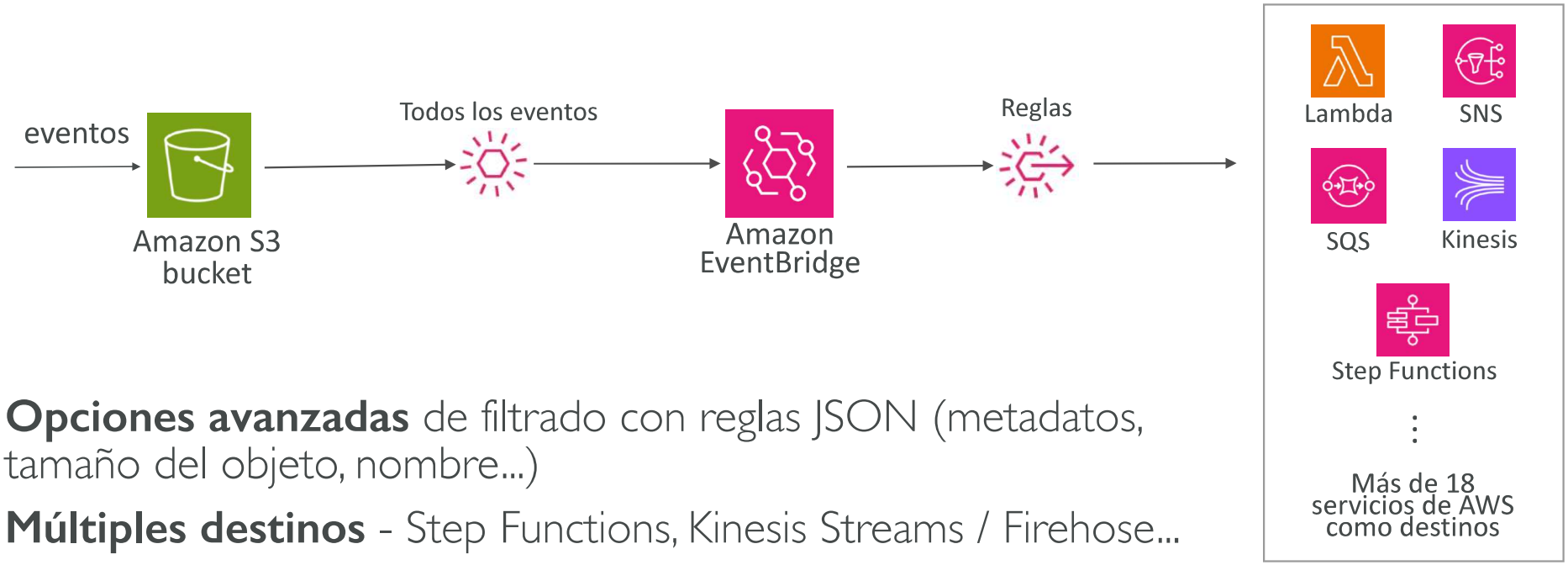
Notificaciones de eventos S3 - Permisos IAM



© Joan Amengual

Todos los dere

Notificaciones de eventos S3 con EventBridge



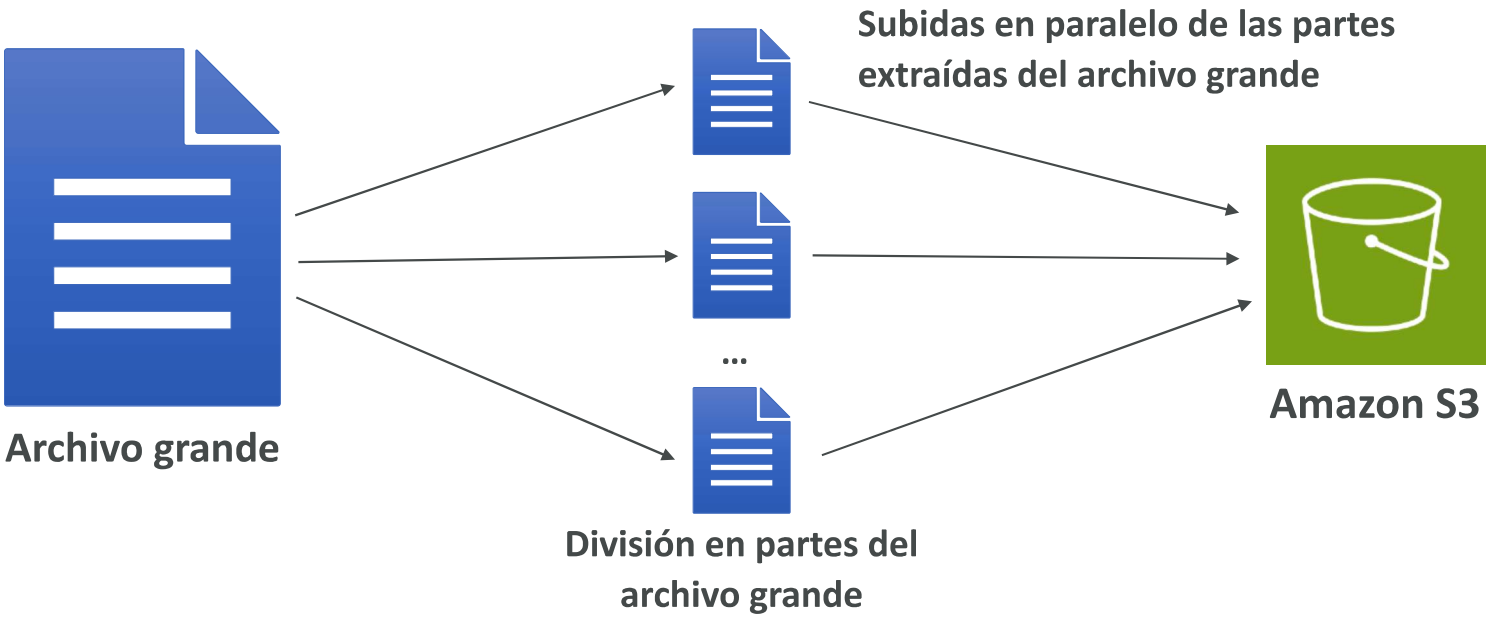
- **Opciones avanzadas** de filtrado con reglas JSON (metadatos, tamaño del objeto, nombre...)
- **Múltiples destinos** - Step Functions, Kinesis Streams / Firehose...
- **Capacidades de EventBridge** - Repetición de eventos, entrega fiable

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Rendimiento S3 (1/2)
Carga multiparte

- Recomendado para archivos > 100MB, obligatorio para archivos > 5GB
- Puede ayudar a paralelizar las subidas (acelerar las transferencias)



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Rendimiento S3 (2/2)
Aceleración de transferencia en S3

- Utiliza la red global de AWS para **minimizar la latencia** de transferencia de datos
- Aprovecha la optimización de ruta para usar el ancho de banda disponible de forma más eficiente
- Se **beneficia de la red de entrega de contenido (CDN)** de AWS para acelerar la entrega de archivos
- Utiliza protocolos seguros para proteger los datos durante la transferencia
- **Reduce los costos operativos** al disminuir el tiempo de transferencia y potencialmente el tráfico saliente
- Ideal para la migración de grandes volúmenes de datos a la nube de AWS
- Compatible con la carga multiparte

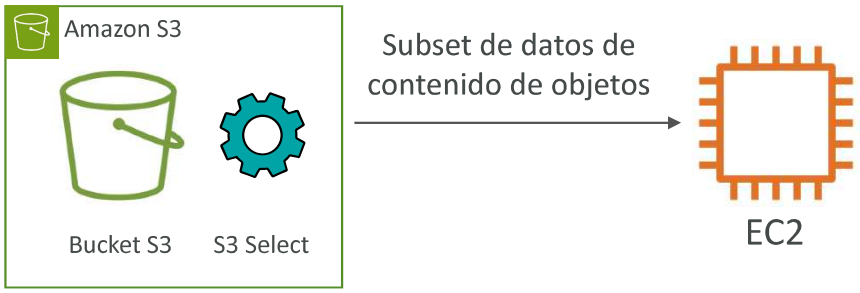


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon S3 Select y Amazon S3 Glacier Select

- **Amazon S3 Select permite utilizar instrucciones SQL para filtrar el contenido de los objetos de S3 y recuperar exactamente el subconjunto de datos que necesitas**
- **Amazon S3 Glacier Select permite ejecutar consultas directamente en los datos almacenados en Amazon Glacier y recupera solo los datos necesarios de los archivos para utilizarlos en el análisis**
- Menos transferencia de red, menos coste de CPU en el lado del cliente
- Puedes filtrar por filas y columnas (simples sentencias SQL)



Ejemplos: Utilizar Amazon S3 Select para buscar y recuperar únicamente los registros que coincidan con una cuenta determinada, o solo los datos de facturación para un cliente específico

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon S3 - Cifrado de objetos



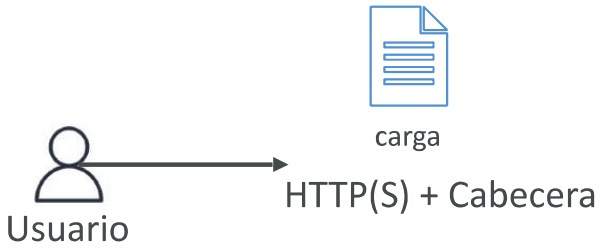
- Puedes cifrar objetos en buckets de S3 utilizando uno de los métodos siguientes
- **Cifrado del lado del servidor (SSE)**
  - **Cifrado del lado del servidor con claves gestionadas por Amazon S3 (SSE-S3)**
    - **Activado por defecto**
    - Cifra los objetos de S3 utilizando claves manejadas, gestionadas y propiedad de AWS
  - **Cifrado del lado del servidor con claves KMS almacenadas en AWS KMS (SSE-KMS)**
    - Aprovecha el servicio de administración de claves de AWS (AWS KMS) para gestionar las claves de cifrado
  - **Cifrado del lado del servidor con claves proporcionadas por el cliente (SSE-C)**
    - Cuando quieras gestionar tus propias claves de cifrado
- **Cifrado del lado del cliente**
- Es importante entender cuáles son para cada situación para exámenes de certificación de AWS

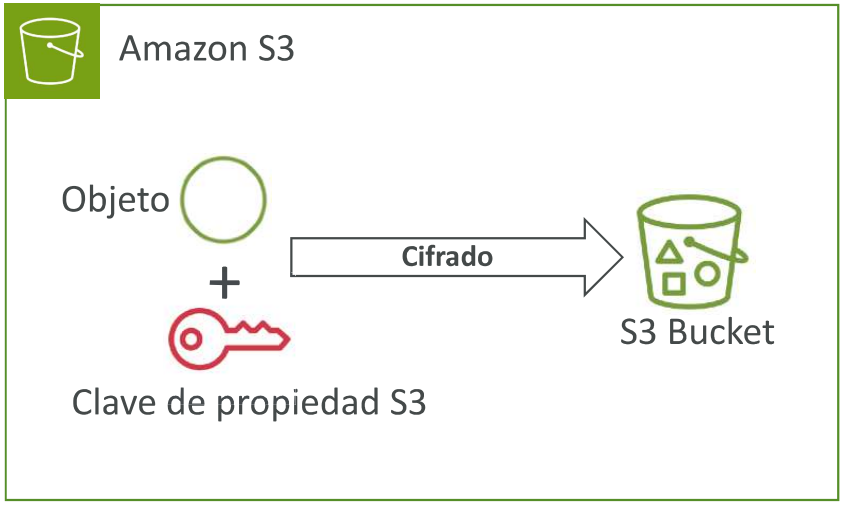
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Cifrado de Amazon S3 - SSE-S3

- Cifrado mediante claves manejadas, gestionadas y propiedad de AWS
- El objeto está cifrado en el lado del servidor
- El tipo de cifrado es **AES-256**
- Debe establecerse la cabecera **"x-amz-server-side-encryption": "AES256"**
- **Activado por defecto para nuevos buckets y nuevos objetos**



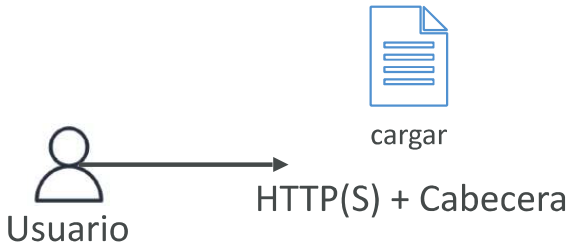


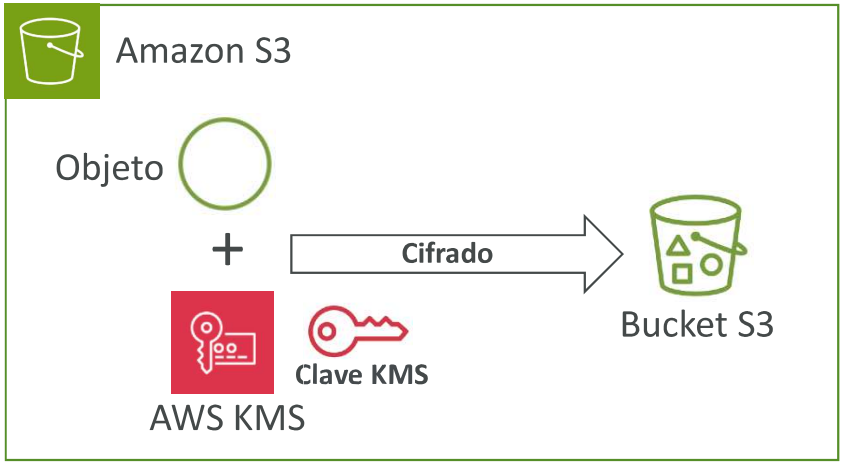
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Cifrado de Amazon S3 - SSE-KMS

- Cifrado mediante claves manejadas y gestionadas por AWS KMS (Key Management Service)
- Ventajas del KMS: control del usuario + auditoría del uso de las claves mediante CloudTrail
- El objeto está cifrado en el lado del servidor
- Debes establecer la cabecera **"x-amz-server-side-encryption": "aws:kms"**



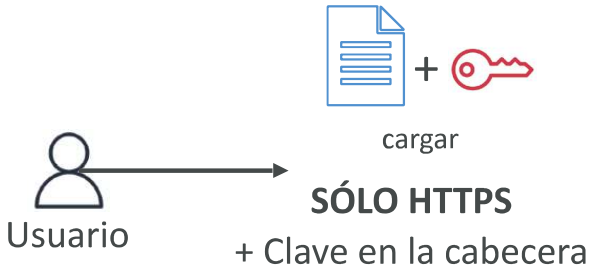


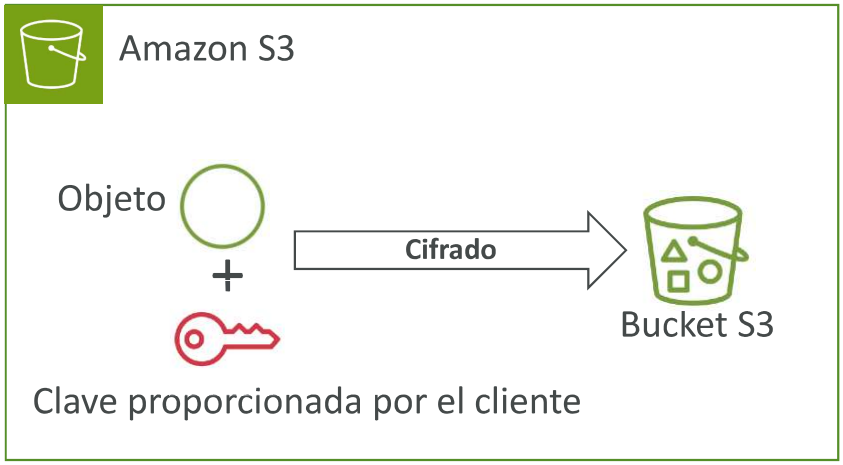
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Cifrado de Amazon S3 - SSE-C

- Cifrado del lado del servidor mediante claves totalmente gestionadas por el cliente fuera de AWS
- Amazon S3 **NO** almacena la clave de cifrado que proporciones
- **Se tiene que utilizar HTTPS**
- La clave de cifrado debe proporcionarse en las cabeceras HTTP, para cada petición HTTP realizada



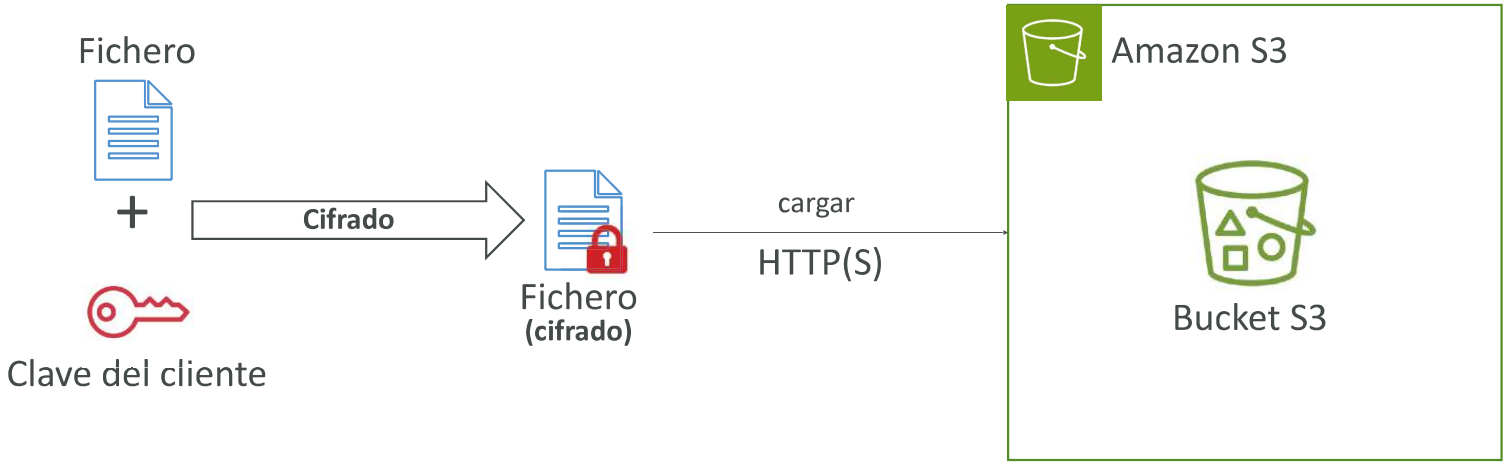


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Cifrado de Amazon S3 - Cifrado del lado del cliente

- Utiliza bibliotecas de clientes como la **biblioteca de cifrado del lado del cliente de Amazon S3**
- Los clientes deben cifrar los datos ellos mismos antes de enviarlos a Amazon S3
- Los clientes deben descifrar los datos ellos mismos al recuperarlos de Amazon S3
- El cliente gestiona completamente las claves y el ciclo de cifrado



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon S3 - Cifrado en tránsito (SSL/TLS)

- El cifrado en vuelo también se llama SSL/TLS
- Amazon S3 expone dos endpoints:
  - **HTTP Endpoint** - no cifrado
  - **HTTPS Endpoint** - cifrado en vuelo
- **Se recomienda HTTPS**
- **HTTPS es obligatorio para SSE-C**
- La mayoría de los clientes usarían el endpoint HTTPS por defecto

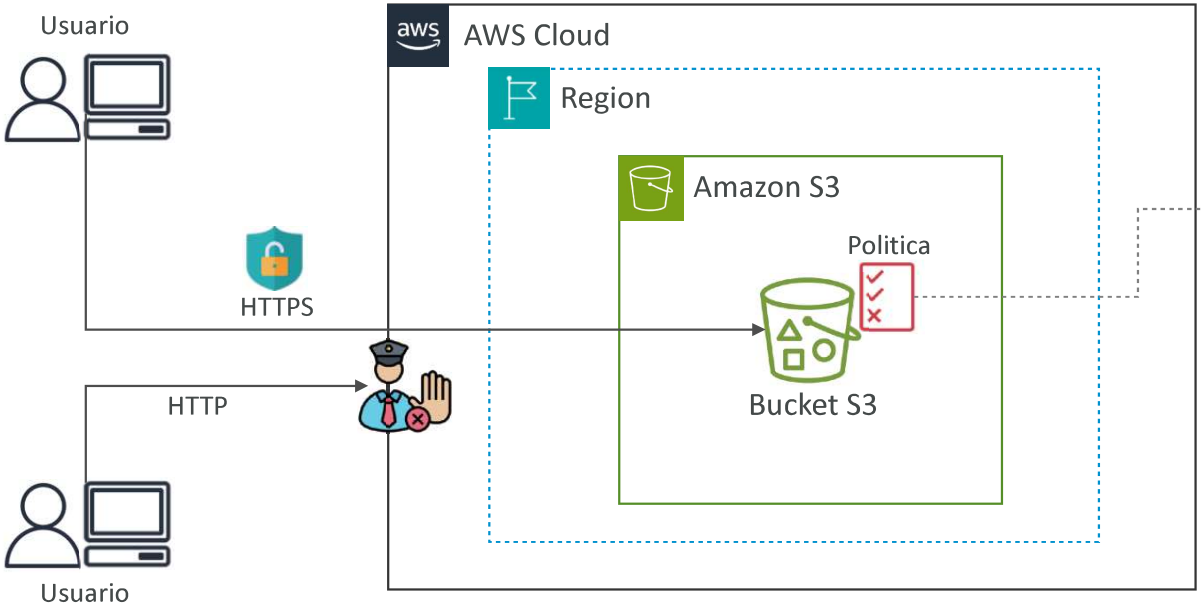


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Forzado del cifrado en tránsito en Amazon S3

Importancia del cifrado en tránsito para la protección contra ataques de tipo "man-in-the-middle"




```
{
  "Version": "2012-10-17",
  "Statement": [
      {
        "Effect": "Deny",
        "Action": "s3:GetObject",
        "Principal": "*",
        "Resource":"arn:aws:s3:::bucket/*",
        "Condition": {
          "Bool": {
            "aws:SecureTransport":"false"
          }
        }
      }
  ]
}
```

Uso de HTTPS para proteger la integridad y la confidencialidad de los datos entre el cliente y el bucket S3

Implementación de políticas de seguridad en S3 para garantizar el cifrado en tránsito

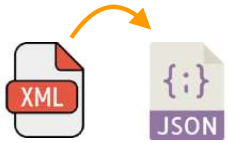
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon S3 Object Lambda

- Utiliza las funciones Lambda de AWS para modificar el objeto antes de que lo recupere la aplicación que lo llama
- Sólo se necesita un bucket de S3, sobre el que creamos **puntos de acceso de S3** y **puntos de acceso de S3 Object Lambda**.
- Casos de uso:

- Convertir entre formatos de datos, como convertir XML a JSON



- Automáticamente ocultar o eliminar información confidencial, como números de tarjetas de crédito o datos personales



- Extraer partes específicas de archivos, como ciertas columnas de un archivo CSV, antes de enviarlos al usuario final



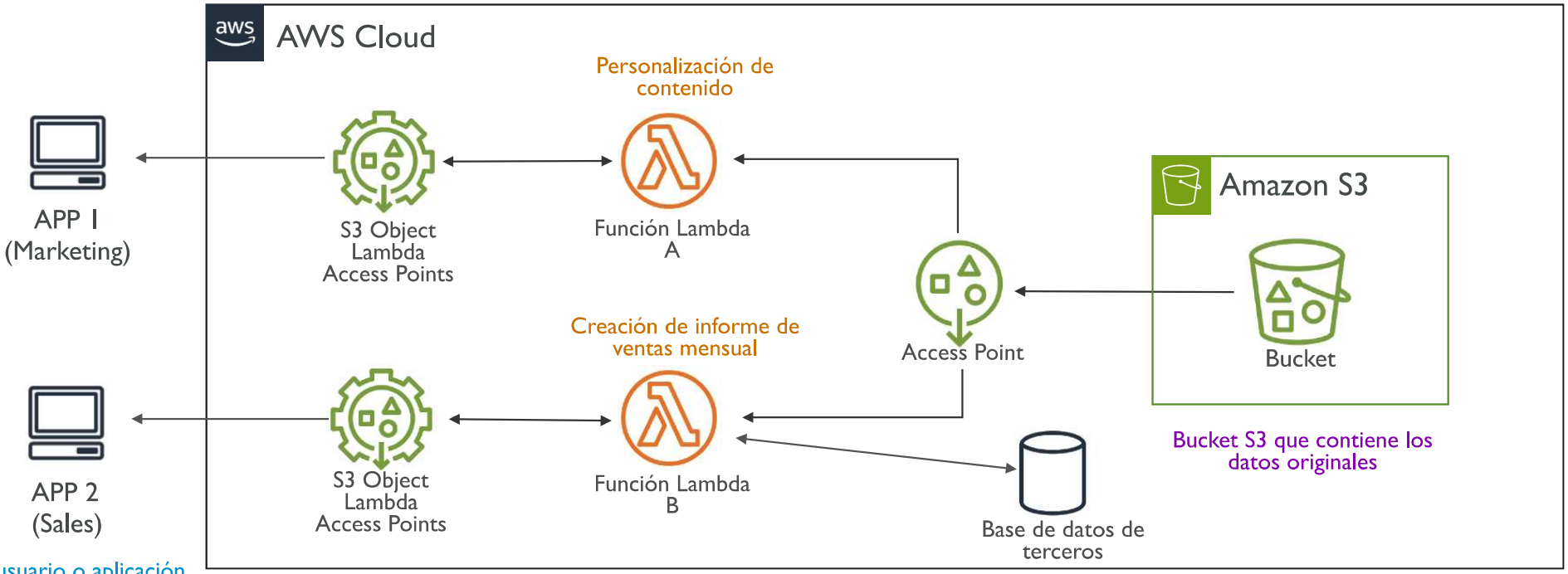
- Redimensionar y poner marcas de agua a las imágenes



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon S3 Object Lambda



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com



Introducción a las bases de datos de AWS

www.blockstellart.com

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Bases de datos relacionales vs. no relacionales

Bases de datos relacionales (SQL)

- Una base de datos relacional (o base de datos SQL) almacena los datos en formato tabular con filas y columnas
- Las columnas contienen atributos de datos, mientras que en las filas hay valores de dato

Bases de datos no relacionales (NoSQL)

- Utilizan diversos modelos de datos para acceder a estos y administrarlos
- Están optimizadas específicamente para aplicaciones que requieren grandes volúmenes de datos, baja latencia y modelos de datos flexibles

https://aws.amazon.com/es/compare/the-difference-between-relational-and-non-relational-databases/

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Bases de datos relacionales vs. no relacionales

RELACIONALES (SQL)



Amazon RDS



Amazon Aurora

NO RELACIONALES (NoSQL)



Amazon DynamoDB



Amazon DocumentDB



Amazon Timestream



Amazon Keyspaces



Amazon Neptune



Amazon ElastiCache

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Bases de datos Relacionales (SQL)

Clave Primaria

P_ID	Nombre	Apellido	Edad
1	Bob	Smith	28
2	Alice	Jones	35
3	James	Davis	37



Tabla de relaciones usando claves

P_ID	M_ID
1	1
2	2
3	2

M_ID	Nombre	Tipo
1	Gin	Perro
2	Mixu	Gato





Clave Compuesta

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Bases de datos No Relacionales (NoSQL)

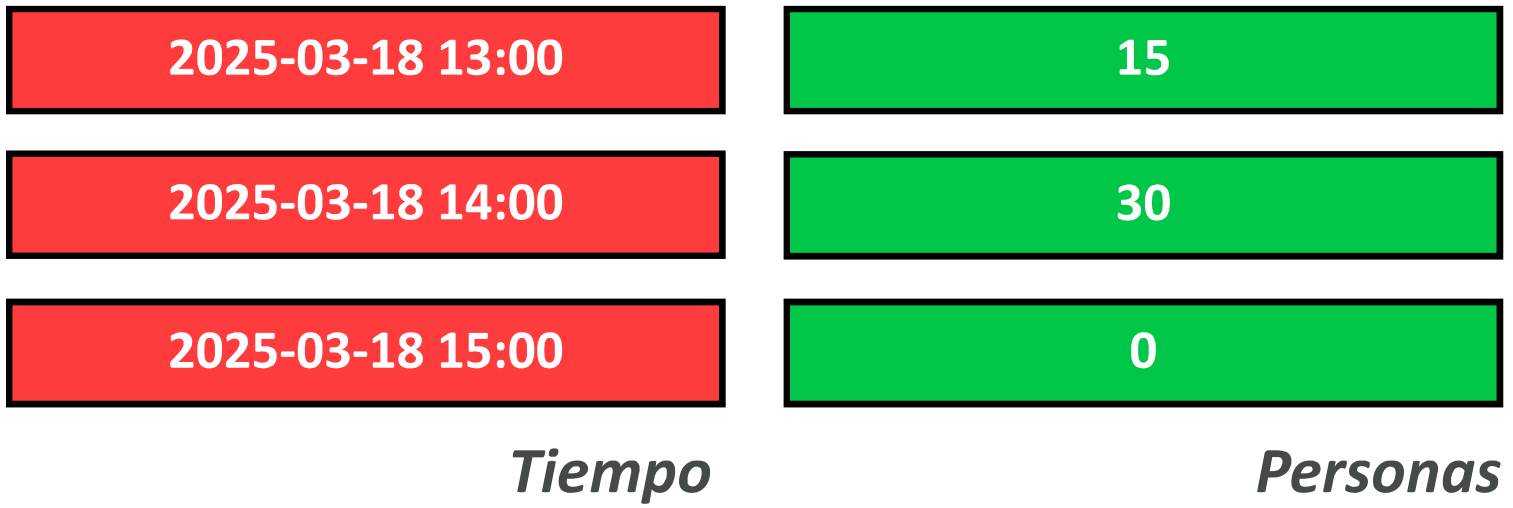
Clave - Valor

No esquema

No estructura

Escalable

Muy rápido

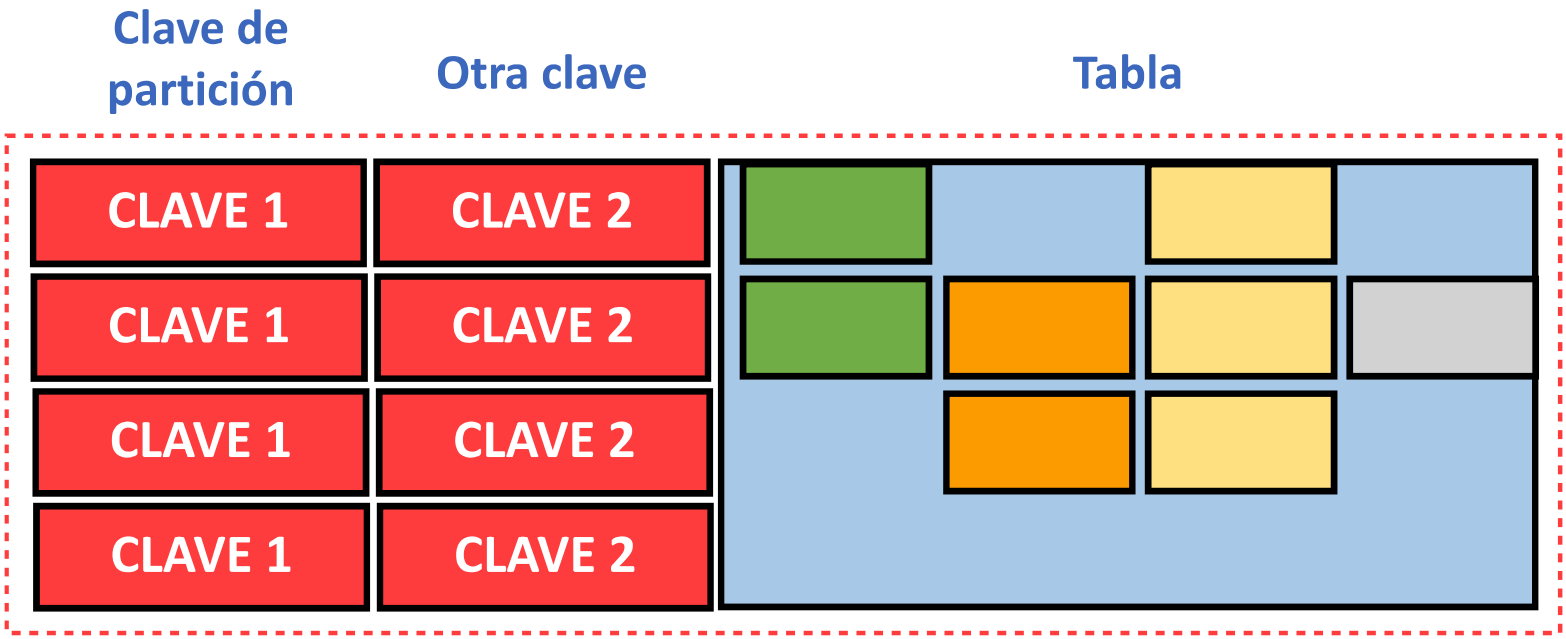


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Bases de datos No Relacionales (NoSQL)

Almacenamiento de columna amplia



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Bases de datos No Relacionales (NoSQL)

Documento

```xml
<publicacion id="p5001">
    <autorId>u10001</autorId>
    <texto>Mi primera publicación</texto>
    <fecha>2024-01-11</fecha>
    <likes>150</likes>
</publicacion>
```

Perfil de usuario en un sistema de redes sociales

```json
{
    "_id": "u10001",
    "nombre": "Juan Pérez",
    "email": "juanperez@example.com",
    "publicacionFavorita": "p5001"
}
```

Publicación en el sistema de redes sociales

Son ideales para aplicaciones que requieren esquemas flexibles y escalabilidad horizontal

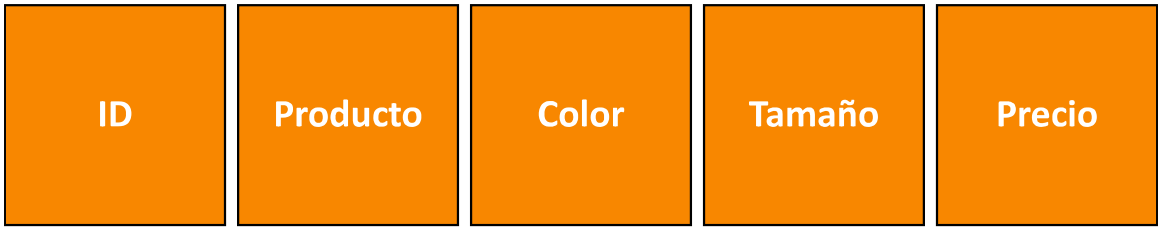
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

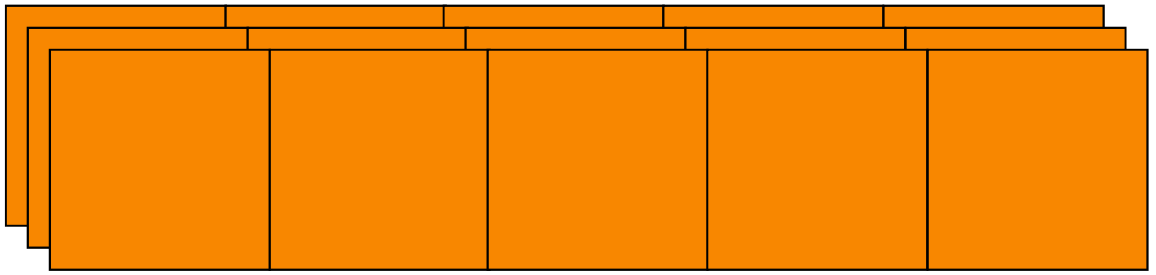
Bases de datos No Relacionales (NoSQL)

Filas (Almacén de filas - MySQL)

FILA



Almacenado como filas



Ideal si operas con filas añadiendo, actualizando, borrando

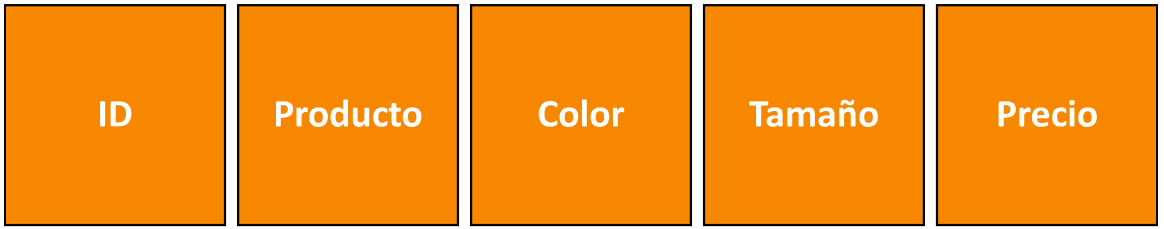
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

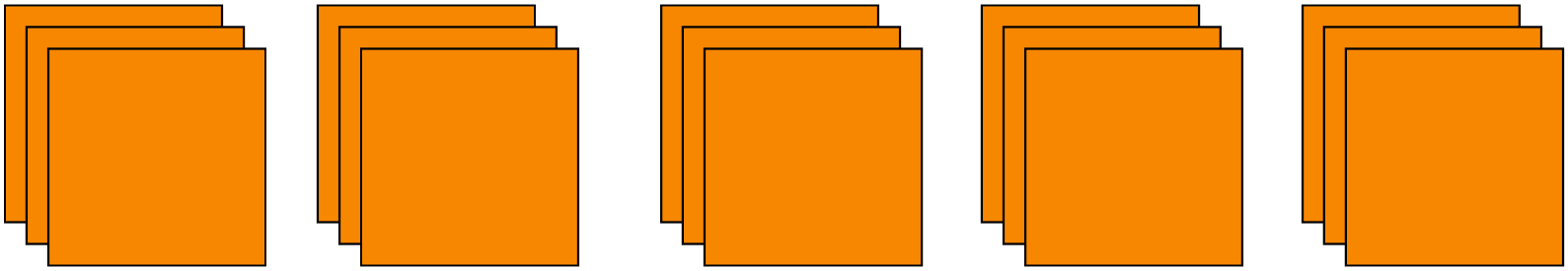
Bases de datos No Relacionales (NoSQL)

Columnas (Almacén de columnas - Redshift)

FILA



Almacenado como columnas



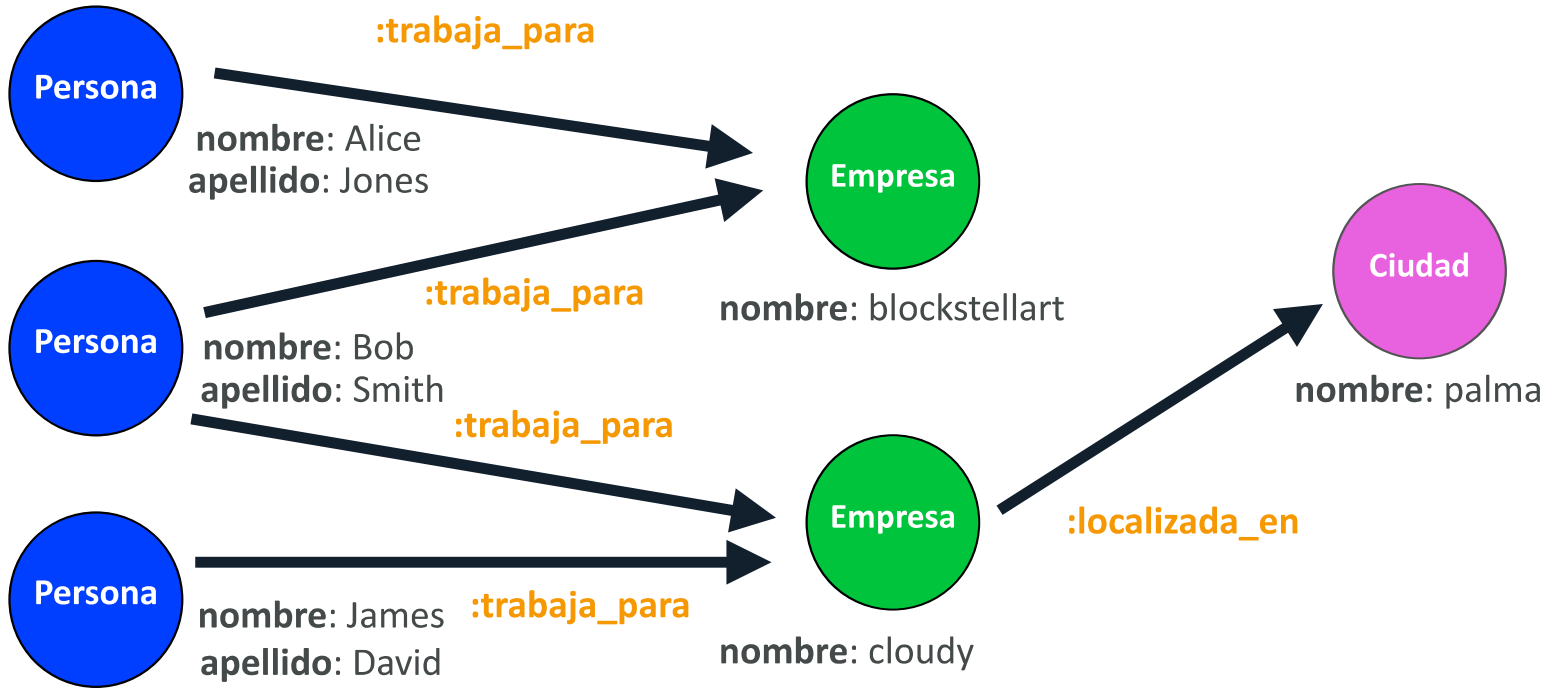
Las columnas se almacenan juntas. Ideal cuando se necesitan todos los valores de un atributo concreto (tamaño)

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Bases de datos No Relacionales (NoSQL)

Grafo



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com



Amazon RDS

www.blockstellart.com

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Visión general de Amazon RDS



- **RDS = Servicio de bases de datos relacionales**
- Es un servicio de bases de datos gestionado para que las bases de datos utilicen SQL como lenguaje de consulta

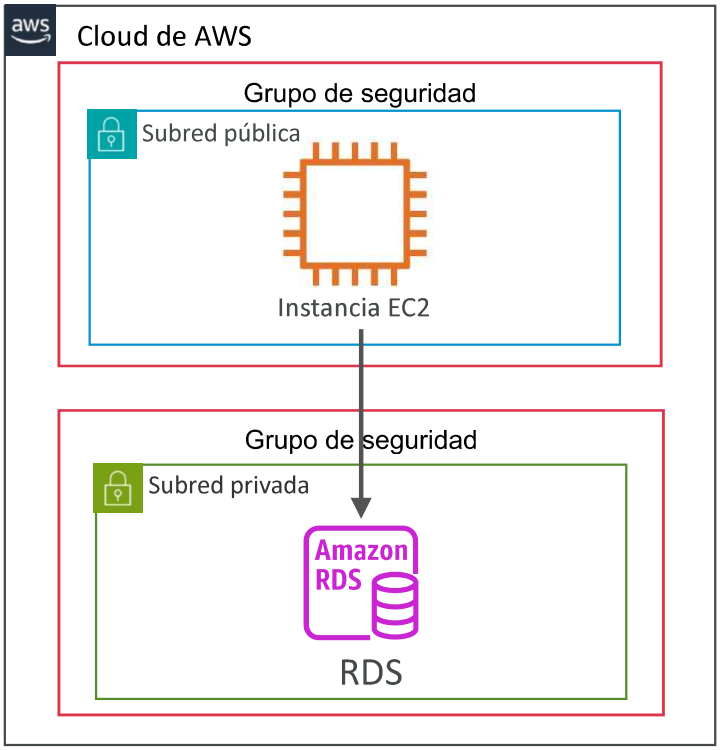
- Amazon RDS admite actualmente los siguientes motores:
  - MariaDB
  - Microsoft SQL Server
  - MySQL
  - Oracle
  - PostgreSQL

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

¿RDS o base de datos en EC2?

- **RDS es un servicio gestionado. Lo que significa…**
  - Aprovisionamiento automatizado
  - Parcheo del sistema operativo
  - Copias de seguridad continuas y restauración a una fecha determinada (Point in Time Restore)
  - Dashboards de monitorización
  - Réplicas de lectura para mejorar el rendimiento de lectura
  - Configuración multi AZ para recuperación de desastres (Disaster Recovery)
  - Capacidad de escalado (vertical y horizontal)
  - Almacenamiento respaldado por EBS
- **PERO no puedes acceder por SSH a tus instancias**



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Autoescalado de almacenamiento de RDS

- **Evita escalar manualmente el almacenamiento de tu base de datos**
- Te ayuda a aumentar el almacenamiento de tu instancia de base de datos RDS de **forma dinámica**
- Cuando RDS detecta que te estás quedando sin almacenamiento en la base de datos escala automáticamente
- Tienes que establecer el **umbral máximo de almacenamiento** (límite máximo de almacenamiento de la base de datos)
- Útil para aplicaciones con **cargas de trabajo imprevisibles**
- Soporta todos los motores de bases de datos RDS (MariaDB, MySQL, PostgreSQL, SQL Server,…)



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Introducción a las réplicas de Amazon RDS

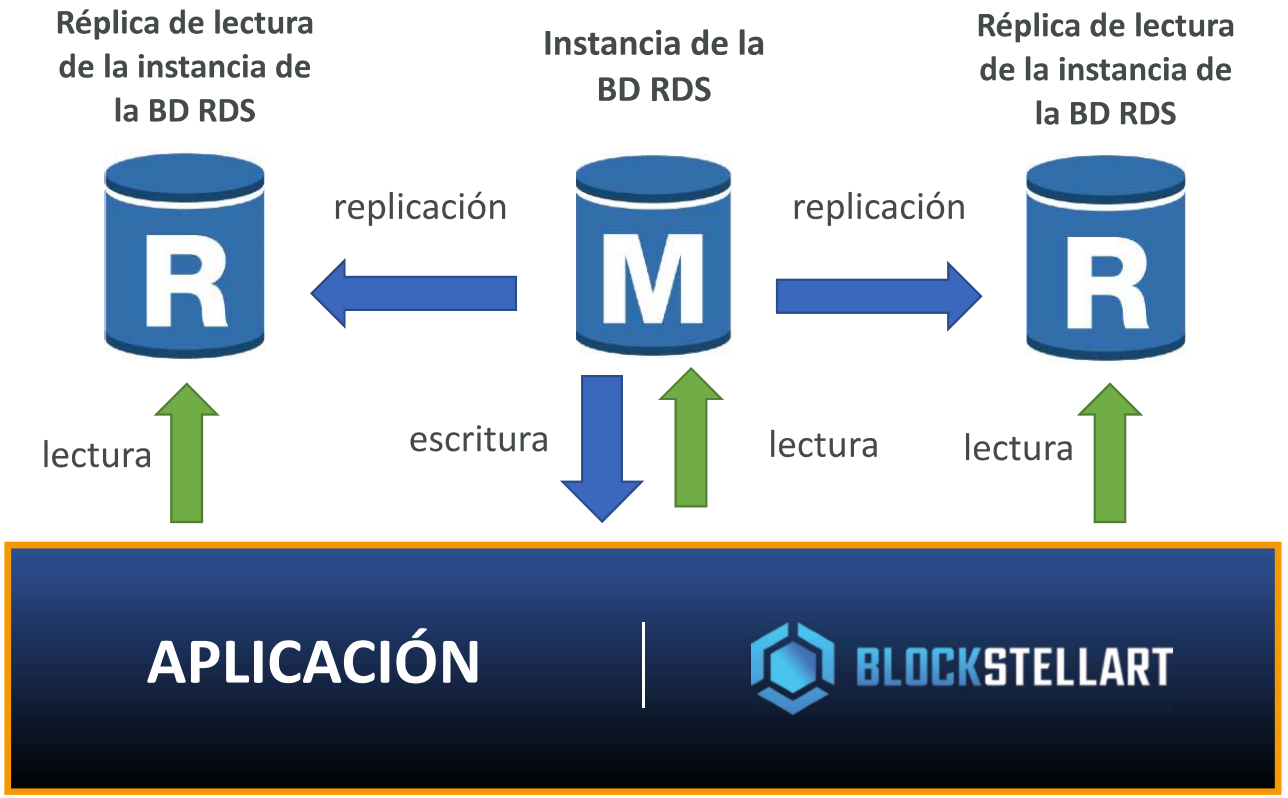
- **Una réplica es una copia de una base de datos principal que se sincroniza continuamente con ella.** Las réplicas se usan principalmente para dos propósitos: **escalabilidad de lectura** y **alta disponibilidad**
  - **Mejora del rendimiento de lectura**: Escalan eficientemente las operaciones de lectura, distribuyendo la carga entre la instancia principal y las réplicas
  - **Alta disponibilidad**: En caso de fallo, una réplica puede convertirse rápidamente en la instancia principal, reduciendo el tiempo de inactividad
  - **Desacoplamiento de cargas de trabajo**: Separan las operaciones de lectura y escritura para optimizar el rendimiento
  - **Facilidad de implementación**: Integración sencilla con varias bases de datos soportadas por RDS. Las aplicaciones deben actualizar la cadena de conexión para aprovechar las réplicas de lectura



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Réplicas de lectura de Amazon RDS



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

RDS Multi AZ

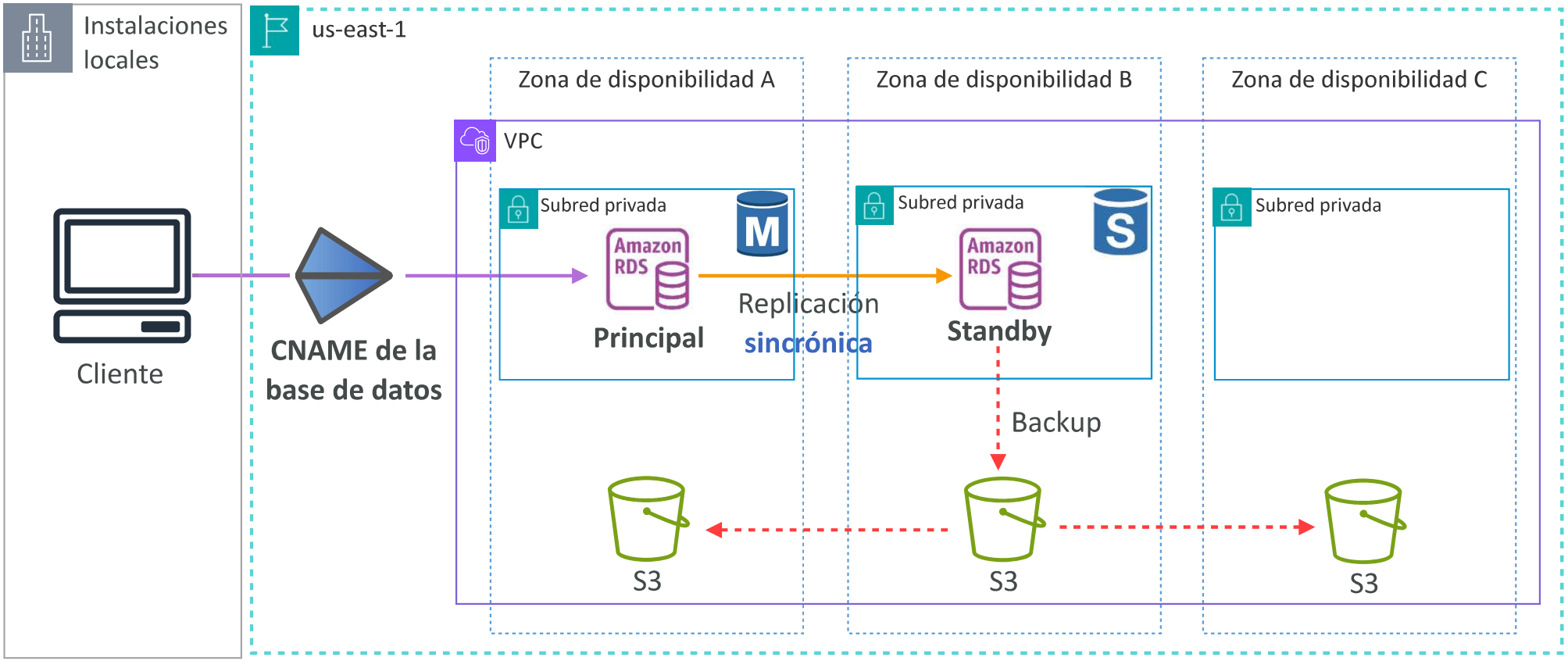
RDS Multi AZ - **Con una instancia en espera**

RDS Multi AZ - **Con dos instancias legibles**

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

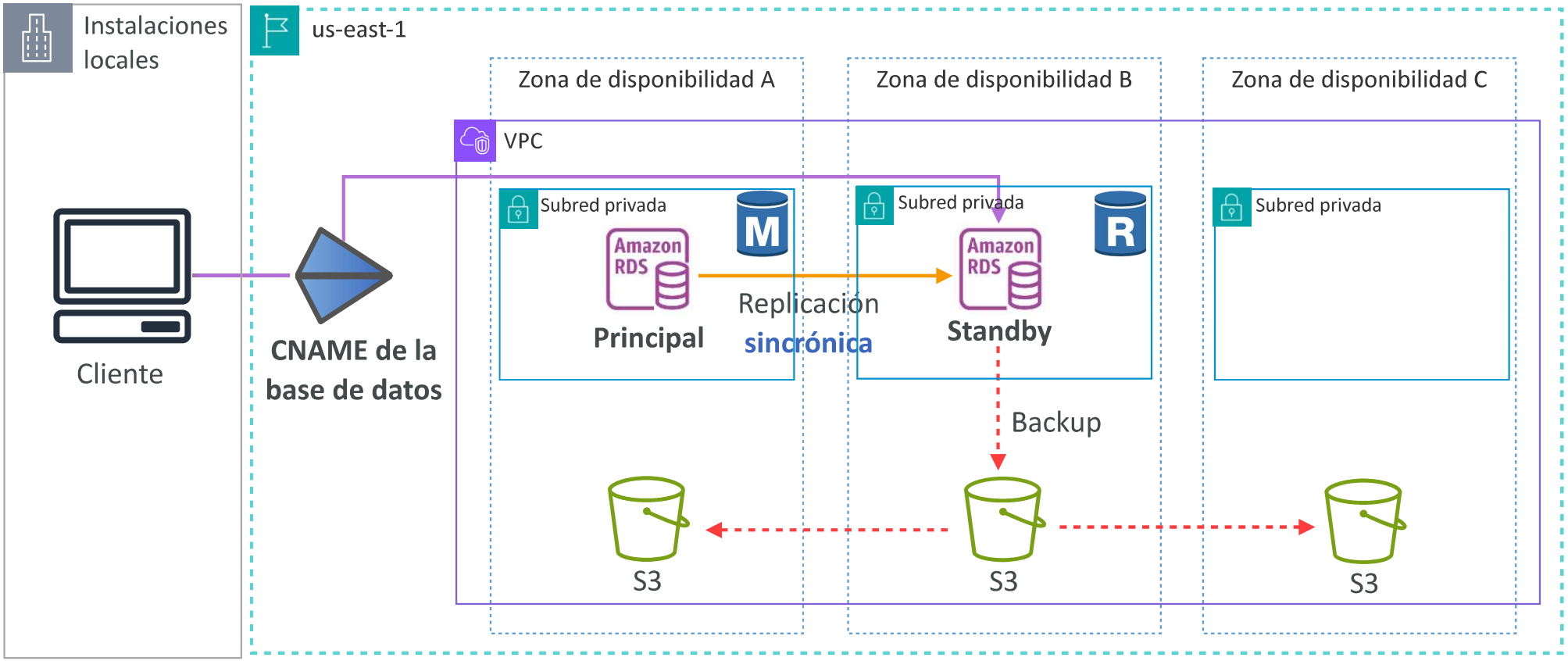
RDS Multi AZ - Con una instancia en espera



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

RDS Multi AZ - Con una instancia en espera



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

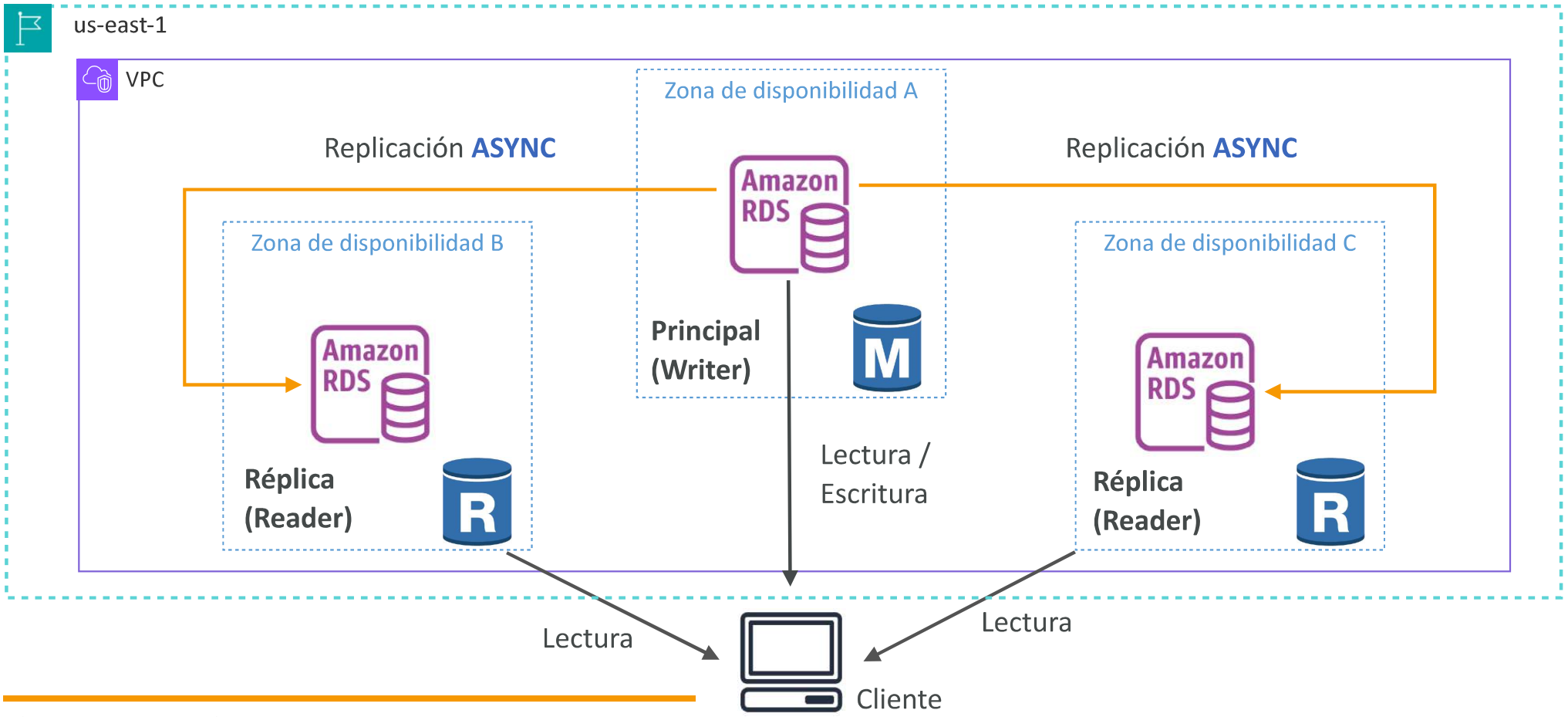
RDS Multi AZ - Con una instancia en espera

- Los datos se replican de la instancia primaria a la instancia de reserva (standby) de **forma sincrónica**
- Este servicio **no es parte de la capa gratuita** (hay un costo adicional por la réplica)
- **Solo se permite una réplica de reserva (StandBy)**
- La réplica de reserva no se puede utilizar para lecturas o escrituras
- El tiempo de failover es de 60 a 120 segundos
- Solo se permite **en la misma región**, pero en diferentes zonas de disponibilidad (AZ) dentro de esa región
- Las copias de seguridad (backups) se toman desde la instancia de reserva para mejorar el rendimiento
- Un fallo en la zona de disponibilidad, un fallo primario, un failover manual, cambios en el tipo de instancia y parches de software son algunos de los escenarios cubiertos

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

RDS Multi AZ - Con dos instancias en espera legibles



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

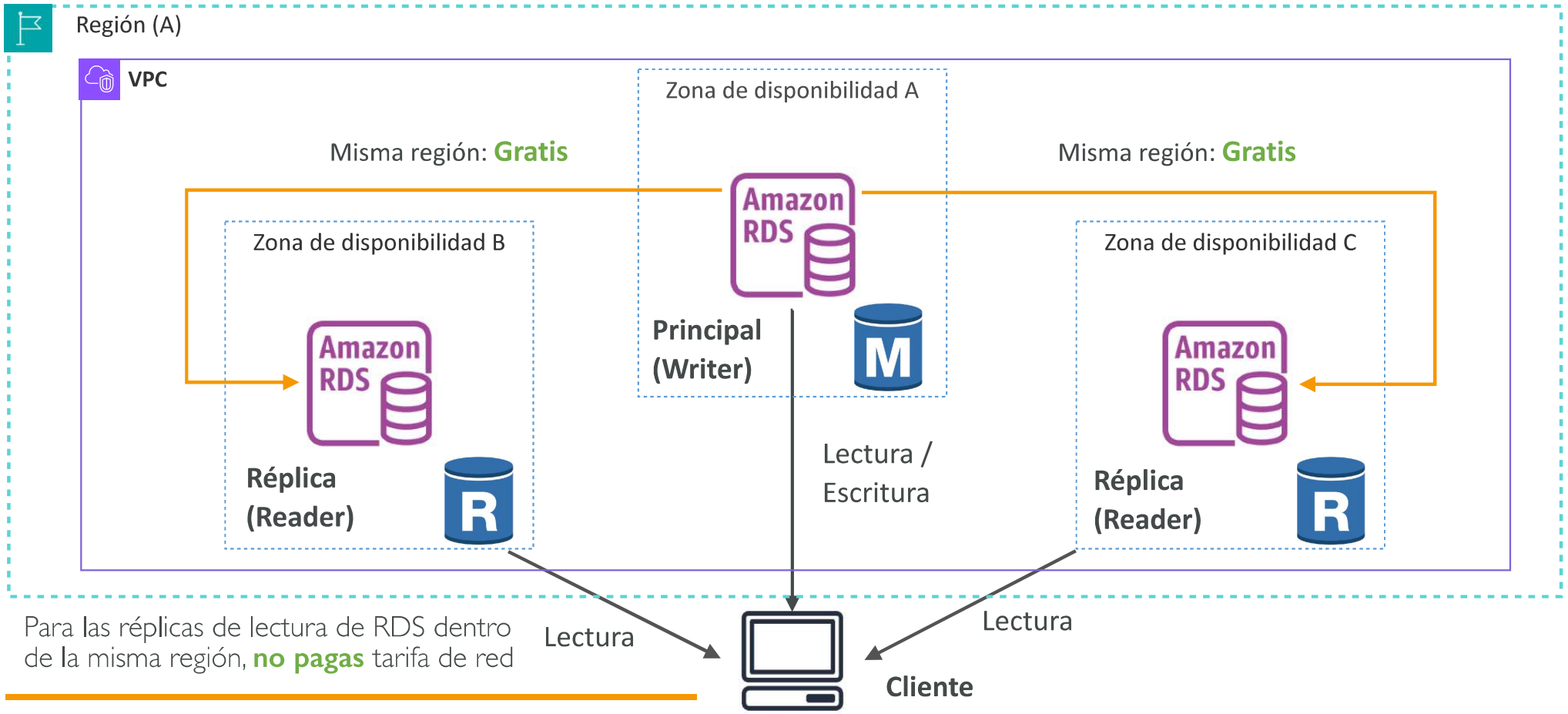
RDS Multi AZ - Con dos instancias en espera legibles

- Un escritor y dos instancias de base de datos lectoras en distintas zonas de disponibilidad (AZs)
- Escrituras rápidas al almacenamiento local => se transfieren a EBS
- Los lectores pueden ser utilizados para lecturas, permitiendo cierta escalabilidad de lectura
- La replicación se realiza a través de registros de transacciones, lo que resulta más eficiente
- El failover es más rápido, aproximadamente 35 segundos más la aplicación del registro de transacción
- Las escrituras se "confirman" cuando un lector ha confirmado

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Coste de la red (misma región)

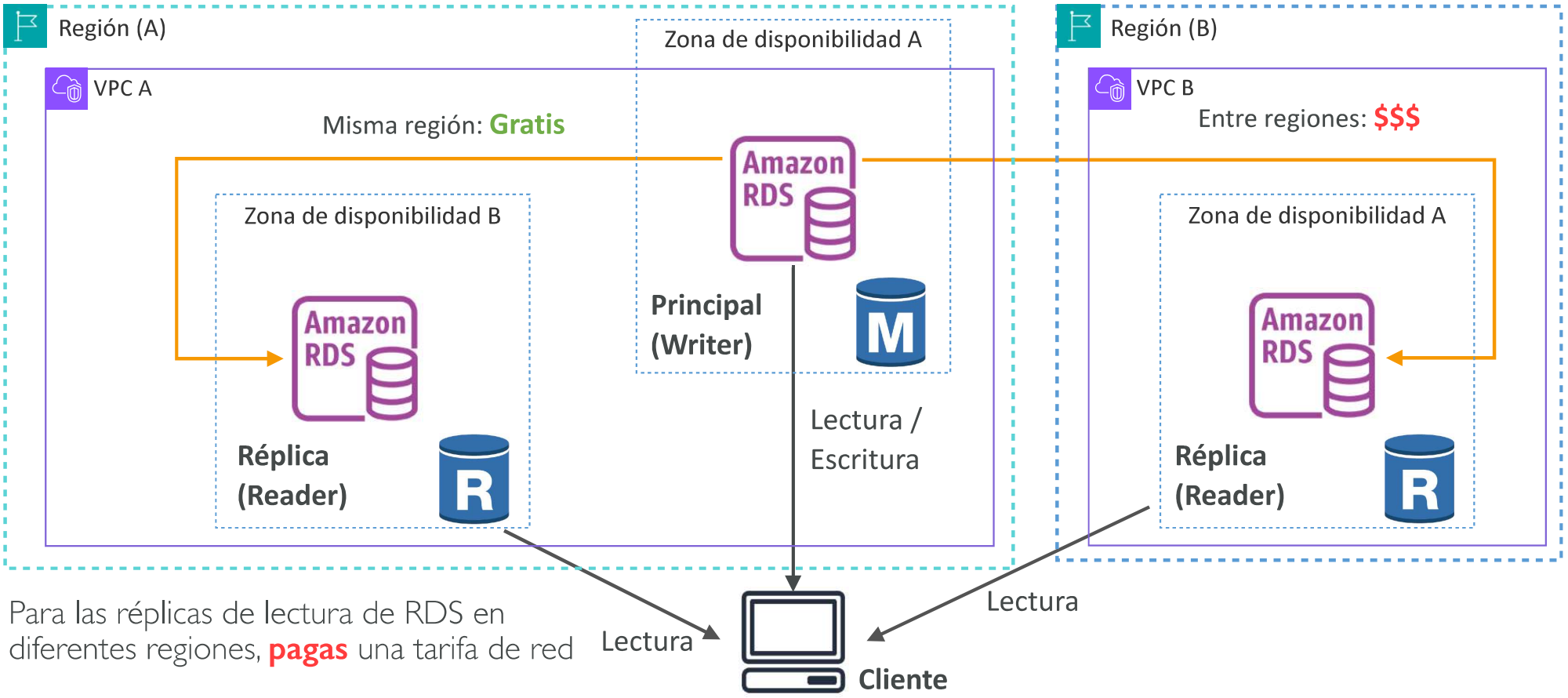


Para las réplicas de lectura de RDS dentro de la misma región, **no pagas** tarifa de red

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Coste de la red (diferentes regiones)



Para las réplicas de lectura de RDS en diferentes regiones, **pagas** una tarifa de red

© Joan Amengual

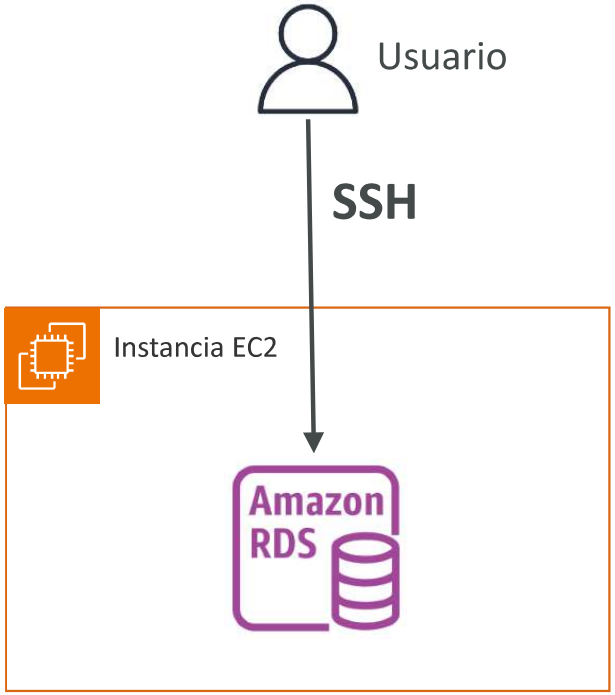
Todos los derechos reservados © Joan Amengual www.blockstellart.com

RDS Personalizado / Custom RDS

- **Amazon RDS** = Si necesitas que toda la base de datos y el sistema operativo estén completamente administrados por AWS
- **Amazon RDS Custom** = Si necesitas derechos administrativos sobre la base de datos y el sistema operativo subyacente para que las aplicaciones dependientes estén disponibles

---

- Base de datos gestionada de **Oracle y Microsoft SQL Server** con personalización del sistema operativo y de la base de datos
- Acceder a la instancia EC2 subyacente mediante **SSH** o **SSM Session Manager**
- **Desactivar el modo de automatización** para realizar tu personalización (mejor tomar una snapshot de la BD antes)
- RDS vs. RDS Custom:
  - RDS: toda la base de datos y el SO serán gestionados por AWS
  - RDS Custom: acceso administrativo completo al SO subyacente y a la base de datos

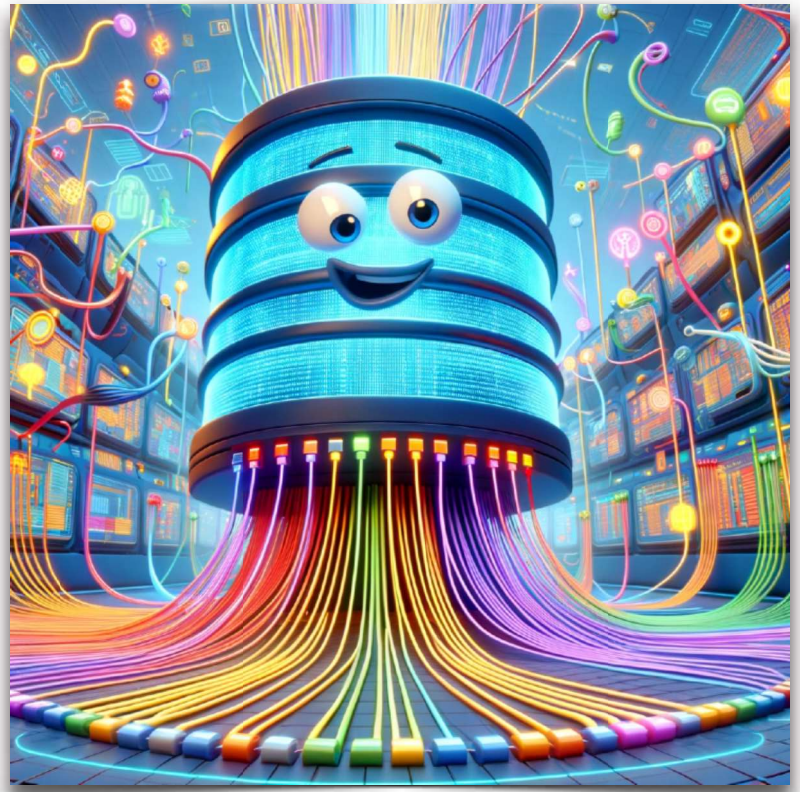


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Proxy de Amazon RDS

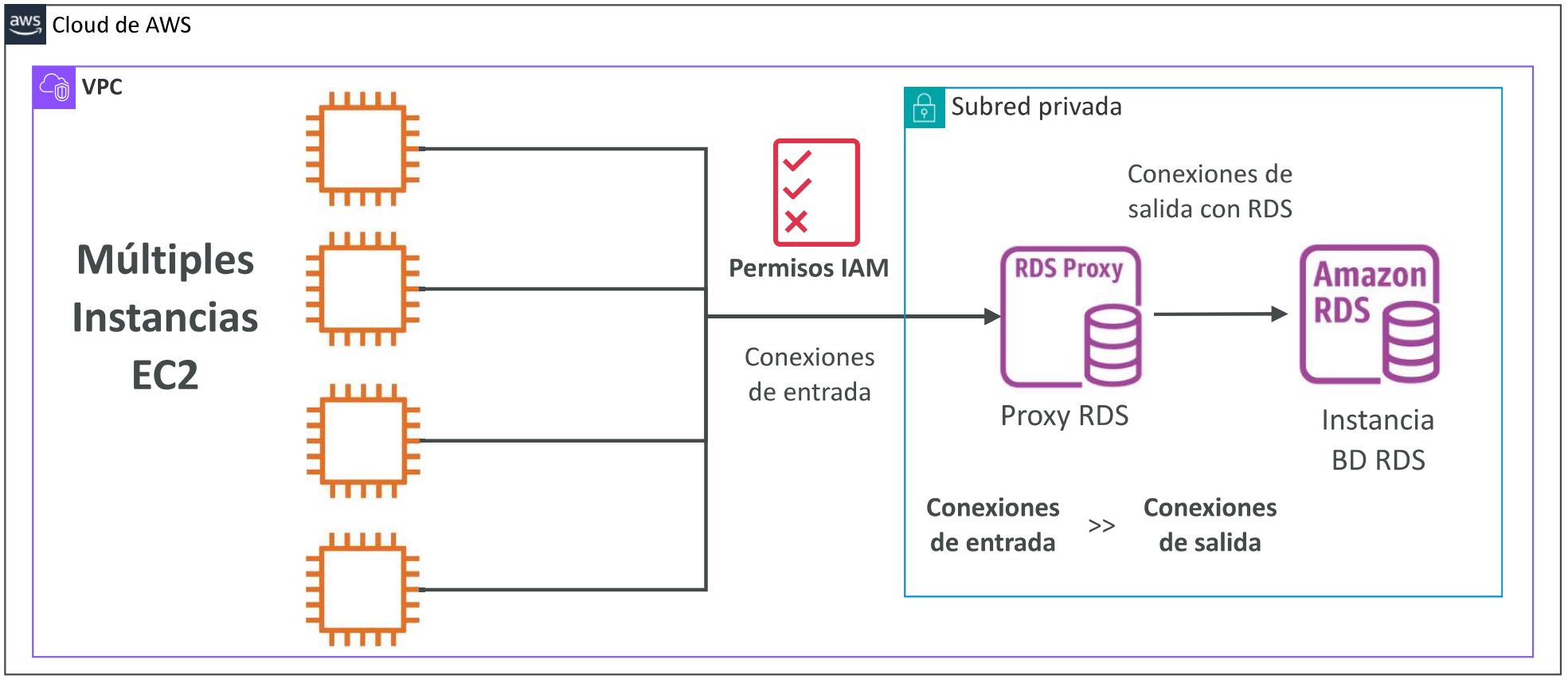
- Proxy de base de datos totalmente gestionado para RDS
- Permite a las apps **agrupar y compartir las conexiones a la base de datos** establecidas
- **Mejora la eficiencia de la base de datos reduciendo el estrés de los recursos de la base de datos (por ejemplo, CPU, RAM) y minimizando las conexiones abiertas (y los tiempos de espera)**
- Reduce el tiempo de conmutación por error de RDS y Aurora hasta en un 66%
- No se requieren cambios de código para la mayoría de las aplicaciones
- **Aplica los permisos IAM para la base de datos**
- **El proxy RDS nunca es accesible al público (debe accederse desde la VPC)**
- **Soporta RDS (MySQL, PostgreSQL, MariaDB) y Aurora (MySQL, PostgreSQL)**



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Proxy de Amazon RDS



© Joan Amengual

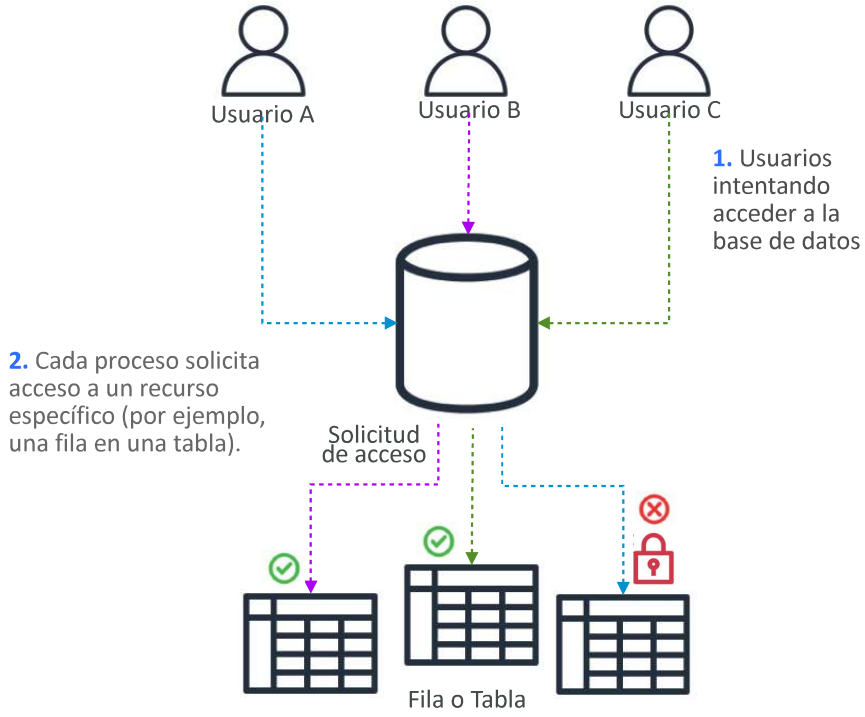
Todos los derechos reservados © Joan Amengual www.blockstellart.com

Uso del comando LOCK

🚨⚠️ Las bases de datos relacionales implícitamente **"bloquean" tablas o filas** para evitar que escriban en ellas al mismo tiempo, o que se lea mientras un proceso de escritura está en curso

Tipos de bloqueos:

1. **Bloqueos compartidos (S LOCK):** Permite a múltiples usuarios leer un elemento bloqueado, pero impide que cualquiera lo modifique mientras el bloqueo esté activo. Pueden ser mantenidos por múltiples transacciones. (*FOR SHARE*)
2. **Bloqueos exclusivos (X LOCK):** Impide que cualquier otro usuario o proceso lea o escriba en el elemento bloqueado hasta que el bloqueo sea liberado. Solo una transacción puede mantener un bloqueo exclusivo. (*FOR UPDATE*)



3. Evaluación del Bloqueo:
- Si no hay bloqueos existentes que impidan el acceso solicitado, el sistema otorga el bloqueo (ya sea compartido o exclusivo).
- Si hay un bloqueo incompatible (por ejemplo, un proceso solicita un bloqueo exclusivo mientras otro tiene un bloqueo compartido en el mismo recurso), el proceso solicitante debe esperar.

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Ejemplos - MySQL

Bloquear una tabla completa:



```
LOCK TABLES employees WRITE;
```

Bloquea la tabla completa 'employees' para operaciones de escritura

Para Liberar el bloqueo



```
UNLOCK TABLES
```

Nota: Redshift también tiene un comando LOCK para el mismo propósito.

Bloqueo compartido (permite lecturas, previene otras escrituras durante esta transacción):



```
SELECT * FROM employees WHERE department = 'Finance' FOR SHARE;
```

Bloqueo exclusivo (previene todas las lecturas y escrituras durante esta transacción):



```
SELECT * FROM employees WHERE employee_id = 123 FOR UPDATE;
```

Asegúrate de que las transacciones y los bloqueos estén completos, o podrías terminar con un "bloqueo mutuo" (deadlock)

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Directrices operativas de Amazon RDS



Monitoreo

Usa CloudWatch para monitorear memoria, CPU, almacenamiento, y latencia de réplica.



Backups

Realiza copias de seguridad automáticas durante los momentos de baja escritura en IOPS.



Multi-AZ & Conmutación por error

1. Establece el TTL en DNS para tus instancias de DB a 30 segundos o menos desde tus aplicaciones.
2. Prueba de conmutación por error antes de que lo necesites.



Performance

Una I/O insuficiente hará que la recuperación después de un fallo sea lenta.

1. Migra a una instancia de DB con más I/O.
2. Mueve a General Purpose o IOPS provisionadas.



Gestión de recursos

Provee suficiente RAM para incluir todo tu conjunto de trabajo.

1. Si tu métrica de ReadIOPS es pequeña y estable, estás bien.



Protección de datos

Los límites de tasa en Amazon API Gateway pueden usarse para proteger tu base de datos.

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

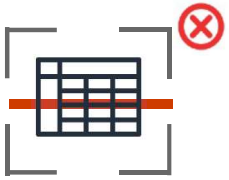
Optimización de consultas en RDS

Utiliza índices para acelerar las sentencias SELECT

- Utiliza planes EXPLAIN para identificar los índices que necesitas.



Evita los escaneos completos de tablas



Utiliza ANALYZE TABLE periódicamente



Optimizaciones específicas del motor



Simplifica las cláusulas WHERE



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Ajustes específicos para bases de datos

MySQL MariaDB



Mantén las tablas bien por debajo de 16TB, idealmente menos de 100GB.



Asegúrate de tener suficiente RAM para sostener los índices de las tablas usadas activamente.



Intenta tener menos de 10,000 tablas.

InnoDB

Usa InnoDB como motor de almacenamiento.

PostgreSQL



Al cargar datos, desactiva las copias de seguridad de la base de datos y Multi-AZ. Ajusta varios parámetros de la base de datos como maintenance_work_mem, max_wal_size, checkpoint_timeout. Desactiva synchronous_commit, activa autovacuum y asegura que las tablas estén registradas.



Utiliza autovacuum.

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Ajustes específicos para bases de datos

Microsoft SQL Server



Utiliza los Eventos de BD de RDS para monitorear failovers.



No habilites el modo de recuperación simple, modo fuera de línea, o modo solo lectura (esto rompe Multi-AZ).



Implementa en todas las Zonas de Disponibilidad (AZ's).

ORACLE

Oracle es una bestia propia. 

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com



Amazon Aurora

www.blockstellart.com

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Visión general de Amazon Aurora



- Aurora es una tecnología propietaria de AWS (no de código abierto)
- Tanto Postgres como MySQL se soportan como base de datos de Aurora (eso significa que tus controladores funcionarán como si Aurora fuera una base de datos Postgres o MySQL)
- Aurora está "optimizada para el Cloud de AWS" y afirma que mejora 5 veces el rendimiento de MySQL en RDS, y más de 3 veces el rendimiento de Postgres en RDS
- El almacenamiento de Aurora crece automáticamente en incrementos de 10 GB, hasta 128 TB.
- Aurora puede usar réplicas de BD con un proceso de replicación muy rápido (retardo de réplica inferior a 10 ms)
- La conmutación por error en Aurora es instantánea
- Aurora cuesta más que RDS (un 20% más), pero es más eficiente

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Alta disponibilidad y escalado de lectura de Aurora

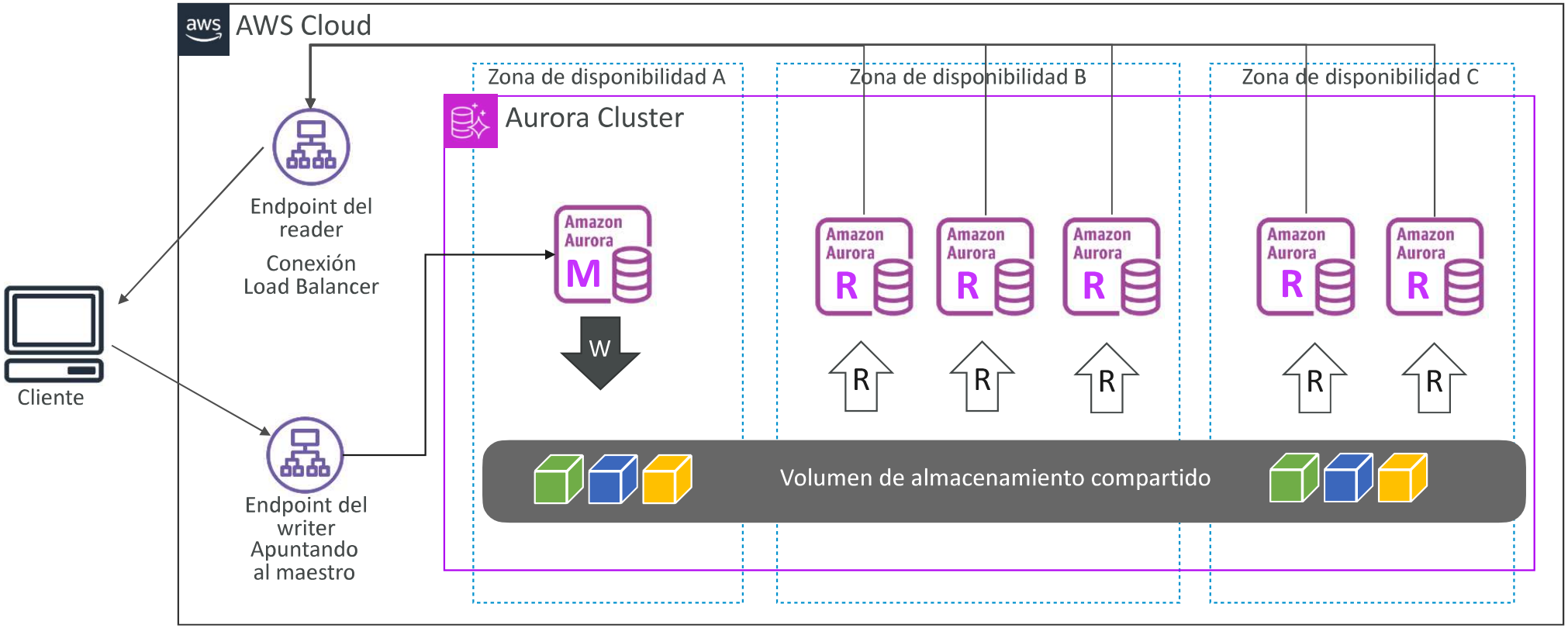
- 6 copias de tus datos en 3 AZ:
  - 4 copias de las 6 necesarias para las escrituras
  - 3 copias de las 6 necesarias para las lecturas
  - Autoreparación con replicación entre pares
  - El almacenamiento está dividido en 100 volúmenes
- Una instancia de Aurora se encarga de las escrituras (maestra)
  - Master + Réplicas de lectura
- Recuperación automática del master en menos de 30 segundos



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Cluster de BD Aurora



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Características de Aurora

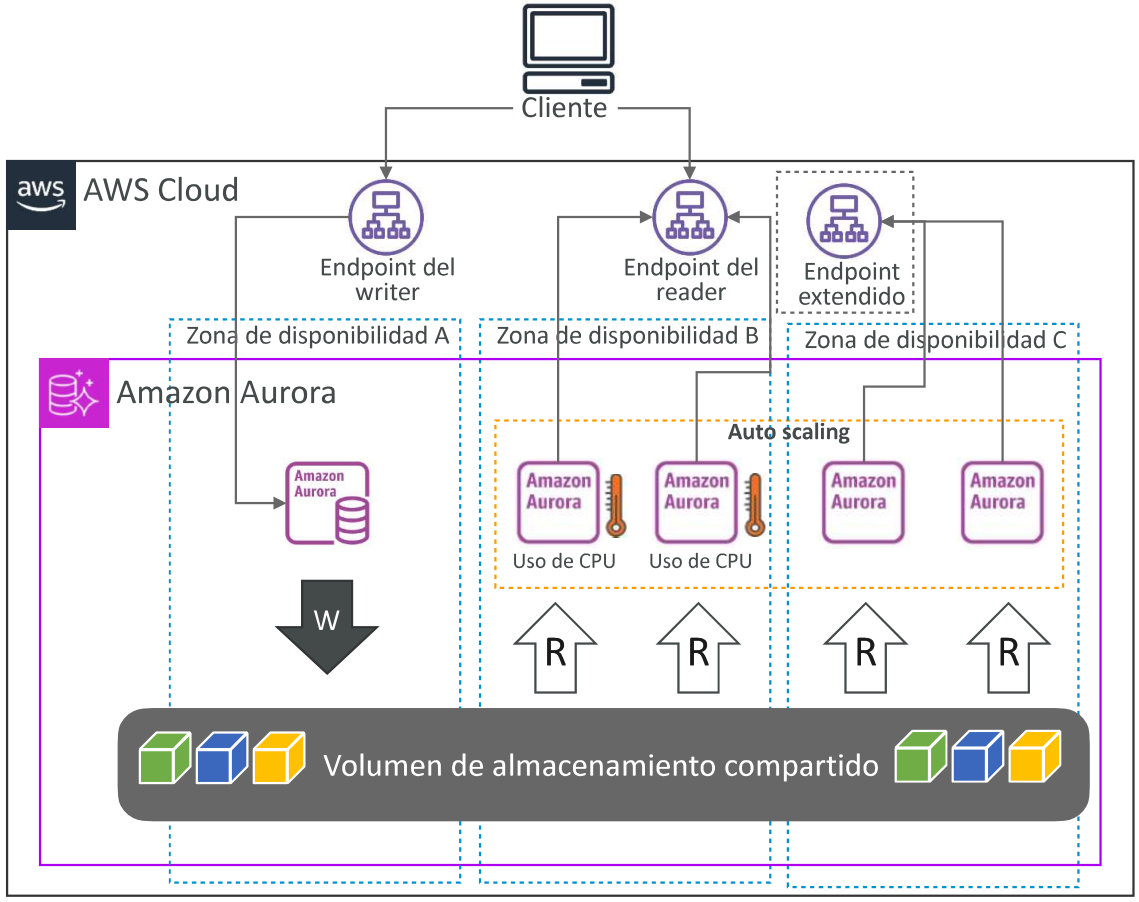
- Conmutación automática por error
- Copia de seguridad y recuperación
- Aislamiento y seguridad
- Cumplimiento de la normativa del sector
- Escalado con un botón
- Parches automáticos con cero tiempo de inactividad
- Supervisión avanzada
- Mantenimiento rutinario
- Restaura los datos en cualquier momento sin usar copias de seguridad

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Aurora - Auto Scaling de réplicas

- Ajusta automáticamente el número de réplicas de lectura en respuesta a cambios en la carga de trabajo
- Aurora distribuye automáticamente las solicitudes de lectura entre las réplicas de lectura activas, optimizando el rendimiento y reduciendo la latencia

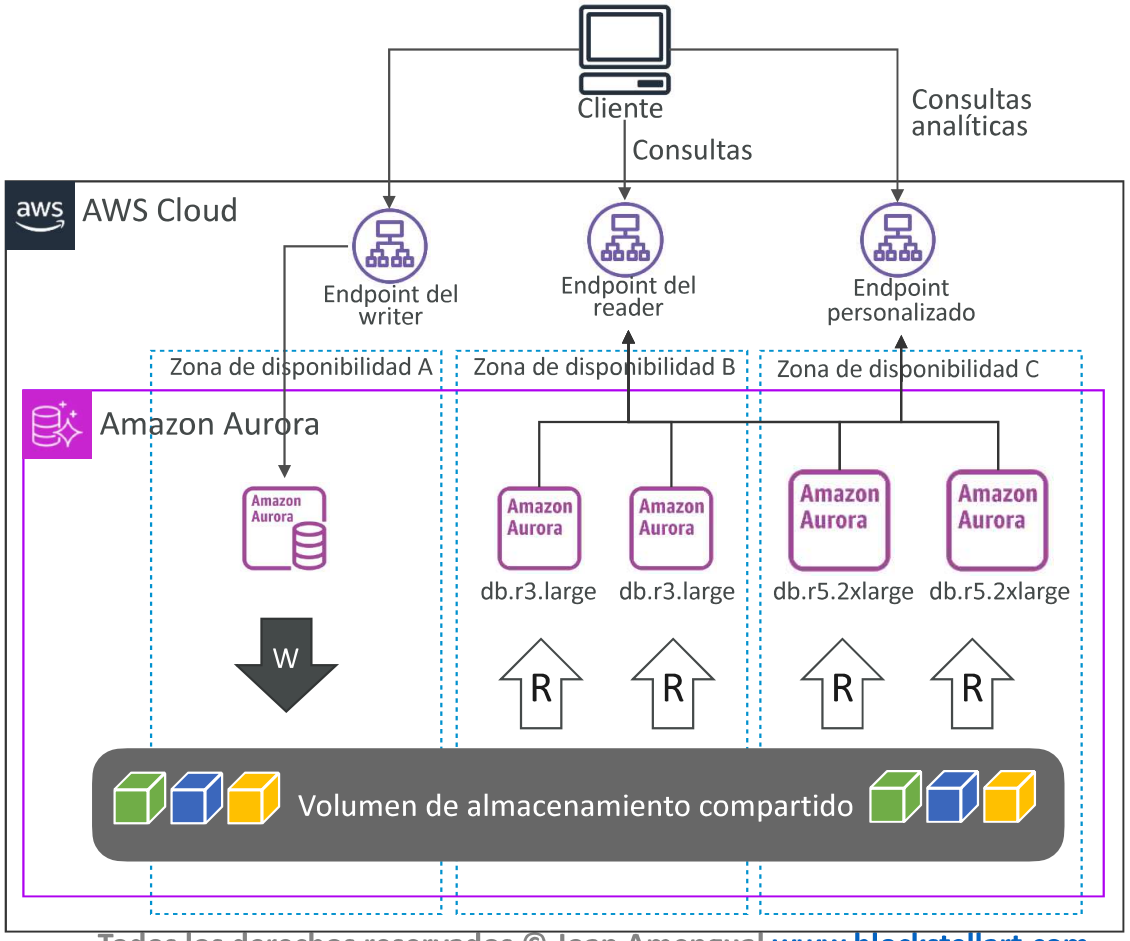


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Aurora - Endpoints personalizados

- Permiten configurar puntos de enlace personalizados para acceder a instancias de Aurora, facilitando la alta disponibilidad y la tolerancia a fallos
- Puedes dirigir el tráfico de lectura a réplicas específicas y el tráfico de escritura a la instancia principal, optimizando el rendimiento de la base de datos
- Ofrecen la flexibilidad de crear diferentes puntos de enlace para distintos tipos de operaciones (lectura, escritura, etc.), según las necesidades de la aplicación

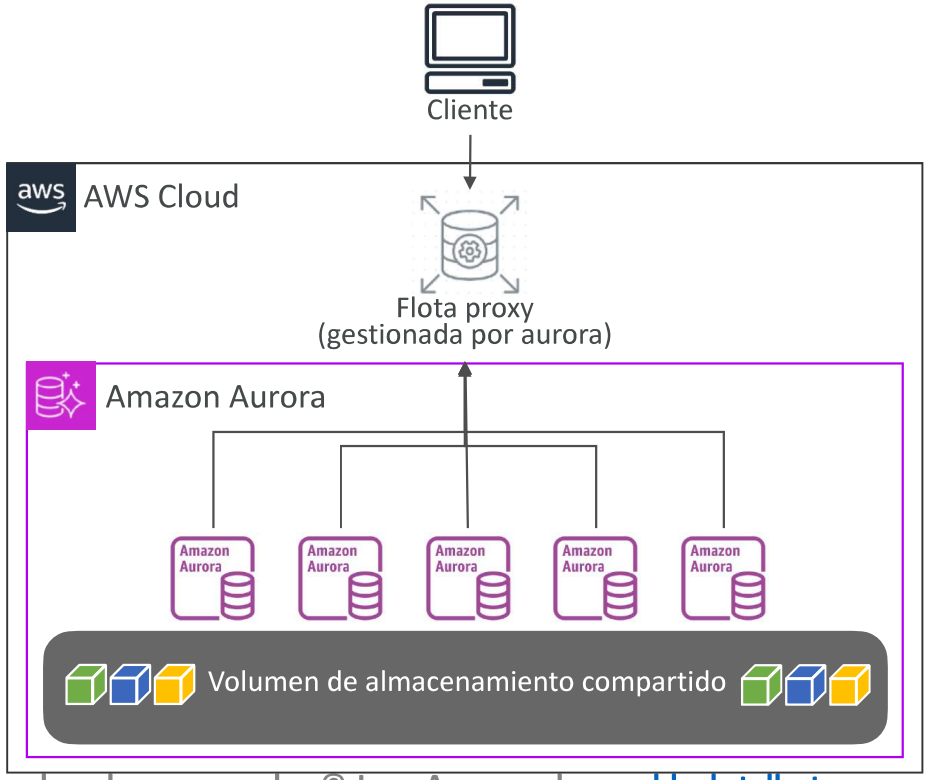


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Aurora - Sin servidor (serverless)

- Ajusta automáticamente la capacidad de la base de datos **en función de la demanda de la aplicación**, eliminando la necesidad de gestión manual
- **Sólo pagas por los recursos que realmente utilizas**, lo que puede reducir significativamente los costos en comparación con las instancias tradicionales
- Puede iniciarse, detenerse y escalarse automáticamente **según los patrones de uso** de la aplicación, optimizando la utilización de recursos
- Elimina la necesidad de gestionar la infraestructura subyacente



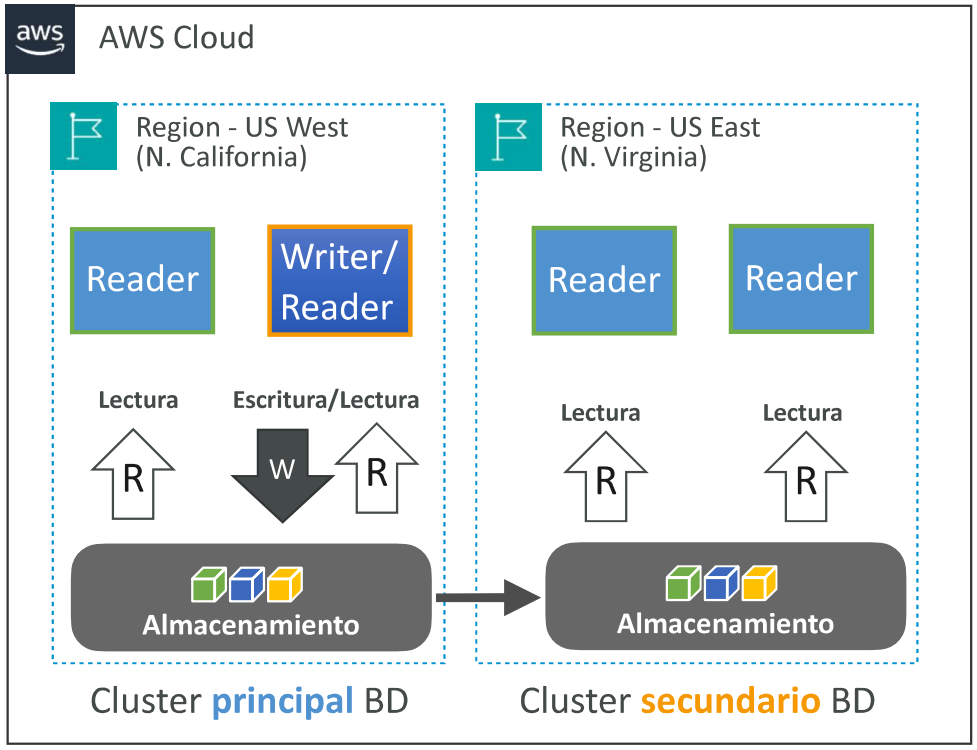
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Aurora - Bases de datos globales

- Las bases de datos globales de Amazon Aurora abarcan múltiples regiones de AWS, lo que permite lecturas globales de baja latencia y proporcionan una recuperación rápida de cualquier interrupción

- **Clúster principal** =1 Región Principal (lectura/escritura)
- **Clústeres secundarios** = Regiones secundarias (solo lectura), la latencia de replicación es de menos de 1 segundo
- Réplicas de lectura por región secundaria
- Ayuda a disminuir la latencia
- Promover otra región (para recuperación ante desastres) tiene un RTO de menos de 1 minuto
- La replicación típica entre regiones toma menos de 1 segundo

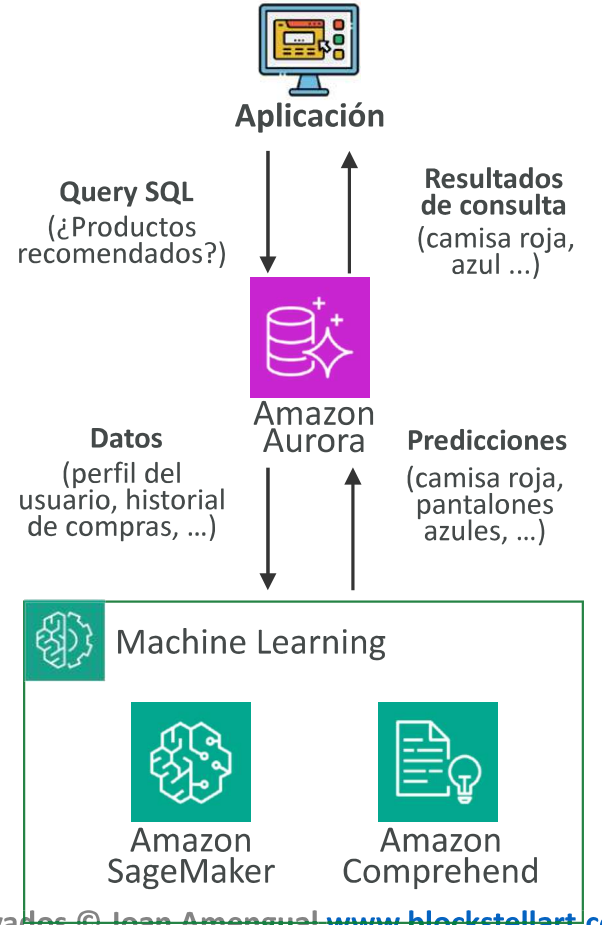


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Aurora - Machine Learning

- Aurora permite utilizar SQL para agregar predicciones basadas en ML sin necesidad de aprender nuevas herramientas ni tener experiencia previa en ML
- Integración optimizada y segura con los servicios de ML de AWS, como SageMaker, Comprehend y Bedrock
- Ideal para aplicaciones en tiempo real y de baja latencia, como detección de fraudes y recomendaciones de productos
- Gestión segura de datos con cifrado de extremo a extremo y control de acceso mediante AWS IAM
- Casos de uso: Recomendaciones personalizadas de productos ,servicios al cliente mejorados, etc.



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Aurora - Copias de seguridad

- Amazon Aurora realiza **copias de seguridad automáticas del cluster de la base de datos** sin impactar el rendimiento de las operaciones de la base de datos
- Permiten la recuperación punto a punto, pudiendo restaurar la base de datos a cualquier segundo dentro del periodo de retención, usualmente **hasta los últimos 35 días**
- Aurora almacena solo los cambios en los bloques de datos desde la última copia de seguridad, optimizando el uso del espacio y reduciendo los costos



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

RDS y Aurora - Opciones de restauración

- **Restaurar una copia de seguridad de RDS / Aurora o una instantánea crea una nueva base de datos**

- **Restauración de base de datos RDS MySQL desde S3**
  - Crea una copia de seguridad de tu base de datos local
  - Guárdala en Amazon S3 (almacenamiento de objetos)
  - Restaura el archivo de copia de seguridad en una nueva instancia de RDS que ejecute MySQL

- **Restauración de cluster Aurora MySQL desde S3**
  - Crea una copia de seguridad de tu base de datos local usando Percona XtraBackup
  - Almacena el archivo de copia de seguridad en Amazon S3
  - Restaura el archivo de copia de seguridad en un nuevo clúster de Aurora que ejecute MySQL



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Aurora - Clonación de base de datos

- **Crea un nuevo clúster de Aurora a partir de uno existente**
- Más rápido que hacer una snapshot y restaurar
- Utiliza el protocolo de copia - escritura
  - Inicialmente, el nuevo clúster de DB utiliza el mismo volumen de datos que el clúster DB original (rápido y eficiente — no se necesita copia)
  - Cuando se realizan actualizaciones en los datos del nuevo clúster DB, se asigna almacenamiento adicional y los datos se copian para separarlos
- Útil para crear una base de datos de "pruebas" a partir de una base de datos de "producción" sin impactar la base de datos de producción



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

RDS y Aurora - Seguridad

- **Cifrado en reposo:**
  - Cifrado de la base de datos maestra y de las réplicas mediante AWS KMS - debe definirse en el momento del lanzamiento.
  - Si la base maestra no está cifrada, las réplicas de lectura no pueden ser cifradas
  - Para cifrar una base de datos no cifrada, pasa por un Snapshot de la base de datos y restaura como cifrada
- **Cifrado en vuelo:** Preparado para TLS por defecto, utiliza certificados root del lado del cliente de AWS TLS
- **Autenticación IAM:** Roles de IAM para conectarse a tu BD (en lugar de username/password)
- **Grupos de seguridad:** Controla el acceso de red a tu RDS / Aurora DB
- **No hay SSH disponible** excepto en RDS Custom
- **Los logs de auditoría pueden ser activados** y enviados a CloudWatch Logs

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

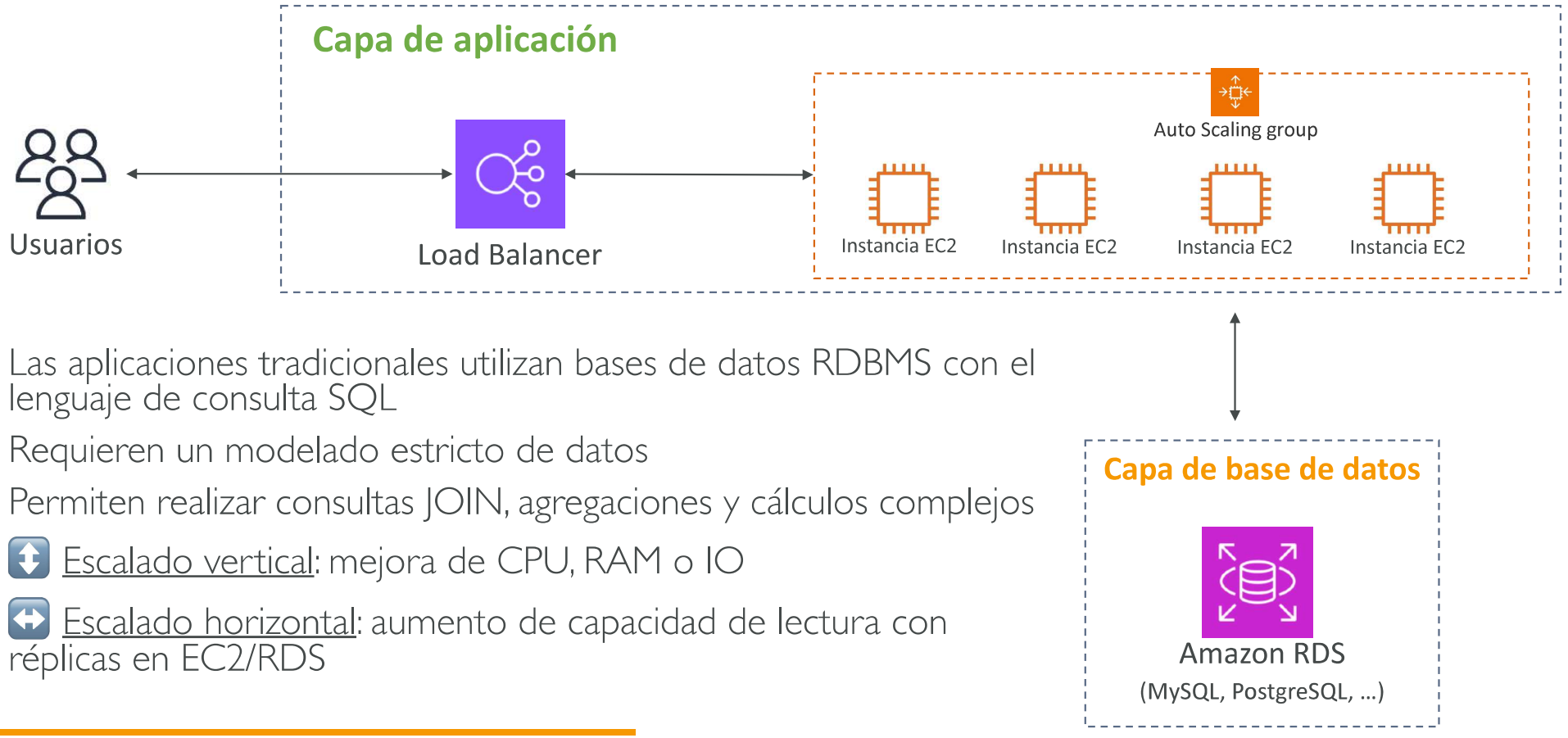


Amazon DynamoDB

www.blockstellart.com

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Diseño tradicional de las arquitecturas



- Las aplicaciones tradicionales utilizan bases de datos RDBMS con el lenguaje de consulta SQL
- Requieren un modelado estricto de datos
- Permiten realizar consultas JOIN, agregaciones y cálculos complejos
- Escalado vertical: mejora de CPU, RAM o IO
- Escalado horizontal: aumento de capacidad de lectura con réplicas en EC2/RDS

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Bases de datos NoSQL

- **Bases de datos NoSQL**
  - No son relacionales y están distribuidas
  - Ejemplos: MongoDB, DynamoDB, Cassandra, Redis
  - No soportan consultas "JOIN"
  - Todos los datos necesarios para una consulta suelen estar en una fila
  - No realizan agregaciones como "SUM", "AVG"
  - Escalan horizontalmente





DynamoDB





redis

- **Comparación con SQL**
  - No hay "correcto o incorrecto" entre NoSQL y SQL
  - Requieren modelar los datos de forma diferente
  - Es necesario pensar en las consultas de los usuarios de manera distinta

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Visión general de DynamoDB



- Totalmente gestionada y altamente disponible con replicación MultiAZ
- Base de datos NoSQL (no relacional)
- Escala para cargas de trabajo masivas
- Soporta millones de solicitudes por segundo, trillones de filas y cientos de TB de almacenamiento
- Rendimiento rápido y consistente
- Integrada con IAM para seguridad, autorización y administración
- Clases de tablas de acceso estándar y acceso poco frecuente (IA)

zoom

Zoom Video Communications, Inc. administró el incremento de 10 millones a 300 millones de participantes diarios en reuniones

Snap Inc. redujo la mediana de latencia en un 20 % con Amazon DynamoDB

Dropbox

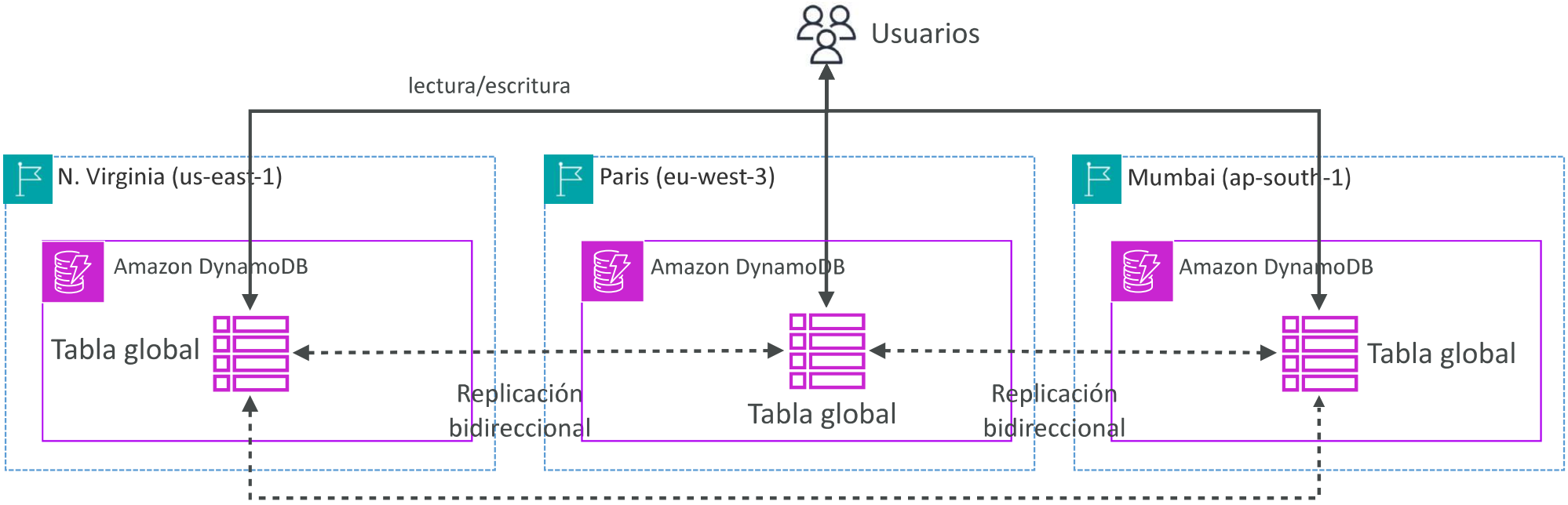
Dropbox se ahorró millones de dólares en costes de expansión con un nuevo sistema de almacenamiento

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Tablas globales de DynamoDB

- Haz que una tabla de DynamoDB sea accesible con **baja latencia** en varias regiones
- Replicación **activa-activa** (**lectura/escritura** en cualquier región de AWS)



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Conceptos básicos

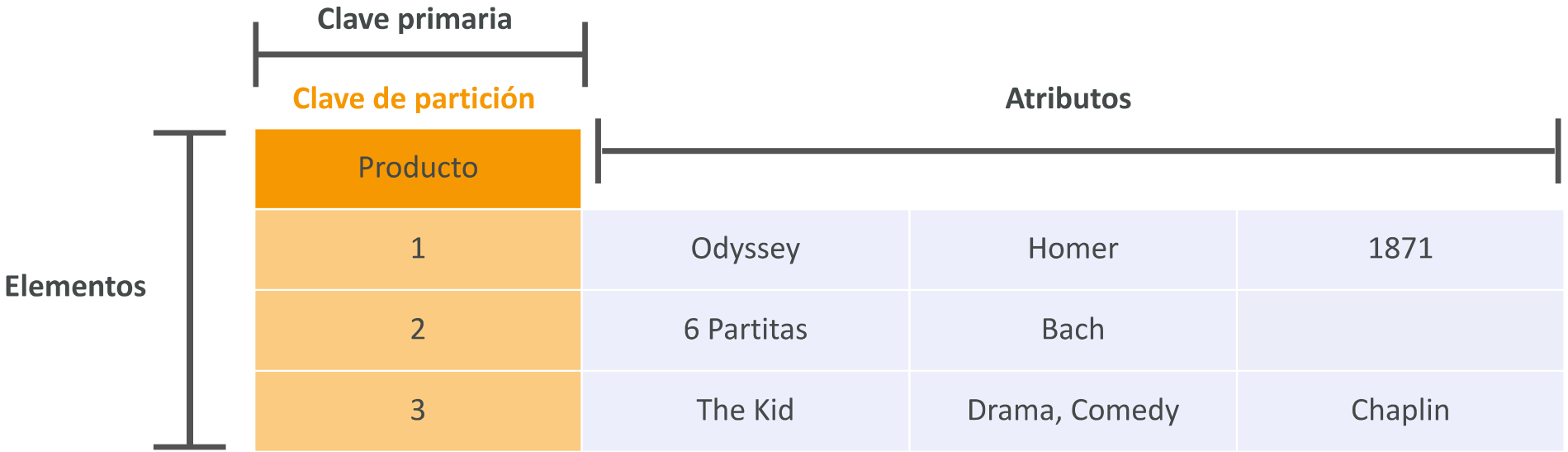
- DynamoDB está formado por **Tablas**
- Almacenan datos organizados en filas, cada una con una **estructura de clave-valor**
- Cada tabla tiene una **Clave Primaria** (se define al crear la tabla)
  - Esencial para cada tabla, establece la unicidad de cada elemento
- Cada tabla puede tener un número infinito de elementos (= filas)
- Cada elemento tiene **atributos**
  - Los atributos pueden ser nulos, permitiendo flexibilidad en los datos almacenados
- El tamaño máximo de un elemento es de **400 KB**

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Claves primarias

- **Opción 1: Clave de partición (HASH)**
  - La clave de partición debe ser única para cada elemento
  - La clave de partición debe ser "diversa" para que los datos estén distribuidos

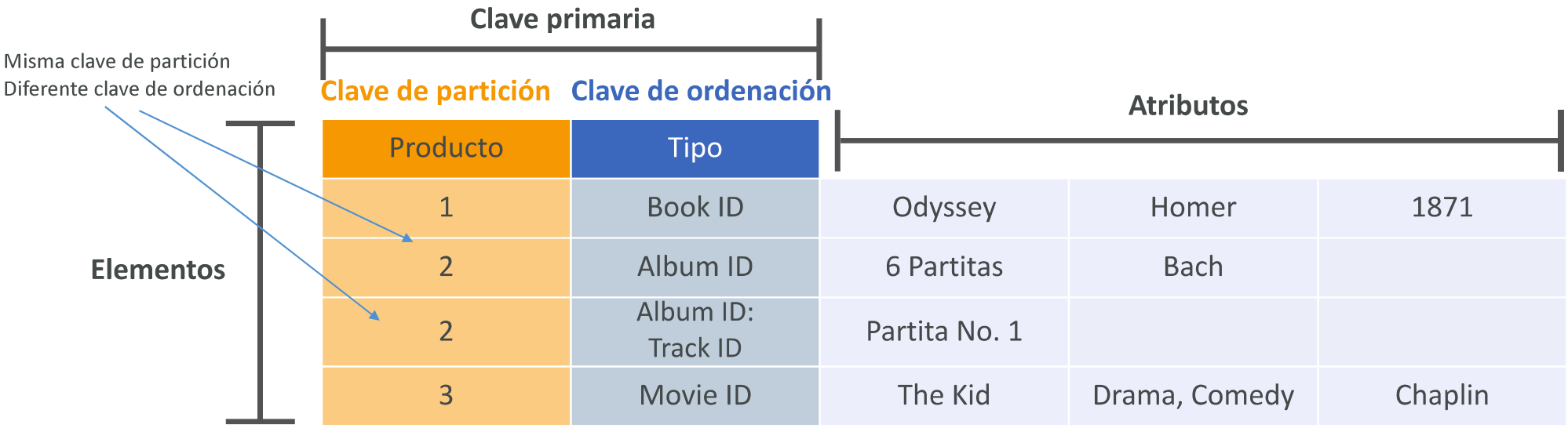


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Claves primarias

- **Opción 2: Clave de partición + Clave de ordenación / clasificación (HASH + RANGO)**
  - La combinación debe ser única para cada elemento
  - Los datos se agrupan por clave de partición



Producto	Tipo			
1	Book ID	Odyssey	Homer	1871
2	Album ID	6 Partitas	Bach	
2	Album ID: Track ID	Partita No. 1		
3	Movie ID	The Kid	Drama, Comedy	Chaplin

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Claves de partición (Ejercicio)

- ¿Cuál es la mejor clave de partición para maximizar la distribución de los datos?
  - movie_id
  - producer_name
  - leader_actor_name
  - movie_language
  - movie_budget
  - movie_genre

- **"movie_id"** tiene la cardinalidad más alta, por lo que es un buen candidato
- **"movie_language"** no admite muchos valores y puede estar sesgado hacia el inglés, por lo que no es una buena opción para la clave de partición



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB
Modos de capacidad de lectura/escritura

Modo aprovisionado (por defecto)

- Especificas el número de lecturas/escrituras por segundo
- Es necesario planificar la capacidad de antemano
- Pagas por unidades de capacidad de lectura (RCU) y unidades de capacidad de escritura (WCU) provisionadas
- Posibilidad de añadir el modo de autoescalado para RCU y WCU

Modo bajo demanda

- Las lecturas/escrituras aumentan/disminuyen automáticamente con tus cargas de trabajo
- No es necesario planificar la capacidad
- Pagas por lo que utilizas, más caro ($$$)
- Ideal para cargas de trabajo impredecibles, picos repentinos pronunciados

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Modos de capacidad R/W - Aprovisionado

- La tabla debe tener **aprovisionadas** unidades de capacidad de lectura y escritura
  - **Unidades de Capacidad de Lectura (RCU)** - rendimiento para lecturas
  - **Unidades de Capacidad de Escritura (WCU)** - rendimiento de escritura
- Opción para configurar el **escalado automático** del rendimiento
- El rendimiento puede superarse temporalmente utilizando la **"Capacidad de ráfaga" (Burst Capacity)**
- Si se ha consumido la capacidad de ráfaga, obtendrás una "**ProvisionedThroughputExceededException**"
- En ese caso, se aconseja hacer un reintento de **retroceso exponencial**

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Modos de capacidad R/W - Bajo demanda

- Las lecturas/escrituras **aumentan/disminuyen automáticamente** con tus cargas de trabajo
- No necesitas planificar la capacidad (WCU / RCU)
- 🚨⚠️ **2,5 veces más caro** que la capacidad aprovisionada (¡con cuidado!)
- WCU y RCU ilimitadas, sin acelerador, más caras
- Te cobran por las lecturas/escrituras que utilizas en términos de RRU y WRU
- **Unidades de petición de lectura (RRU)** - rendimiento de las lecturas (igual que RCU)
- **Unidades de petición de escritura (WRU)** - rendimiento de las escrituras (igual que WCU)
- Casos de uso:
  - Cargas de trabajo desconocidas
  - Tráfico de aplicación impredecible, ...

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB – Unidades de capacidad de escritura (WCU)

- Una Unidad de Capacidad de Escritura (WCU) representa una escritura por segundo para un elemento de hasta **1 KB** de tamaño
- 1 WCU permite realizar una escritura por segundo para un ítem de hasta 1 KB
- Si los elementos son mayores de 1 KB, se consumen más WCUs
- Para ítems que excedan 1 KB, el número de WCUs necesarias **aumenta en proporción al tamaño del ítem**. Por ejemplo, un ítem de 2 KB requerirá 2 WCUs por escritura por segundo
- El **tamaño del ítem se redondea al próximo KB** completo para calcular las WCUs. Un ítem de 1.5 KB se tratará como un ítem de 2 KB



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB – Unidades de capacidad de escritura (WCU)

- **Escribimos 20 elementos por segundo, con un tamaño de elemento de 3 KB**

SOLUCIÓN

$$\text{Necesitamos } 20 * (\frac{3\ KB}{1\ KB}) = 60\ WCUs$$

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB – Unidades de capacidad de escritura (WCU)

- **Escribimos 6 elementos por segundo, con un tamaño de elemento de 6,5 KB**

SOLUCIÓN

Necesitamos $6 * (\frac{7\ KB}{1\ KB}) = 42\ WCUs$ (6,5 se redondea al KB superior)

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB – Unidades de capacidad de escritura (WCU)

- **Escribimos 180 elementos por minuto, con un tamaño de elemento de 2 KB**

SOLUCIÓN

$$\text{Necesitamos } \left(\frac{180}{60}\right) * \left(\frac{2\ KB}{1\ KB}\right) = 6\ WCUs$$

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

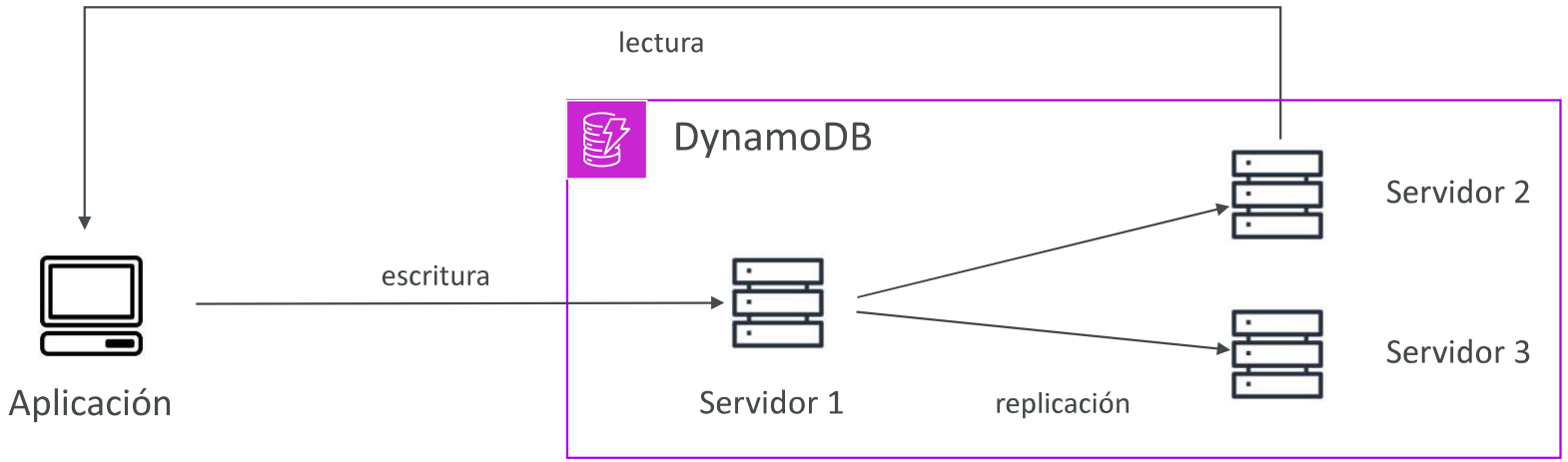
Lectura eventualmente consistente vs. Lectura fuertemente consistente

Lectura eventualmente consistente (por defecto)

- Si leemos justo después de una escritura, es posible que obtengamos algunos datos obsoletos debido a la replicación

Lectura fuertemente consistente

- Si leemos justo después de una escritura, obtendremos los datos correctos
- Pon el parámetro **"ConsistentRead"** a **True** en las llamadas a la API (GetItem, BatchGetItem, Query, Scan)
- Consume el doble de RCU



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB – Unidades de capacidad de lectura (RCUs)

- Una Unidad de Capacidad de Lectura (RCU) representa **una lectura fuertemente consistente** por segundo, o **dos lecturas eventualmente consistentes** por segundo, para un elemento de hasta **4 KB** de tamaño
- Para ítems que superen los 4 KB, se necesitarán múltiples RCUs para una sola lectura, dependiendo del tamaño del ítem.



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB – Unidades de capacidad de lectura (RCUs)

- **16 Lecturas eventualmente consistentes por segundo, con un tamaño de elemento de 12 KB**

SOLUCIÓN

Necesitamos $\left(\frac{16}{2}\right) * \left(\frac{12\ KB}{4\ KB}\right) = 24\ RCUs$

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB – Unidades de capacidad de lectura (RCUs)

- **10 lecturas fuertemente consistentes por segundo, con un tamaño de elemento de 4 KB**

SOLUCIÓN

Necesitamos $10 * \left(\frac{4\ KB}{4\ KB} \right) = 10\ RCUs$

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB – Unidades de capacidad de lectura (RCUs)

- **10 lecturas fuertemente consistentes por segundo, con un tamaño de elemento de 6 KB**

SOLUCIÓN

$$\text{Necesitamos } 10 * \left(\frac{8\ KB}{4\ KB}\right) = 20\ RCUs$$

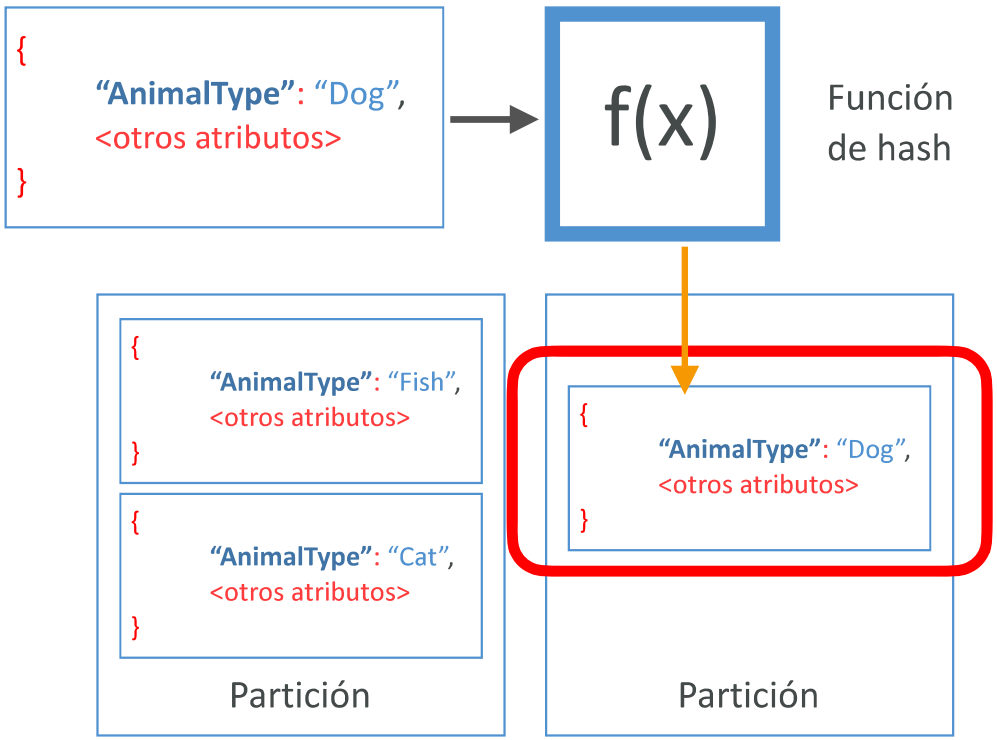
- *Debemos redondear 6 KB a 8 KB*

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Particiones y distribuciones

- Los datos se almacenan en particiones
- Las particiones se diseñan para distribuir las operaciones de manera uniforme entre ellas, evitando cuellos de botella y mejorando el rendimiento global
- Para agregar un elemento, DynamoDB aplica una función hash
- El resultado de la función hash determina en qué partición se almacenará el elemento
- Los elementos no se almacenan de forma ordenada; su ubicación depende del valor hash de su clave de partición



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Particiones y distribuciones

- Para calcular el número de particiones:
  - $\#particiones_{capacidad} = \left(\frac{RCUs_{Total}}{3000}\right) + (\frac{WCUs_{Total}}{1000})$
  - $\#particiones_{tamaño} = \frac{TamañoTotal}{10GB}$
  - $\#particiones = \text{ceil}(\max\left(\#particiones_{capacidad}, \#particiones_{tamaño}\right))$
- **Las WCUs y RCUs se reparten uniformemente por las particiones**

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB – Throttling (Estrangulamiento)

- Si superamos las RCUs o WCUs aprovisionadas, obtenemos la siguiente excepción: "***ProvisionedThroughputExceededException***"
- 🤔 ¿Cuáles pueden ser los motivos?
  - **Claves calientes** - una clave de partición se lee demasiadas veces (ej: elemento popular)
  - **Particiones calientes**
  - **Elementos muy grandes**, recuerda que RCU y WCU dependen del tamaño de los elementos
- ✅ ¿Cómo se puede solventar el problema?:
  - **Retroceso exponencial** cuando se encuentra una excepción (disponible en el SDK)
  - **Distribuye las claves de partición** tanto como sea posible
  - Si hay problemas de RCU, podemos **utilizar el Acelerador de DynamoDB (DAX)**

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Escritura de datos

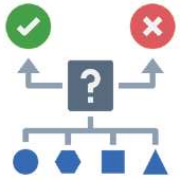
- **PutItem**
  - Crea o sustituye un elemento con la misma clave primaria
  - Consume Unidades de Capacidad de Escritura (WCU)



- **UpdateItem**
  - Modifica atributos de un elemento o añade uno nuevo si no existe
  - Útil para contadores atómicos que incrementan sin condiciones



- **Conditional Writes**
  - Realiza escrituras condicionales que solo proceden si se cumplen ciertas condiciones, sin impactar el rendimiento



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Lectura de datos

- **GetItem**
  - Lectura basada en clave primaria
  - La clave primaria puede ser **HASH** o **HASH+RANGE**
  - Lectura eventualmente consistente (por defecto)
  - Opción de utilizar lecturas fuertemente consistentes (puede tardar más)
  - Se puede especificar una **ProjectionExpression** para recuperar sólo determinados atributos



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Lectura de datos (Consulta)

- La consulta devuelve elementos basados en:
- **KeyConditionExpression**:
  - Clave de partición:
    - Utiliza el operador '=' (obligatorio)
  - Clave de ordenación:
    - Soporta '=', '<', '<=', '>', '>=', 'BETWEEN', 'BEGINS WITH' (opcional)
- **FilterExpression**:
  - Aplica filtros después de recuperar datos, solo para atributos no clave
- Resultados:
  - Devuelve hasta el número de elementos especificados en Limit o hasta 1 MB de datos.



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Lectura de datos (Escaneado)

- Escanea toda la tabla, después filtra (ineficiente)
- **Hasta 1 MB** de datos, usa paginación para más
- Controla el impacto con Limit o reduciendo el tamaño de resultados
- Mejora la velocidad usando múltiples "workers" para escanear segmentos simultáneamente, lo que aumenta el RCU usado pero no el impacto de los escaneos
- Usa **ProjectionExpression** y **FilterExpression** para eficiencia sin afectar RCU



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Eliminación de datos

- **DeleteItem**
  - Borrar un elemento individual
  - Posibilidad de realizar una eliminación condicional



- **DeleteTable**
  - Borrar una tabla entera y todos sus elementos
  - Eliminación mucho más rápida que llamar a **DeleteItem** en todos los ítems



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Operaciones por lotes

- Te permite ahorrar en latencia reduciendo el número de llamadas a la API
- Las operaciones se realizan en paralelo para una mayor eficiencia
- Parte de un lote puede fallar, en cuyo caso hay que volver a intentarlo con los elementos fallidos

- **BatchWriteItem**
  - Hasta 25 **PutItem** y/o **DeleteItem** en una misma llamada
  - Hasta 16 MB de datos escritos, hasta 400 KB de datos por elemento
  - No se pueden actualizar los elementos (utiliza **UpdateItem**)
  - **UnprocessedItems** para operaciones de escritura fallidas (retroceso exponencial o añadir WCU)

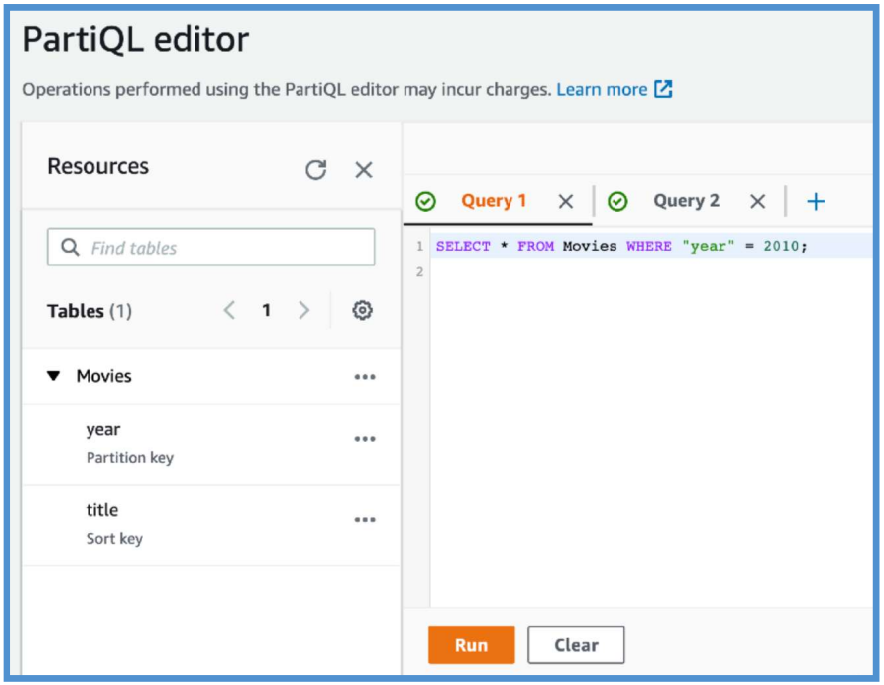
- **BatchGetItem**
  - Devuelve elementos de una o varias tablas
  - Hasta 100 elementos, hasta 16 MB de datos
  - Los elementos se recuperan en paralelo para minimizar la latencia
  - **UnprocessedKeys** para operaciones de lectura fallidas (retroceso exponencial o añadir RCU)

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB – PartiQL

- Facilita las consultas en **DynamoDB usando la sintaxis SQL**, soporta datos estructurados, semiestructurados y anidados
  - Optimiza el rendimiento y minimiza la latencia de las consultas
  - Se conecta con otros servicios de AWS para análisis avanzados
  - Ejecuta consultas PartiQL desde:
    - Consola de administración de AWS
    - NoSQL Workbench para DynamoDB
    - API de DynamoDB
    - AWS CLI
    - AWS SDK



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Escrituras condicionales

- Son operaciones que solo se ejecutan si cumplen con una serie de condiciones preestablecidas sobre los atributos de los datos

- Utilizan una sintaxis especial para definir las condiciones que deben cumplirse antes de que la operación de escritura pueda proceder, como *attribute_exists* o *attribute_not_exists*

- **Nota: Las expresiones de filtro filtran los resultados de las consultas de lectura, mientras que las expresiones de condición son para las operaciones de escritura**

```
params = {
    'ConditionExpression': (
        'attribute_not_exists(expirationDate) AND '
        'attribute_exists(address) AND '
        'attribute_type(details, :typeMap) AND '
        'quantity > :minQ AND '
        'NOT contains(historicalIds, :id) AND '
        'begins_with(details.address.street, :streetPrefix) AND '
        'details.address.zipCode = :zipCode AND '
        '(:val IN attribute_list) AND '
        'ProductCategory IN (:cat1, :cat2) AND '
        'price BETWEEN :low AND :high AND '
        'size(details.description) <= :maxSize AND '
        '(revenue - cost) > :profit'
    )}
```

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Escrituras condicionales - Ejemplo en actualizar elemento

Actualizaciones condicionales

```
aws dynamodb update-item \
    --table-name ProductCatalog \
    --key '{"Id": {"N": "456"}}' \
    --update-expression "SET Price = Price - :discount" \
    --condition-expression "Price > :limit" \
    --expression-attribute-values file://values.json
```

Valores iniciales

```
{
    "Id": {"N": "456" },
    "ProductCategory": {"S": "Sporting Goods" },
    "Price": {"N": "650" }
}
```

values.json

```
{
    ":discount": { "N": "75"},
    ":limit": {"N": "500"}
}
```

- Si el valor inicial de Price es **650**, la operación UpdateItem reduce el Price a **575**.
- Si ejecuta la operación UpdateItem de nuevo, el valor de Price se reduce a **500**.
- Si se ejecuta una tercera vez, la expresión de condición se evalúa en **false** y la actualización no se lleva a cabo.

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Escrituras condicionales - Eliminar elemento

- **attribute_not_exists**
  - Sólo tiene éxito si el atributo aún no existe (sin valor)

```
aws dynamodb delete-item \
    --table-name ProductCatalog \
    --key '{"Id": {"N": "456"}}' \
    --condition-expression "attribute_not_exists(Price)"
```

- **attribute_exists**
  - Opuesto a attribute_not_exists

```
aws dynamodb delete-item \
    --table-name ProductCatalog \
    --key '{"Id": {"N": "456"}}' \
    --condition-expression "attribute_exists(ProductReviews.OneStar)"
```

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Escrituras condicionales - Condición compleja

Condición avanzada

```
aws dynamodb delete-item \
    --table-name ProductCatalog \
    --key '{"Id":{"N":"456"}}' \
    --condition-expression "(ProductCategory IN (:cat1, :cat2)) and (Price between :lo and :hi)" \
    --expression-attribute-values file://values.json
```

Valores iniciales

```
{
    "Id": {"N": "456" },
    "ProductCategory": {"S": "Sporting Goods" },
    "Price": {"N": "650" }
}
```

- En este ejemplo, la comparación de ProductCategory se evalúa en true, pero la comparación de Price se evalúa en false. Esto hace que la expresión de condición se evalúe en false y, por consiguiente, la operación DeleteItem no se lleva a cabo.

values.json

```
{
    ":cat1": {"S": "Sporting Goods"},
    ":cat2": {"S": "Gardening Supplies"},
    ":lo": {"N": "500"},
    ":hi": {"N": "600"}
}
```

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Escrituras condicionales - Comprobación del valor inicial de cadena

- **begins_with** – comprobar si el prefijo coincide
- **contains** – comprobar si una string está contenida en otra string

Comprobación del valor inicial de cadena

```
aws dynamodb delete-item \
    --table-name ProductCatalog \
    --key '{"Id": {"N": "456"}}' \
    --condition-expression "begins_with(Pictures.FrontView, :v_sub)" \
    --expression-attribute-values file://expression-attribute-values.json
```

expression-attribute-values.json

```
{
    ":v_sub":{"S":"http://"}
}
```

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Índices secundarios

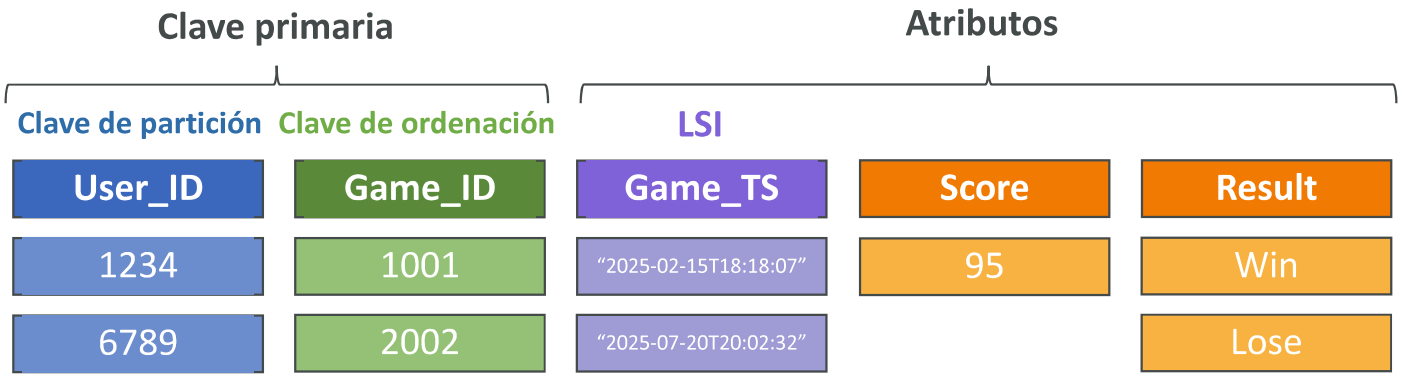
- Algunas aplicaciones solo necesitan consultar datos mediante la clave principal
- Sin embargo, puede haber situaciones en las que una **clave de ordenación alternativa** sería útil
- Para que la aplicación disponga de diversas claves de ordenación entre las que elegir, puedes **crear uno o varios índices secundarios** en una tabla de DynamoDB
- DynamoDB admite dos tipos de índices secundarios:
  - **Índice secundario local**: se considera "local" en el sentido de que el ámbito de todas sus particiones se corresponde con una partición de la tabla base que tiene el mismo valor de clave de partición
  - **Índice secundario global**: se considera "global" porque las consultas que se realizan en el índice pueden abarcar todos los datos de la tabla base y todas las particiones

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Índice Secundario Local (LSI)

- **Clave de ordenación alternativa** para tu tabla (misma **clave de partición** que tabla base)
- La clave de ordenación consiste en un atributo escalar (String, Number o Binary)



- Hasta 5 Índices Secundarios Locales (LSI) por tabla
- **Deben definirse al crear la tabla**
- **Proyecciones de atributos** - pueden contener algunos o todos los atributos de la tabla base **(KEYS_ONLY, INCLUDE, ALL)**

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Índice Secundario Global (GSI)

- **Clave primaria alternativa (HASH o HASH+RANGE)** de la tabla base
- Aceleran las consultas sobre atributos no clave, mejorando la eficiencia y velocidad de acceso a los datos
- **Proyecciones de atributos** - algunos o todos los atributos de la tabla base **(ALL, KEYS_ONLY, INCLUDE)**
- Es necesario asignar RCUs y WCUs adecuadas para mantener un rendimiento óptimo del índice.
- **Los GSIs no están restringidos por la clave primaria de la tabla base, permitiendo crear índices completamente independientes para consultas especializadas**
- Facilitan el acceso a grandes volúmenes de datos de manera eficiente



TABLE (consulta por "User_ID")



INDEX GSI (consulta por "Game_ID")

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Índices y estrangulamiento

- **Índice Secundario Local (LSI)**
  - Utiliza las WCUs y RCUs de la tabla principal
  - Sin consideraciones especiales de estrangulamiento
- **Índice Secundario Global (GSI)**
  - 🚨⚠️**Si se estrangulan las escrituras en el GSI, ¡se estrangulará la tabla principal!**
    - **Aunque las WCU de las tablas principales estén bien*
  - Recomendaciones:
    - Elige con cuidado tu clave de partición GSI
    - Asigna cuidadosamente la capacidad de tu WCU



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB Accelerator - DAX

- **Caché en memoria** totalmente gestionada para DynamoDB
- **Mejora del rendimiento x 10** - latencia de un milisegundo a microsegundos - al acceder a tus tablas de DynamoDB



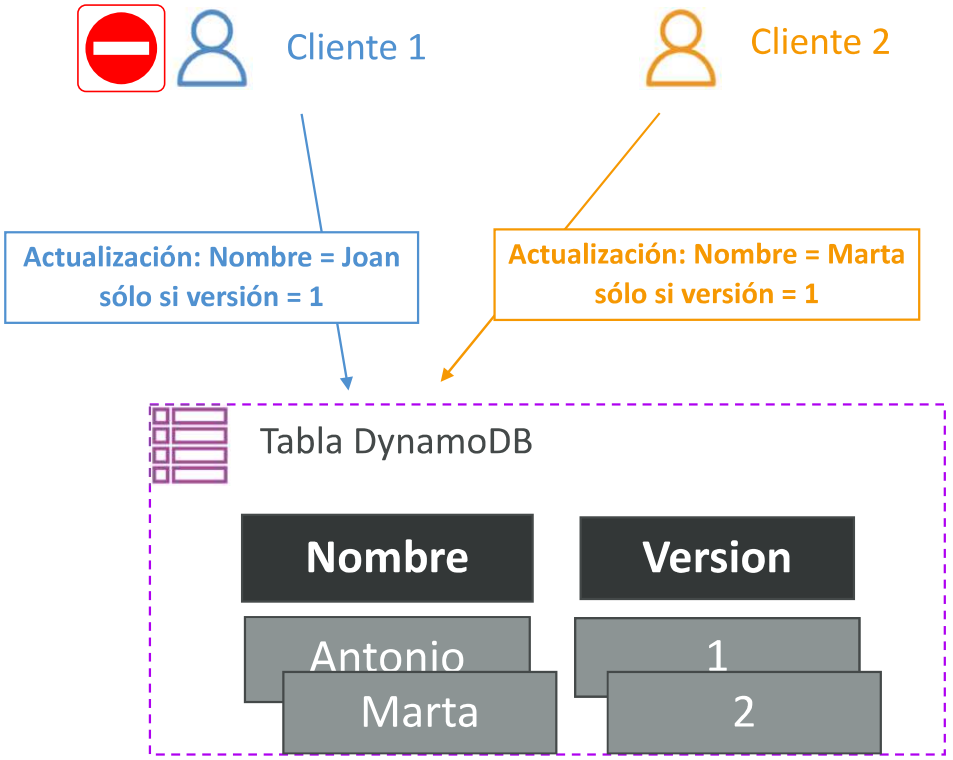
- Seguridad, alta escalabilidad y alta disponibilidad
- Diferencia con ElastiCache: **DAX sólo se utiliza y se integra con DynamoDB**, mientras que ElastiCache puede utilizarse para otras bases de datos

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Bloqueo optimista

- Estrategia para asegurarse de que el elemento del lado del cliente que se va a actualizar (o eliminar) sea el mismo que figura en Amazon DynamoDB
- Las escrituras en la base de datos se **protegen contra posibles sobrescrituras** de otros y viceversa
- Con el bloqueo optimista, cada elemento tiene un atributo que actúa como número de versión
- Si recuperas un elemento de una tabla, la aplicación registra el número de versión de ese elemento



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

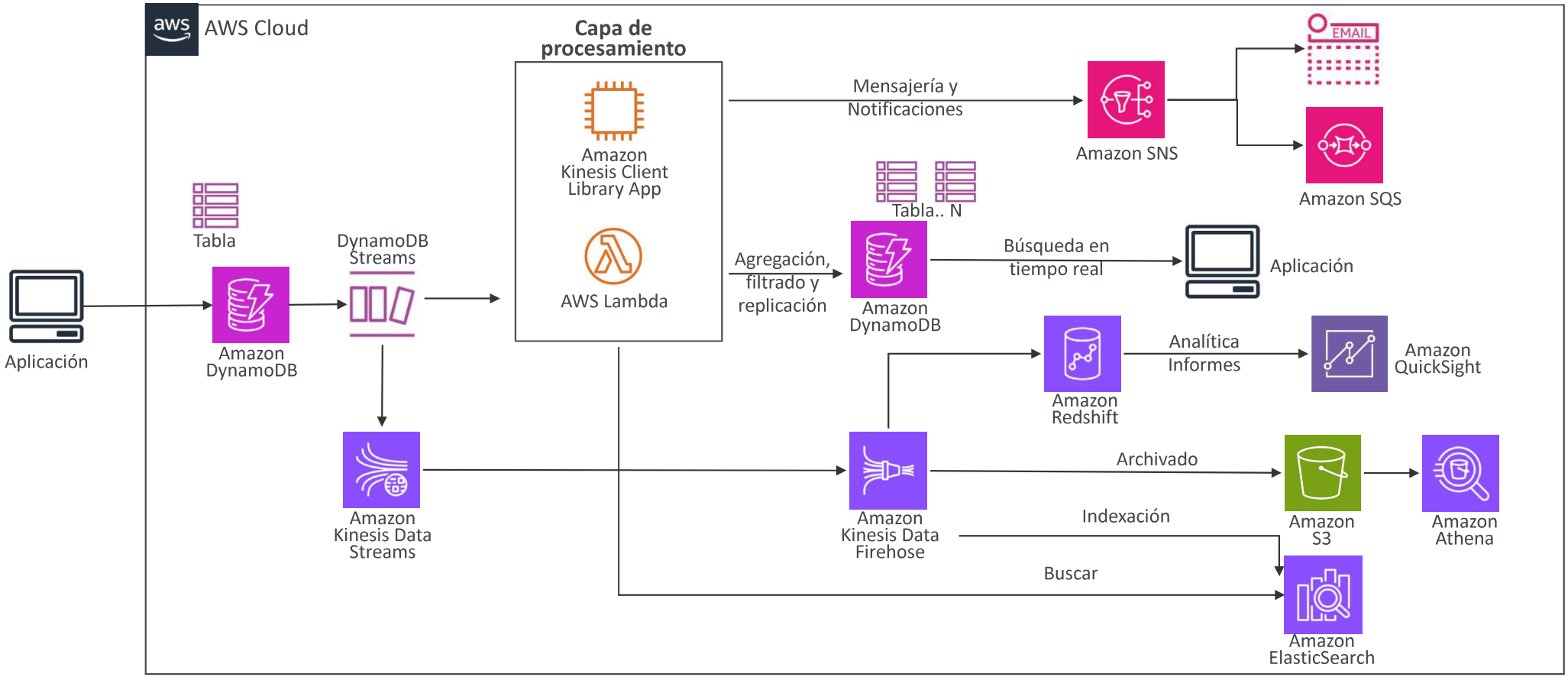
DynamoDB Streams

- Captura una **secuencia en orden cronológico de las modificaciones** (inserciones, actualizaciones y borrados) de los elementos en una tabla de DynamoDB
- Permite acceder a los datos de los cambios en los últimos **24 horas**, incluyendo el contenido de los elementos modificados
- Las aplicaciones pueden obtener acceso a este registro y ver los elementos de datos tal y como se encontraban antes y después de la modificación
- Facilita la creación de arquitecturas impulsadas por eventos,
- **Casos de uso**:
  - Reaccionar a los cambios en tiempo real (correo electrónico de bienvenida a los usuarios)
  - Sincronizar datos en tiempo real entre tablas de DynamoDB en diferentes regiones
  - Analíticas, activación de funciones Lambda, auditoría de cambios, etc…

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB Streams

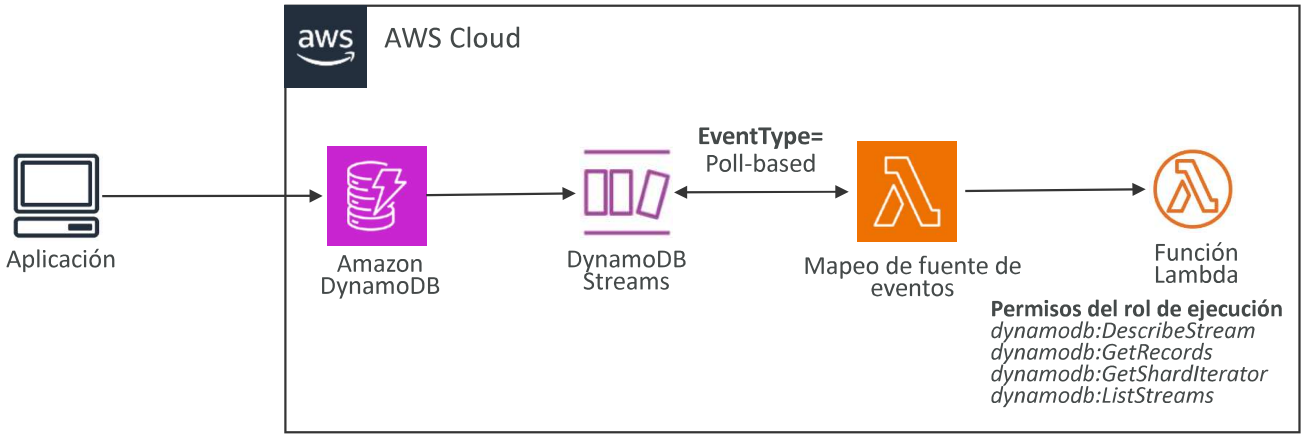


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB Streams y AWS Lambda

- Necesitas definir un **mapeo de fuente de eventos** para leer de un DynamoDB Streams
  - **El mapeo de fuente de eventos automáticamente invoca la función Lambda en respuesta a eventos específicos detectados en la fuente de eventos*
- Necesitas asegurarte de que la función Lambda tiene los **permisos adecuados**
- **Tu función Lambda se invoca de forma sincrónica**

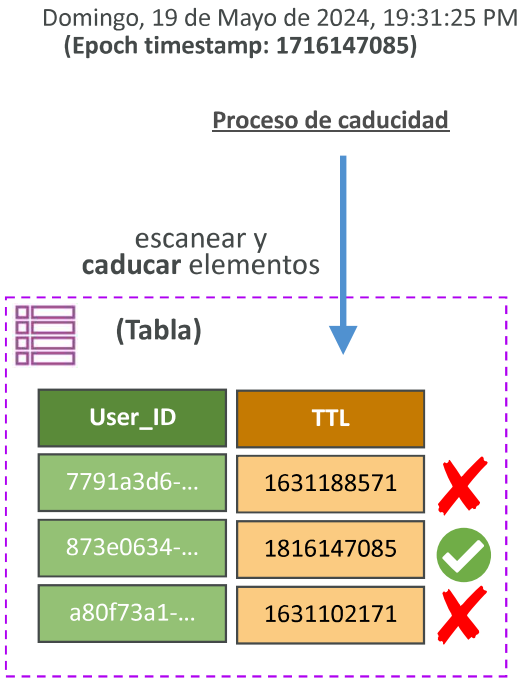



```
{
 "Records": [
  {
   "eventID": "1",
   "eventVersion": "1.0",
   "dynamodb": {
    "Keys": {
     "Id": {
      "N": "101"
     }
    },
    "NewImage": {
     "Message": {
      "S": "New item!"
     },
     "Id": {
      "N": "101"
     }
    },
    "StreamViewType": "NEW_AND_OLD_IMAGES",
    "SequenceNumber": "111",
    "SizeBytes": 26
   }
  }
 ]
}
```


Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB – Time To Live (TTL)

- Elimina automáticamente los elementos después de una fecha de caducidad
- No consume ninguna WCU (es decir, no tiene coste adicional)
- El atributo TTL debe ser un tipo de dato **“Number"** con valor **"Unix Epoch timestamp”**
- Los artículos caducados se eliminan en las 48 horas siguientes a su caducidad
- Los elementos caducados, que no se han borrado, aparecen en las lecturas/consultas/escaneos (si no los quieres, fíltralos)
- Casos de uso: reducir los datos almacenados conservando sólo los elementos actuales, cumplir las obligaciones normativas, ...



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

CLI de DynamoDB - Detalles relevantes

- Con la CLI de DynamoDB puedes ejecutar consultas y escaneos directamente desde la línea de comandos

- Por ejemplo, si deseamos escanear una tabla:

```
aws dynamodb scan --table-name TuNombreDeTabla
```

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

CLI de DynamoDB - Detalles relevantes

- Algunos detalles importantes son:
  - **--projection-expression**: uno o más atributos a recuperar
  - **--filter-expression**: filtra los elementos antes de devolvértelos

- Por ejemplo:

```
aws dynamodb scan \
    --table-name TuNombreDeTabla \
    --projection-expression "Atributo1, Atributo2" \
    --filter-expression “Atributo2 = 'ValorEspecifico'"
```

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

CLI de DynamoDB - Detalles relevantes

- Opciones generales de paginación de la CLI de AWS:
  - **--page-size**: especifica que la CLI de AWS recupere la lista completa de elementos, pero con un mayor número de llamadas a la API en lugar de una sola (por defecto: 1000 elementos)
  - **--max-items**: número máximo de elementos a mostrar en la CLI (devuelve **NextToken**)
  - **--starting-token**: especifica el último **NextToken** para recuperar el siguiente conjunto de elementos

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Transacciones DynamoDB

- Operaciones coordinadas de **todo o nada**
- Por ejemplo, **añadir/actualizar/eliminar** en varios elementos de una o varias tablas
- Proporciona atomicidad, consistencia, aislamiento y durabilidad (ACID)
- **Consume el doble de WCUs y RCUs**
  - DynamoDB realiza 2 operaciones por cada elemento (preparar y confirmar)
- Casos de uso:
  - Transacciones financieras
  - Gestión de pedidos
  - Juegos Multijugador



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Transacciones de DynamoDB - Cálculo de capacidad

- 🚨🚨🚨**¡MUY IMPORTANTE para el examen!**🚨🚨🚨
- **Ejemplo 1**: 5 escrituras transaccionales por segundo, con un tamaño de elemento de 4 KB

SOLUCIÓN

$$\text{Necesitamos } 5 * \left(\frac{4\ KB}{1\ KB}\right) * 2\ (transactional\ cost) = 40\ WCUs$$

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Transacciones de DynamoDB - Cálculo de capacidad

- 🚨🚨🚨**¡MUY IMPORTANTE para el examen!**🚨🚨🚨
- **Ejemplo 2**: 10 lecturas de transacciones por segundo, con un tamaño de elemento de 9 KB

SOLUCIÓN

Necesitamos $10 * \left(\frac{12\ KB}{4\ KB}\right) * 2\ (transactional\ cost) = 60\ RCUs$

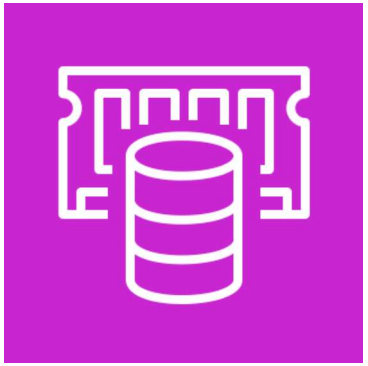
**(9 se redondea al 4 KB superior)*

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB vs. ElastiCache

ElastiCache



- ElastiCache es en memoria, pero DynamoDB es sin servidor
- Ambos son almacenes de claves/valores

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB vs. EFS

EFS



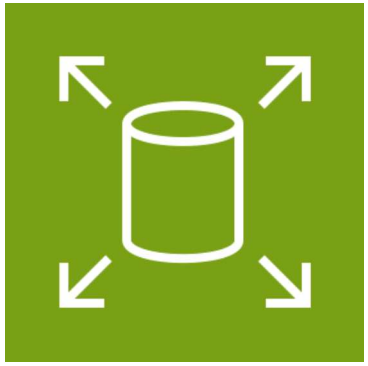
- EFS debe conectarse a las instancias EC2 como una unidad de red

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB vs. EBS / Instance Store

EBS



- EBS e Instance Store sólo pueden utilizarse para el almacenamiento en caché local, no para el almacenamiento en caché compartido

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB vs. S3

S3



- S3 tiene mayor latencia y no está pensado para objetos pequeños

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Fragmentación de escritura en DynamoDB

- Imagina que tenemos una aplicación de votación con dos candidatos, el **candidato A** y el **candidato B**
- 🚨⚠️ Si la **clave de partición** es **"Candidate_ID"**, esto da lugar a dos particiones, lo que generará problemas (por ejemplo, partición en caliente)
- Una estrategia que permite distribuir mejor los elementos uniformemente entre las particiones
- **Añade un sufijo al valor de la clave de partición**



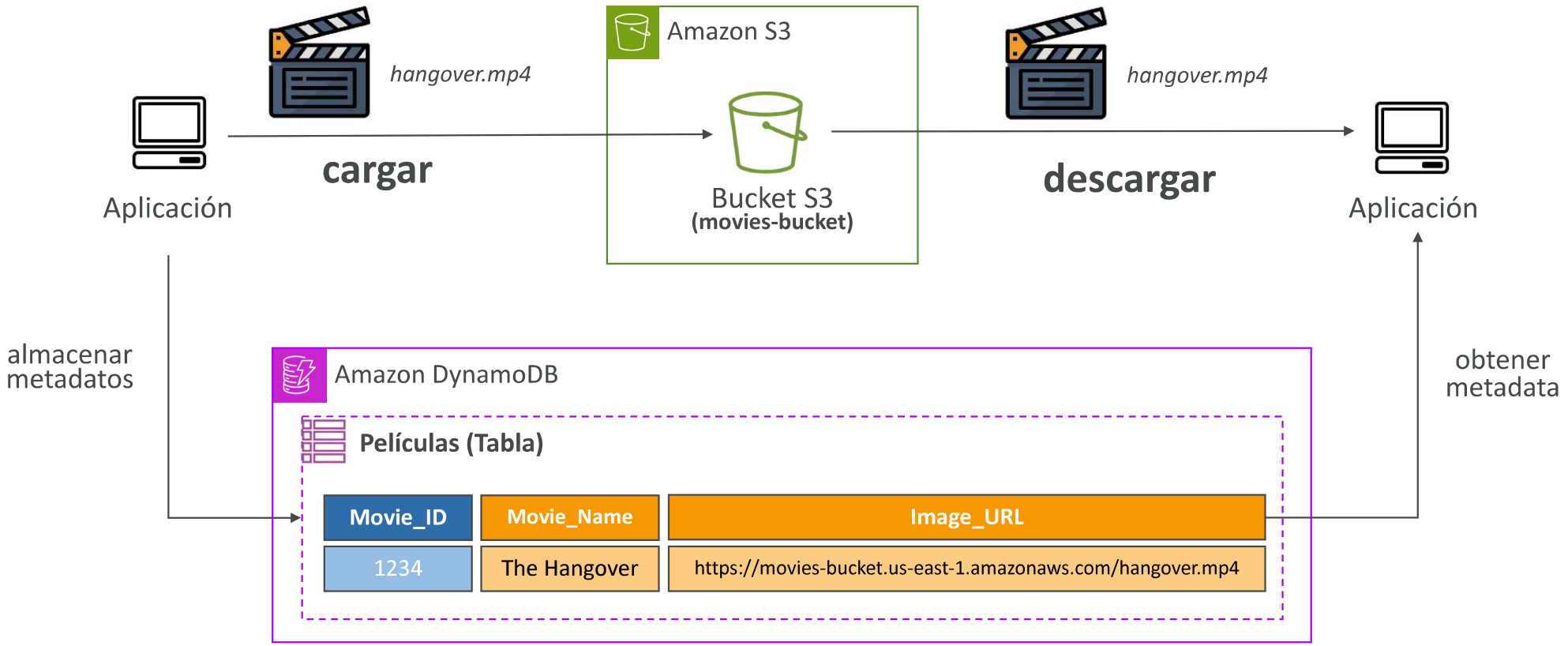
Clave de partición | Atributos

Candidate_ID	Voter_ID
Candidate_A-98	439812123A
Candidate_B-77	219834522b

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Patrón de objetos grandes



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Indexación de metadatos de objetos S3

- Asegurar la consistencia eventual entre los datos almacenados en S3 y sus metadatos indexados en DynamoDB
- Configurar triggers de AWS Lambda para actualizar índices en DynamoDB automáticamente al modificar objetos en S3
- Utilizar políticas de IAM para controlar el acceso a los metadatos en DynamoDB basado en roles de usuario



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Seguridad y otras características

- **Seguridad**
  - **Endpoints VPC**: Acceso seguro a DynamoDB sin necesidad de conexión a Internet
  - **Control de acceso**: Gestión completa mediante AWS IAM para un control granular
- **Cifrado**
  - **En reposo**: Utiliza AWS Key Management Service (KMS) para cifrar datos almacenados
  - **En tránsito**: Protección mediante SSL/TLS para garantizar la seguridad de los datos durante su transmisión
- **Resiliencia de datos**
  - **Copia de seguridad y restauración**: Opciones integradas para respaldar y restaurar datos fácilmente
  - **Recuperación Puntual (PITR)**: Similar a RDS, permite la restauración de datos a cualquier punto en el tiempo, sin impactar el rendimiento

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Seguridad y otras características

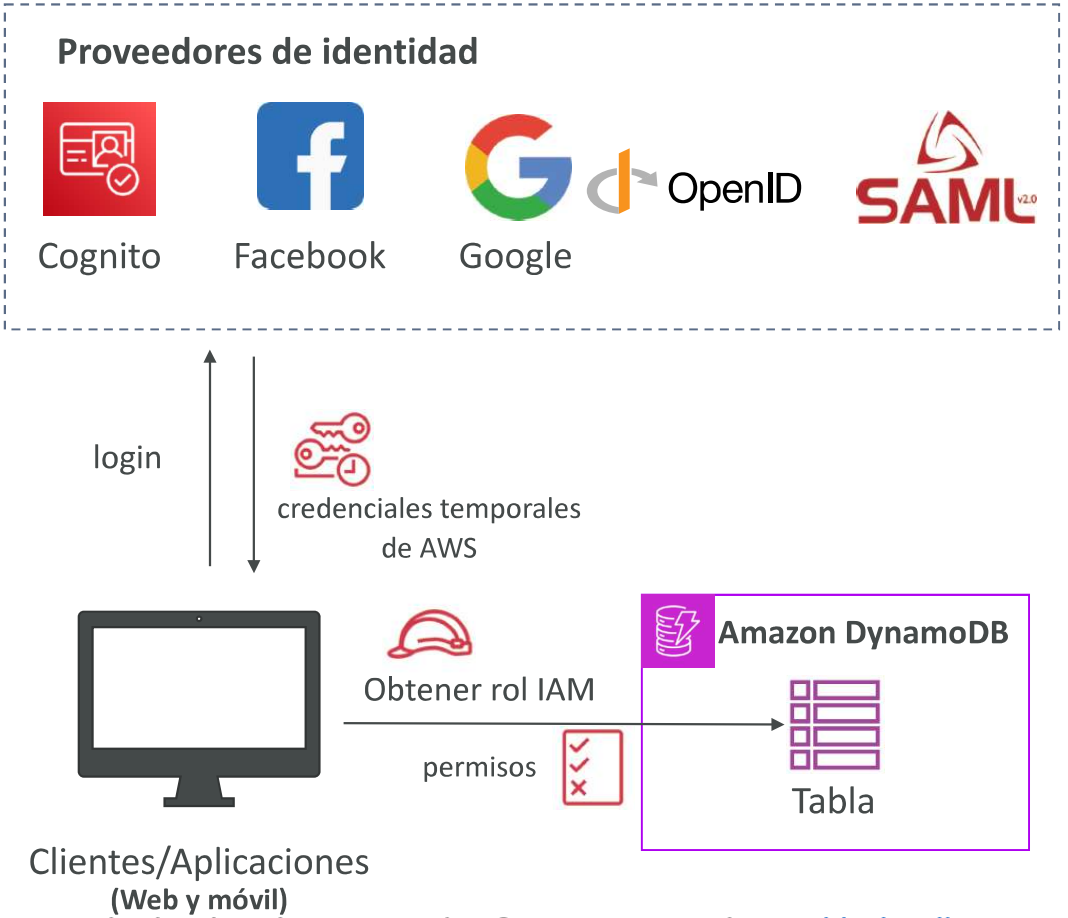
- **Tablas globales**
  - Configuración multiregión, multiactiva, completamente replicada para un alto rendimiento y disponibilidad
- **DynamoDB local**
  - Permite el desarrollo y pruebas de aplicaciones localmente sin necesidad de conectarse a la red de AWS
- **Migración a DynamoDB**
  - AWS Database Migration Service (DMS): Facilita la migración a DynamoDB desde diversas fuentes como MongoDB, Oracle, MySQL y Amazon S3

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Interacción con DynamoDB

- Antes de interactuar con DynamoDB, los **usuarios se autentican** a través de proveedores de identidad externos, incluidos Amazon Cognito User Pools, Google, Facebook, OpenID Connect y SAML
- Tras la autenticación, los usuarios obtienen roles de AWS IAM que definen sus permisos específicos para acceder y operar sobre DynamoDB



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DynamoDB - Control de acceso detallado

- Existe la opción de especificar condiciones al conceder permisos mediante una política de IAM:
  - Conceder permisos para que los usuarios puedan obtener acceso de solo lectura a determinados elementos y atributos de una tabla o un índice secundario
  - Conceder permisos para que los usuarios puedan obtener acceso de solo escritura a determinados atributos de una tabla, según la identidad del usuario en cuestión

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/specifying-conditions.html



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com



Amazon Redshift

www.blockstellart.com

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Visión general de Redshift

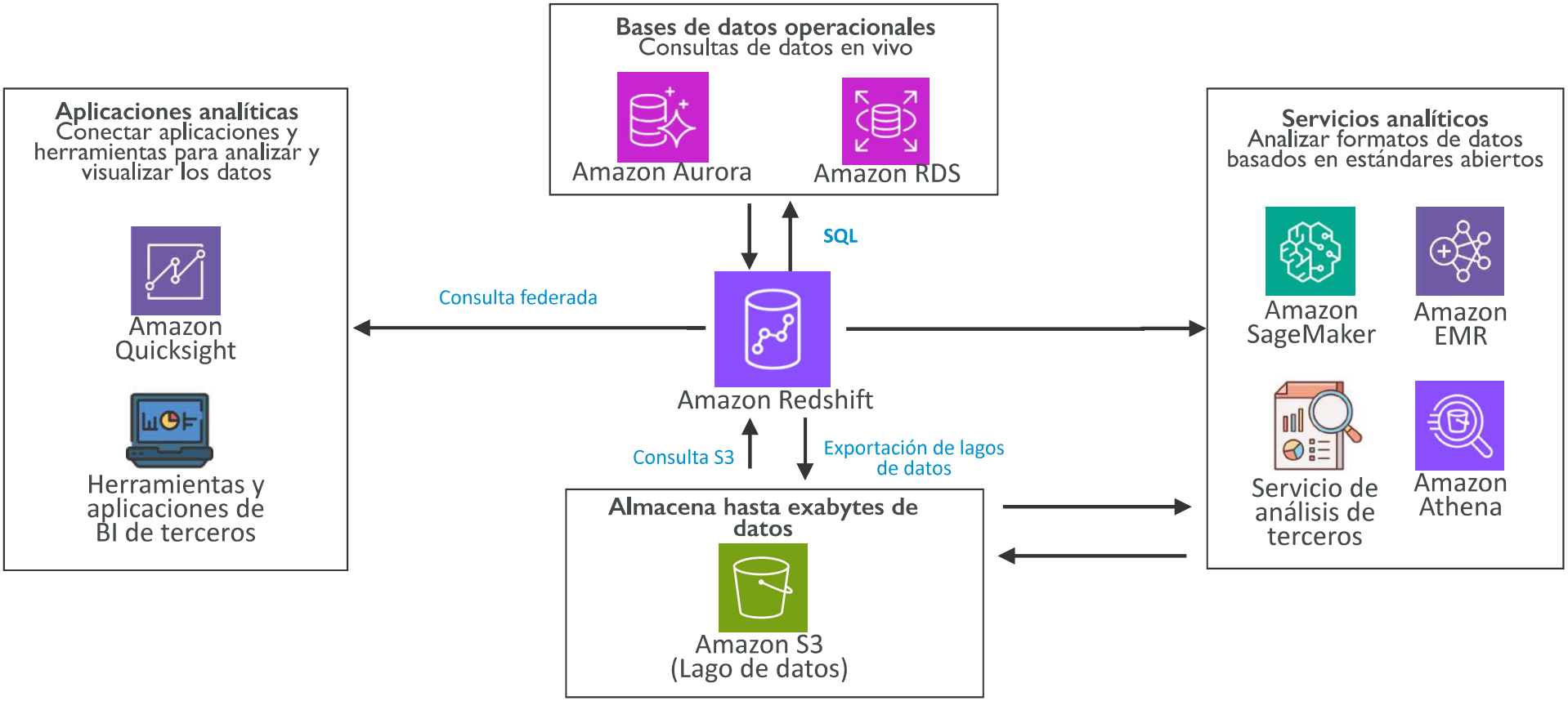


- Redshift se basa en PostgreSQL, **pero no se utiliza para OLTP**
- **Es OLAP - procesamiento analítico en línea (análisis y almacenamiento de datos)**
- Carga los datos una vez cada hora, no cada segundo
- Rendimiento 10 veces superior al de otros almacenes de datos, escala a PBs de datos
- Almacenamiento de datos **en columnas** (en lugar de en filas)
- Ejecución de consultas en paralelo masivo (MPP), con alta disponibilidad
- Paga a medida que avanzas en función de las instancias aprovisionadas
- Tiene una interfaz SQL para realizar las consultas
- Las herramientas de BI, como AWS Quicksight o Tableau, se integran con ella

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Visión general de Redshift



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Casos de uso de Redshift



Acelerar las cargas de trabajo analíticas



Modernización del almacén de datos



Almacenar datos históricos de transacciones bursátiles



Analizar impresiones de anuncios y clics



Almacén de datos unificado y lago de datos



Analizar datos de ventas globales



Agregar datos de juegos



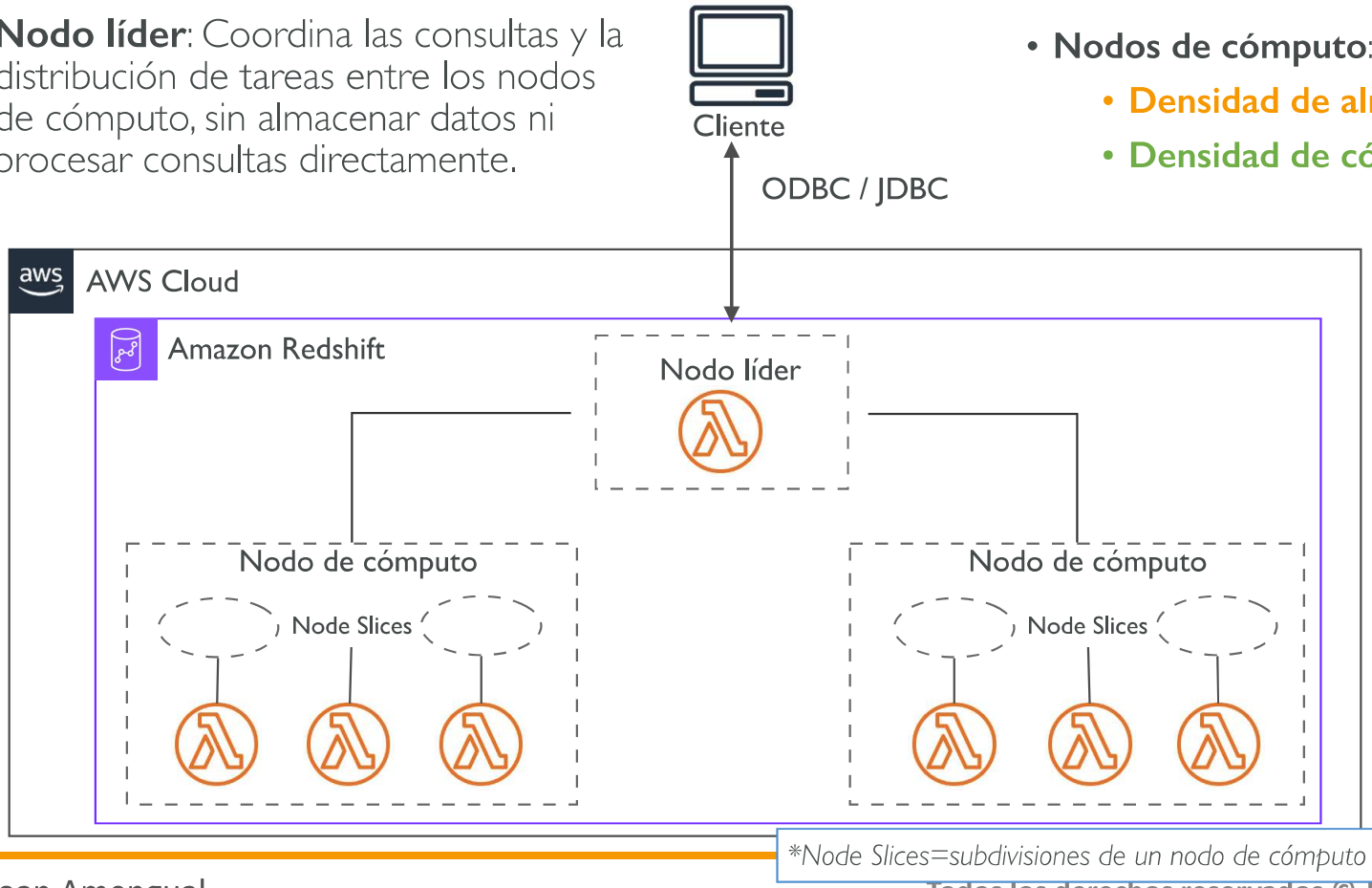
Analizar tendencias sociales

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Arquitectura de Redshift

- **Nodo líder**: Coordina las consultas y la distribución de tareas entre los nodos de cómputo, sin almacenar datos ni procesar consultas directamente.



*Node Slices=subdivisiones de un nodo de cómputo

- **Nodos de cómputo**:
  - **Densidad de almacenamiento (DS)**
  - **Densidad de cómputo (DC)**

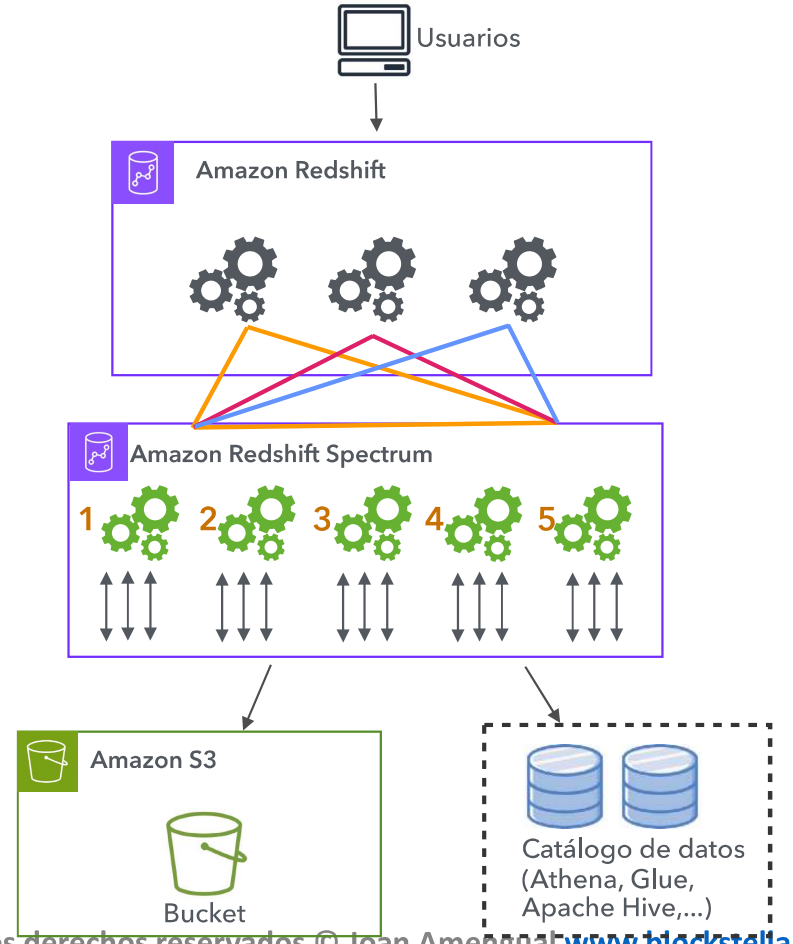
- **Densidad de almacenamiento (DS)**: Ideal para grandes cantidades de datos, con más almacenamiento pero menos capacidad de procesamiento
- **Densidad de cómputo (DC)**: Optimizados para un procesamiento rápido de consultas, ofreciendo más CPU y RAM pero menos espacio de almacenamiento

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Redshift Spectrum

- Permite la consulta de exabytes de datos no estructurados en S3 sin necesidad de cargarlos previamente
- Soporta múltiples usuarios y consultas simultáneas sin limitaciones
- Capacidad de aumentar recursos horizontalmente para manejar más carga de trabajo sin degradar el rendimiento
- Los recursos de almacenamiento y computación están desacoplados, lo que permite optimizar costos y escalabilidad
- Soporta una amplia variedad de formatos de datos
- Compatible con compresión Gzip y Snappy para una gestión eficiente del almacenamiento



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Rendimiento de Redshift

- **Procesamiento Paralelo Masivo (MPP)**: Aprovecha la arquitectura MPP para ejecutar consultas complejas de manera eficiente y rápida sobre grandes volúmenes de datos
- **Almacenamiento de datos en columnas**: Optimiza el rendimiento de las consultas al almacenar datos en formato columnar, lo que permite un acceso más rápido a las columnas específicas necesarias para las consultas
- **Compresión de columnas**: Mejora la eficiencia del almacenamiento y la velocidad de las consultas al comprimir los datos en las columnas, reduciendo así el espacio de almacenamiento necesario y acelerando el procesamiento de datos

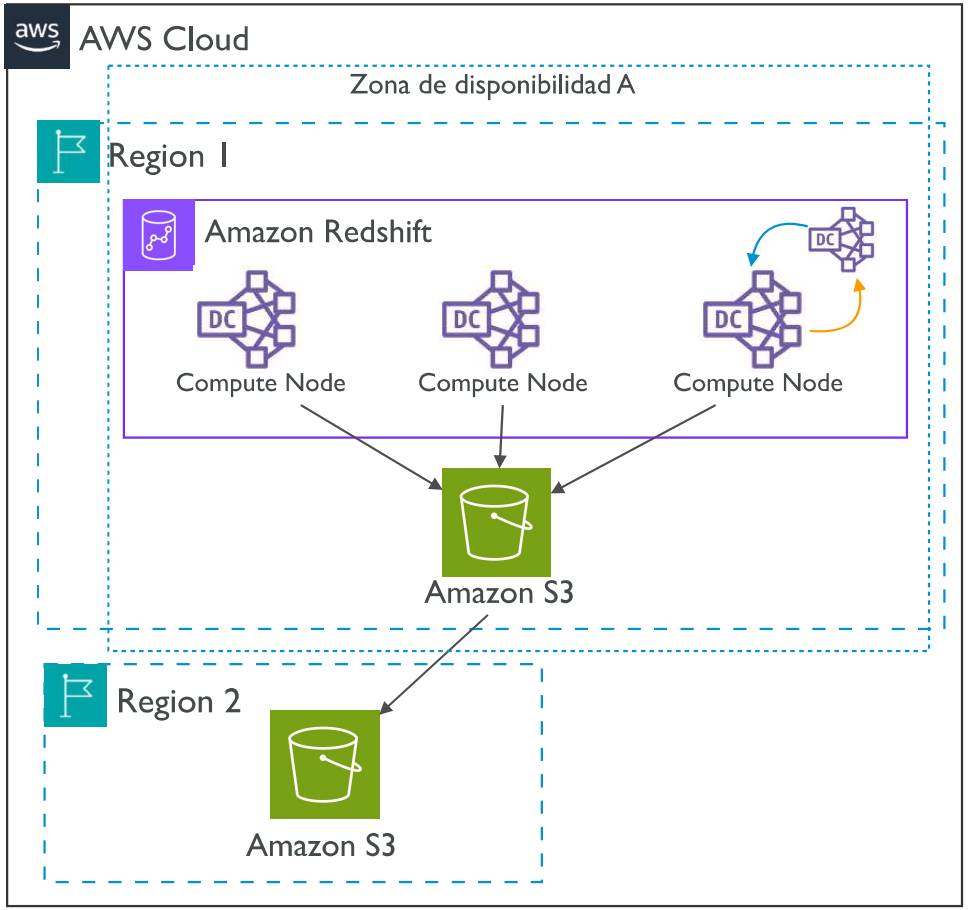


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Durabilidad de Redshift

- **Replicación dentro del clúster**: Garantiza la integridad de los datos mediante la duplicación dentro del mismo clúster
- **Respaldo en S3**:
  - Replicación asincrónica hacia otra región para mayor seguridad
- **Snapshots automatizados**: Los snapshots se crean automáticamente para proteger los datos
- **Reemplazo automático de discos/nodos fallidos**: Los componentes dañados son sustituidos automáticamente sin interrumpir el servicio
- **Limitaciones**:
  - Originalmente limitado a una sola AZ
  - Clústeres RA3 ahora admiten Multi-AZ para mayor resiliencia

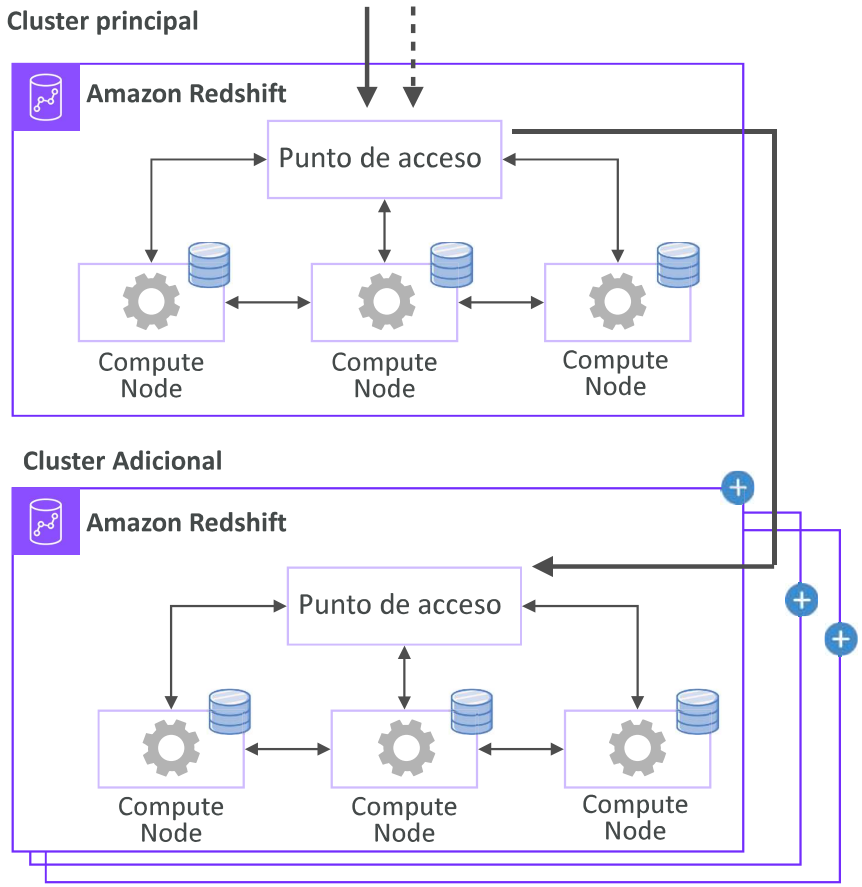


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Escalado de Redshift

- **Escalado vertical y horizontal:** Se puede escalar tanto en capacidad como en número de nodos según la demanda
- **Durante el escalado**:
  - Se crea un nuevo clúster mientras el antiguo sigue disponible para lecturas
  - El CNAME se cambia al nuevo clúster (unos minutos de inactividad)
  - Los datos se trasladan en paralelo a los nuevos nodos de cómputo



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Estilos de distribución de Redshift

- Cuando creas una tabla, puedes designar uno de los siguientes estilos de distribución:
  - AUTO, EVEN, KEY o ALL
- Si no se especificas un estilo de distribución, Amazon Redshift usa la distribución **AUTO**

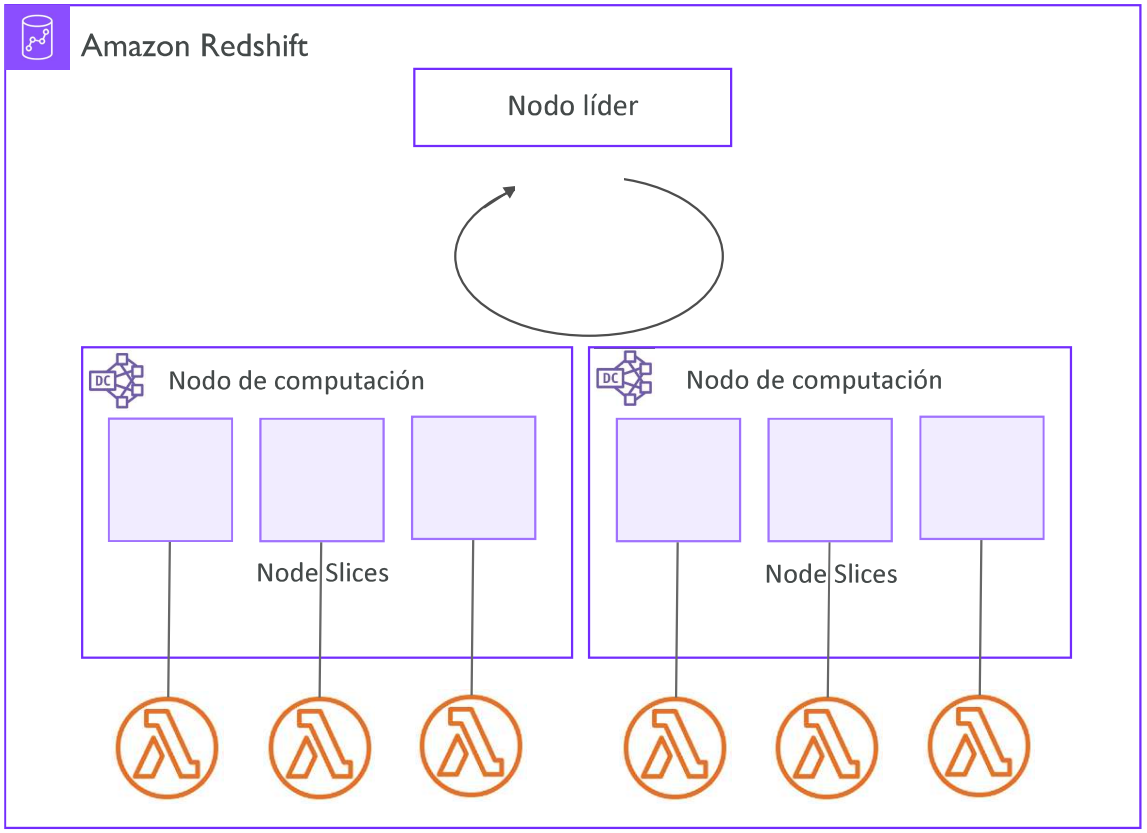
- **AUTO**
  - Redshift determina automáticamente el estilo de distribución basándose en el tamaño de los datos
- **EVEN**
  - Las filas se distribuyen equitativamente a través de las particiones en un método round-robin
- **KEY**
  - Las filas se distribuyen basándose en una columna específica
- **ALL**
  - La tabla completa se replica en cada nodo



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

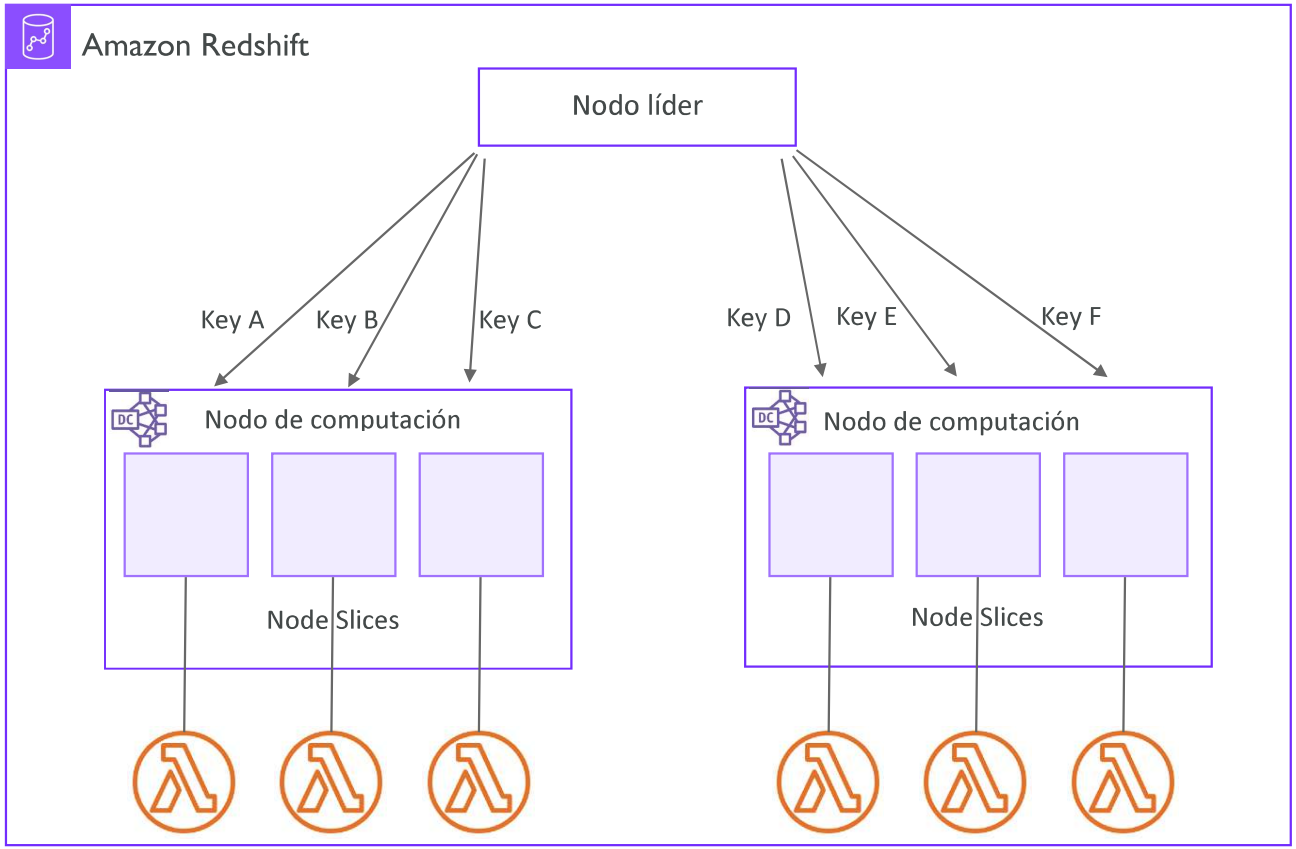
Distribución EVEN



- La distribución **EVEN** permite que los datos y las cargas de trabajo se distribuyan uniformemente entre todos los nodos disponibles
- Cada nodo maneja una porción de la carga total, y dentro de cada nodo, los fragmentos (Node Slices) procesan tareas específicas
- Esta arquitectura permite una distribución equitativa (EVEN) de la carga de trabajo, mejorando la escalabilidad y la resiliencia del sistema

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Distribución KEY

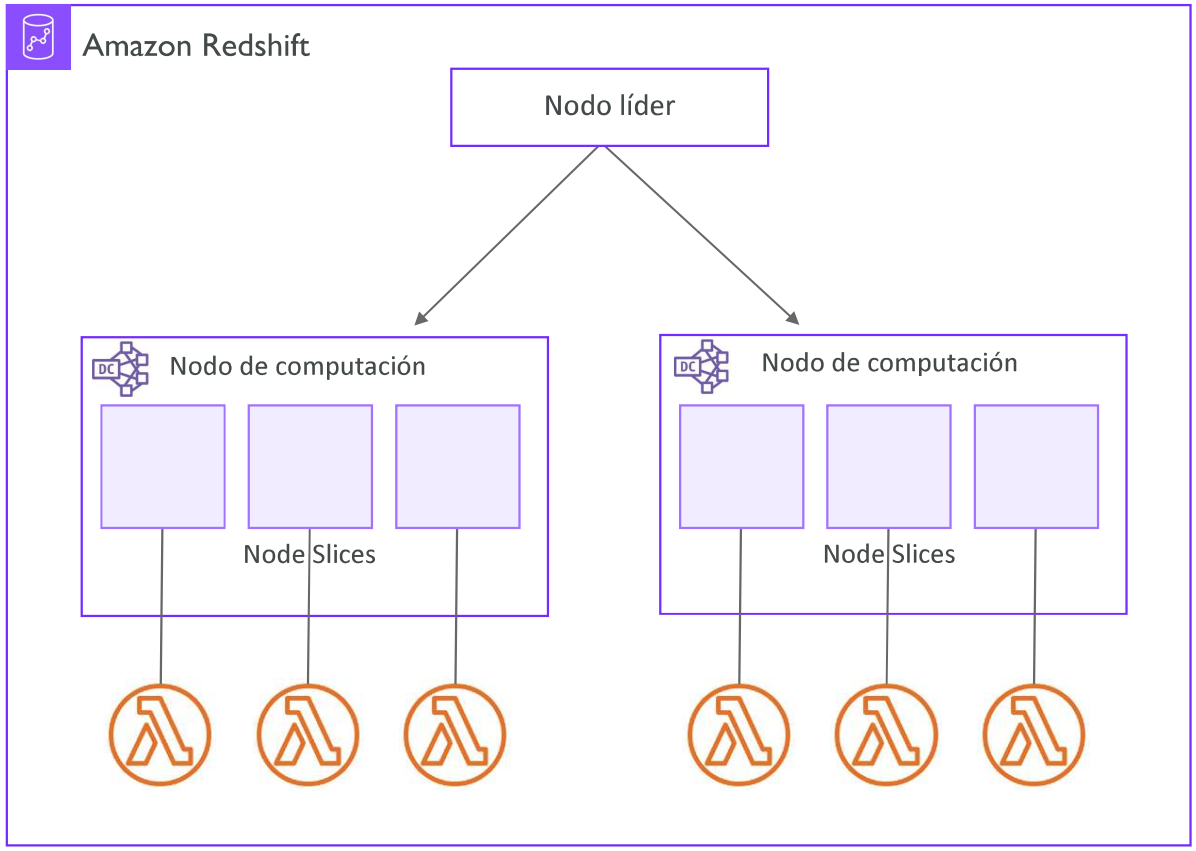


- La distribución **KEY** permite que los datos se distribuyan de manera eficiente entre todos los nodos disponibles basándose en las claves específicas
- Cada nodo maneja una porción de la carga total, y dentro de cada nodo, los fragmentos (Node Slices) procesan tareas específicas asignadas por clave
- Esta arquitectura permite una distribución basada en claves (KEY), mejorando la eficiencia y el balance de carga del sistema

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Distribución ALL



- La distribución **ALL** permite que los datos se distribuyan uniformemente entre todos los nodos disponibles
- Cada nodo maneja una porción de la carga total, y dentro de cada nodo, los fragmentos (Node Slices) procesan tareas específicas
- Las funciones Lambda se utilizan para manejar eventos y realizar tareas asignadas de manera eficiente
- Esta arquitectura es ideal para aplicaciones que requieren alta disponibilidad y escalabilidad, ya que permite el procesamiento paralelo de tareas y la rápida respuesta a las solicitudes de los usuarios

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Importación / Exportación de datos

- Comando **COPY**
  - Desde S3, EMR, DynamoDB, hosts remotos
  - S3 requiere un archivo manifiesto y rol de IAM
- Comando **UNLOAD**
  - Descarga datos de una tabla a archivos en S3

- **Otras funcionalidades**:
  - Integración sin ETL de Amazon Aurora
    - Replicación automática desde Aurora a Redshift
  - Ingesta de Streaming en Redshift
    - Desde Kinesis Data Streams o MSK

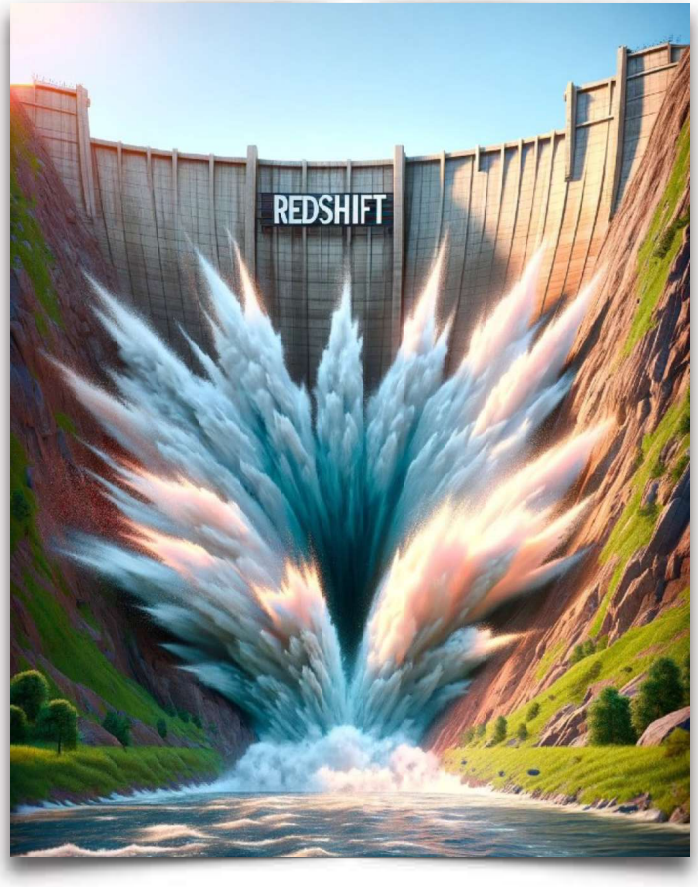


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Comando COPY en profundidad

- **Usa el comando COPY para cargar grandes cantidades de datos desde fuera de Redshift**
- Si los datos ya están en Redshift en otra tabla:
  - Usa: *INSERT INTO <…> SELECT*
  - O también puedes usar: *CREATE TABLE AS*
- COPY puede descifrar datos mientras se cargan desde S3
- Se admite la compresión Gzip, lzop y bzip2 para acelerarlo aún más
- Opción de compresión automática
  - Analiza los datos que se están cargando y determina el esquema de compresión óptimo para almacenarlos
- Caso especial: tablas con muchas filas, pocas columnas
  - Carga con una sola transacción COPY si es posible
  - De lo contrario, las columnas de metadatos ocultas consumen demasiado espacio

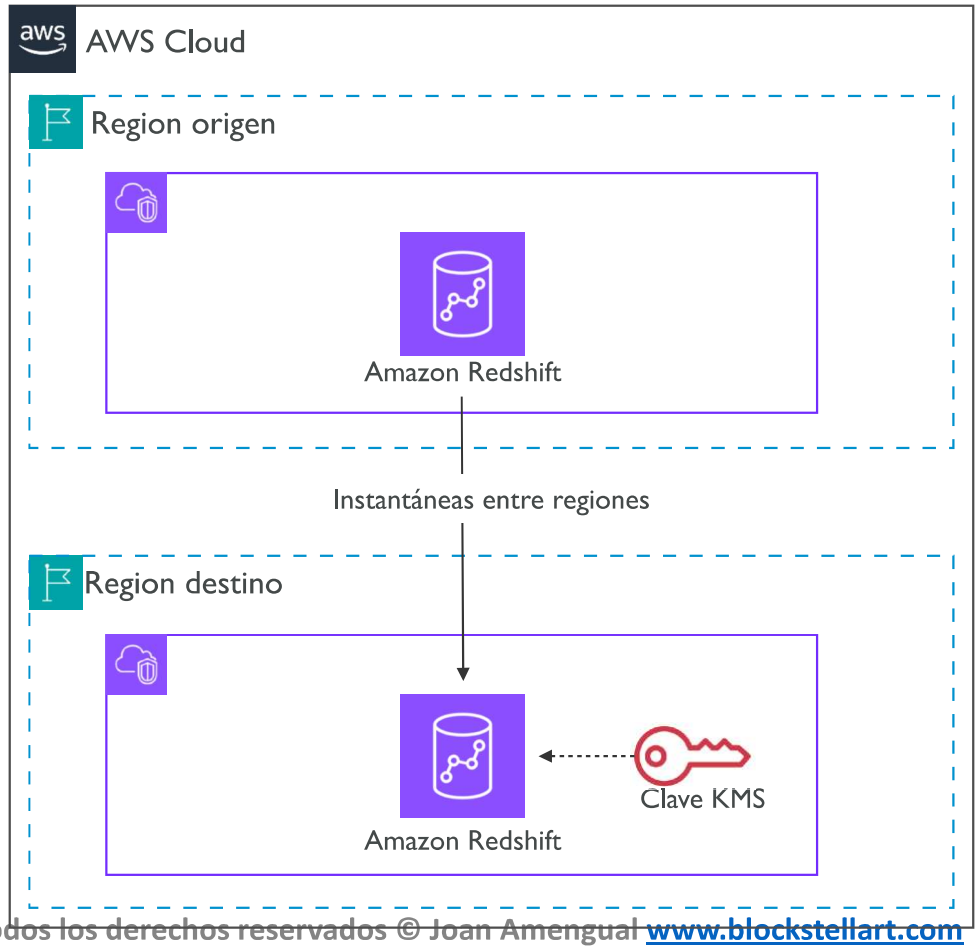


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Concesiones de copia de Redshift para copias de instantáneas entre regiones (COPY GRANT)

- Supongamos que tienes un clúster de Redshift cifrado con KMS y una instantánea del mismo
- Quieres copiar esa instantánea a otra región para realizar una copia de seguridad
- En la región de destino de AWS:
  - Crea una clave KMS si aún no tienes una
  - Especifica un nombre único para tu concesión de copia de instantáneas
  - Especifica el ID de la clave KMS para la cual estás creando la concesión de copia
- En la región de origen de AWS:
  - Habilita la copia de instantáneas a la concesión de copia que acabas de crear

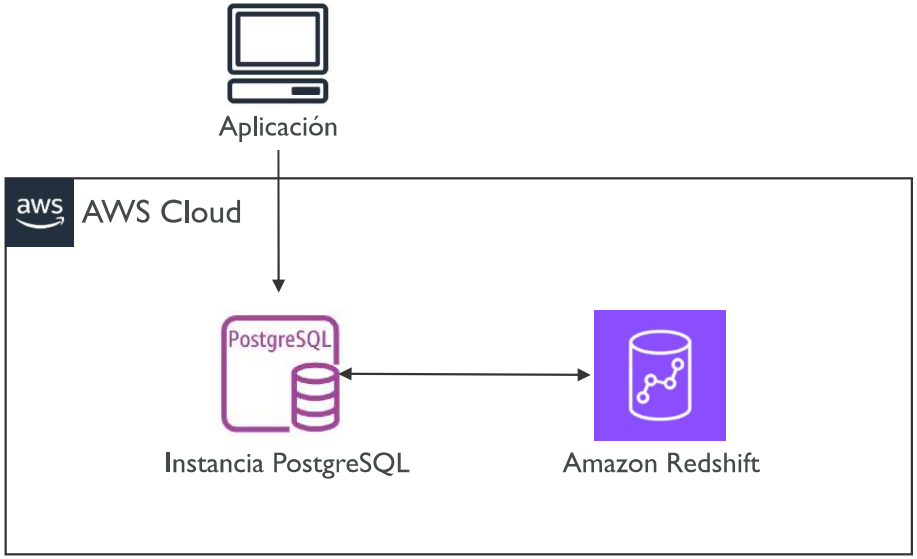


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

DBLINK

- Conectar Redshift a PostgreSQL (posiblemente en RDS)
- Buena forma de copiar y sincronizar datos entre PostgreSQL y Redshift




```
CREATE EXTENSION postgres_fdw;
CREATE EXTENSION dblink;
CREATE SERVER foreign_server
  FOREIGN DATA WRAPPER postgres_fdw
  OPTIONS (host '<amazon_redshift_ip>', port '<puerto>', dbname '<nombre_base_de_datos>', sslmode 'require');
CREATE USER MAPPING FOR <nombre_usuario_postgresql_rds>
 SERVER foreign_server
 OPTIONS (user '<nombre_usuario_redshift_amazon>', password '<contraseña>');
```

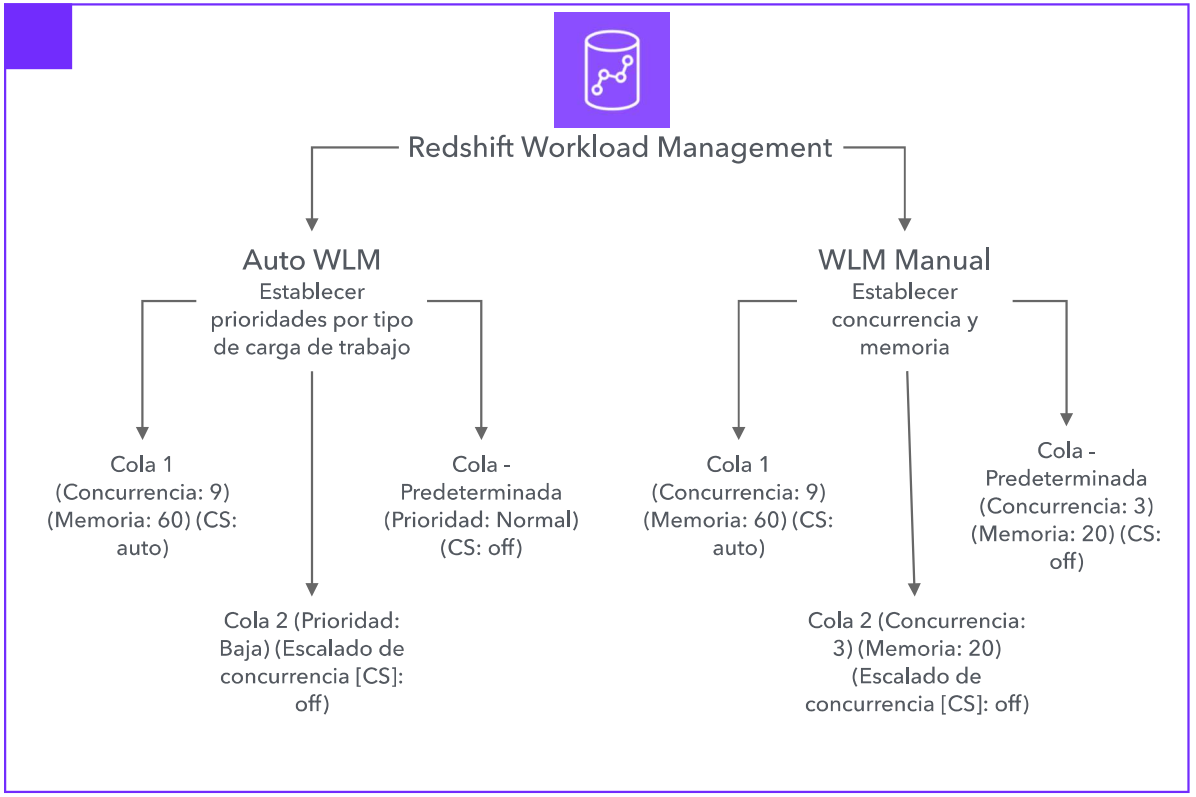
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Redshift Workload Management (WLM) / Administración de cargas de trabajo (WLM)

- Priorizar consultas cortas y rápidas frente a consultas largas y lentas
- Colas de consultas
- A través de la consola, CLI o API

En Auto WLM, la concurrencia y la memoria son gestionadas por Amazon Redshift

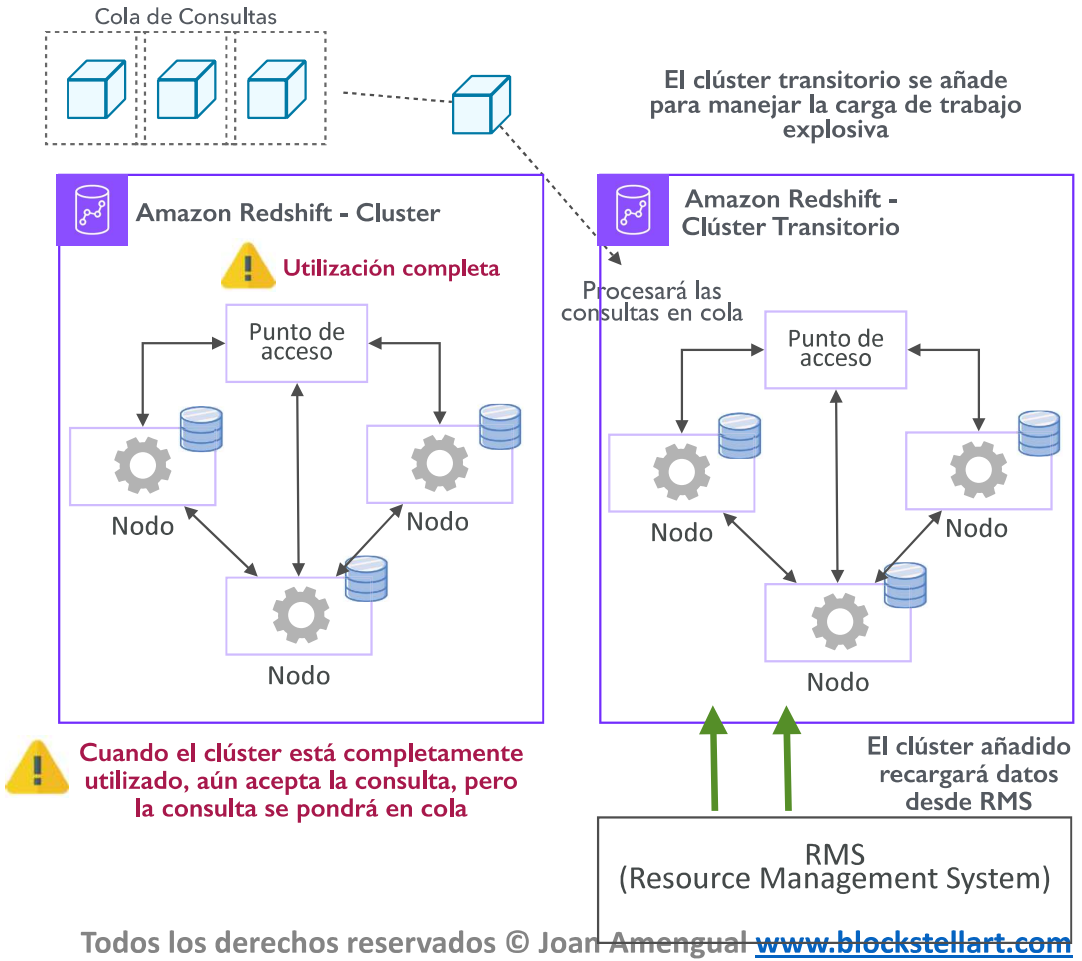


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Escalado de concurrencia

- Añade automáticamente capacidad al clúster para manejar el aumento de consultas de lectura concurrentes
- Soporta prácticamente usuarios y consultas concurrentes ilimitadas

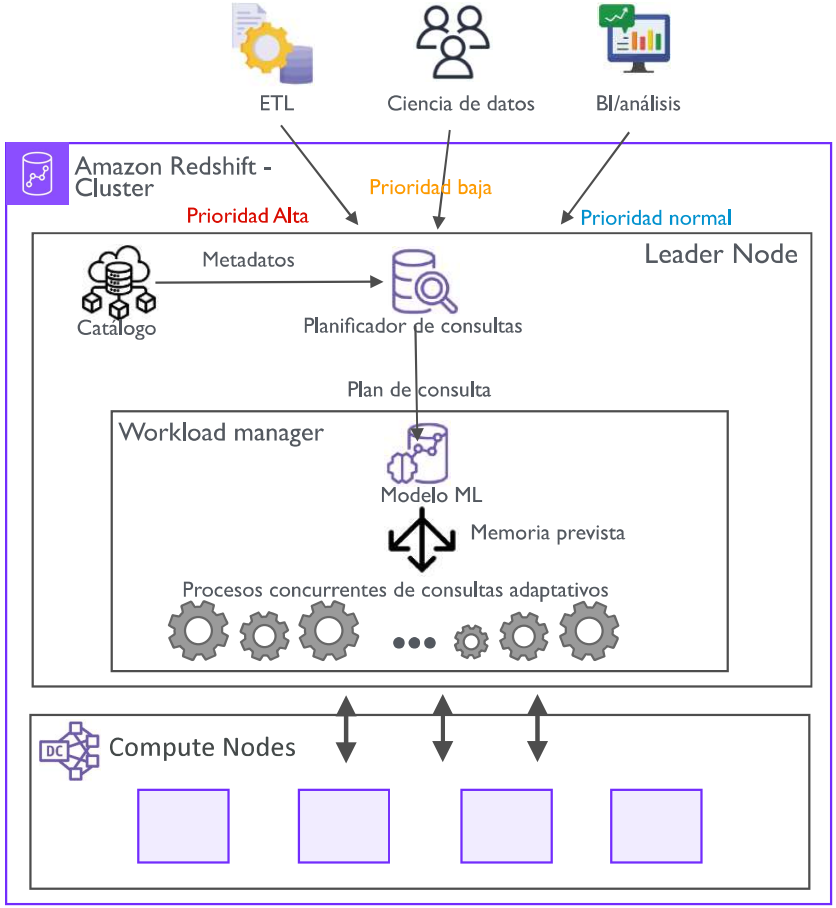


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

AutoWLM
Administración automática de cargas de trabajo (WLM)

- El **AutoWLM** de Redshift es responsable del control de admisión, la programación y la asignación de recursos
- Después de recibir la consulta, AutoWLM convierte el plan de ejecución y las estadísticas optimizadas en un formato vectorial
- Redshift utiliza el resultado del modelo para colocar la consulta en la cola: basándose en el tiempo de ejecución predicho
- AutoWLM emplea un mecanismo de turno rotativo ponderado para programar consultas de mayor prioridad más frecuentemente que las de baja prioridad



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Manual WLM
Administración manual de cargas de trabajo (WLM)

- Con WLM manual, puedes administrar el rendimiento del sistema y la experiencia de los usuarios mediante la modificación de la configuración de la WLM para crear colas independientes para las consultas de ejecución prolongada y las de ejecución corta

- Amazon Redshift configura las siguientes colas de consultas:
  - **Una cola de superusuario**: La cola de superusuario está reservada únicamente para superusuarios y no se puede configurar
  - **Una cola de usuario predeterminada** : La cola predeterminada está configurada inicialmente para ejecutar cinco consultas de forma simultánea

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Manual WLM
Administración manual de cargas de trabajo (WLM)

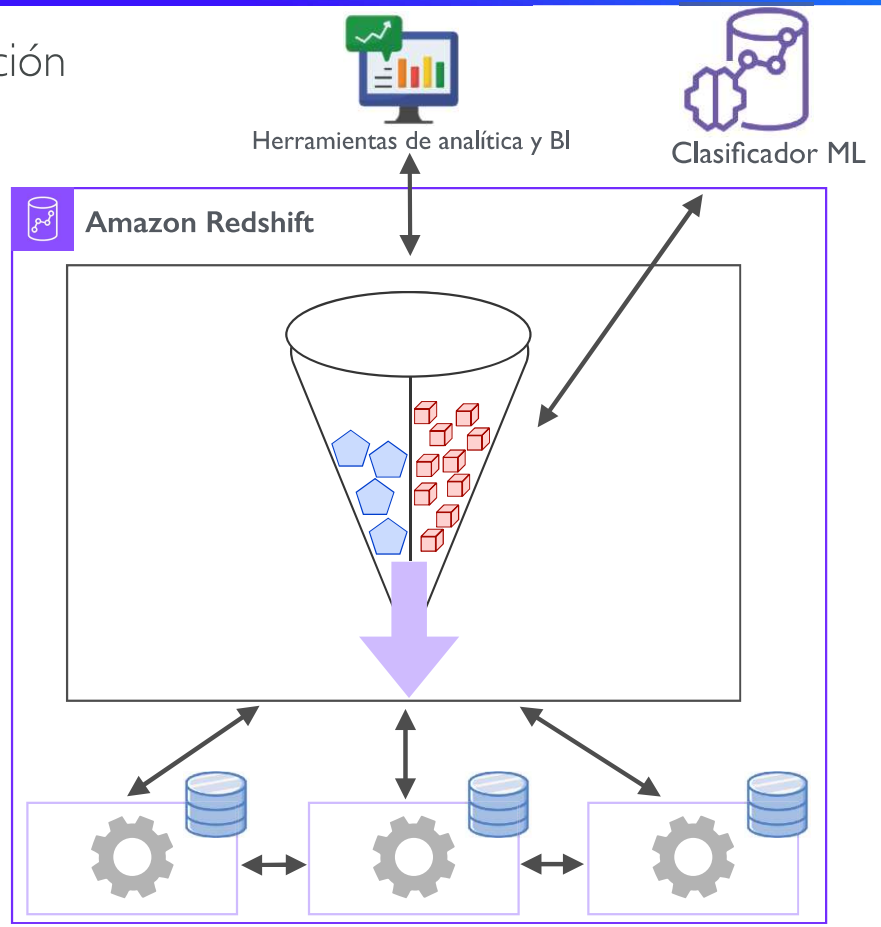
- Puedes añadir colas de consultas adicionales a la configuración de WLM predeterminada, hasta 8 colas y hasta un nivel de concurrencia de 50
- Para cada cola de consultas, puedes configurar lo siguiente:
  - Modo de escalado de simultaneidad
  - Nivel de simultaneidad
  - Grupos de usuarios
  - Grupos de consultas
  - Porcentaje de memoria de WLM por utilizar
  - Tiempo de espera de WLM
  - Salto de cola de consultas de WLM
  - Reglas de monitorización de consultas

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Short Query Acceleration (SQA)
Aceleración de Consultas Cortas (SQA)

- Prioriza consultas de corta duración sobre las de larga duración
- Las consultas cortas se ejecutan en un espacio dedicado, no esperan en la cola detrás de consultas largas
- Se puede usar en lugar de las colas de WLM para consultas cortas
- Funciona con:
  - **CREATE TABLE AS (CTAS)**
  - **Consultas de solo lectura (instrucciones SELECT)**
- Utiliza un algoritmo de machine learning para analizar cada una de las consultas que reúnan los requisitos necesarios y predecir su tiempo de ejecución
- WLM asigna dinámicamente un valor para el tiempo de ejecución máximo de SQA en función del análisis de la carga de trabajo del clúster
- Como alternativa, puede especificar un valor fijo comprendido entre 1 y 20 segundos



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Comando VACUUM

- Reordena las filas y recupera espacio en una tabla especificada o en todas las tablas de la base de datos actual
- Parámetros destacables de VACUUM:
  - **VACUUM FULL**
    - Combina la reclamación de espacio y la reordenación de los datos
  - **VACUUM DELETE ONLY**
    - Se enfoca únicamente en recuperar espacio eliminando filas que ya no son necesarias
  - **VACUUM SORT ONLY**
    - Realiza solo la reordenación de los datos según las claves de ordenación y no intenta recuperar espacio
  - **VACUUM REINDEX**
    - Realiza primero una re-análisis de la distribución de las columnas de clave de ordenación para identificar si hay desajustes o ineficiencias
    - Luego realiza un VACUUM completo

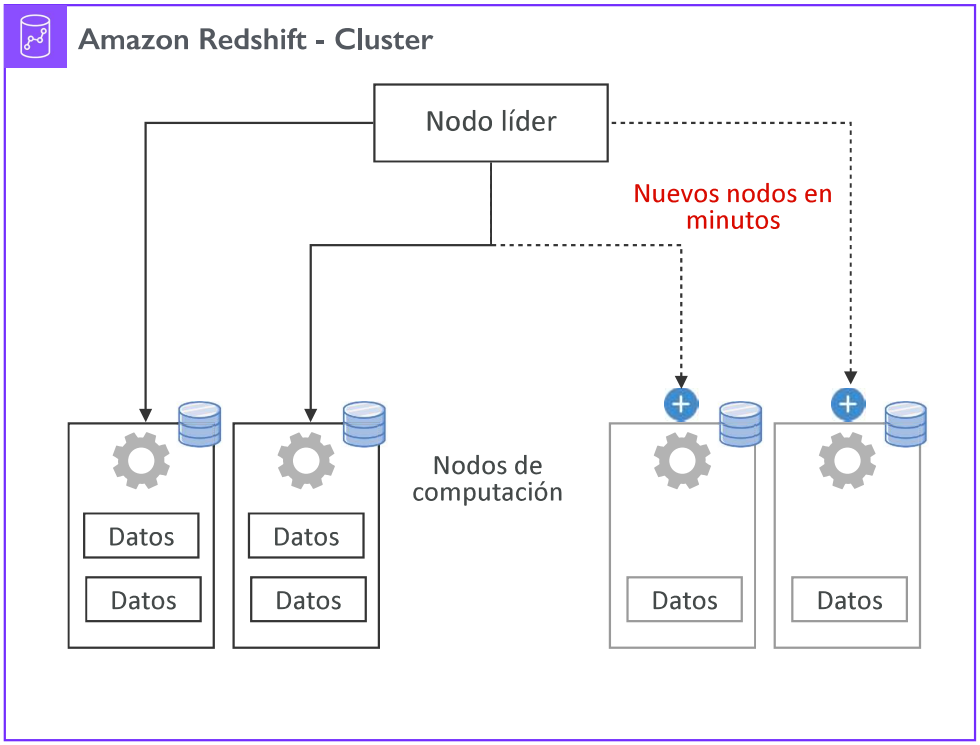


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Redimensionamiento de clústeres de Redshift

- **Redimensionamiento elástico**
  - Agregar o eliminar nodos del mismo tipo rápidamente
  - El clúster está inactivo por unos minutos
  - Intenta mantener las conexiones abiertas durante el tiempo de inactividad
  - Limitado a duplicar o reducir a la mitad para algunos tipos de nodos dc2 y ra3.
- **Redimensionamiento clásico**
  - Cambiar el tipo de nodo y/o el número de nodos
  - El clúster es de solo lectura durante horas o días
- **Instantánea, restauración, redimensionamiento**
  - Usado para mantener el clúster disponible durante un redimensionamiento clásico (Copiar clúster, redimensionar nuevo clúster)



© Joan Amengual

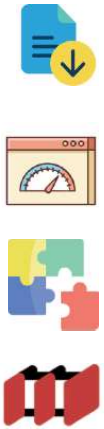
Todos los derechos reservados © Joan Amengual www.blockstellart.com

Novedades de Redshift

- **Nodos RA3 con almacenamiento gestionado**
  - Permiten escalado independiente de computación y almacenamiento
  - Basado en SSD



- **Exportación de data lake de Redshift**
  - Descarga consultas de Redshift a S3 en formato Apache Parquet
  - Parquet es 2 veces más rápido para descargar y consume hasta 6 veces menos almacenamiento
  - Compatible con Redshift Spectrum, Athena, EMR, SageMaker
  - Particionado automáticamente



- **Tipos de datos espaciales**
  - GEOMETRY, GEOGRAPHY



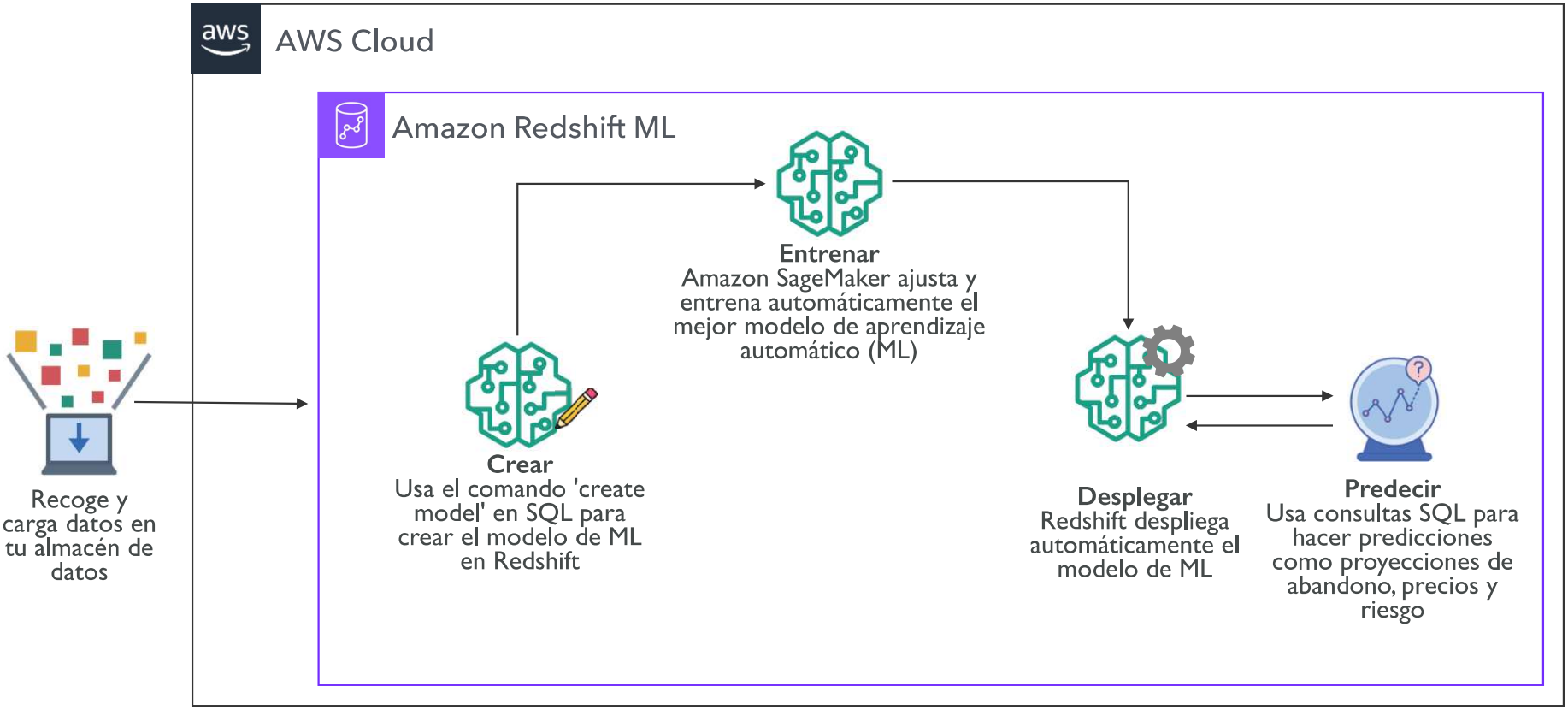
- **Compartición de datos entre regiones**
  - Compartir datos en vivo entre clústeres de Redshift sin copiar
  - Requiere el nuevo tipo de nodo RA3
  - Seguro, entre regiones y entre cuentas



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon Redshift ML



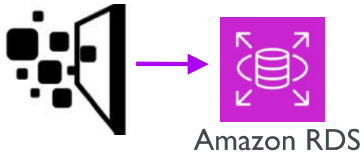
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Anti-patrones de Redshift

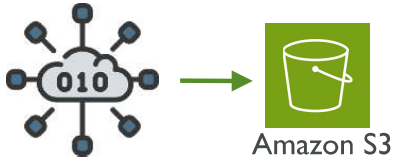
- **Conjuntos de datos pequeños**

  Usar RDS en su lugar



- **Datos BLOB**

  Almacenar referencias a archivos binarios grandes en S3, no los archivos en sí mismos.



- **OLTP**

  Usar RDS o DynamoDB en su lugar



- **Datos no estructurados**

  Primero realizar ETL con EMR, etc.

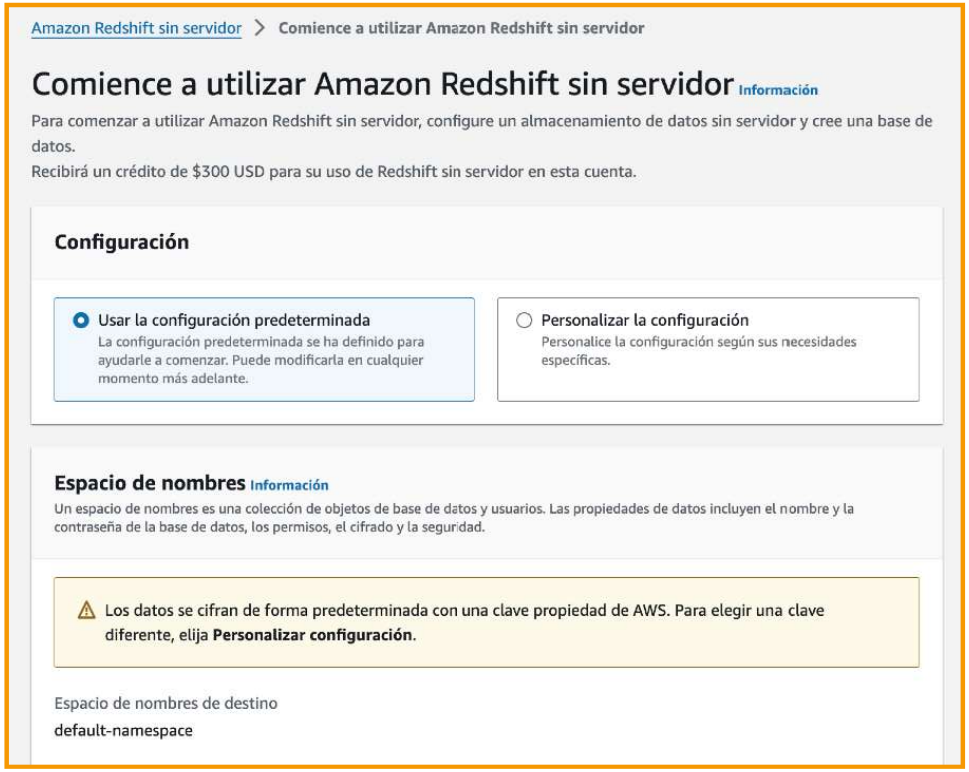


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Redshift Serverless

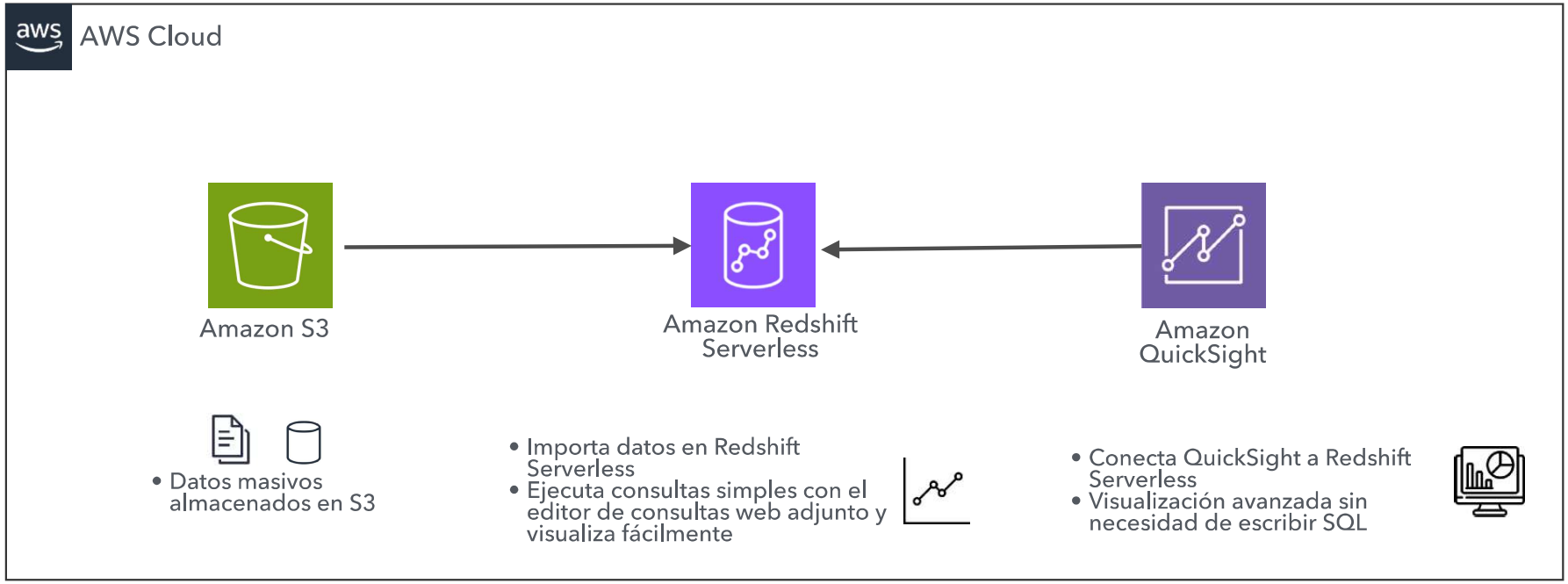
- **Escalado automático y aprovisionamiento para la carga de trabajo**
- Optimiza costes y el rendimiento
  - Pagas solo cuando está en uso
- Utiliza ML para mantener el rendimiento a través de cargas de trabajo variables y esporádicas
- Fácil configuración de entornos de desarrollo y prueba
- Obtienes un punto final sin servidor, conexión JDBC/ODBC, o simplemente consultas a través del editor de consultas de la consola



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Redshift Serverless

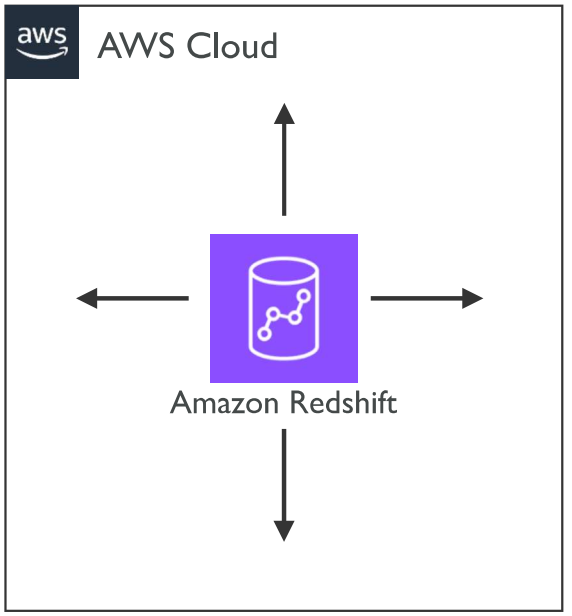


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Escalado de recursos en Redshift Serverless

- La capacidad se mide en **Unidades de Procesamiento de Redshift (RPU's)**
- Pagas por horas de RPU + almacenamiento
- **RPU's base**
  - Puedes ajustar la capacidad base
  - Por defecto está en AUTO
  - Pero puedes ajustar de 32 a 512 RPU's para mejorar el rendimiento de las consultas
- **RPU's máximas**
  - Puedes establecer un límite de uso para controlar costos
  - O, aumentarlo para mejorar el rendimiento

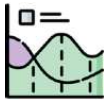


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Redshift Serverless

Hace todo lo que Redshift puede, **excepto**:



Redshift Spectrum



Grupos de parámetros



Gestión de cargas de trabajo



Integración con socios de AWS



Ventanas de mantenimiento / rutas de versiones

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Monitorización de Redshift Serverless

- **Vistas** de monitoreo
  - SYS_QUERY_HISTORY
  - SYS_LOAD_HISTORY
  - SYS_SERVERLESS_USAGE
  - ...y muchas más



- **Logs** de CloudWatch
  - Logs de conexión y de usuario habilitados por defecto
  - Datos de logs de actividad de usuario opcionales
  - Bajo /aws/redshift/serverless/



- **Métricas** de CloudWatch
  - QueriesCompletedPerSecond, QueryDuration, QueriesRunning, etc.
  - Dimensiones: DatabaseName, latency (short/medium/long), QueryType, stage

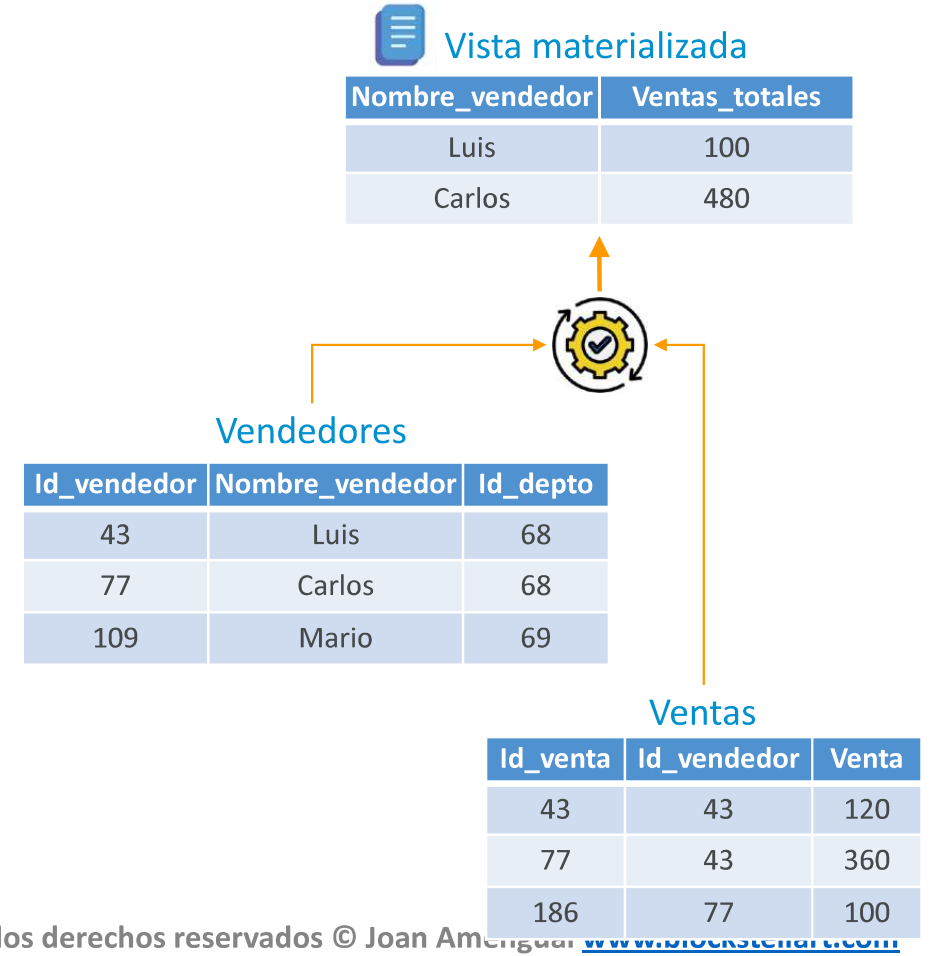


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Vistas materializadas de Redshift

- Contienen resultados precomputados basados en consultas SQL sobre una o más tablas base
- Proporcionan una forma de acelerar consultas complejas en un entorno de almacén de datos, especialmente en tablas grandes
- Puedes consultar vistas materializadas como cualquier otra tabla o vista
- Las consultas devuelven resultados más rápido ya que utilizan resultados precomputados sin acceder a las tablas base

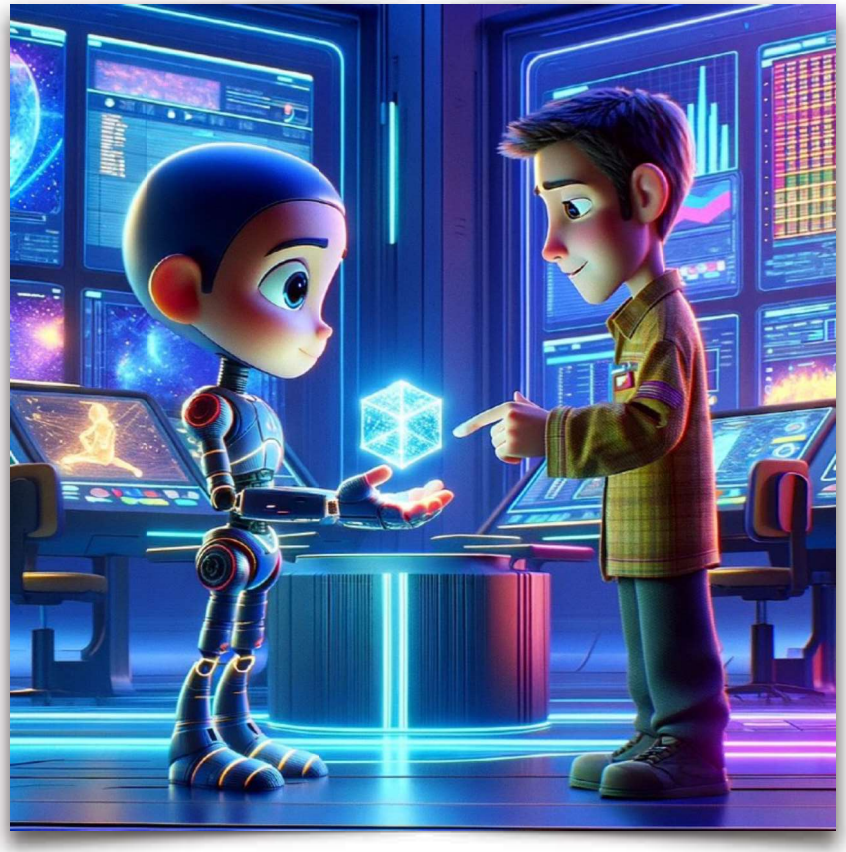


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Compartir datos de Redshift

- Comparte datos de manera segura entre clústeres de Redshift con fines de lectura
- ¿Por qué?
  - Aislamiento de cargas de trabajo
  - Colaboración entre grupos
  - Compartir datos entre desarrollo/ pruebas/producción
- Se pueden compartir bases de datos, esquemas, tablas, vistas, etc.

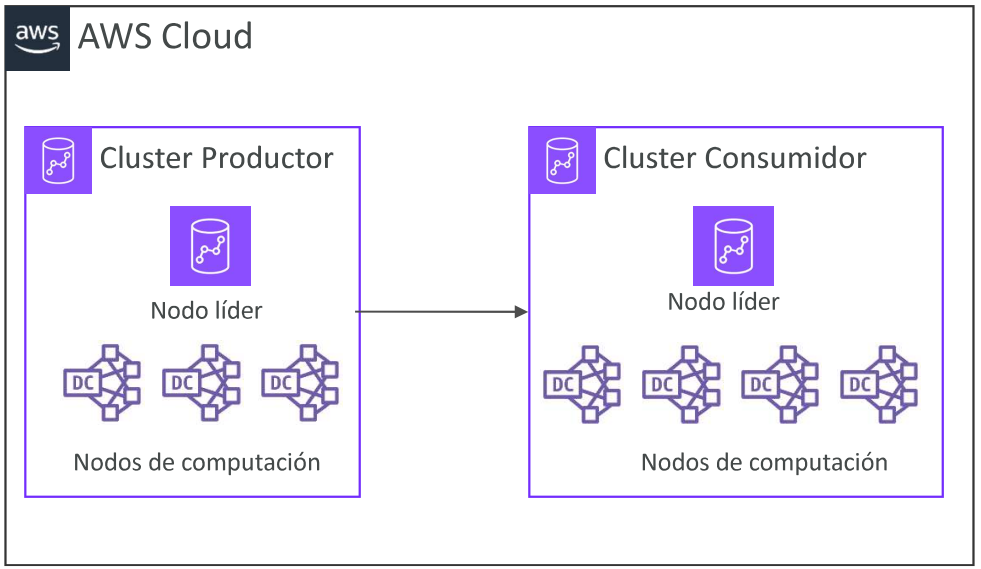


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Compartir datos de Redshift

- **Arquitectura productor / consumidor**
  - El productor controla la seguridad
  - Aislamiento para asegurar que el rendimiento del productor no sea afectado por los consumidores
  - Los datos están en vivo y son transaccionalmente consistentes
- Ambos deben estar cifrados, deben usar nodos RA3
- Compartir datos entre regiones implica cargos de transferencia



© Joan Amengual

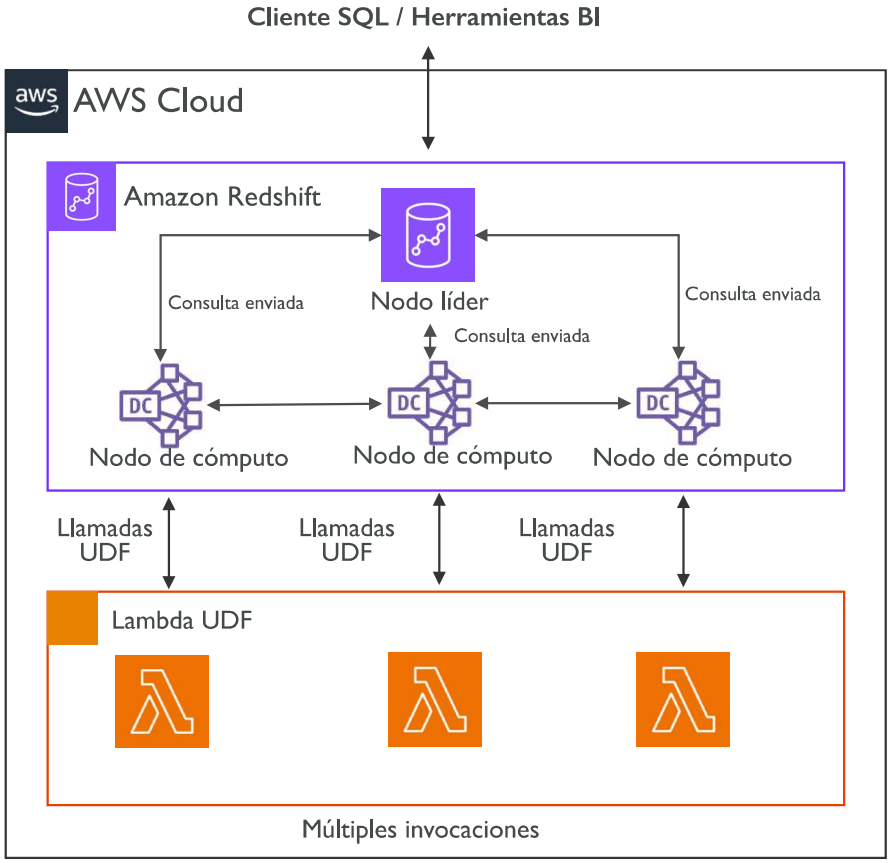
Todos los derechos reservados © Joan Amengual www.blockstellart.com

Redshift Lambda UDF

- Usa funciones personalizadas en AWS Lambda dentro de consultas SQL
  - ¡Usando cualquier lenguaje que desees!
  - Puedes hacer una variedad de acciones:
    - Llama a otros servicios (¿IA?)
    - Accede a sistemas externos
    - Integra con el servicio de ubicación

```
SELECT a, b FROM t1 WHERE lambda_multiply(a, b) = 32;

CREATE EXTERNAL FUNCTION
        custom_func(INT, INT)
RETURNS INT VOLATILE LAMBDA
        'lambda_sum'
         IAM_ROLE
        'arn:aws:iam::1234554321:role/Redshift-Sum-Test';
```

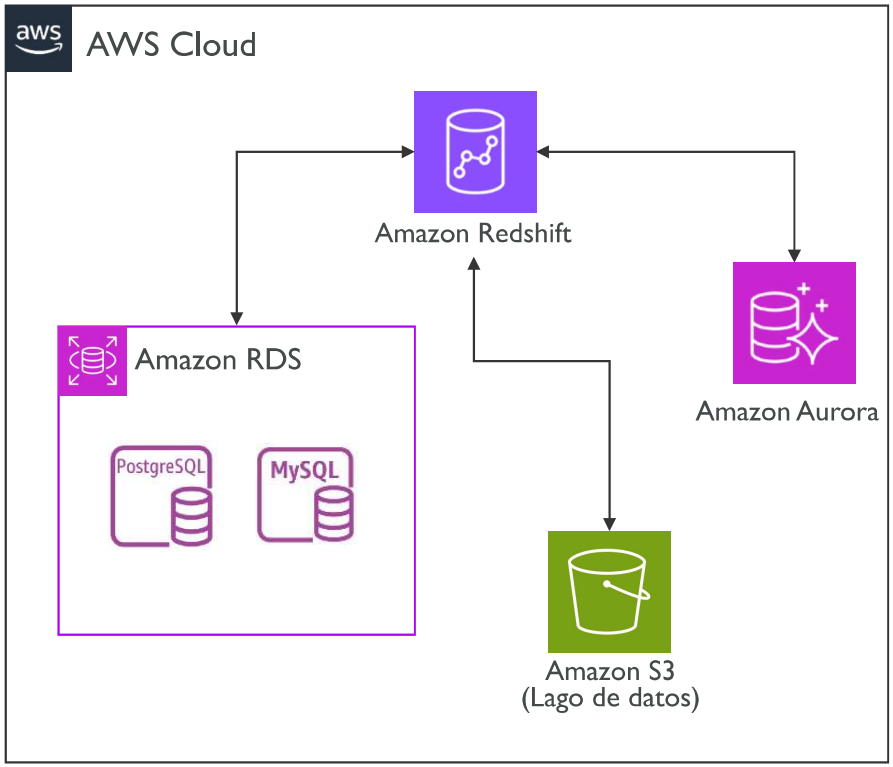


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Consultas federadas de Redshift

- Consulta y analiza a través de bases de datos, almacenes y Data Lakes
- Vincula Redshift a Amazon RDS o Aurora para PostgreSQL y MySQL
  - Incorpora datos en vivo en RDS a tus consultas de Redshift
  - Evita la necesidad de pipelines ETL
  - Puedes consultar RDS/Aurora desde Redshift, pero no al revés
- Desplaza la computación a bases de datos remotas para reducir el movimiento de datos
- Acceso de solo lectura a fuentes de datos externas
- Los costos serán incurridos en las bases de datos externas



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Tablas y vistas del sistema Redshift



- Contiene información sobre cómo está funcionando Redshift
- Tipos de tablas/vistas del sistema
  - **Vistas SYS**: Monitorean el uso de consultas y cargas de trabajo
  - **Tablas STV**: Datos transitorios que contienen instantáneas de los datos actuales
  - **Vistas SVV:** Metadatos sobre objetos de la BD que hacen referencia a las tablas STV
  - **Vistas STL**: Generadas a partir de registros persistidos en disco
  - **Vistas SVCS**: Detalles sobre consultas en clústeres principales y de escalado de concurrencia
  - **Vistas SVL**: Detalles sobre consultas en clústeres principales
- Muchas vistas y tablas de monitoreo del sistema son solo para clústeres aprovisionados, no para serverless

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com



Otras bases de datos

www.blockstellart.com

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon DocumentDB



- Aurora es una "implementación de AWS" de PostgreSQL / MySQL ...
- **DocumentDB es lo mismo que MongoDB (que es una base de datos NoSQL)**



- MongoDB se utiliza para almacenar, consultar e indexar datos JSON
- "Conceptos de despliegue" similares a los de Aurora
- Totalmente gestionado, de alta disponibilidad con replicación a través de 3 AZ
- El almacenamiento de DocumentDB crece automáticamente en incrementos de 10 GB, hasta 128 TB

- Escala automáticamente a cargas de trabajo con millones de peticiones por segundo

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon Neptune

- Base de datos **gráfica** totalmente gestionada
- Facilita la construcción y ejecución de aplicaciones que trabajan con conjuntos de **datos altamente conectados**
- Un **conjunto de datos de grafos** popular sería una **red social**
  - Los usuarios tienen amigos
  - Las publicaciones tienen comentarios
  - Los comentarios tienen likes de los usuarios
  - Los usuarios comparten y les gustan las publicaciones
- Puede almacenar hasta miles de millones de relaciones y consultar el gráfico con una latencia de milisegundos
- Alta disponibilidad con réplicas a través de múltiples AZs
- Casos de uso:
  - Aplicaciones de redes sociales
  - Recomendaciones de productos
  - Redes de conocimiento



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon Keyspaces (para Apache Cassandra)



- *Apache Cassandra = base de datos distribuida NoSQL de código abierto*
- Amazon Keyspaces = servicio compatible con Apache Cassandra, que permite a los usuarios ejecutar sus aplicaciones Cassandra en AWS sin tener que gestionar la infraestructura subyacente
- Ofrece compatibilidad con CQL (Cassandra Query Language)

- Casos de uso:
  - Aplicaciones web a gran escala
  - Datos de series temporales
  - Sistemas de recomendación en tiempo real
  - Análisis de fraude
  - Juegos en línea

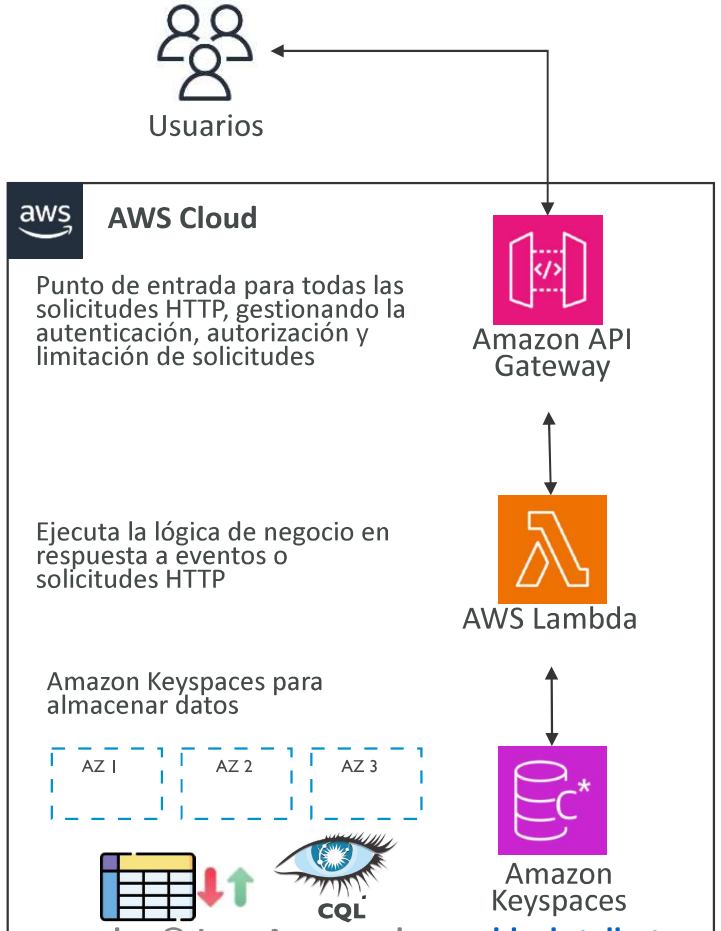


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon Keyspaces (para Apache Cassandra)

- Sin servidor, escalable, de alta disponibilidad, totalmente administrado
- Escala automáticamente las tablas hacia arriba/abajo en función del tráfico de la aplicación
- Las tablas se replican en múltiples AZ
- Latencia de un milisegundo a cualquier escala, miles de solicitudes por segundo
- Modo bajo demanda o modo provisionado con autoescalado
- Cifrado, copia de seguridad, recuperación puntual (PITR) de hasta 35 días

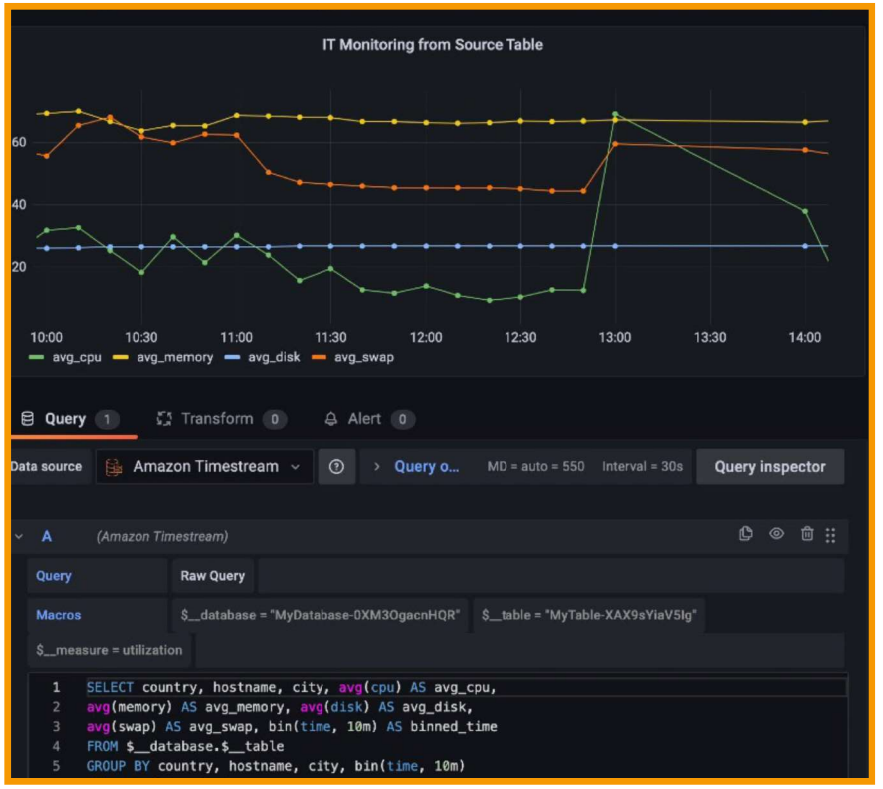


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon Timestream

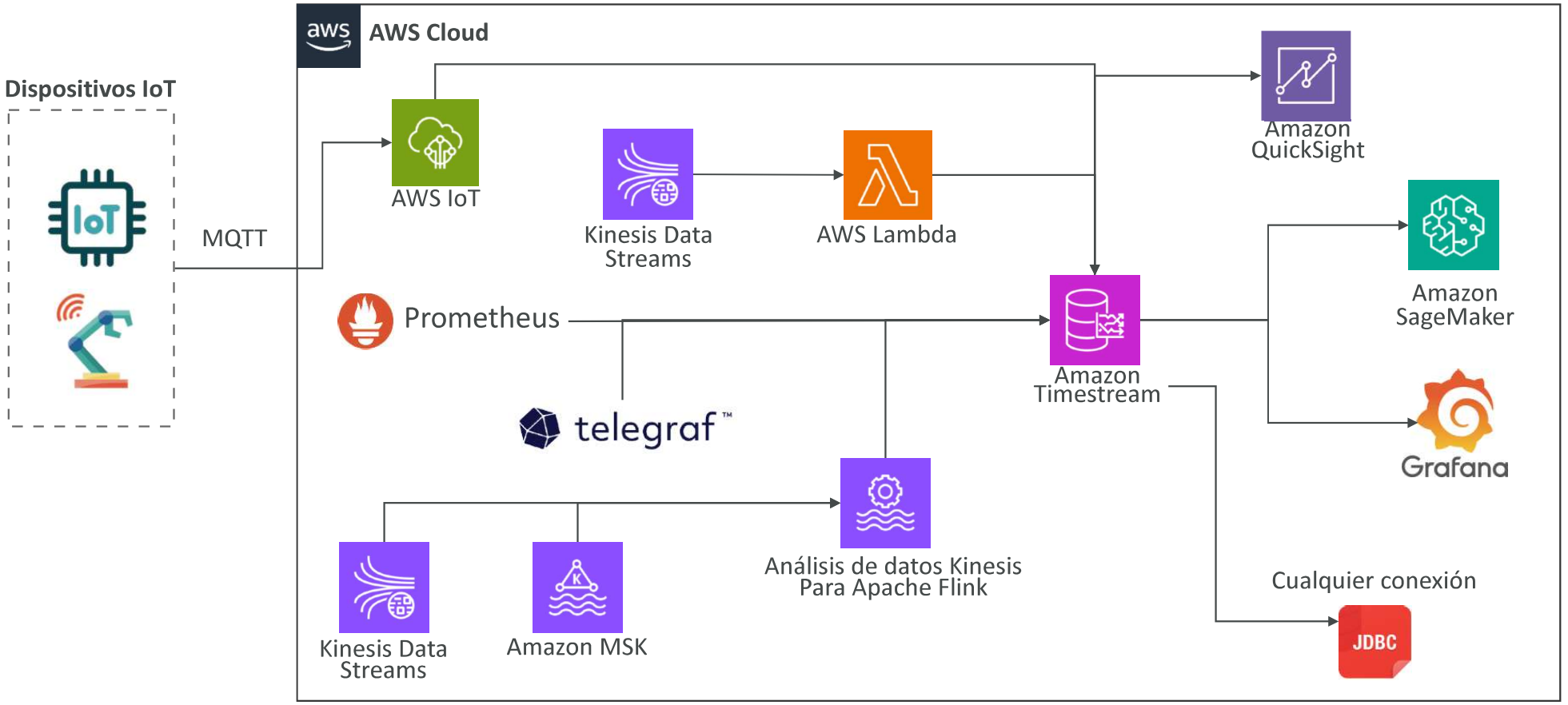
- Base de datos de **series temporales** totalmente gestionada, rápida, escalable y sin servidor (compatibilidad con SQL)
- Se amplía y reduce automáticamente para ajustar la capacidad
- Almacena y analiza billones de eventos al día
- 1000 veces más rápida y 1/10 del coste de las bases de datos relacionales
- Almacenamiento de datos por niveles: los datos recientes se guardan en la memoria y los históricos en un almacenamiento de coste optimizado
- Cifrado en tránsito y en reposo
- Casos de uso:
  - Aplicaciones IoT
  - Análisis en tiempo real



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon Timestream – Arquitectura



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon QLDB

- QLDB significa "Quantum Ledger Database" (base de datos de libros contables)
- Historial secuenciado de todos los cambios de datos de aplicaciones
- Sistema **inmutable**: ninguna entrada puede ser eliminada o modificada, verificable criptográficamente
- Totalmente gestionada, sin servidor, de alta disponibilidad, con replicación en múltiples AZ
- Se utiliza para **revisar el historial de todos los cambios realizados en los datos de tu aplicación** a lo largo del tiempo
- Diferencia con Amazon Managed Blockchain: **No hay componente de descentralización en QLDB**



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon QLDB

Una aplicación se conecta a un libro mayor y ejecuta transacciones que insertan, actualizan y eliminan un documento en una tabla denominada *cars*

1

Aplicación

A continuación, los datos se materializan en la tabla con vistas integradas. Estas vistas permiten consultar tanto el estado actual como el historial completo del vehículo, y se asigna un número de versión a cada revisión

3

Estado actual (antes de la eliminación)

ID	Fabricante	Modelo	Año	VIN	Propietario
1	Tesla	Modelo S	2012	123456789	John Stiles

Historial

ID	Version	Fecha de Inicio	Fabricante	Modelo	Año	VIN	Propietario
1	1	7/16/2012	Tesla	Modelo S	2012	123456789	Jane Doe
1	2	8/3/2013	Tesla	Modelo S	2012	123456789	John Stiles
1	3	9/2/2016	Eliminado				

Diario

2

Los datos se escriben primero en el diario en orden secuencial

INSERTAR coches
ID: 1
H(T1)
Fabricante: Tesla
Modelo: Model S
Año: 2012
VIN: 123456789
Propietario: Jane Doe
Metadatos: {
Fecha: 16/07/2012
}
H (T_1)

ACTUALIZAR coches
ID: 1
Propietario: John Stiles
Metadatos: {
Fecha: 03/08/2013,
}
H (T_2)

ELIMINAR coches
ID: 1
Metadatos: {
Fecha: 02/09/2016
}
H (T_3)

Procesador de Eventos (streaming)

4

Se puede exportar o transmitir datos directamente desde QLDB

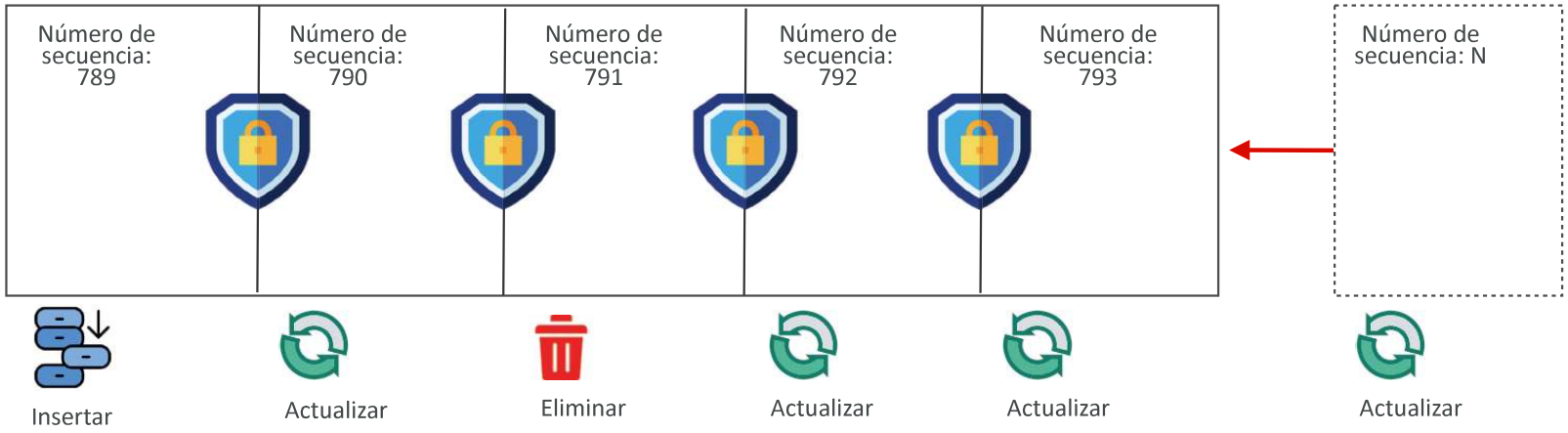
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon QLDB

- QLDB es únicamente un anexo, mantiene un registro completo de todos los cambios en los datos que no se puede modificar ni sobrescribir
- No es posible alterar los datos confirmados mediante API ni ningún otro método

Los registros no pueden alterarse



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

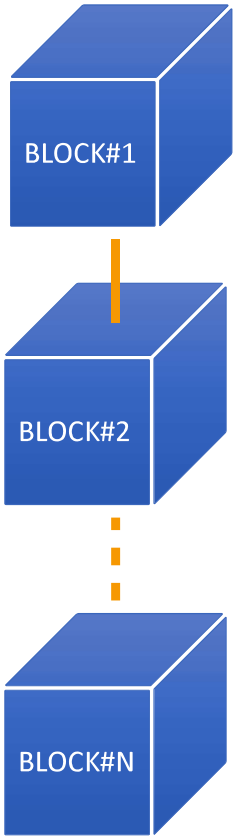
Amazon Managed Blockchain



- Blockchain permite crear aplicaciones en las que varias partes pueden ejecutar transacciones **sin necesidad de una autoridad central de confianza**
- Facilita la creación y gestión de redes blockchain escalables utilizando tecnologías populares como **Hyperledger Fabric** y **Ethereum**
- Amazon Managed Blockchain es un servicio gestionado para:
  - Unirte a redes públicas de blockchain
  - O crear tu propia red privada escalable

ethereum HYPERLEDGER FABRIC

- Ajusta automáticamente los recursos necesarios para satisfacer las demandas de la red, permitiendo un rendimiento óptimo sin intervención manual
- Permite la creación de redes blockchain en múltiples regiones de AWS



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com



Docker en AWS: ECR, ECS, Fargate, EKS

www.blockstellart.com

Todos los derechos reservados © Joan Amengual www.blockstellart.com

¿Qué es Docker?

- Docker es una plataforma de desarrollo de software para desplegar aplicaciones
- Las aplicaciones se empaquetan en **contenedores** que pueden ejecutarse en cualquier sistema operativo
- **Las aplicaciones se ejecutan igual, independientemente de dónde se ejecuten**
  - Cualquier máquina
  - No hay problemas de compatibilidad
  - Comportamiento predecible
  - Menos trabajo
  - Más fácil de mantener y desplegar
  - Funciona con cualquier lenguaje, cualquier sistema operativo y cualquier tecnología
- Amplía y reduce los contenedores muy rápidamente (en segundos)



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Docker en un sistema operativo

- Los contenedores de Docker se ejecutan igual en cualquier infraestructura:
  - Máquina local
  - Centro de datos corporativo
  - Cloud
- Son ligeros
  - En comparación con las máquinas virtuales
- Docker proporciona aislamiento para los contenedores



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Conceptos clave sobre Docker



Dockerfile - Es un archivo de texto con las instrucciones necesarias para crear una imagen (algo similar a un plano de construcción)



Imagen - Es un archivo construido por capas, que contiene todas las dependencias para ejecutarse



Contenedor - Es la instancia de una image en un ambiente aislado



Portable - Autónomo, siempre funciona como se espera



Ligero - Se utiliza el sistema operativo principal y se comparten las capas del sistema de archivos.

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Almacenamiento de imágenes de Docker

- Las imágenes Docker se almacenan en **repositorios Docker**





Amazon ECR

- **Docker Hub (https://hub.docker.com)**
  - Repositorio **público**
  - Repositorio **privado (para planes Pro, Team y Business)**
  - Imágenes base para muchas tecnologías o sistemas operativos (por ejemplo, Ubuntu, MySQL, ...)

- **Amazon ECR (Registro elástico de contenedores de Amazon)**
  - Repositorio **público** (**Galería pública de Amazon ECR https://gallery.ecr.aws**)
  - Repositorio **privado**

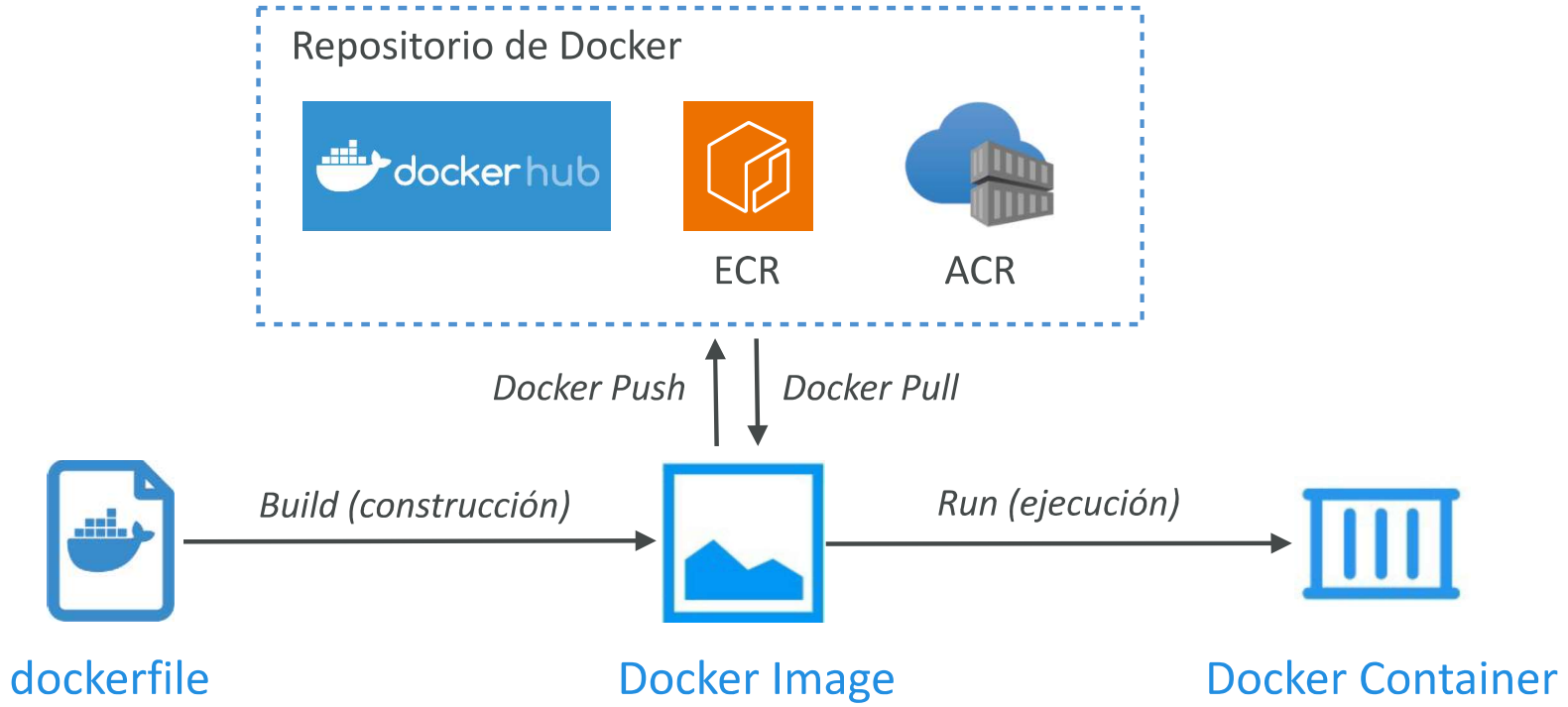


- **Azure Container Registry (ACR)**

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Conceptos clave sobre Docker



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

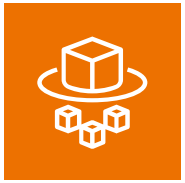
Gestión de contenedores Docker en AWS



Amazon Elastic Container Service (Amazon ECS):
Plataforma de contenedores propia de Amazon



Servicio Amazon Elastic Kubernetes (Amazon EKS):
Kubernetes administrado por Amazon (código abierto)



AWS Fargate: Plataforma de contenedores sin servidor propia de Amazon y funciona con ECS y con EKS



Amazon ECR: Almacena imágenes de contenedores

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Visión general de Amazon ECS

- **Amazon ECS es un servicio de orquestación de contenedores completamente administrado que facilita la implementación, la administración y el escalado de aplicaciones en contenedores.**

*Lanzar contenedores Docker en AWS = Lanzar **tareas ECS** en clústeres ECS

- Tipos de lanzamiento de contenedores con Amazon ECS:
  - **Instancias EC2**: debe aprovisionar y mantener la infraestructura (las instancias EC2)
  - **Fargate**: No aprovisionas la infraestructura (no hay instancias EC2 que administrar)

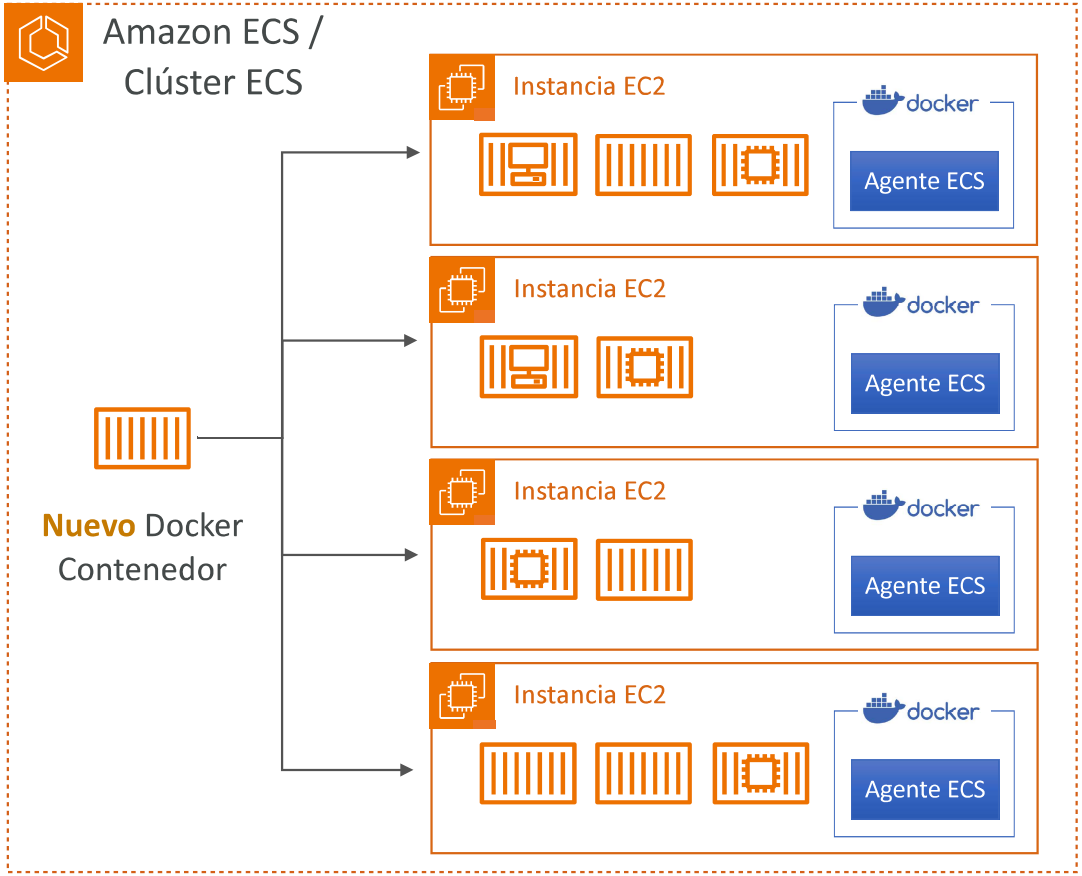


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Lanzamiento de ECS usando instancias EC2

- **Tipo de lanzamiento EC2: debe aprovisionar y mantener la infraestructura (las instancias EC2)**
- Puedes seleccionar el tipo de instancia EC2 y la cantidad que mejor se adapte a tus necesidades de carga de trabajo y rendimiento
- Tienes control sobre la configuración del sistema operativo y la capacidad de instalar software personalizado en las instancias
- Cada Instancia EC2 debe ejecutar el Agente ECS para registrarse en el Cluster ECS
- AWS se encarga de iniciar / detener los contenedores

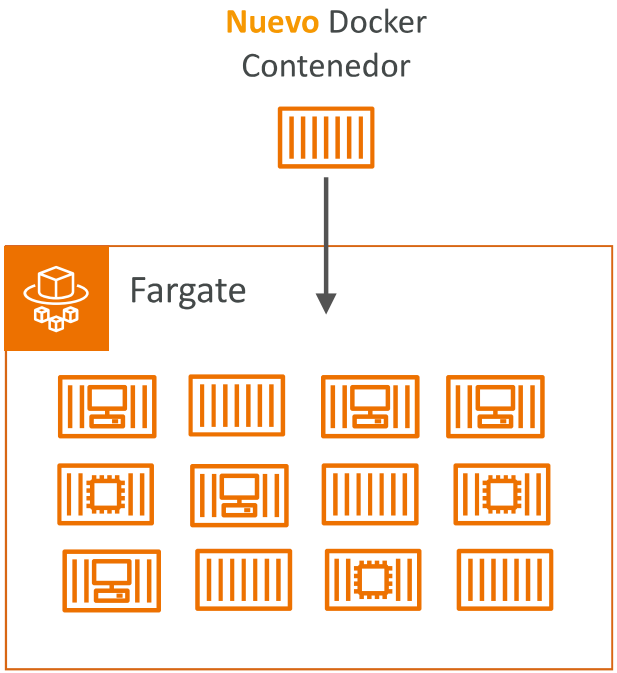


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Lanzamiento de ECS usando Fargate

- **No aprovisionas la infraestructura**
- **No hay instancias EC2 que administrar**
- **Todo es Serverless (sin servidor)**
- Sólo tienes que crear definiciones de tareas
- Fargate permite escalar automáticamente y pagar por uso
- AWS ejecuta las tareas ECS por ti en función de la CPU / RAM que necesites
- Fargate aísla cada contenedor, mejorando la seguridad en la ejecución de tus aplicaciones



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon ECS – Tareas

- **Se trata de un archivo de texto en formato JSON que describe los parámetros y uno o varios contenedores que forman la aplicación.**
- Entre los parámetros que se pueden especificar en una definición de tareas se incluyen los siguientes:
  - El **tipo de lanzamiento** a utilizar, que determina la infraestructura en la que se alojan las tareas
  - La **imagen de Docker** que se va a utilizar con cada contenedor en su tarea
  - La cantidad de **CPU y de memoria** que se va a utilizar con cada tarea o cada contenedor dentro de una tarea
  - El modo de **red de Docker** que utilizar para los contenedores
  - Los **volúmenes de datos** que se utilizan con los contenedores en la tarea
  - El **rol de IAM** que las tareas utilizan
  - Comportamiento de la tarea si el contenedor finaliza o falla
  - El comando que el contenedor ejecuta al iniciarse



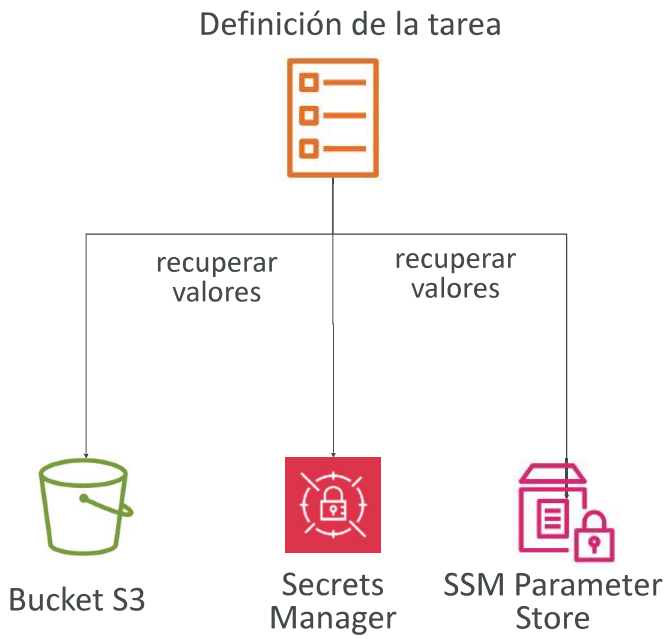
```
{
  "family": "my-task",
  "containerDefinitions": [
    {
      "name": "my-container",
      "image": "my-image",
      "memory": 512,
      "portMappings": [
        {
          "containerPort": 80,
          "hostPort": 80,
          "protocol": "tcp"
        }
      ],
      "essential": true
    }
  ]
}
```

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon ECS – Tareas con variables de entorno

- Variable de entorno
  - **Hardcoded** - por ejemplo, URLs
  - **Almacén de Parámetros SSM** - variables sensibles (por ejemplo, claves API, configuraciones compartidas)
  - **Secrets Manager** - variables sensibles (por ejemplo, contraseñas de BD)
- Archivos de entorno (masivos) - Amazon S3




```
{
  "family": "my-task",
  "containerDefinitions": [
    {
      "name": "my-container",
      "image": "my-image",
      "memory": 512,
      "environment": [
        {
          "name": "ENV_VAR_1",
          "value": "value1"
        },
        {
          "name": "ENV_VAR_2",
          "value": "value2"
        }
      ],
      "portMappings": [
        {
          "containerPort": 80,
          "hostPort": 80,
          "protocol": "tcp"
        }
      ],
      "essential": true
    }
  ]
}
```

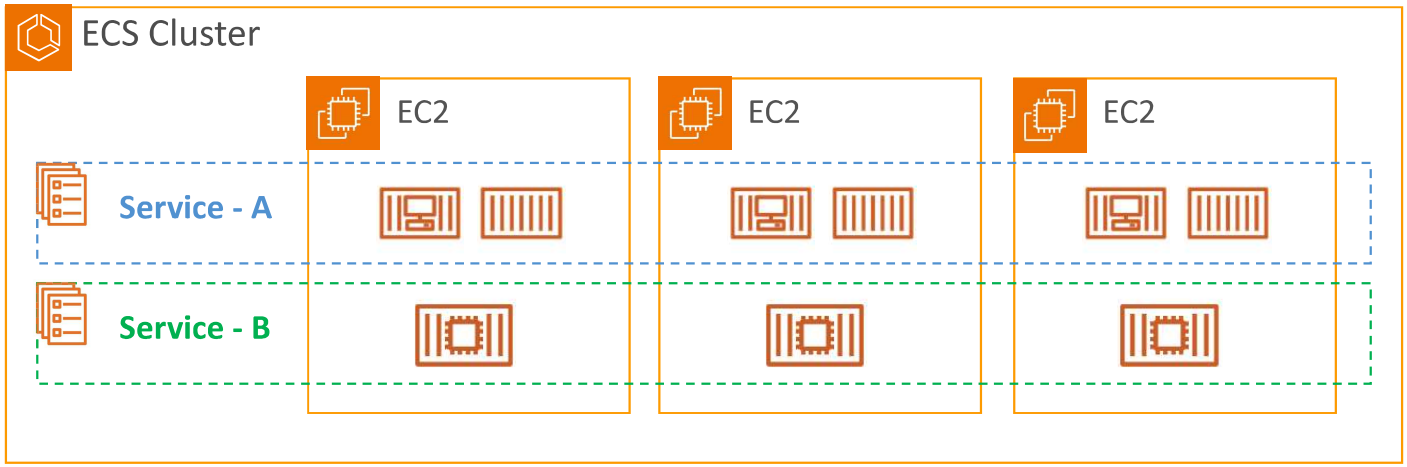
© Joan Amengual

Todos los derechos reservados © Jo

Amazon ECS – Servicios



- Definen cuántas tareas deben ejecutarse y cómo deben ejecutarse
- Garantizan que el número de tareas deseado se ejecuta en toda nuestra flota de instancias EC2
- Pueden vincularse a Elastic Load Balancer (ALB o NLB) si es necesario
- ¡Vamos a crear nuestro primer servicio!



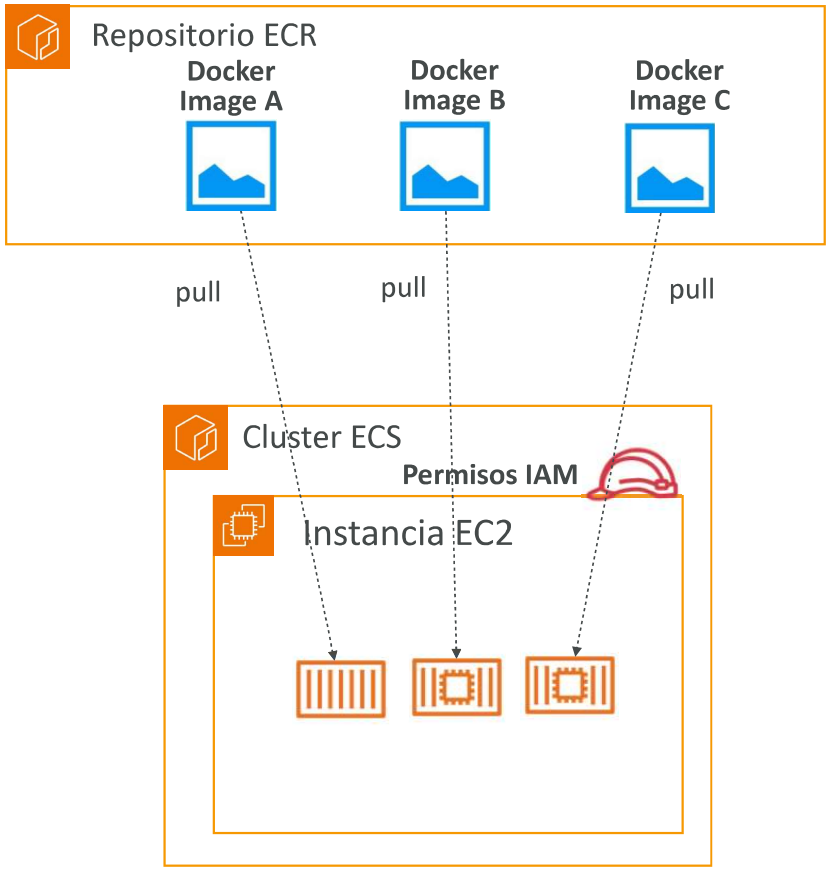
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon ECR

- ECR = Elastic Container Registry
- Almacenar y administrar imágenes Docker en AWS
- Repositorio **privado** y **público** (**Amazon ECR Public Gallery** https://gallery.ecr.aws)

- Totalmente integrado con ECS, respaldado por Amazon S3
- El acceso se controla a través de IAM (errores de permiso => política)
- Soporta escaneo de vulnerabilidades de imágenes, versionado, etiquetas de imágenes, ciclo de vida de imágenes, ...



Guardamos las imágenes de docker en uno o varios repositorios de Amazon ECR

Lanzamiento de los contenedores de docker en instancias EC2 o Fargate

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon ECR - Uso de la CLI de AWS

- **Comando de inicio de sesión**
  - CLI de AWS v2

```
aws ecr get-login password --region region | docker login --username
AWS --password-stdin aws_account_id.dkr.ecr.region.amazonaws.com
```

- **Comandos Docker**
  - Push

```
docker push aws_account_id.dkr.ecr.region.amazonaws.com/demo:latest
```

  - Pull

```
docker pull aws_account_id.dkr.ecr.region.amazonaws.com/demo:latest
```

  - **En caso de que una instancia EC2 (o tú) no pueda extraer una imagen Docker, comprueba los permisos IAM**

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Visión general de Amazon EKS

- EKS = Amazon Elastic Kubernetes
- Es una forma de lanzar **clústeres Kubernetes administrados en AWS**
- Kubernetes es un sistema de código abierto para el despliegue, escalado y gestión automáticos de aplicaciones en contenedores (normalmente Docker)
- Es una alternativa a ECS, objetivo similar pero API diferente
- EKS soporta **EC2** si quieres desplegar nodos trabajadores o **Fargate** para desplegar contenedores sin servidor
- **Caso de uso**: si tu empresa ya utiliza Kubernetes on-premises o en otra nube, y quiere migrar a AWS utilizando Kubernetes
- **Kubernetes es agnóstico a la nube (puede utilizarse en cualquier nube - Azure, GCP…)**



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon EKS - Tipos de nodos

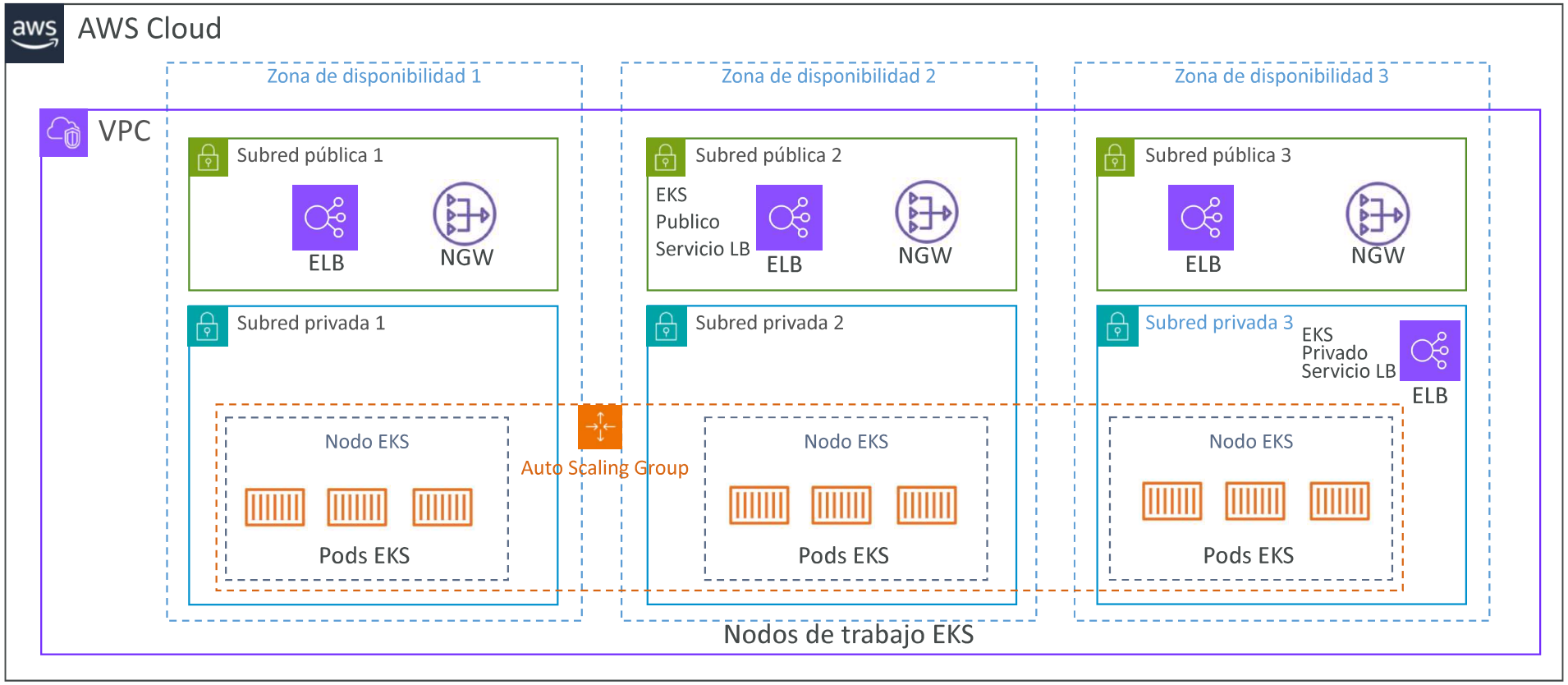
- **Grupos de nodos gestionados**
  - Crea y gestiona nodos (instancias EC2) para ti
  - Los nodos forman parte de un ASG gestionado por EKS
  - Admite instancias bajo demanda o spot (instancias puntuales)
- **Nodos autogestionados**
  - Nodos creados por ti y registrados en el clúster EKS y gestionados por un ASG
  - Admite instancias bajo demanda o spot (instancias puntuales)
- **AWS Fargate**
  - No requiere mantenimiento; no se administran nodos
  - EKS lanzará los contenedores directamente en Fargate



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon EKS - Diagrama de arquitectura



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com



AWS Glue

www.blockstellart.com

Todos los derechos reservados © Joan Amengual www.blockstellart.com

¿Qué es AWS Glue?



- AWS Glue es un servicio serverless de integración de datos completamente gestionado que facilita la preparación y la carga de datos para su análisis
- Compatible con:


Amazon Redshift


Amazon DynamoDB


Amazon RDS


Amazon S3


Compatible con la mayoría de otras bases de datos SQL

- Trabajos ETL personalizados

Activados por:


Eventos


Programados


Bajo demanda

- Completamente gestionado

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Casos de uso de AWS Glue

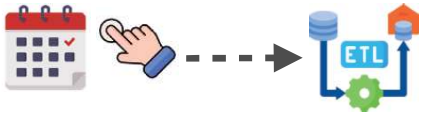
1. **Consultas en un lago de datos S3:**
   - Permite realizar análisis de datos sin la necesidad de mover los datos físicamente, aprovechando AWS Glue para la preparación de los datos



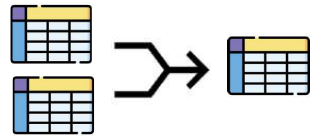
2. **Análisis de datos de registro:**
   - Proporciona la capacidad de transformar y enriquecer los datos de registro almacenados en un almacén de datos utilizando scripts ETL



3. **Pipelines ETL impulsados por eventos:**
   - Facilita la automatización de procesos ETL en respuesta a la llegada de nuevos datos, usando funciones Lambda para ejecutar trabajos de AWS Glue



4. **Vista unificada de datos:**
   - Utiliza AWS Glue Data Catalog para centralizar y gestionar metadatos, lo que permite una fácil búsqueda y descubrimiento de conjuntos de datos distribuidos en diferentes almacenes de datos

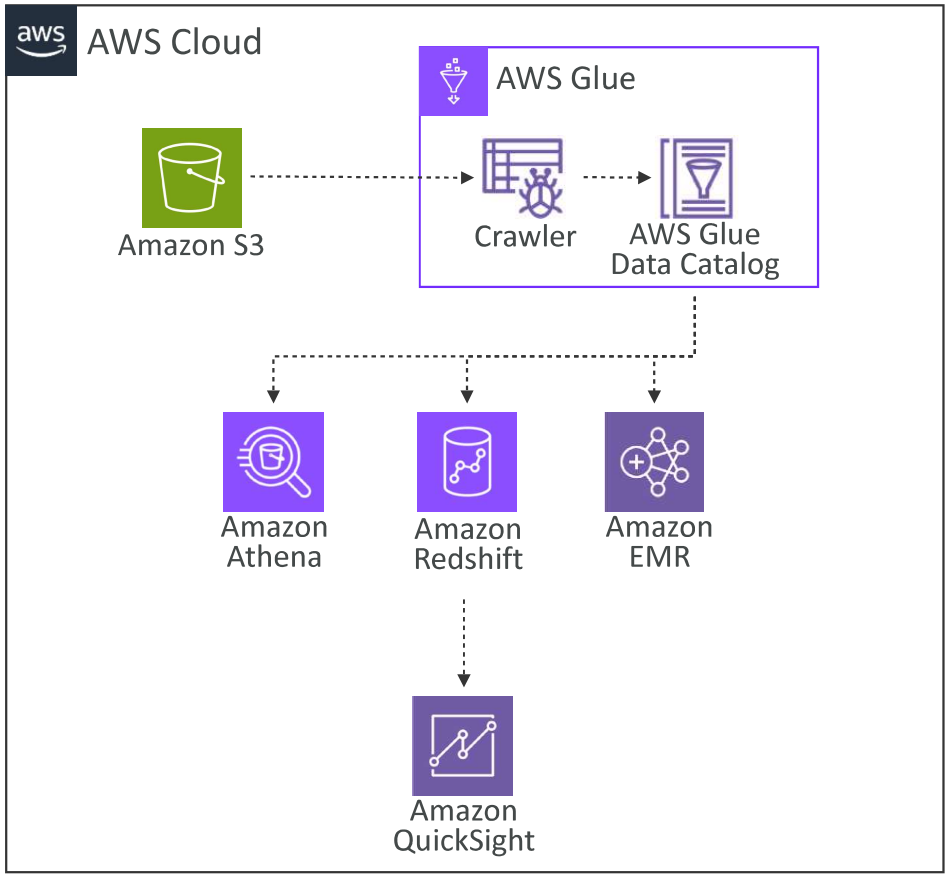


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

AWS Glue Crawler / Data Catalog

- Glue crawler escanea datos en S3, crea esquemas
- Puede ejecutarse periódicamente
- El catálogo de datos de Glue (Data Catalog):
  - Solo almacena la definición de la tabla
  - Los datos originales permanecen en S3
- Una vez catalogados, puedes tratar tus datos no estructurados como si fueran estructurados:
  - Amazon Redshift Spectrum
  - Amazon Athena
  - Amazon EMR
  - Amazon QuickSight



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

AWS Glue y Particiones S3

- El crawler de Glue extraerá particiones basadas en cómo están organizados tus datos en S3
- Piensa de antemano en cómo consultarás tu lago de datos en S3
- Por ejemplo, si el crawler detecta múltiples carpetas en un bucket de S3 que siguen una estructura de carpetas jerárquica por fecha (año, mes, día), determinará automáticamente estas carpetas como particiones



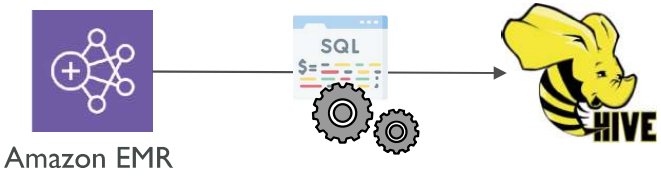
S3://ventas/año=2026/mes=Ene/día=1
S3://ventas/año=2026/mes=Ene/día=2
S3://ventas/año=2026/mes=Feb/día=1
S3://ventas/año=2026/mes=Feb/día=2

© Joan Amengual

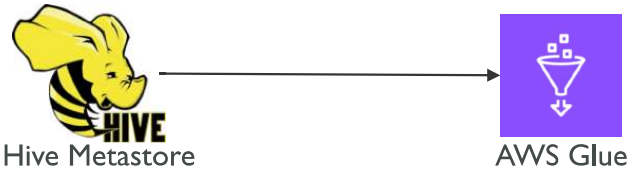
Todos los derechos reservados © Joan Amengual www.blockstellart.com

AWS Glue + Hive

- Hive facilita la **consulta y la gestión de grandes conjuntos de datos** que residen en almacenamiento distribuido utilizando SQL
  - Hive te permite ejecutar **consultas tipo SQL desde EMR**
  - El catálogo de datos de Glue puede servir como un **metastore** de Hive
  - También puedes **importar un metastore de Hive** en Glue







- *Un metastore de Hive es un sistema de almacenamiento centralizado que guarda los metadatos de las estructuras y ubicaciones de datos en un sistema de almacenamiento distribuido*

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

AWS Glue ETL

- **Facilita la preparación, transformación y carga de datos para análisis y procesamiento (Scala o Python)**
- Encriptación:
  - Del lado del servidor (en reposo)
  - SSL (en tránsito)
- Puede ser impulsado por eventos
- Puede aprovisionar **Unidades adicionales de Procesamiento de Datos (DPU's)** para aumentar el rendimiento de los trabajos de Spark subyacentes
  - Habilitar las métricas de trabajos puede ayudarte a entender la capacidad máxima en DPU's que necesitas
- Errores reportados a CloudWatch
  - Podrían integrarse con SNS para notificaciones



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

AWS Glue ETL

- Transformar datos, limpiar datos, enriquecer datos (antes de realizar el análisis)



Generar código ETL en Python o Scala, puedes modificar el código



Puedes proporcionar tus propios scripts de Spark o PySpark



El destino puede ser S3, JDBC (RDS, Redshift), o en el Catálogo de Datos de Glue

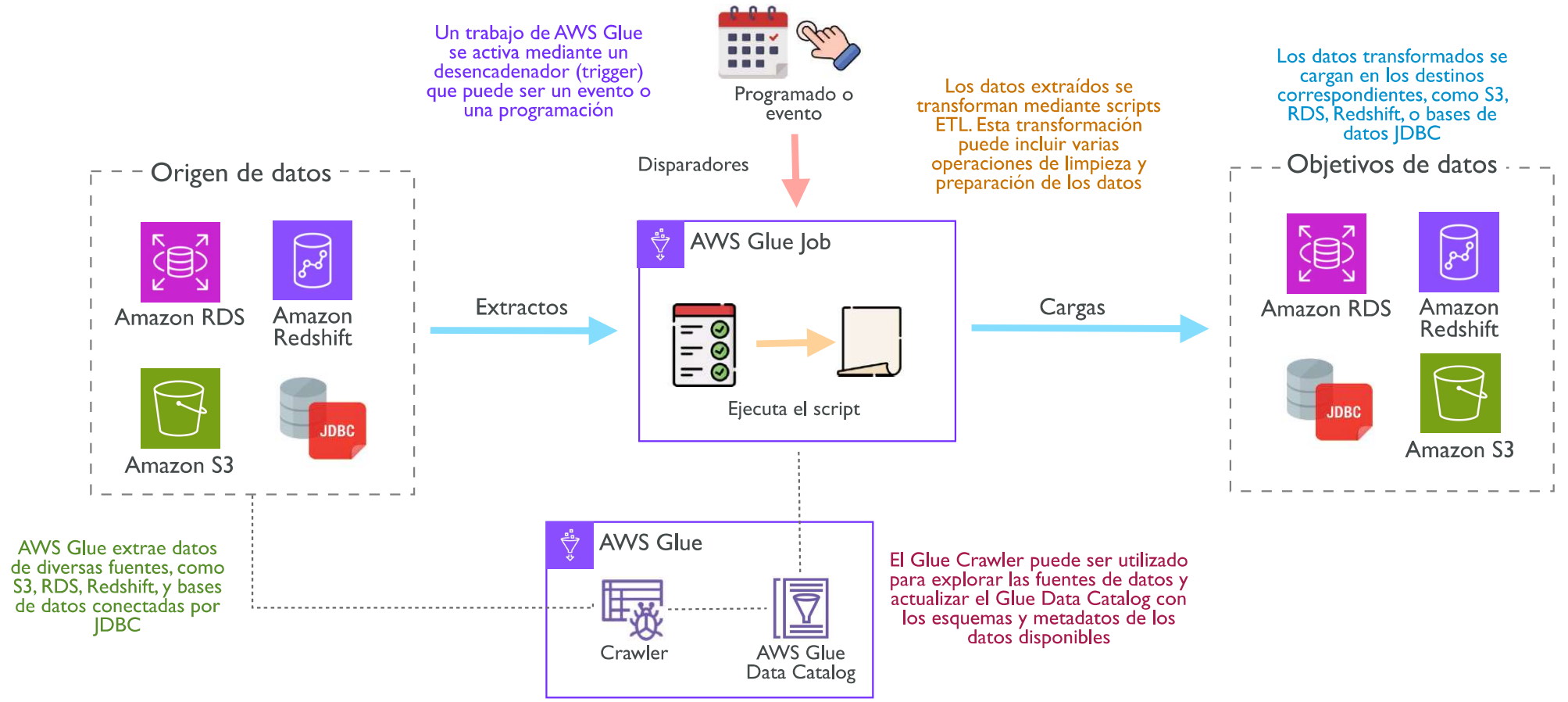
- Totalmente gestionado, rentable, paga solo por los recursos consumidos
- Los trabajos se ejecutan en una plataforma Spark sin servidor
- Glue Scheduler para programar los trabajos
- Glue Triggers para automatizar la ejecución de trabajos basados en "eventos"



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Diagrama del uso de AWS Glue ETL



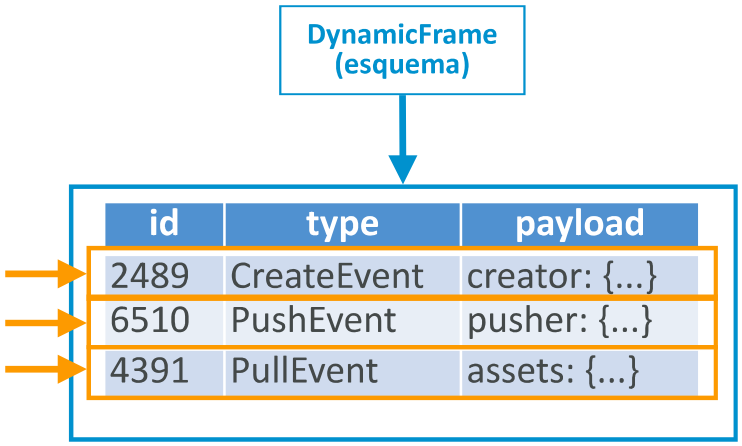
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

AWS Glue ETL - DynamicFrame

- Un **DynamicFrame** es un marco que maneja datos heterogéneos. Básicamente una colección de **DynamicRecords**
  - Los DynamicRecords se describen a sí mismos, tienen un esquema
  - Muy parecido a un DataFrame de Spark, pero con más funcionalidades ETL
  - APIs en Scala y Python

El esquema del DynamicFrame incluye las columnas comunes: id, type y payload



Cada registro dinámico tiene un id, un type y un payload. El payload puede contener diferentes tipos de información dependiendo del type.

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

AWS Glue ETL - Transformaciones

- Transformaciones Incluidas:



DropFields, DropNullFields: Elimina campos (nulos)



Filter: Especifica una función para filtrar registros

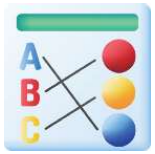


Join: Para enriquecer datos



Map: Añadir campos, eliminar campos, realizar búsquedas externas

- Transformaciones de Machine Learning:



FindMatches ML: Identifica registros duplicados o coincidentes en tu conjunto de datos

- Conversiones de formato: CSV, JSON, Avro, Parquet, ORC, XML



- Transformaciones de Apache Spark (ejemplo: K-Means):
  - Puede convertir entre Spark DataFrame y Glue DynamicFrame



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

AWS Glue ETL - ResolveChoice

- Es una función en AWS Glue ETL que maneja las ambigüedades en un *DynamicFrame*, como cuando hay múltiples campos con el mismo nombre o cuando los tipos de datos no son consistentes
- Opciones de transformación:

1. **make_cols**: Crea una nueva columna para cada tipo. Por ejemplo, dos campos con el mismo nombre
2. **cast**: Convierte todos los valores al tipo especificado
3. **make_struct**: Crea una estructura que contiene cada tipo de dato
4. **project**: Proyecta cada tipo a un tipo dado

```
"myList": [
   {
      "price": 100.00
   },
   {
      "price": "$100.00"
   }
]
```

```
Dataframe1 = df.resolveChoice(choice = "make_cols")
Dataframe2 = df.resolveChoice(specs = [("myList[].price",
"make_struct"), ("columnA", "cast:double")])
```

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Modificación del catálogo de datos

- Los scripts ETL pueden actualizar tu esquema y particiones si es necesario

1

Añadir nuevas particiones

Volver a ejecutar el crawler, o

Hacer que el script use las opciones enableUpdateCatalog y partitionKeys

2

Actualizar el esquema de la tabla

Volver a ejecutar el crawler, o

Usar enableUpdateCatalog / updateBehavior desde el script

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

AWS Glue ETL - Modificación del catálogo de datos

3

Actualizar el esquema de la tabla

- Volver a ejecutar el crawler, o
- Usar enableUpdateCatalog / updateBehavior desde el script

4

Crear nuevas tablas

- enableUpdateCatalog / updateBehavior con setCatalogInfo

5

Restricciones

- Solo S3
- Solo json, csv, avro, parquet
- Parquet requiere código especial
- Los esquemas anidados no son soportados

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Puntos finales de desarrollo de AWS Glue

- **Desarrollo de scripts ETL en Notebooks:** AWS Glue permite desarrollar scripts ETL en entornos de notebooks, como:
  - Apache Zeppelin y SageMaker
- Una vez que el script está listo, se puede crear un trabajo ETL que lo ejecute usando Spark y Glue





- **Conexión al Endpoint en una VPC:** El endpoint de desarrollo está ubicado en una VPC y está controlado por grupos de seguridad. Puedes conectarte utilizando diferentes métodos:
  - Apache Zeppelin en tu máquina local
  - Servidor de notebook Zeppelin en EC2 (a través de la consola de Glue)
  - Notebook de SageMaker
  - Ventana de terminal
  - PyCharm edición profesional
  - Usa direcciones IP elásticas para acceder a una dirección de endpoint privada







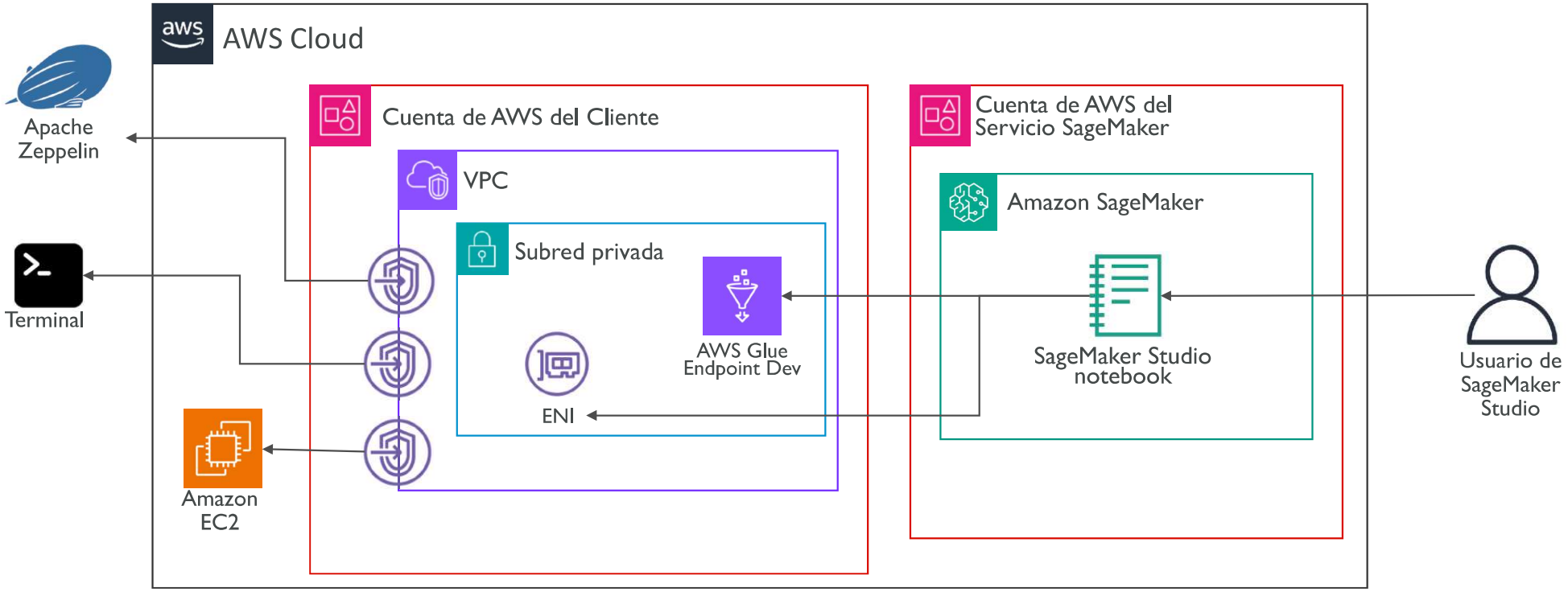




© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Puntos finales de desarrollo de AWS Glue



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

AWS Glue - Ejecución de trabajos

- Horarios basados en el tiempo (estilo cron)
- Marcadores de trabajo



Persiste el estado desde la ejecución del trabajo



Funciona con fuentes S3 en una variedad de formatos



Previene el reprocesamiento de datos antiguos



Funciona con bases de datos relacionales a través de JDBC (si las claves primarias están en orden secuencial)



Permite procesar solo datos nuevos al volver a ejecutar un trabajo en un horario



Solo maneja nuevas filas, no filas actualizadas

- Eventos de CloudWatch



Dispara una función Lambda o una notificación SNS cuando un trabajo ETL tiene éxito o falla



Invoca la ejecución de una instancia EC2, envía eventos a Kinesis, activa una función Step Function

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

AWS Glue - Modelo de costos

- **Facturación por segundo** para crawlers y trabajos ETL



- El **primer millón de objetos** almacenados y accesos son **gratuitos** para el catálogo de datos de Glue



- Los **endpoints de desarrollo** para desarrollar código ETL **se cobran por minuto**



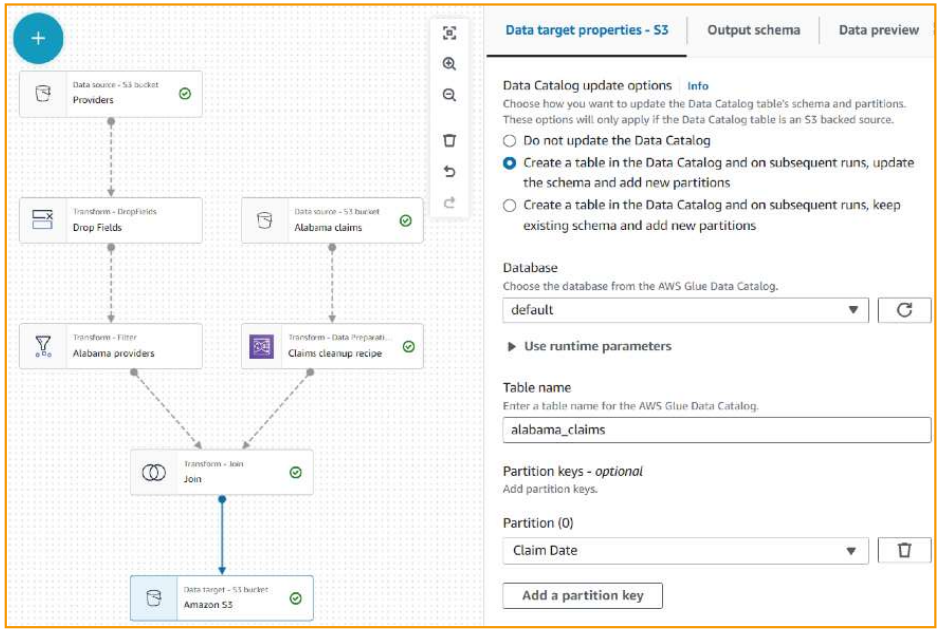


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

AWS Glue Studio

- **Interfaz visual para flujos de trabajo ETL**
- **Editor de trabajos visuales**
  - Crear DAGs (grafos acíclicos dirigidos) para flujos de trabajo complejos
  - Las fuentes incluyen S3, Kinesis, Kafka, JDBC
  - Transformar / muestrear / unir datos, etc…
  - Destino a S3 o Glue Data Catalog
  - Soporta particionamiento
- **Panel de control visual de trabajos**
  - Vistas generales, estado, tiempos de ejecución

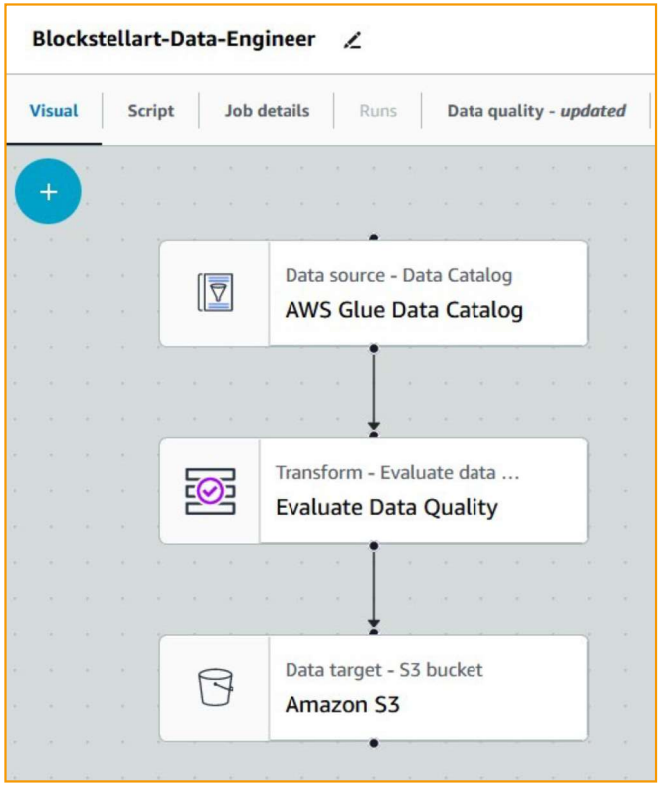


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Calidad de datos de AWS Glue

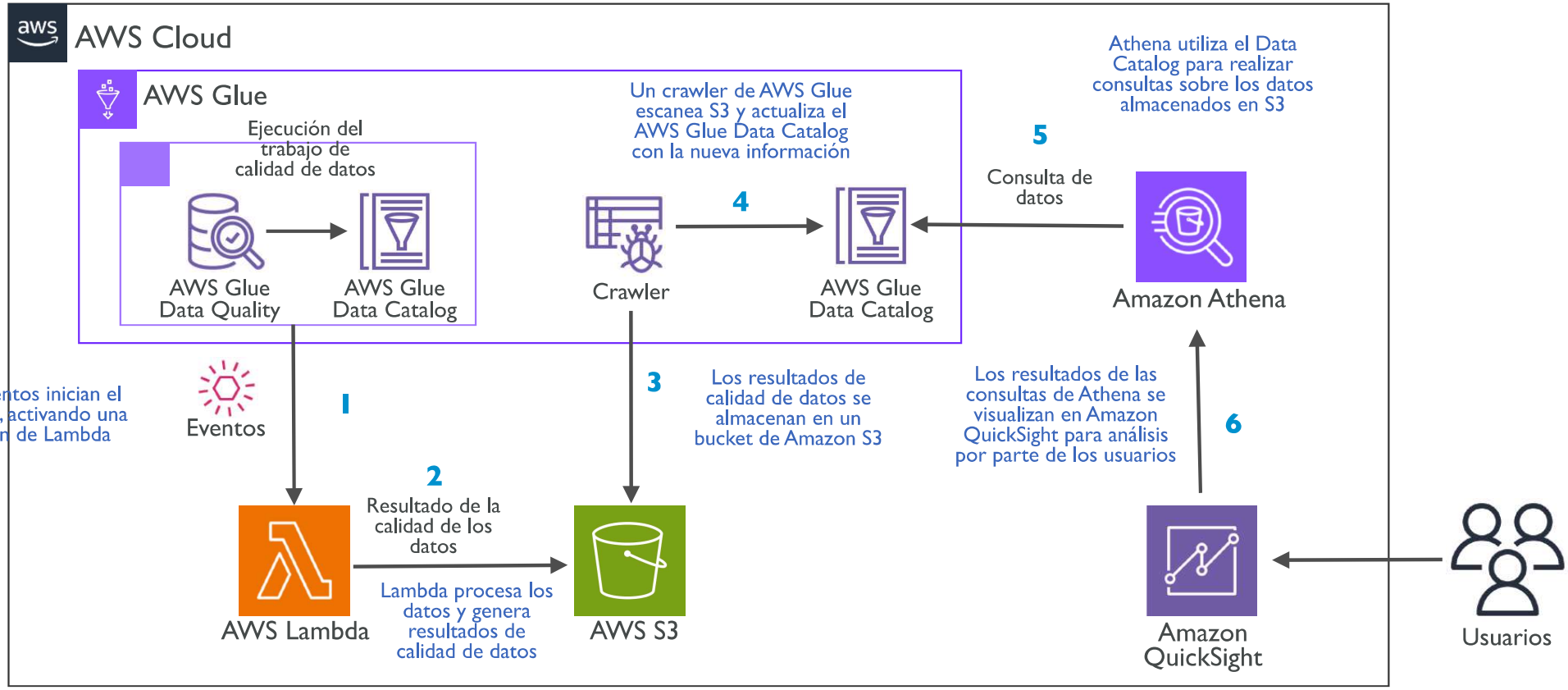
- Las reglas de calidad de datos pueden ser creadas manualmente o recomendadas automáticamente
- Se integra en los trabajos de Glue
- Utiliza el Data Quality Definition Language (DQDL)
- Los resultados pueden ser usados para fallar el trabajo, o simplemente ser reportados a CloudWatch



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Calidad de datos de AWS Glue



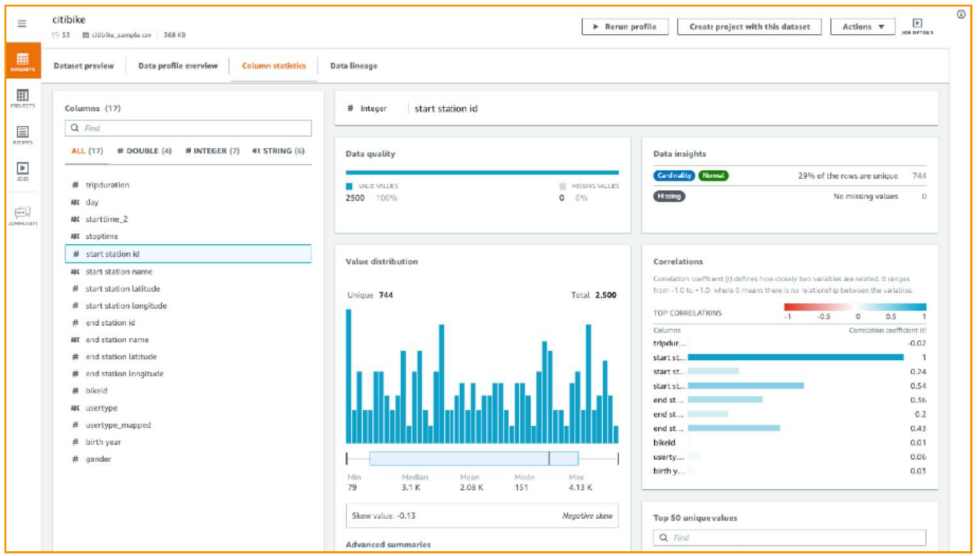
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

AWS Glue DataBrew

AWS Glue DataBrew es una herramienta que facilita la **preparación y limpieza de datos** a través de una interfaz visual intuitiva.

1. Transformaciones visuales (más de 250)
2. Recetas de transformación
3. Reglas de calidad de datos
4. Crear conjuntos de datos con SQL personalizado desde Amazon Redshift y Snowflake
5. Integración y seguridad
   - Puede integrarse con KMS (solo con claves maestras del cliente)
   - SSL en tránsito
   - IAM puede restringir quién puede hacer qué
   - CloudWatch y CloudTrail



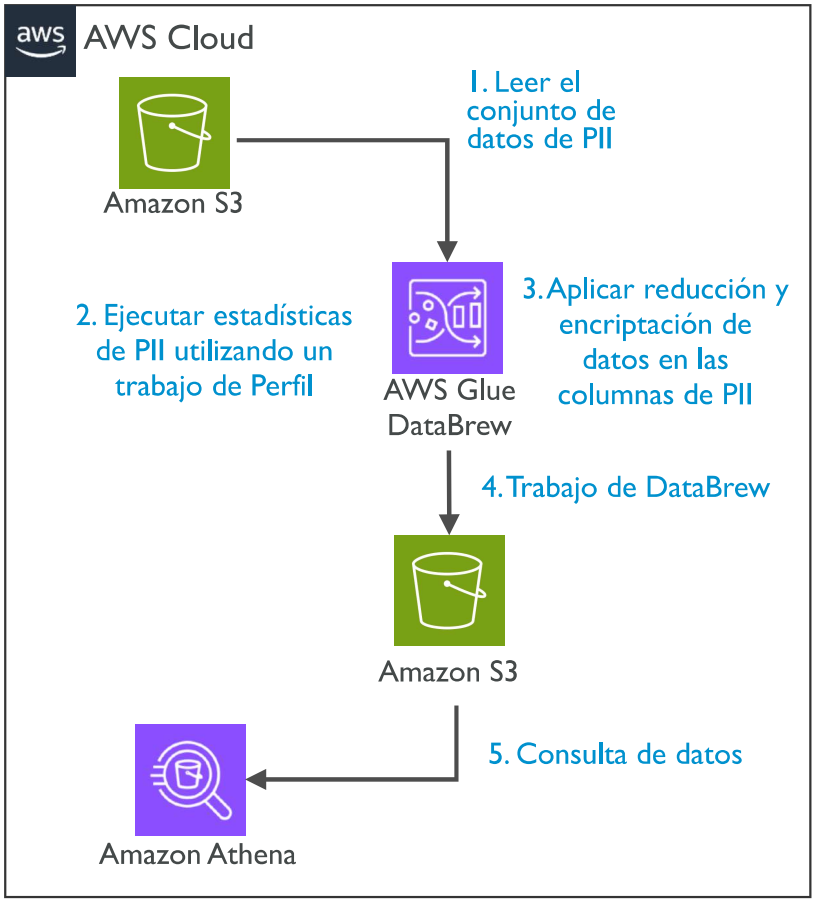
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Manejo de Información Personalmente Identificable (PII) en Transformaciones de DataBrew

AWS Glue DataBrew proporciona diversas técnicas para manejar y proteger **Información Personalmente Identificable (PII)** durante las transformaciones de datos

1. Habilitar estadísticas de PII
2. Sustitución (REPLACE_WITH_RANDOM...)
3. Mezcla (SHUFFLE_ROWS)
4. Encriptación determinística (DETERMINISTIC_ENCRYPT)
5. Encriptación probabilística (ENCRYPT)
6. Desencriptación (DECRYPT)
7. Anulación o eliminación (DELETE)
8. Enmascaramiento (MASK_CUSTOM, _DATE, _DELIMITER, _RANGE)
9. Hashing (CRYPTOGRAPHIC_HASH)

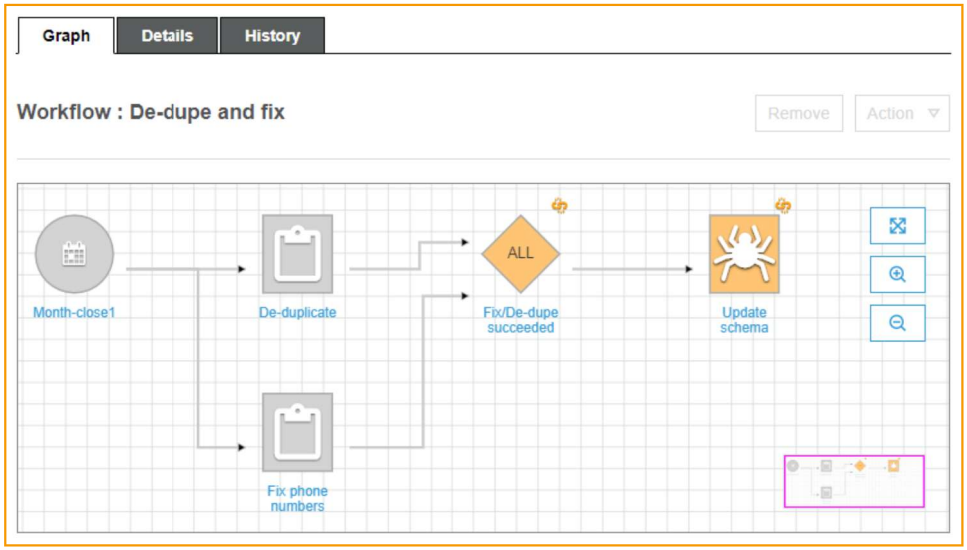


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Flujos de trabajo de AWS Glue

- Diseña procesos ETL con múltiples trabajos y múltiples crawlers que se ejecutan juntos
- Crea desde un modelo (blueprint) de AWS Glue, desde la consola o la API
- Esto es solo para orquestar operaciones ETL complejas utilizando Glue

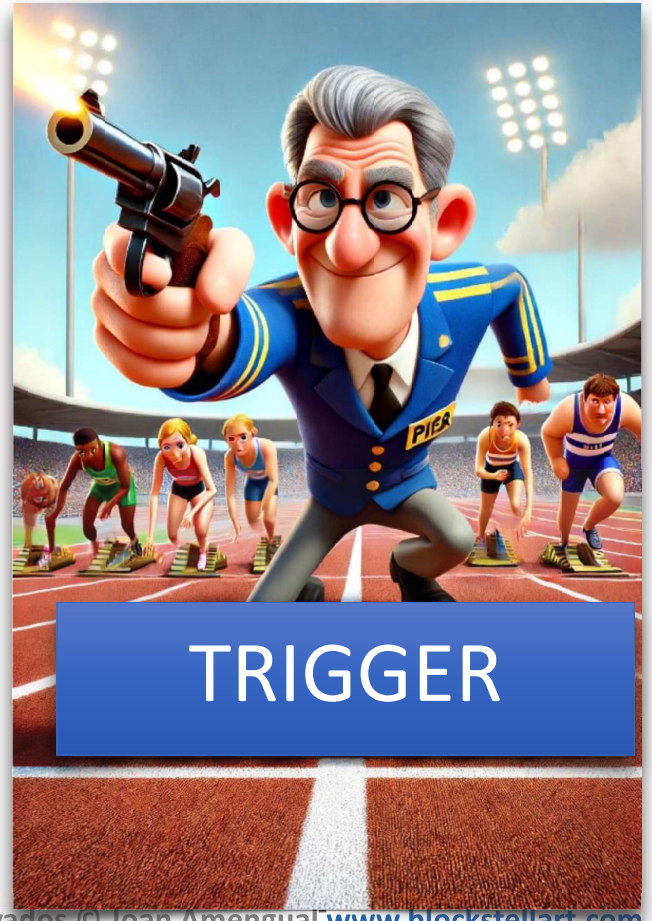


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Triggers de flujo de trabajo de AWS Glue

- **Disparadores dentro de los flujos de trabajo inician trabajos o crawlers**
  - O pueden activarse cuando los trabajos o crawlers se completan
- **Programación** (Basado en una expresión cron)
- **A demanda**
- **Eventos de EventBridge**
  - Por ejemplo, la llegada de un nuevo objeto en S3
  - Inicia en un solo evento o en un lote de eventos
  - Condiciones opcionales para el lote
    - Tamaño del lote (número de eventos)
    - Ventana del lote (dentro de X segundos, el valor predeterminado es 15 minutos)



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

AWS Lake Formation

AWS Lake Formation simplifica y acelera la **creación de un lago de datos** seguro en Amazon S3



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

AWS Lake Formation



- Permite **configurar un lago de datos seguro** de manera eficiente



- Gestión de permisos granular para un acceso seguro y específico a los datos



- Soporte para transacciones ACID en múltiples tablas, optimizando el manejo y análisis de datos



- Cifrado y filtros de datos a nivel de columna, fila y celda para proteger la información sensible



- Compatible con AWS Glue, S3, Athena, y Redshift para una gestión de datos robusta

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

AWS Lake Formation - Precios

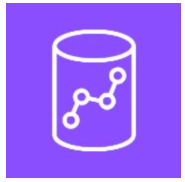
- AWS Lake Formation **no tiene costo directo,** se puede usar la herramienta sin tarifas adicionales específicas por su uso
- Pero los **servicios subyacentes** tienen costos asociados


AWS Glue


Amazon S3


Amazon Athena

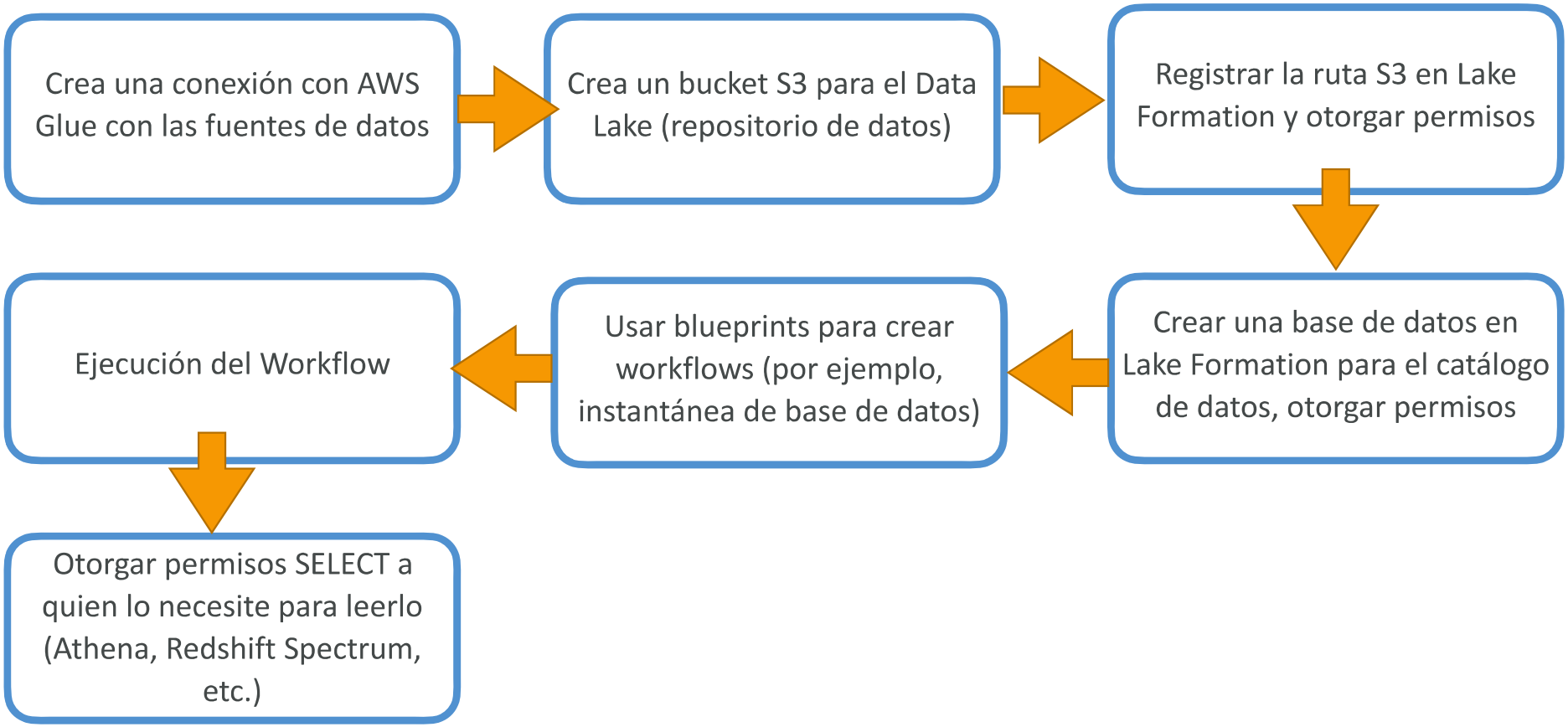

Amazon EMR


Amazon Redshift

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Construcción de un Data Lake en AWS



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

AWS Lake Formation - Detalles importantes

- **Permisos entre cuentas en AWS Lake Formation**
  - El **receptor debe estar configurado como administrador** de un Data Lake
  - Se **puede utilizar AWS Resource Access Manager** (AWS RAM) para cuentas externas a tu organización
  - **Permisos IAM** para acceso entre cuentas
- Se **necesitan permisos IAM en la clave de encriptación KMS** para catálogos de datos cifrados en Lake Formation
- **Se necesitan permisos IAM** para crear flujos de trabajo (Workflows)



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

AWS Lake Formation -Tablas gobernadas y seguridad

- Soporta **tablas gobernadas** que permiten transacciones ACID entre múltiples tablas



Nuevo tipo de tabla S3



No se puede cambiar la selección de gobernada posteriormente



Funciona con datos en streaming también (Kinesis)



Se pueden hacer consultas con Athena

- **Optimización de almacenamiento** con compactación automática
- **Control de acceso granular** con seguridad a nivel de fila y celda
  - Aplicable tanto para tablas gobernadas como S3
- Las **características mencionadas generan cargos adicionales** basados en el uso

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

AWS Lake Formation - Permisos de datos



- Se pueden vincular a usuarios/roles IAM o cuentas AWS externas



- Se pueden seleccionar permisos específicos para tablas o columnas



- Se pueden usar etiquetas de políticas en bases de datos, tablas o columnas

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com



Amazon Kinesis

www.blockstellart.com

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Visión general de Kinesis



- Facilita la **recopilación**, el **procesamiento** y el **análisis** de datos de flujo continuo en tiempo real
- Ingesta de datos en tiempo real como: Registros de aplicaciones, métricas, secuencias de clics de sitios web, datos telemétricos de IoT...

Kinesis Data Streams: captura, procesa y almacena flujos de datos



Amazon Data Firehose: carga flujos de datos en almacenes de datos de AWS



- **Managed Apache Flink:** (sucesor de Kinesis Data Analytics): analiza flujos de datos con SQL o Apache Flink



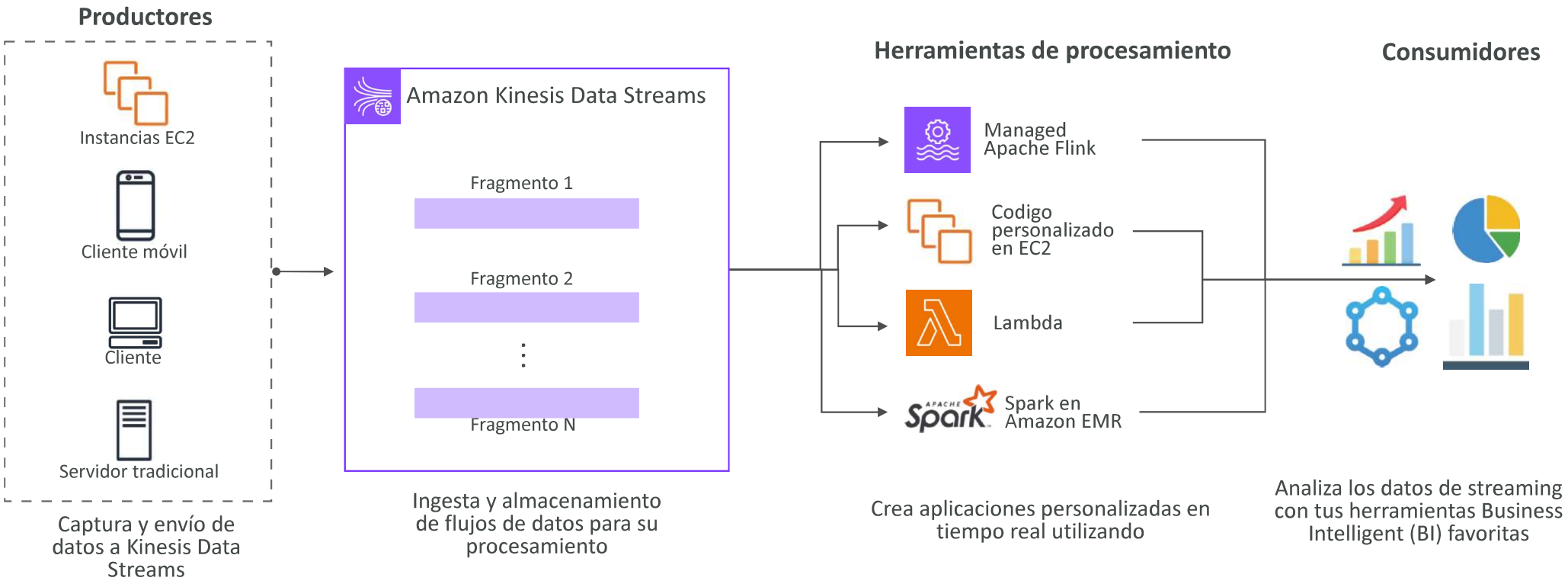
- **Kinesis Video Streams**: captura, procesa y almacena transmisiones de vídeo



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Kinesis Data Streams



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Kinesis Data Streams



- Retención entre 1 día y 365 días
- Posibilidad de volver a procesar (reproducir) los datos
- Una vez que los datos se insertan en Kinesis, no pueden borrarse (inmutabilidad)
- Los datos que comparten la misma partición van al mismo fragmento (ordenación)

- **Productores**:
  - SDK de AWS, biblioteca de productores de Kinesis (KPL), agente de Kinesis
- **Consumidores**:
  - Escribe el tuyo propio: Kinesis Client Library (KCL), AWS SDK
  - Administrados: AWS Lambda, Amazon Data Firehose, Managed Apache Flink

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Kinesis Data Streams - Modos de capacidad

1

- **Modo aprovisionado**

- Tú eliges el número de shards aprovisionados, escala manualmente o usando API
- Cada fragmento recibe 1 MB/s (o 1000 registros por segundo)
- Cada fragmento recibe 2 MB/s de salida (consumo en fan-out clásico o mejorado)
- Se paga por cada fragmento aprovisionado por hora

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Kinesis Data Streams - Modos de capacidad

2

- **Modo bajo demanda**

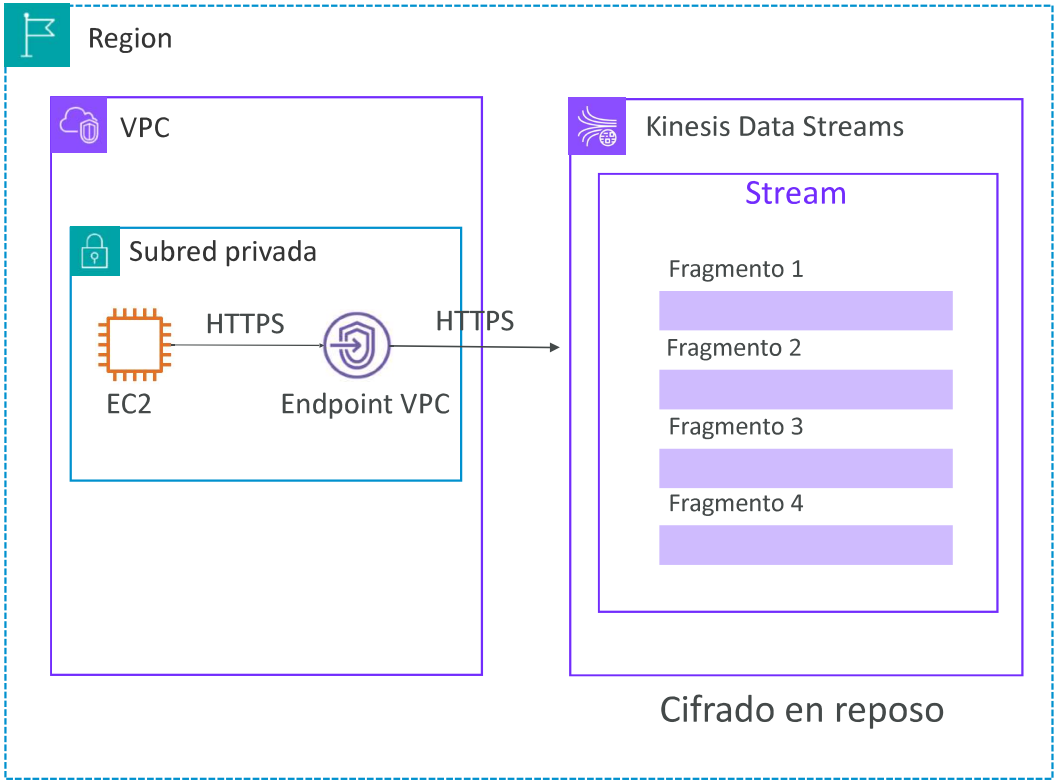
- No es necesario aprovisionar ni gestionar la capacidad
- Capacidad provisionada por defecto (4 MB/s de entrada o 4000 registros por segundo)
- Escala automáticamente en función del pico de rendimiento observado durante los últimos 30 días
- Pago por flujo por hora y entrada/salida de datos por GB

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Seguridad de Kinesis Data Streams

- Control de acceso / autorización mediante políticas IAM
- Cifrado en vuelo mediante endpoints HTTPS
- Cifrado en reposo mediante KMS
- Puedes implementar el cifrado/ descifrado de datos en el lado del cliente (más difícil)
- Endpoints VPC disponibles para que Kinesis acceda dentro de VPC



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

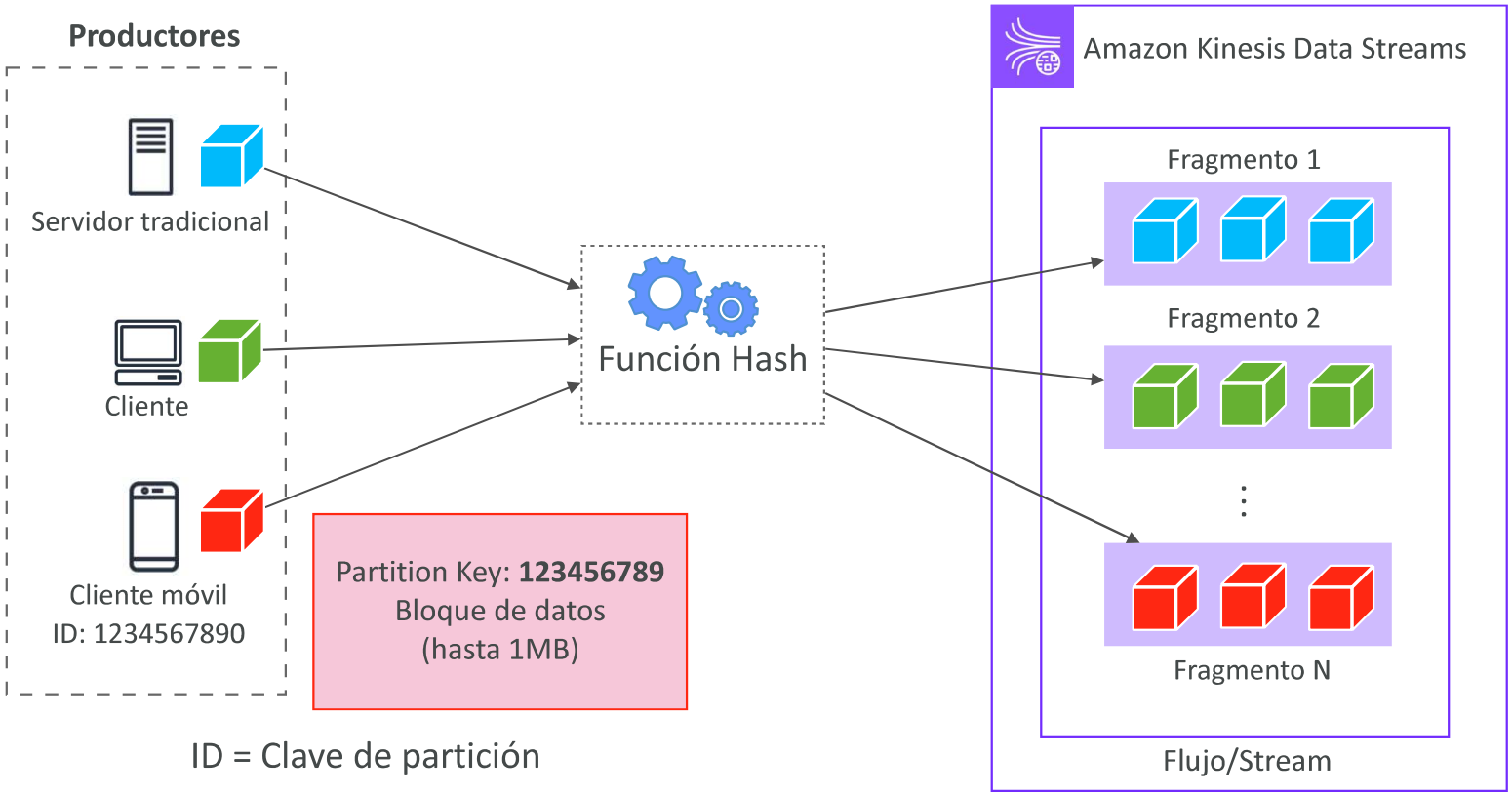
Productores Kinesis

- Coloca registros de datos en streams de datos
- El registro de datos consta de:
  - Número de secuencia (único por clave de partición dentro del fragmento)
  - Clave de partición (debe especificarse al poner los registros en el stream)
  - Bloque de datos (hasta 1 MB)
- Productores:
  - **AWS SDK**: productor simple
  - **Biblioteca de productores Kinesis (KPL)**
  - **Agente Kinesis**: monitoriza los logs
- Velocidad de escritura: 1 MB/seg o 1000 registros/seg por fragmento
- Utiliza el procesamiento por lotes con la API PutRecords para reducir costes y aumentar el rendimiento

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

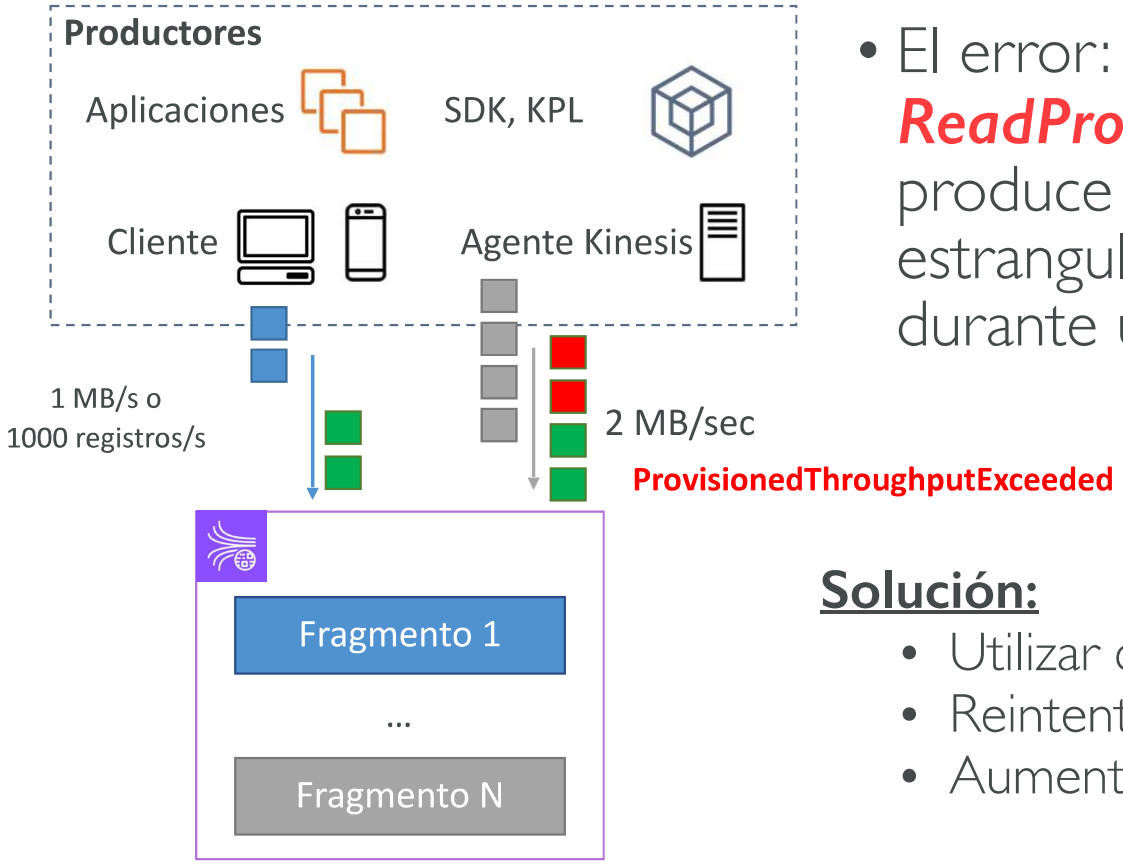
Productores Kinesis



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Kinesis - ProvisionedThroughputExceeded



- El error: ***ReadProvisionedThroughputExceeded*** se produce cuando Kinesis Data Streams estrangula las llamadas a GetRecords durante un periodo

Solución:

- Utilizar clave de partición altamente distribuida
- Reintentos con backoff exponencial
- Aumentar los fragmentos (escalado)

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Consumidores de Kinesis Data Streams



- AWS Lambda



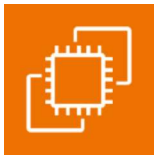
- Managed Apache Flink



- Amazon Data Firehose



- Consumidor personalizado (AWS SDK)



- Amazon EC2



- Kinesis Client Library (KCL): biblioteca para simplificar la lectura del stream de datos



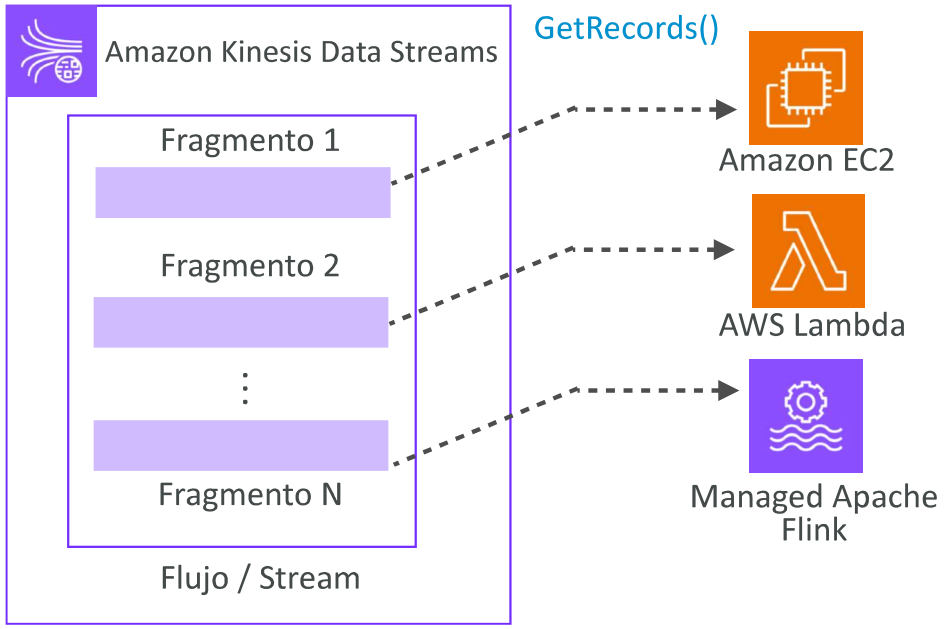
- AWS Glue

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

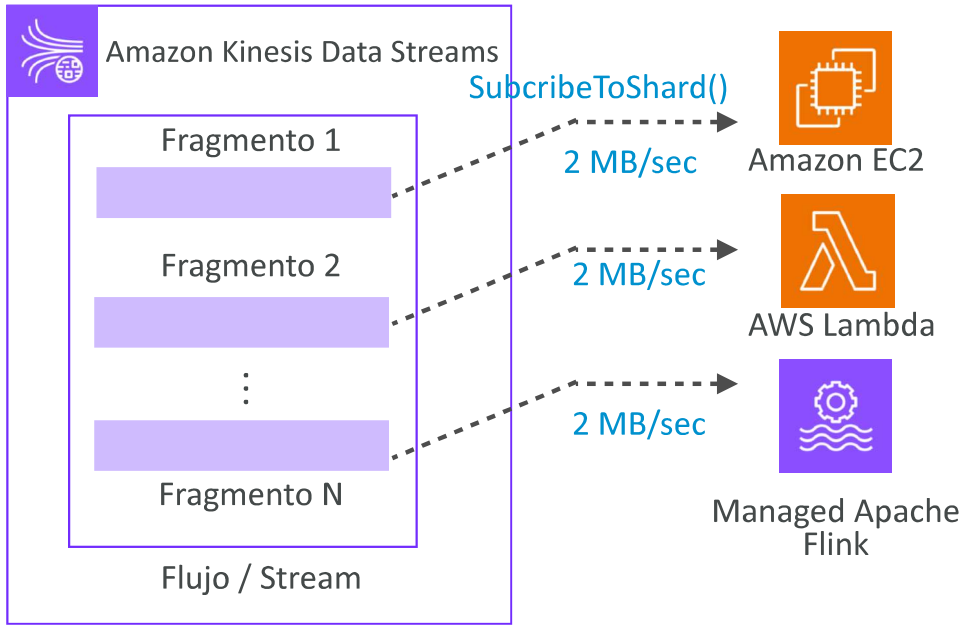
Consumidores Kinesis - Consumidor personalizado

Consumidor en Fan-out compartido (clásico)



2 MB/s por fragmento en todos los consumidores

Consumidor en Fan-out (mejorado)



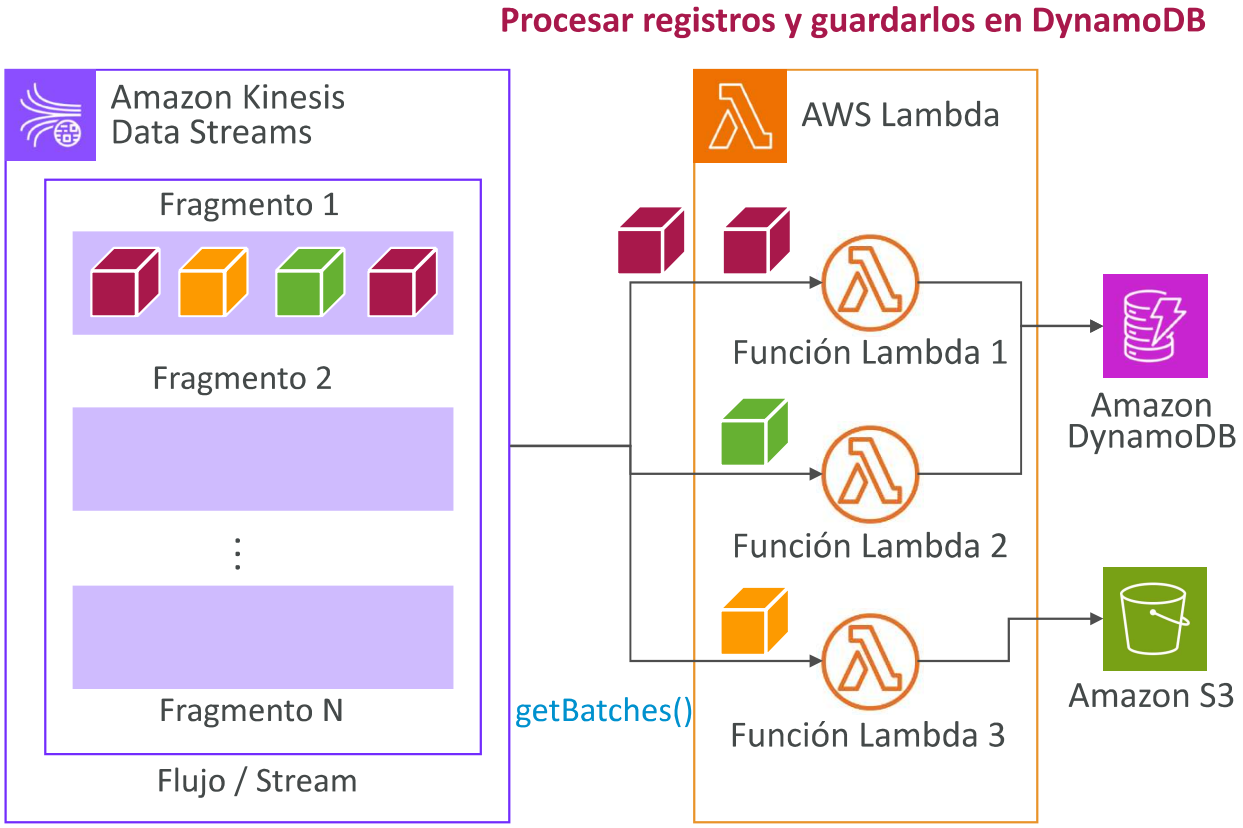
2 MB/s por consumidor por fragmento

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Consumidores Kinesis - AWS Lambda

- Soporta consumidores en Fan-Out clásico y mejorado
- Lee registros por lotes (batches)
- Puedes configurar el **tamaño del lote** y la **ventana del lote**
- Si se produce un error, Lambda reintenta hasta que tenga éxito o los datos caduquen
- Puede procesar hasta 10 lotes por fragmento simultáneamente

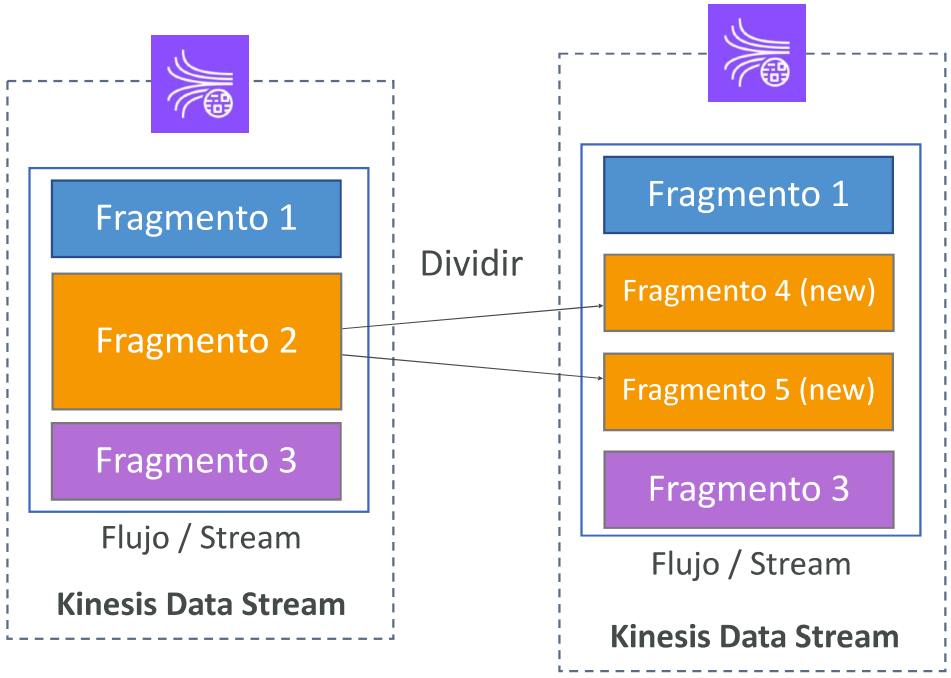


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Operación Kinesis - Dividir fragmentos

- Se utiliza para dividir un "fragmento caliente"
- El antiguo fragmento se cierra y se elimina cuando caducan los datos
- No hay escalado automático (aumenta/disminuye la capacidad manualmente)
- No se puede dividir en más de dos shards en una sola operación



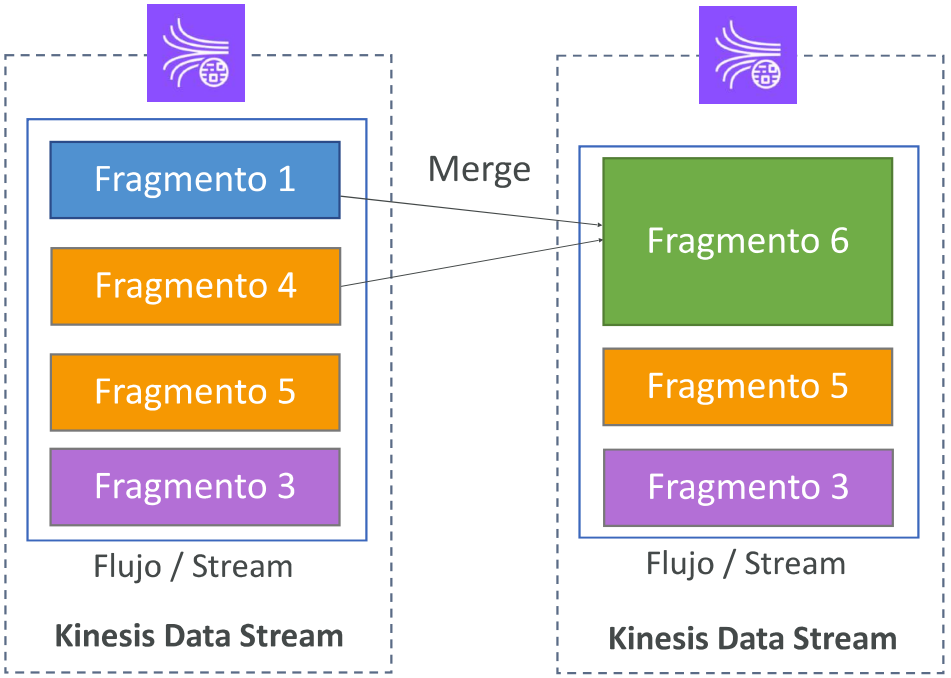
Aumentar la capacidad y el coste

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Operación Kinesis - Merge (fusión) de fragmentos

- Reduce la capacidad del stream y ahorra costes
- Puede utilizarse para agrupar dos shards con poco tráfico (shards fríos)
- Los shards antiguos se cierran y se borrarán cuando caduquen los datos
- No se pueden fusionar más de dos shards en una sola operación

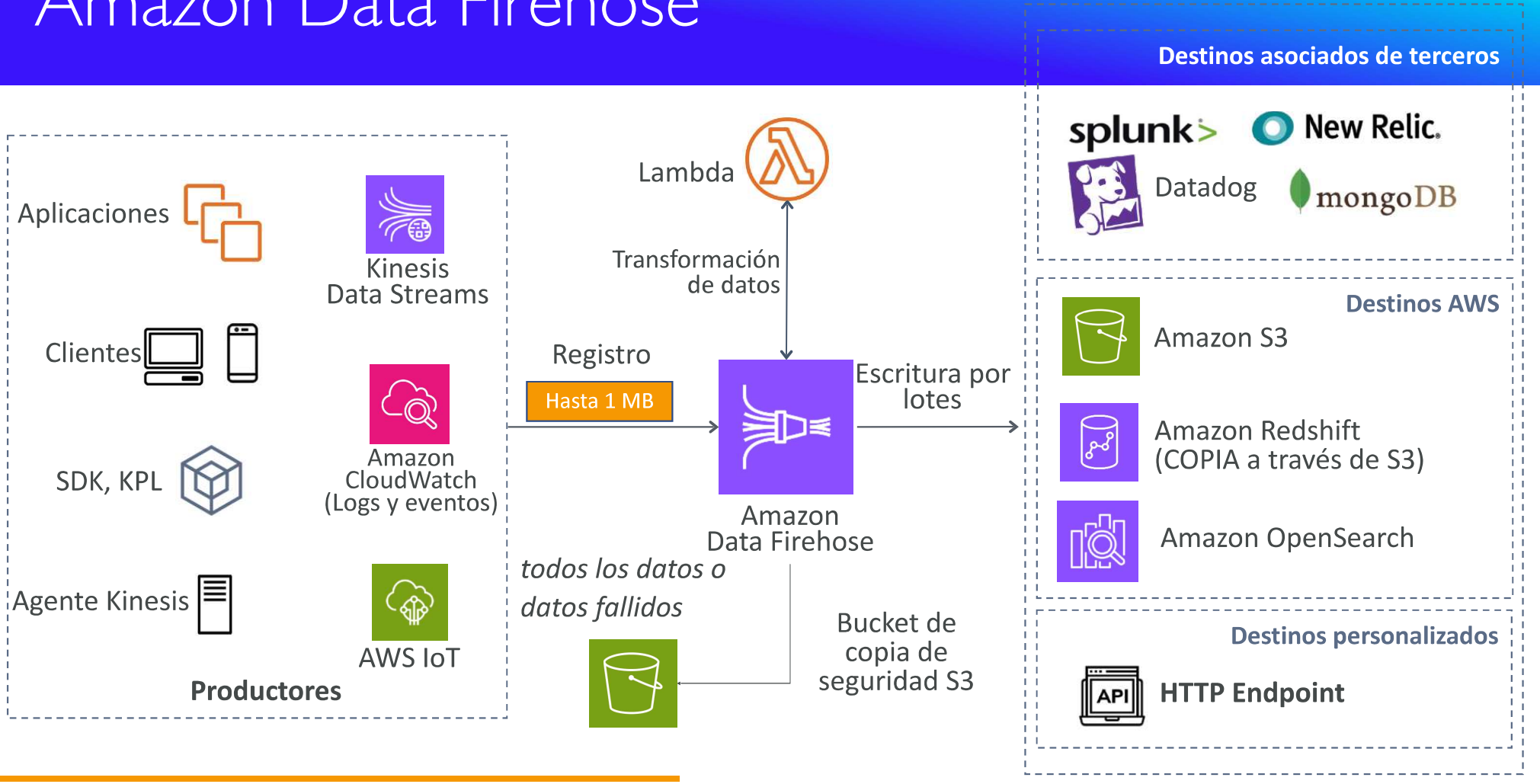


Disminuir la capacidad y el coste

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon Data Firehose



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon Data Firehose



- Servicio completamente gestionado, sin administración, escalado automático, sin servidor
  - AWS: Redshift / Amazon S3 / OpenSearch
  - Socio de terceros: Splunk / MongoDB / DataDog / NewRelic / …
  - Personalizado: envía a cualquier endpoint HTTP
- Paga por los datos que pasan por Firehose
- **Casi en tiempo real**
  - Intervalo de buffer: de 0 segundos (sin buffer) a 900 segundos
  - Tamaño del buffer: mínimo 1MB
- Soporta muchos formatos de datos, conversiones, transformaciones, compresión
- Soporta transformaciones de datos personalizadas usando AWS Lambda
- Puede enviar datos fallidos o todos los datos a un bucket S3 de respaldo

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Kinesis Data Streams vs Amazon Data Firehose



Kinesis Data Streams

- Servicio de streaming para la ingesta a escala
- Escribir código personalizado (productor / consumidor)
- En tiempo real (~200 ms)
- Gestión del escalado (división/fusión de fragmentos)
- Almacenamiento de datos de 1 a 365 días
- Capacidad de reproducción

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Kinesis Data Streams vs Amazon Data Firehose



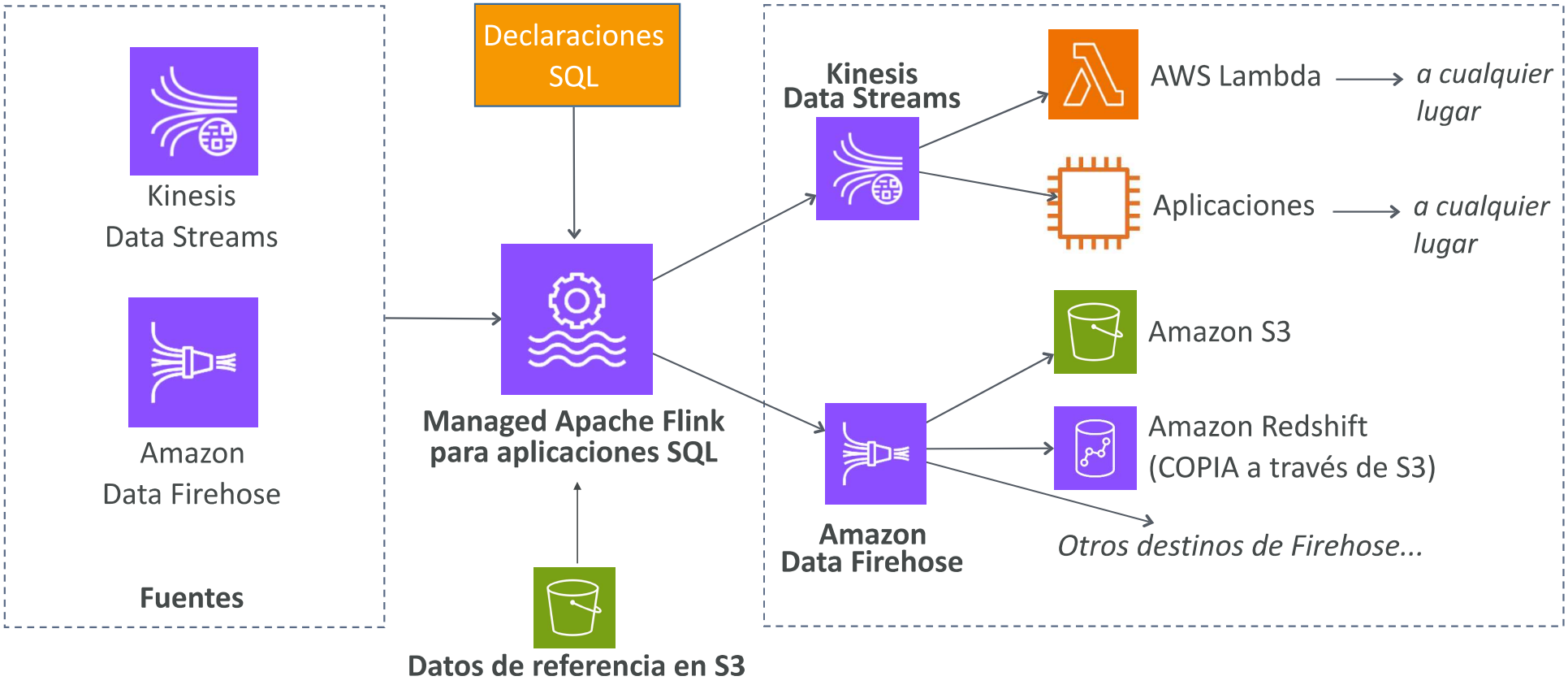
Amazon Data Firehose

- Carga de datos de streaming en S3 / Redshift / terceros / custom HTTP
- Totalmente gestionado
- Casi en tiempo real
- Escalado automático
- Sin almacenamiento de datos
- No soporta capacidad de repetición

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Managed Apache Flink (antes Kinesis Data Analytics)



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Managed Apache Flink (aplicación SQL)



- Análisis en tiempo real en **Kinesis Data Streams & Amazon Data Firehose** mediante SQL
- Añadir datos de referencia de Amazon S3 para enriquecer los datos de streaming
- Totalmente administrado, sin servidores que aprovisionar
- Escalado automático
- Paga por la tasa de consumo real
- Salida:
  - Kinesis Data Streams: crea flujos a partir de las consultas analíticas en tiempo real
  - Amazon Data Firehose: envío de resultados de consultas analíticas a destinos
- Casos de uso:
  - Análisis de series temporales
  - Dashboards en tiempo real
  - Métricas en tiempo real

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com



Amazon Athena

www.blockstellart.com

Todos los derechos reservados © Joan Amengual www.blockstellart.com

¿Qué es Amazon Athena?



- Servicio interactivo de consultas que permite analizar datos directamente en Amazon S3 utilizando SQL estándar sin necesidad de configurar servidores
  - No es necesario cargar los datos en Athena, permanecen en S3
- Servicio de consultas interactivas sin servidor
- Consultas ad-hoc sobre datos - paga solo por los datos consumidos
- **Esquema al leer** - traducción similar a una tabla
- Los datos originales **nunca se cambian** - permanecen en S3
- El esquema traduce los datos => similar a relacional cuando se leen
- Modelo de precio de pago por uso (Pay-as-you-go)
  - Ahorra mucho dinero usando formatos columnar (ORC, Parquet)

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Visión general de Amazon Athena



Admite formatos estándar de datos estructurados, semiestructurados y no estructurados

Origen de datos

El origen de datos es guardado en S3

Amazon S3

Esquema

Tabla

Tabla

Tabla

Las "tablas" se definen de antemano en un catálogo de datos y los datos se proyectan a través cuando se leen. Permite consultas similares a SQL sobre datos sin transformar los datos de origen

Amazon Athena

Athena puede leer directamente muchos formatos de datos de AWS como CloudTrail, registros de ELB y registros de flujo

La salida se puede enviar a herramientas de visualización

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Formatos de datos y casos de uso

- Soporta muchos formatos de datos:



CSV, TSV (legibles por humanos)



JSON (legible por humanos)



ORC (columnar, dividible)



Parquet (columnar, dividible)



Avro (divisible)



Zlib, LZO, Gzip

- Casos de uso:



Análisis de logs



ETL (Exracción, Transformación y Carga)



Lagos de datos

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

[DEMO] Amazon Athena



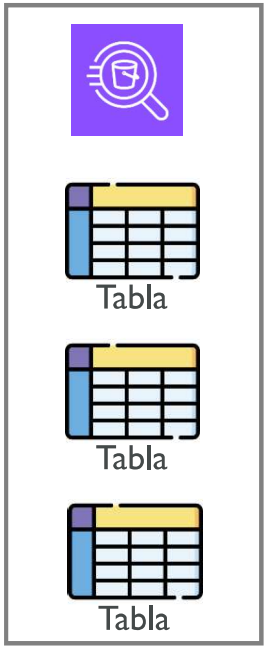
Coordenadas GPS

La facturación se basa en los datos consumidos. No hay costo base ni tiempo de configuración

```
SELECT * from planet WHERE type = 'node'
AND tags['amenity'] IN ('veterinary')
AND lat BETWEEN X AND Y
AND lon BETWEEN Z AND W;
```



Veterinários del área local



Las 'TABLAS' se definen con antelación y, a medida que se leen los datos, estos se transforman sobre la marcha en esa estructura de tabla - los datos originales no se modifican.



Amazon S3
(s3://osm-pds)



Proporciona un conjunto de datos abiertos mantenidos y entregados como un conjunto de archivos de datos grandes - no estructurados en tablas

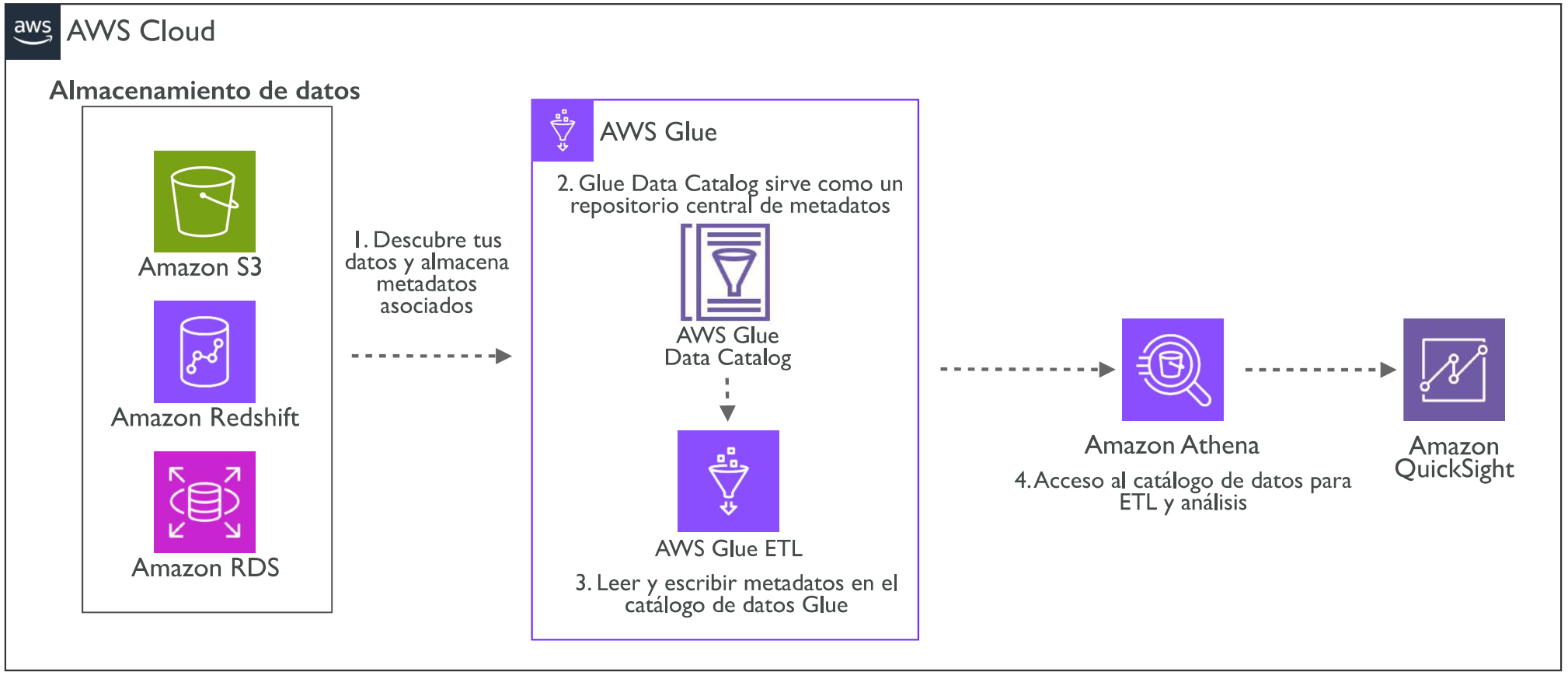


Los datos del planeta son ~100GB (en el momento de la creación) y están alojados en un bucket S3 gestionado por el proyecto

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon Athena + AWS Glue



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Grupos de trabajo de Athena

- Los grupos de trabajo de Athena se utilizan para controlar el acceso a recursos de consulta
- Cada cuenta tiene un grupo de trabajo principal y los permisos predeterminados permiten que todos los usuarios autenticados accedan a este grupo de trabajo
  - Control de acceso a las consultas y seguimiento de costos por grupo de trabajo
- Integración con IAM, CloudWatch, SNS
- Cada grupo de trabajo puede tener su propio:



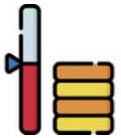
Historial de consultas



Políticas de IAM personalizadas



Configuraciones de cifrado



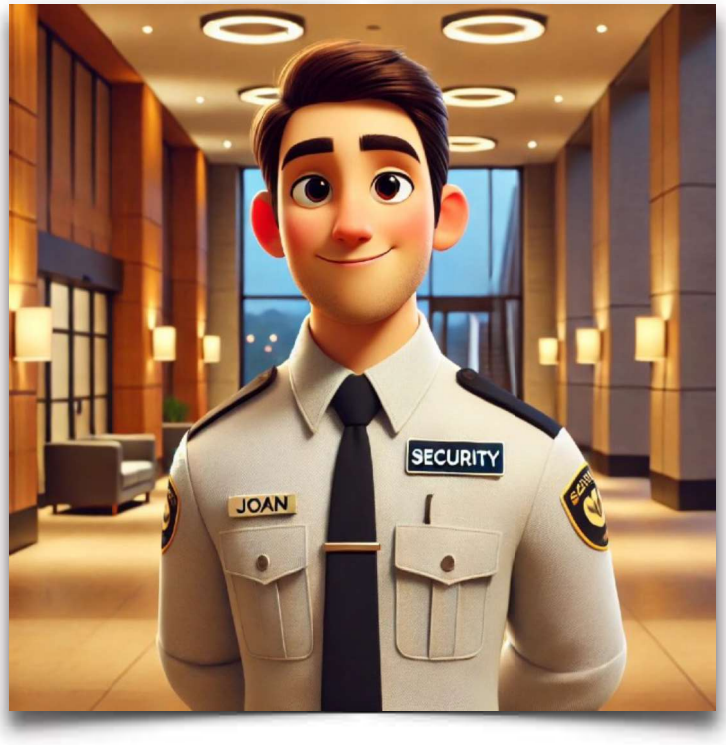
Límites de datos (puede limitar cuánto datos pueden escanear)

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Seguridad de Athena

- **Control de acceso**
  - IAM, ACLs, políticas de buckets S3
  - *AmazonAthenaFullAccess / AWSQuicksightAthenaAccess*
- Admite **opciones de cifrado** para conjuntos de datos y resultados de consulta en Amazon S3:
  - Cifrado del lado del servidor con clave administrada por S3 (SSE-S3)
  - Cifrado del lado del servidor con clave de KMS (SSE-KMS)
  - Cifrado del lado del cliente con clave de KMS (CSE-KMS)
- **Cifrado en tránsito** (TLS) entre Athena y S3
- **Monitoreo y auditoría**
  - Utiliza AWS CloudTrail para registrar todas las acciones realizadas en Athena y S3
- **Rotación de claves**
  - Implementa políticas de rotación de claves en KMS para asegurar que las claves de cifrado se actualicen regularmente

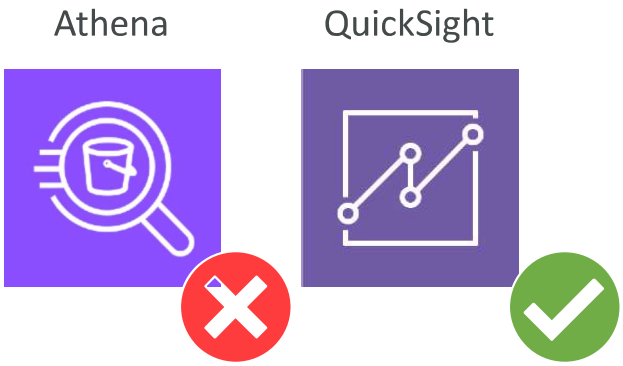


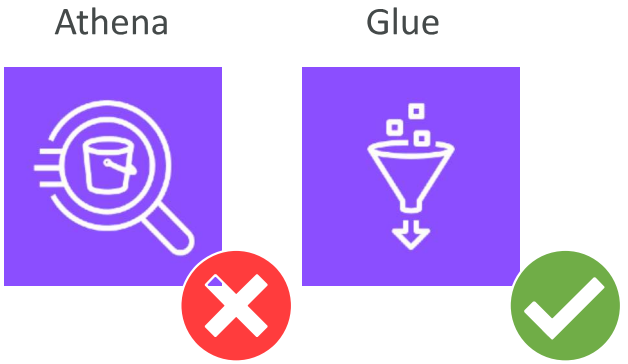
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Consideraciones de Amazon Athena

- **Informes altamente formateados / visualización**
  - Amazon **Athena no está diseñado para la creación de informes** altamente formateados o visualizaciones complejas. Si tu objetivo es generar **dashboards interactivos o informes** visuales, **Amazon QuickSight** es la herramienta adecuada.
- **ETL (Extracción, Transformación y Carga)**
  - Athena puede realizar algunas operaciones de transformación de datos, sin embargo no está optimizado para procesos ETL complejos. **AWS Glue** es la mejor opción.





© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon Athena - Optimización del rendimiento



- **Número de archivos grandes vs. archivos pequeños**

  Tener un pequeño número de archivos grandes en lugar de muchos archivos pequeños mejora el rendimiento de las consultas en Athena



- **Uso de particiones**

  Si estás gestionando particiones manualmente, utiliza el comando: *ALTER TABLE ADD PARTITION* para añadir nuevas particiones a la tabla



- **Usa datos columnar (ORC, Parquet)**

  Los formatos de almacenamiento columnar como ORC y Parquet están optimizados para consultas de análisis.

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Transacciones ACID en Amazon Athena

**ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad)*

- **Impulsadas por Apache Iceberg**

  Solo añade table_type = 'ICEBERG' en tu comando CREATE TABLE

```
CREATE TABLE nombre_tabla(...)
WITH (
  table_type = 'ICEBERG', ...);
```

- **Tablas gobernadas en Lake Formation**



Esto es útil para organizaciones que requieren control de acceso granular y cumplimiento normativo junto con capacidades de transacción.

- **Modificaciones a nivel de fila**



Los usuarios pueden realizar modificaciones a nivel de fila sin comprometer la integridad de los datos

- **Compatibilidad**

Iceberg es compatible con AWS EMR, Apache Spark y cualquier otra plataforma que soporte el formato de tabla Iceberg





- **Compactación periódica**

Es un proceso que reescribe y organiza los datos para mejorar el rendimiento de las consultas y mantener la eficiencia del almacenamiento

```
OPTIMIZE table REWRITE DATA USING
BIN_PACK WHERE catalog = 'c1';
```

- **Operaciones de viaje en el tiempo**

Permite a los usuarios recuperar estados anteriores

```
SELECT * FROM nombre_tabla FOR
SYSTEM_TIME AS OF 'timestamp';
```

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Acceso detallado de Athena al catálogo de datos de AWS Glue

- **Seguridad a nivel de BD y tabla basada en IAM**
  - Utilizar políticas de IAM para controlar el acceso a las bases de datos y tablas en el catálogo de datos
- **Limitaciones**
  - Aunque la seguridad basada en IAM es amplia, no permite restricciones a versiones específicas de tablas. Esto significa que las políticas de IAM controlan el acceso a toda la tabla y no pueden restringir el acceso a versiones particulares de los datos
- **Política mínima requerida**
  - Como mínimo, necesitas al menos una política que otorgue acceso a tu base de datos y al catálogo de datos en cada región



```
{
 "Sid": "DatabasePermissions",
 "Effect": "Allow",
 "Action": [
  "glue:GetDatabase",
  "glue:GetDatabases",
  "glue:CreateDatabase"
 ],
 "Resource": [
  "arn:aws:glue:us-east-1:123456789012:catalog",
  "arn:aws:glue:us-east-1:123456789012:database/
default"
 ]
}
```

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Acceso detallado de Athena al catálogo de datos de AWS Glue

- **Restricción de acceso a operaciones específicas**
  - Esto proporciona un control detallado sobre quién puede realizar acciones críticas como:

    ALTER o CREATE DATABASE

    CREATE TABLE

    DROP DATABASE o DROP TABLE

    MSCK REPAIR TABLE

    SHOW DATABASES o SHOW TABLES

```
{
 "Effect": "Allow",
 "Action": [
  "glue:GetDatabase",
  "glue:GetTable",
  "glue:DeleteTable",
  "glue:GetPartitions",
  "glue:GetPartition",
  "glue:DeletePartition"
 ],
 "Resource": [
  "arn:aws:glue:us-east-1:123456789012:catalog",
  "arn:aws:glue:us-east-1:123456789012:database/example_db",
  "arn:aws:glue:us-east-1:123456789012:table/example_db/test"
 ]
}
```

- Documentación adicional: https://docs.aws.amazon.com/athena/latest/ug/fine-grained-access-to-glue-resources.html

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

CREATE TABLE AS SELECT (CTAS)

- **Uso de CTAS en Amazon Athena**
  - La instrucción **CREATE TABLE AS SELECT (CTAS)** es una forma poderosa de crear nuevas tablas en Athena a partir de los resultados de una consulta SQL
  - Es una práctica común en muchas bases de datos relacionales, no solo en Athena
- **Crear subconjuntos de tablas**
  - Puedes usar CTAS para crear una nueva tabla que contenga solo un subconjunto de los datos de otra tabla
- **Conversión de formatos de datos**
  - Por ejemplo, puedes convertir datos de una tabla almacenada en formato CSV a una tabla en formato Parquet u ORC

Crear una nueva tabla en formato Parquet con compresión Snappy

```
CREATE TABLE nueva_tabla
WITH (
  format = 'Parquet',
  write_compression = 'SNAPPY'
)
AS SELECT *
FROM tabla_antigua;
```

Crear una nueva tabla en formato ORC y almacenarla en un lugar específico en S3

```
CREATE TABLE mi_orc_tabla_ctas
WITH (
  external_location = 's3://mi_resultado_athena/
mi_orc_tabla_stas/',
  format = 'ORC'
)
AS SELECT *
FROM tabla_antigua;
```

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com



Apache Spark

www.blockstellart.com

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Visión general de Apache Spark

- Motor de procesamiento de datos rápido y de código abierto



- Soporta APIs para Java, Scala, Python y R
- Ofrece procesamiento unificado para ETL, análisis interactivo, aprendizaje automático y procesamiento de gráficos
- Permite procesamiento en memoria, acelerando tareas hasta 100 veces más que Hadoop MapReduce
- Integra con Hadoop, bases de datos NoSQL y servicios en la nube

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

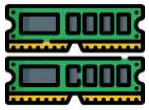
Visión general de Apache Spark

- **Marco de procesamiento distribuido para big data**



Está diseñado para manejar grandes volúmenes de datos

- Caché en memoria y ejecución de consultas optimizadas



Permite realizar operaciones de datos a alta velocidad en comparación con MapReduce tradicional

- **Soporte multilinguaje**



Soporta varios lenguajes de programación como Java, Scala, Python y R

- **Reutilización de código en múltiples tareas**



Procesamiento por lotes, consultas interactivas, análisis en tiempo real, aprendizaje automático y procesamiento de gráficos

- **No está destinado para OLTP**



Su enfoque principal es el procesamiento de datos en lotes y en tiempo real para análisis

- **Spark Streaming**





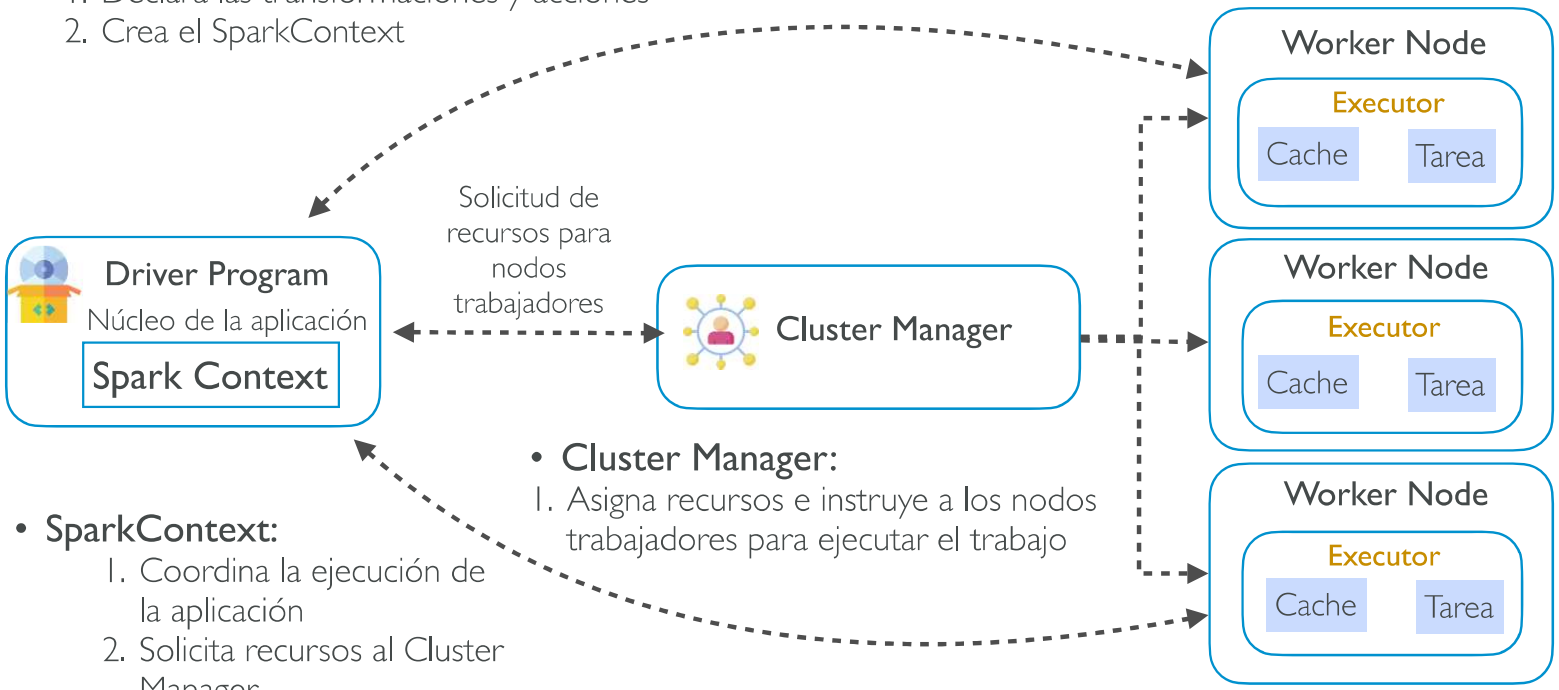
Procesamiento de datos en tiempo real. Se integra fácilmente con Amazon Kinesis y Apache Kafka

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

¿Cómo funciona Apache Spark?

- **Driver Program**
  1. Declara las transformaciones y acciones
  2. Crea el SparkContext



- **Executors**
  1. Ejecutan las tareas asignadas
  2. Almacenan los datos y resultados
  3. Mantienen la caché en memoria

- **Cluster Manager:**
  1. Asigna recursos e instruye a los nodos trabajadores para ejecutar el trabajo

- **SparkContext:**
  1. Coordina la ejecución de la aplicación
  2. Solicita recursos al Cluster Manager

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Componentes de Apache Spark



Permite a los desarrolladores ejecutar consultas SQL en los datos almacenados en DataFrames y Datasets



Permite el procesamiento de flujos de datos en tiempo real



Proporciona varias implementaciones de algoritmos ML, incluyendo clasificación, regresión, clustering, filtrado colaborativo y minería de patrones

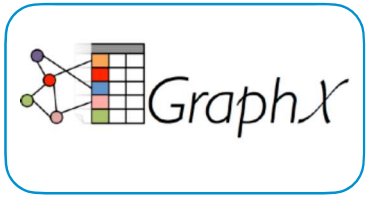


Permite a los usuarios de R aprovechar la capacidad de procesamiento distribuido de Spark para manipular y analizar grandes conjuntos de datos

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Componentes de Apache Spark



Permite a los desarrolladores realizar cálculos iterativos y consultas en gráficos grandes

Apache Spark Core API

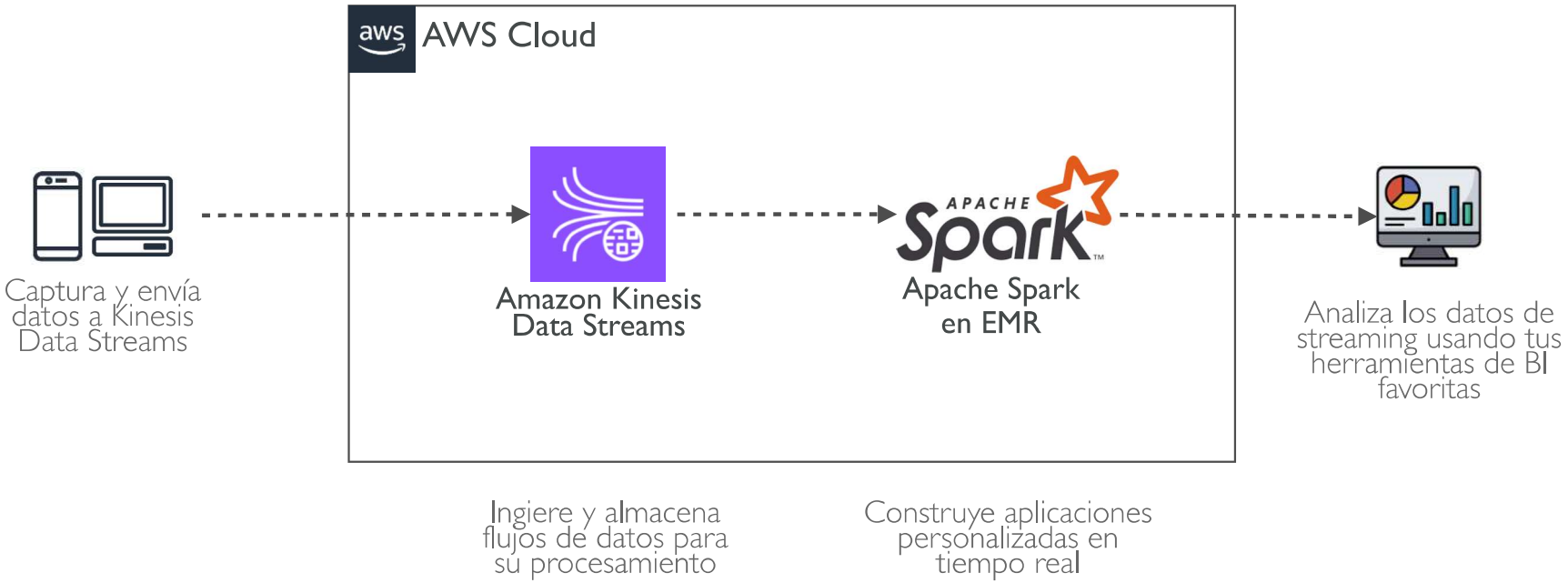


- Proporciona las funcionalidades básicas para la programación en Spark:
  - Gestión de memoria, la recuperación ante fallos, la programación y la monitorización de trabajos, etc

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Apache Spark + Amazon Kinesis



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Apache Spark + Amazon Redshift

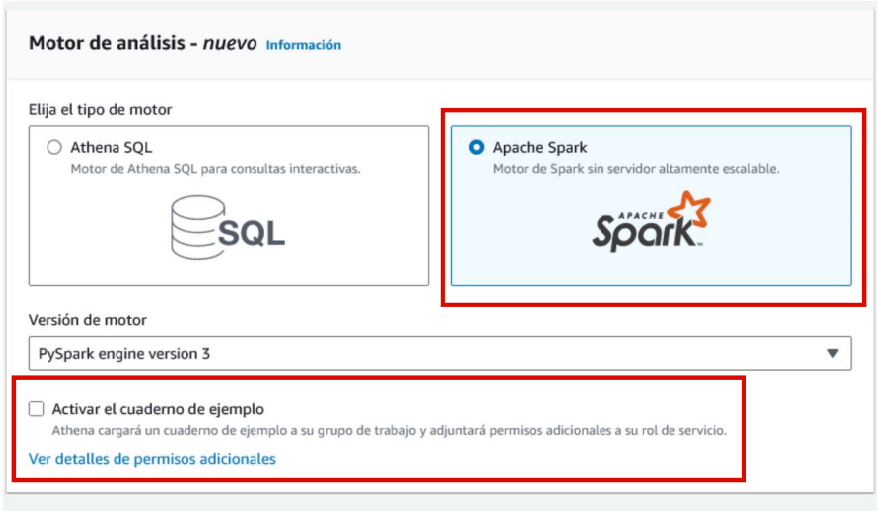


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Apache Spark + Amazon Athena

- Seleccionable como un motor de análisis alternativo (vs. Athena SQL)
- Acceso programático mediante API / CLI
- Puedes ajustar los DPU's para el coordinador y los tamaños de los ejecutores
- Precio basado en el uso de cómputo y DPU por hora



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com



Amazon EMR

www.blockstellart.com

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Visión general de Amazon EMR



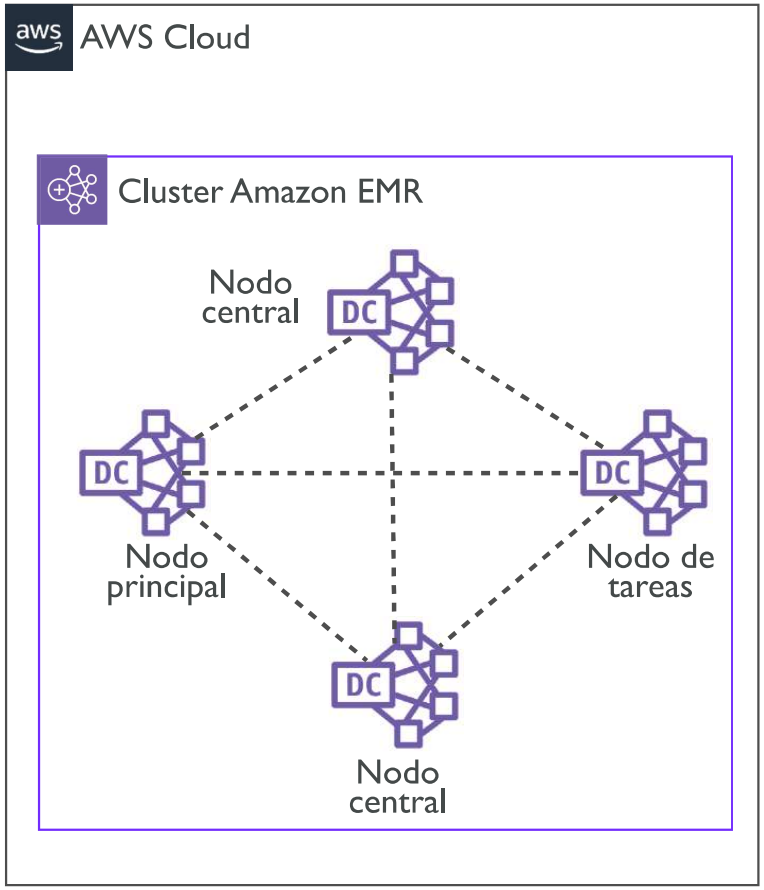
- **EMR = Elastic MapReduce**
- EMR ayuda a crear clusters Hadoop (**Big Data**) para analizar y procesar una gran cantidad de datos
- Los clusters pueden estar formados por **cientos de instancias EC2**
- También es compatible con Apache Spark, HBase, Presto, Flink…
- EMR se encarga de todo el aprovisionamiento y la configuración
- Autoescalado e integrado con instancias Spot
- Casos de uso: procesamiento de datos, Machine Learning, indexación web, big data...

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Clúster EMR

- **Nodo principal**: Gestiona el clúster
  - Monitorea el estado de las tareas y la salud del clúster
  - Instancia EC2 única (puede ser incluso un clúster de un solo nodo)
  - También conocido como "nodo líder"
- **Nodo central**: Aloja los datos en HDFS y ejecuta tareas
  - Puede ser escalado hacia arriba y hacia abajo
- **Nodo de tareas**: Ejecuta tareas, no aloja datos (**Opcional*)
  - No hay riesgo de pérdida de datos al removerlo
  - Buen uso de instancias de spot



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Uso de EMR

- **Clústeres transitorios**: Se crean para realizar una tarea específica y luego se terminan una vez que todos los pasos se completan
  - Carga de datos, procesamiento, almacenamiento - luego se apagan
  - Ahorra dinero



- **Clústeres de larga duración**: Permanecen activos durante un período prolongado y deben ser terminados manualmente
  - Almacén de datos con procesamiento periódico en grandes conjuntos de datos
  - Pueden activar nodos de tareas utilizando instancias de spot para capacidad temporal
  - Pueden usar instancias reservadas en clústeres de larga duración para ahorrar dinero
  - Protección contra terminación activada por defecto, auto-terminación desactivada

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Integración EMR en AWS



Amazon EC2

Proporciona las instancias que componen los nodos del clúster



Amazon VPC

Configura la red virtual donde se lanzan las instancias



Amazon S3

Almacena los datos de entrada y salida



Amazon CloudWatch

Monitorea el rendimiento del clúster y configura alarmas



Amazon CloudTrail

Audita las solicitudes realizadas al servicio



AWS IAM

Configura permisos y seguridad



AWS Data Pipeline

Programa y arranca tus clústeres

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Almacenamiento EMR

- Uno de los aspectos clave de Amazon EMR es cómo maneja el almacenamiento de datos
- Hay varias opciones disponibles para almacenar y gestionar datos en un clúster de EMR:
  - **EBS (Elastic Block Store)**: EBS proporciona almacenamiento en bloque que puede ser utilizado por instancias EC2 dentro de un clúster de EMR
  - **HDFS (Hadoop Distributed File System)**: HDFS es el sistema de archivos distribuido utilizado por Hadoop para almacenar grandes volúmenes de datos en clústeres
  - **EMRFS (EMR File System)**: EMRFS es una implementación de Hadoop Compatible File System (HCFS) que permite a EMR acceder a Amazon S3 como si fuera un sistema de archivos Hadoop
  - **Discos Locales (Local Disks)**: Además de HDFS y S3, EMR puedes utilizar los discos locales de las instancias EC2
  - **Glue Data Catalog**: AWS Glue Data Catalog es un servicio que permite la gestión y catalogación de metadatos para los datos almacenados en S3 y otras fuentes

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Almacenamiento EMR - EBS

- **EBS (Elastic Block Store)**
  - Permite el uso de EMR en tipos solo EBS (M4, C4)
  - Los volúmenes EBS se eliminan cuando se termina el clúster
  - Los volúmenes EBS solo se pueden adjuntar al lanzar un clúster
  - Si se desmonta manualmente un volumen EBS, EMR lo trata como un fallo y lo reemplaza



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Almacenamiento EMR - HDFS

- **HDFS (Hadoop Distributed File System)**
  - Múltiples copias almacenadas en instancias del clúster para redundancia
  - Archivos almacenados como bloques (tamaño por defecto de 128 MB)
  - Los datos en HDFS se pierden cuando el clúster se termina (efímeros)
  - Útil para el almacenamiento intermedio o cargas de trabajo con I/O aleatorio significativo
  - Hadoop intenta procesar los datos donde están almacenados en HDFS

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Almacenamiento EMR - EMRFS

- **EMRFS (EMR File System)**
  - Acceso a S3 como si fuera HDFS
  - Permite almacenamiento persistente después de la terminación del clúster
  - Opcional para la consistencia de S3
    - Usa DynamoDB para rastrear la consistencia
    - Puede ser necesario ajustar la capacidad de lectura/escritura en DynamoDB

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Almacenamiento EMR - Discos locales

- **Discos locales**
  - Además de HDFS y S3, Amazon EMR puede usar los discos locales de las EC2
  - Útil para operaciones de E/S intensivas debido a la baja latencia y alto rendimiento de los discos locales
  - Los datos en los discos locales también se pierden cuando la instancia se termina, similar a HDFS

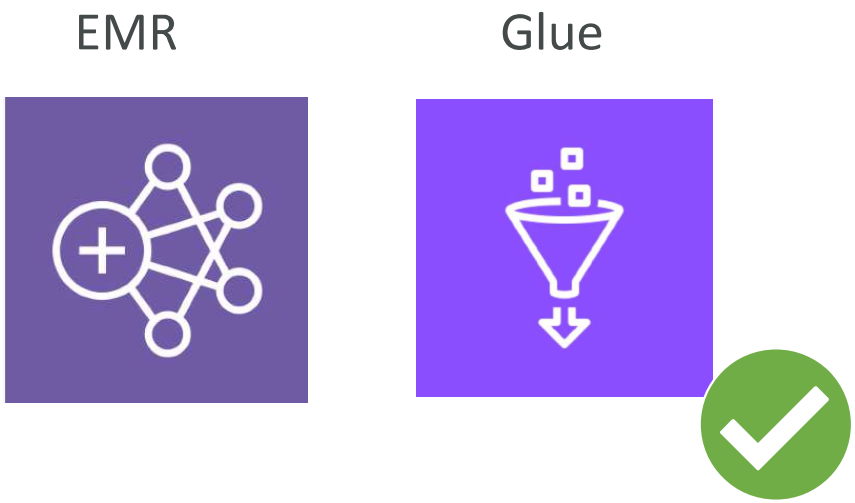


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Almacenamiento EMR - Glue Data Catalog

- **Glue Data Catalog**
  - Amazon EMR puede integrarse con el catálogo de datos de AWS Glue para gestionar metadatos
  - Permite la definición de esquemas y gestión de metadatos para datos almacenados en S3 y otras fuentes
  - Facilita operar con otras herramientas de AWS y aplicaciones de análisis de datos



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Ventajas de EMR



- **Cobros por hora**
  - EMR cobra por hora de uso
  - Además, se aplican cargos de EC2



- **Provisión de nuevos nodos**
  - Provisión de nuevos nodos automáticamente si falla un nodo central



- **Gestión de nodos de tareas**
  - Añadir y eliminar nodos de tareas sobre la marcha
  - Aumenta la capacidad de procesamiento, pero no la capacidad de HDFS



- **Redimensionamiento de nodos centrales**
  - Permite redimensionar los nodos centrales en ejecución
  - Aumenta tanto la capacidad de procesamiento como la de HDFS



- **Adición y eliminación de nodos centrales**
  - Los nodos centrales pueden ser añadidos o eliminados
  - Eliminar nodos centrales conlleva el riesgo de pérdida de datos

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

EMR Serverless



- **Elección de versión y runtime**
  - Elige una versión de EMR y un runtime (Spark, Hive, Presto)



- **¿Por qué es importante?**
  - No necesitas estimar cuántos workers se necesitan para tus cargas de trabajo; se provisionan automáticamente según sea necesario



- **Envío de consultas / scripts**
  - Envía consultas o scripts a través de solicitudes de ejecución de trabajos

- **Gestión de capacidad por EMR**



  - EMR gestiona la capacidad subyacente automáticamente
  - EMR calcula los recursos necesarios para tu trabajo y programa los trabajadores en consecuencia
  - Todo dentro de una sola región (a través de múltiples zonas de disponibilidad)

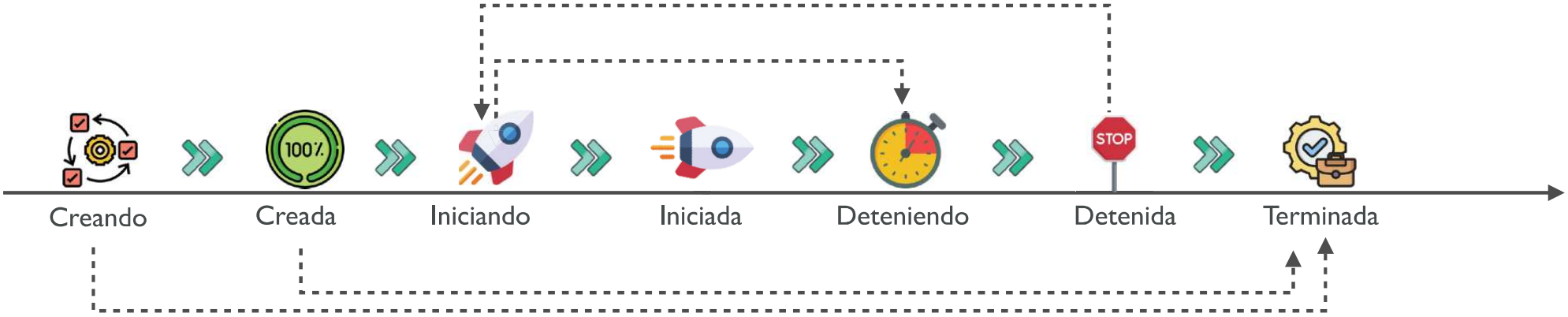


- **¿Realmente es serverless?**
  - Aunque es sin servidor, aún debes considerar los nodos de trabajadores y cómo se configuran

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Ciclo de vida de la app con EMR Serverless



¡Esto no es todo automático!
Debes llamar a las API, tales como:

- CreateApplication (Crear aplicación)
- StartApplication (Iniciar aplicación)
- StopApplication (Detener aplicación)
- Y, lo más importante, DeleteApplication (Eliminar aplicación) para evitar cargos excesivos

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Técnicas de escalado en EMR

Escalado automático en EMR



- Basado en reglas de escalado personalizadas usando métricas de CloudWatch
- Solo soporta grupos de instancias

Escalado administrado en EMR



- Soporta grupos de instancias y flotas EC2
- Escala instancias spot, bajo demanda y en un plan de ahorro dentro del mismo clúster
- Disponible para cargas de trabajo en Spark, Hive y YARN

Estrategia de escalado hacia abajo



- Reducir capacidad:
  - Primero elimina nodos de tareas, luego nodos centrales, sin pasar los límites mínimos establecidos
  - Las instancias spot siempre se eliminan antes que las instancias bajo demanda

Estrategia de escalado hacia arriba



- Añadir capacidad:
  - Primero añade nodos centrales, luego nodos de tareas, hasta alcanzar el límite máximo especificado

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com



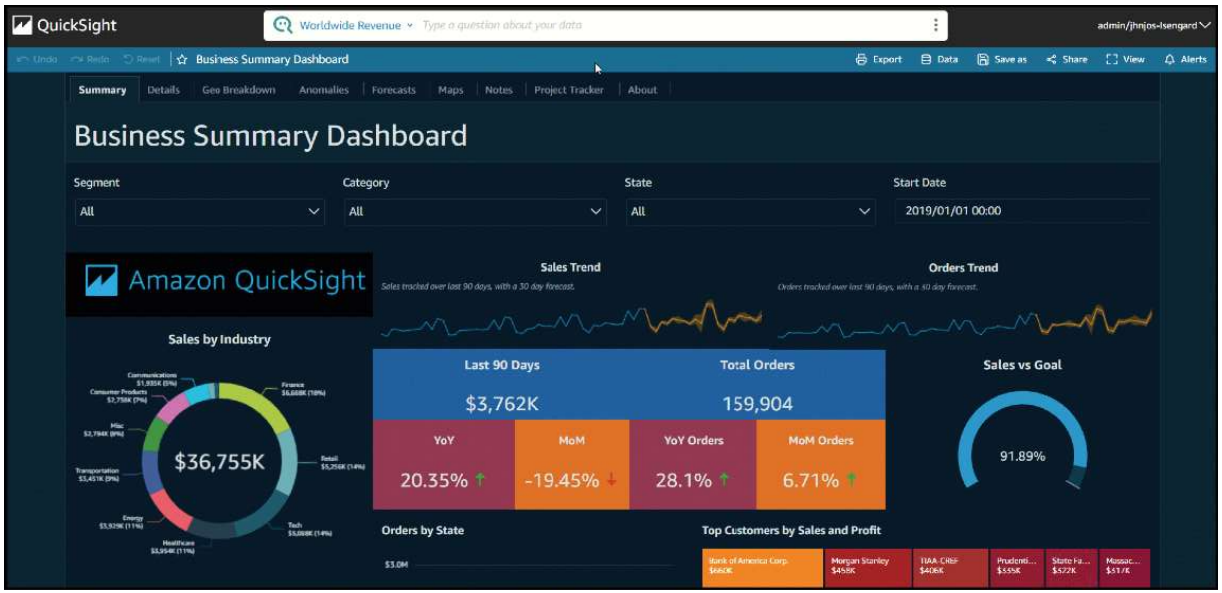
Amazon QuickSight

www.blockstellart.com

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Visión general de Amazon QuickSight

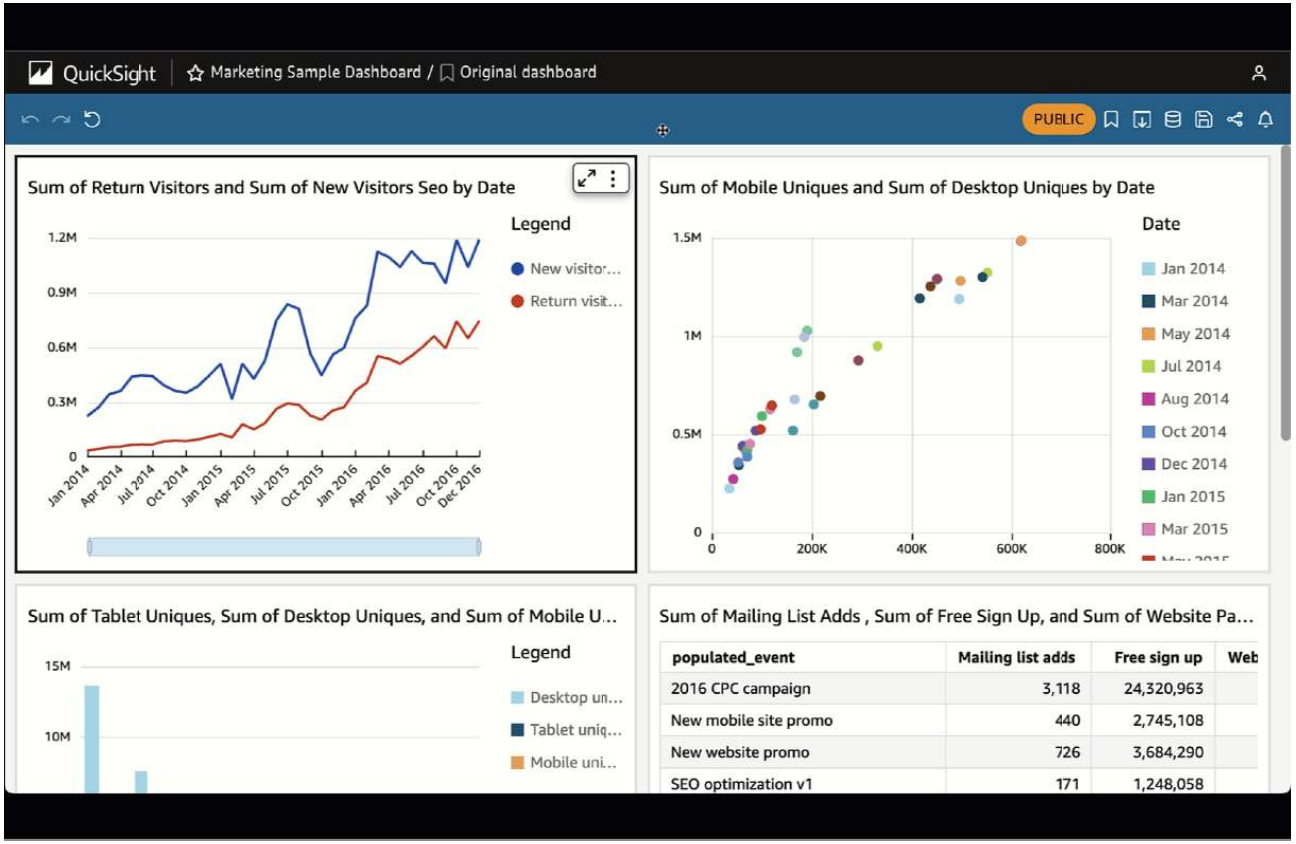
- Servicio de **análisis empresarial** en la nube que permite a los usuarios crear y compartir visualizaciones interactivas y dashboards de datos (sin servidor)
- QuickSight se integra de manera nativa con otras soluciones de AWS (S3, RedShift, RDS, etc.)
- Permite:
  - Crear visualizaciones
  - Realizar análisis ad-hoc
  - Recibir alertas sobre anomalías detectadas
  - Obtener rápidamente conocimientos empresariales a partir de datos
  - En cualquier momento, en cualquier dispositivo (navegadores, móviles, etc.)



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

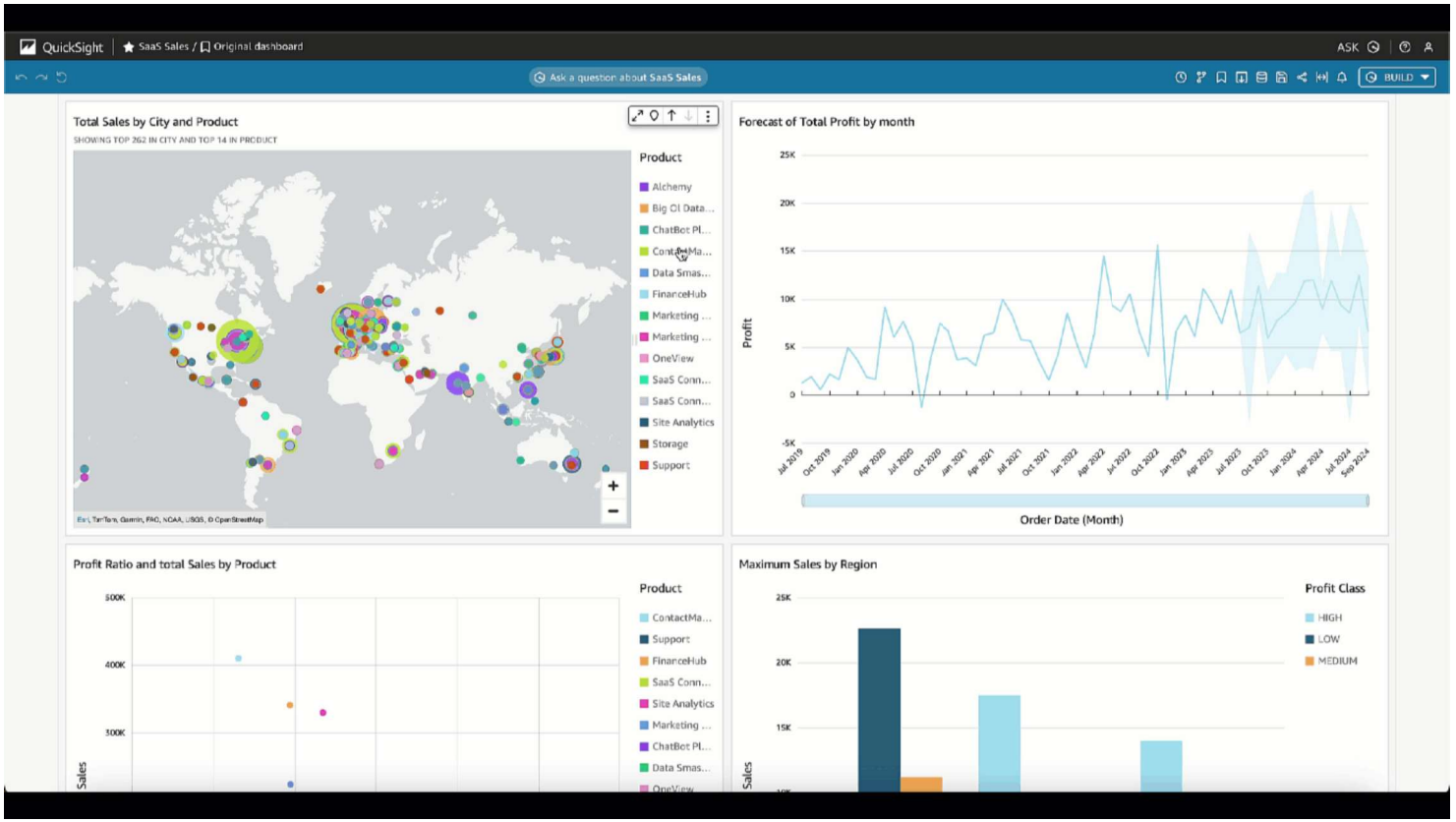
Visión general de Amazon QuickSight



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon QuickSight ❤️ Amazon Q



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Fuentes de datos de QuickSight



Amazon RedShift



Amazon Aurora



Amazon RDS



Amazon Athena



Amazon OpenSearch



AWS IoT Analytics



Bases de datos alojadas en Amazon EC2



S3

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Casos de uso de Amazon QuickSight

Análisis de ventas y marketing



Visualiza métricas clave de rendimiento (KPI) y tendencias de ventas



Analiza campañas de marketing y su impacto



Cualquier fuente de datos JDBC/ODBC



Aplicaciones SaaS, como Salesforce



Bases de datos locales

Análisis financiero



Monitorea ingresos, gastos y rentabilidad



Crea informes financieros detallados

Operaciones y logística



Rastrea el inventario y la cadena de suministro



Optimiza rutas de entrega y tiempos de respuesta

Gestión de recursos humanos



Analiza datos de empleados, como retención y rendimiento



Visualiza estadísticas de capacitación y desarrollo

Monitoreo de TI y Seguridad



Visualiza métricas de rendimiento del sistema



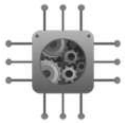
Monitorea actividades de seguridad y cumplimiento

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

SPICE: Motor de cálculo en memoria súper rápido y paralelo

Conjuntos de datos importados a SPICE



Motor de cálculo en memoria, paralelo y súper rápido



Utiliza almacenamiento columnar, en memoria y generación de código máquina



Acelera consultas interactivas en grandes conjuntos de datos

Beneficios de SPICE para los usuarios



Cada usuario obtiene 10GB de SPICE



Altamente disponible y duradero



Escalable a cientos de miles de usuarios

Mejora el rendimiento de consultas



Puede acelerar consultas grandes que se agotarían en modo de consulta directa (accediendo a Athena directamente)



Nota: Si la importación de datos a SPICE toma más de 30 minutos, la operación se agotará

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Seguridad de QuickSight

- Acceso a recursos:
  - Debes **asegurarte de que QuickSight esté autorizado** para usar Athena y buckets de S3
  - Esto se **puede gestionar dentro de la consola de QuickSigh**
- Acceso a datos:
  - Se **pueden crear políticas IAM para restringir qué datos en S3** pueden acceder los usuarios de QuickSight

- ⚠️🚨 **NOTA IMPORTANTE:**
  - Si QuickSight no tiene **permisos para descifrar** los datos almacenados en S3, no podrá acceder a los datos, incluso si puede listar / ver los metadatos del bucket

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Seguridad de QuickSight con Redshift

- **PROBLEMÁTICA:**
  - Por defecto, QuickSight solo puede acceder a datos almacenados **EN LA MISMA REGIÓN** en la que QuickSight está ejecutándose
  - Entonces, si QuickSight se ejecuta en una región y Redshift en otra, eso es un problema.

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Seguridad de QuickSight con Redshift

- **SOLUCIÓN:**
  - Crea un nuevo grupo de seguridad con una regla de entrada que autorice el acceso desde el rango de IP de los servidores de QuickSight en esa región.
  - Esos rangos están documentados en https://docs.aws.amazon.com/quicksight/latest/user/regions.html

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Precios de QuickSight Enterprise Edition

Autores

Los autores de QuickSight pueden conectarse a los datos, crear dashboards e informes y compartir contenido con otros usuarios

Autor

- 24 USD por usuario/mes
- 18 USD por usuario al mes con compromiso anual

Autor Pro

- 50 USD por usuario/mes

Lectores

Solo se les puede conceder acceso como **espectadores** a los dashboards. Pueden ver, exportar e imprimir el dashboard, pero no pueden guardarlo como un análisis. Pueden ver, filtrar y ordenar los datos del panel.

Lector

- 3 USD por usuario/mes

Lector Pro

- 20 USD por usuario/mes

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Precios de QuickSight Enterprise Edition

Precios de capacidad para aplicaciones integradas

Los precios de capacidad permiten a los clientes de QuickSight comprar sesiones de lector o preguntas de Amazon Q en bloque, sin tener que aprovisionar usuarios individuales en QuickSight

Capacidad de lector

- Desde 250 USD para 500 sesiones al mes

Capacidad de preguntas de Amazon Q

- Desde 250 USD para 500 preguntas al mes

https://aws.amazon.com/es/quicksight/pricing/

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Precios de QuickSight Standard Edition

Tipo	Precio	Incluye capacidad SPICE
Plan anual	$ 9 por usuario/mes	10 GB por usuario
Plan mensual	$ 12 por usuario/mes	0,25 por GB de capacidad adicional

https://aws.amazon.com/es/quicksight/pricing/

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com



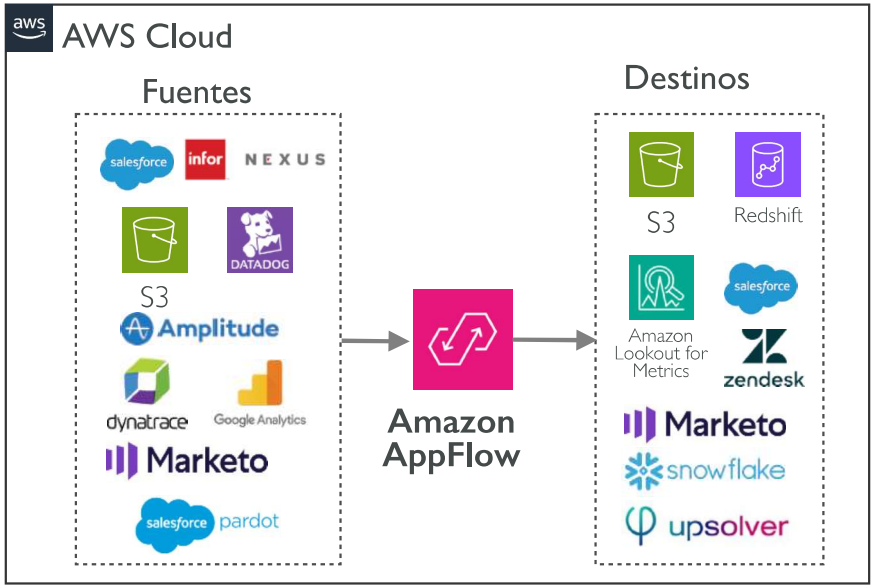
Amazon AppFlow

www.blockstellart.com

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Visión general de Amazon AppFlow

- Facilita la **transferencia** segura y automática de datos **entre servicios AWS y aplicaciones SaaS**
- Elimina la necesidad de desarrollar y mantener integraciones personalizadas a través de una interfaz gráfica sencilla y amigable
- Ofrece capacidades integradas de **transformación de datos** (limpiar, filtrar y enriquecer datos durante su transferencia)
- Garantiza la seguridad de los datos en tránsito mediante cifrado y cumple con diversos estándares de cumplimiento (HIPAA y GDPR)
- Frecuencia de transferencia (programada, eventos, bajo demanda)



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Beneficios de Amazon AppFlow



Seguridad y confiabilidad: Garantía de que los datos se transfieren de manera segura y conforme a los estándares de AWS



Facilidad de uso: Elimina la necesidad de escribir y mantener integraciones personalizadas



Flexibilidad y escalabilidad: Permite ajustar y escalar las transferencias de datos según las necesidades del negocio



Automatización: Automatización de procesos entre aplicaciones con flujos de trabajo basados en eventos

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com



Amazon MWAA

www.blockstellart.com

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Visión general de Amazon MWAA

- Amazon MWAA = Amazon Managed Workflows for Apache Airflow

- Creación, programación y monitoreo de **flujos de trabajo** complejos
- Interfaz fácil de usar para gestionar flujos de trabajo como código
- Los flujos de trabajo se definen como código Python que crea un **DAG** (Grafo Acíclico Dirigido)
- Gestiona la infraestructura necesaria, eliminando la necesidad de instalación y mantenimiento

Apache Airflow

Casos de uso:



Flujos de trabajo complejos



Coordina trabajos de ETL

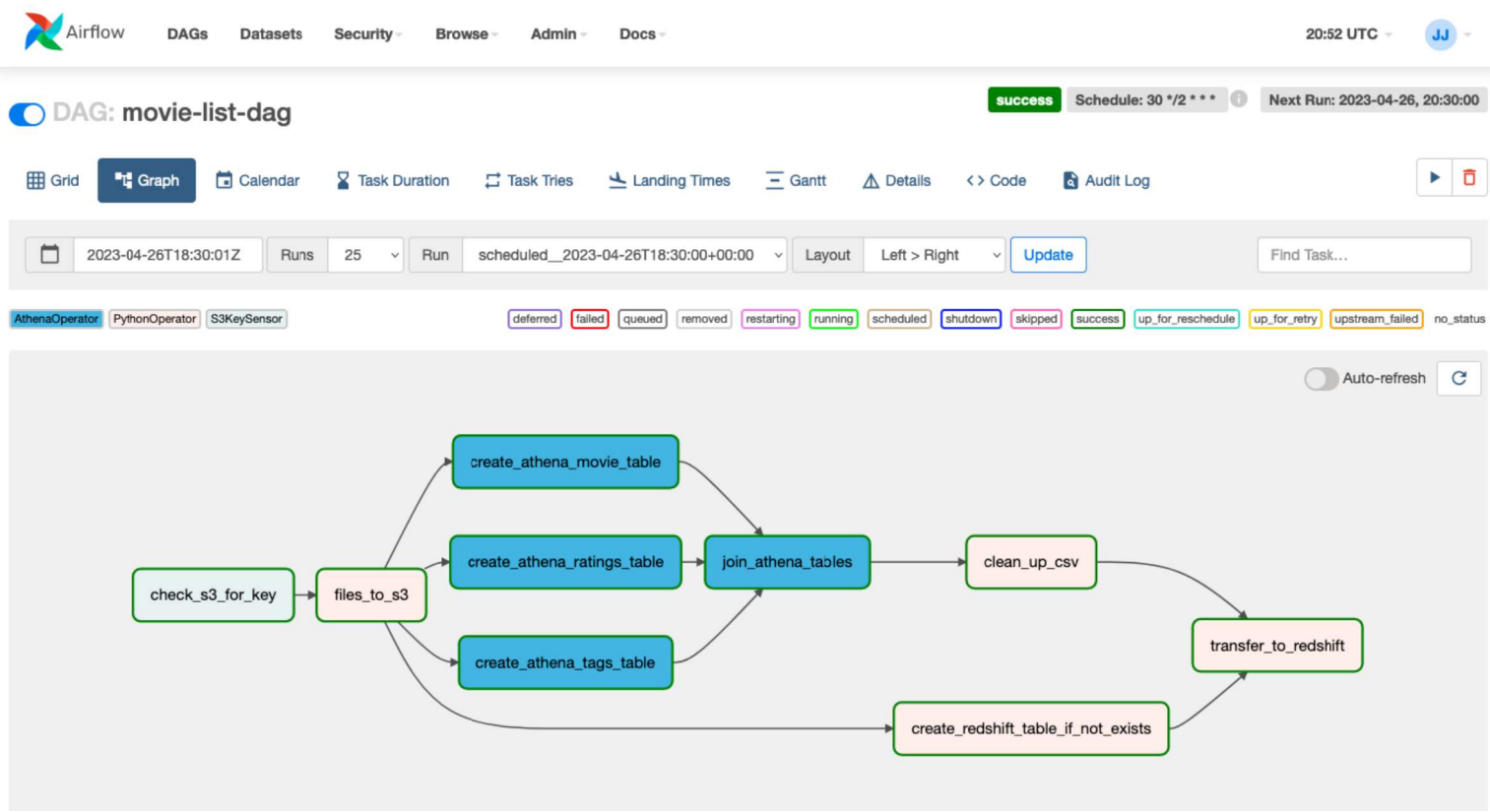


Prepara datos de ML

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Flujos de trabajo de Amazon MWAA



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com



Amazon MSK

www.blockstellart.com

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Visión general de Amazon MSK

- Amazon MSK = Amazon Managed Streaming for Apache Kafka
- **Se encarga del aprovisionamiento, configuración y mantenimiento de clústeres de Apache Kafka**
- Puedes usar aplicaciones y herramientas existentes de Kafka sin cambios en el código
- Asegura una alta disponibilidad al reemplazar automáticamente los nodos no saludables y replicar datos a través de múltiples AZs
- Integración con Lambda, Amazon CloudWatch y Glue, facilitando la construcción y monitoreo de apps de streaming

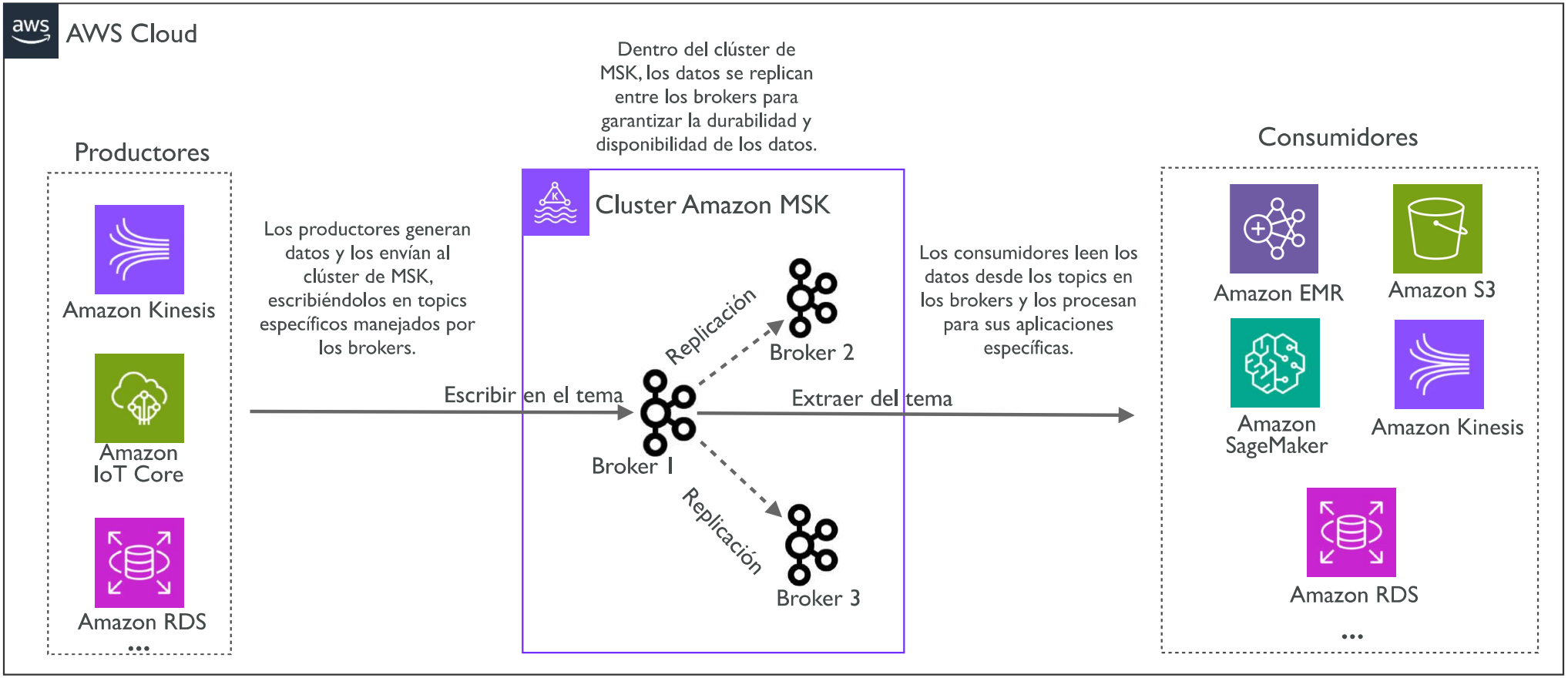


- Precios:
  - Pago por uso, con cargos basados en el tiempo que las instancias están funcionando
  - Almacenamiento utilizado
  - Tarifas estándar de transferencia de datos

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Visión general de Apache Kafka

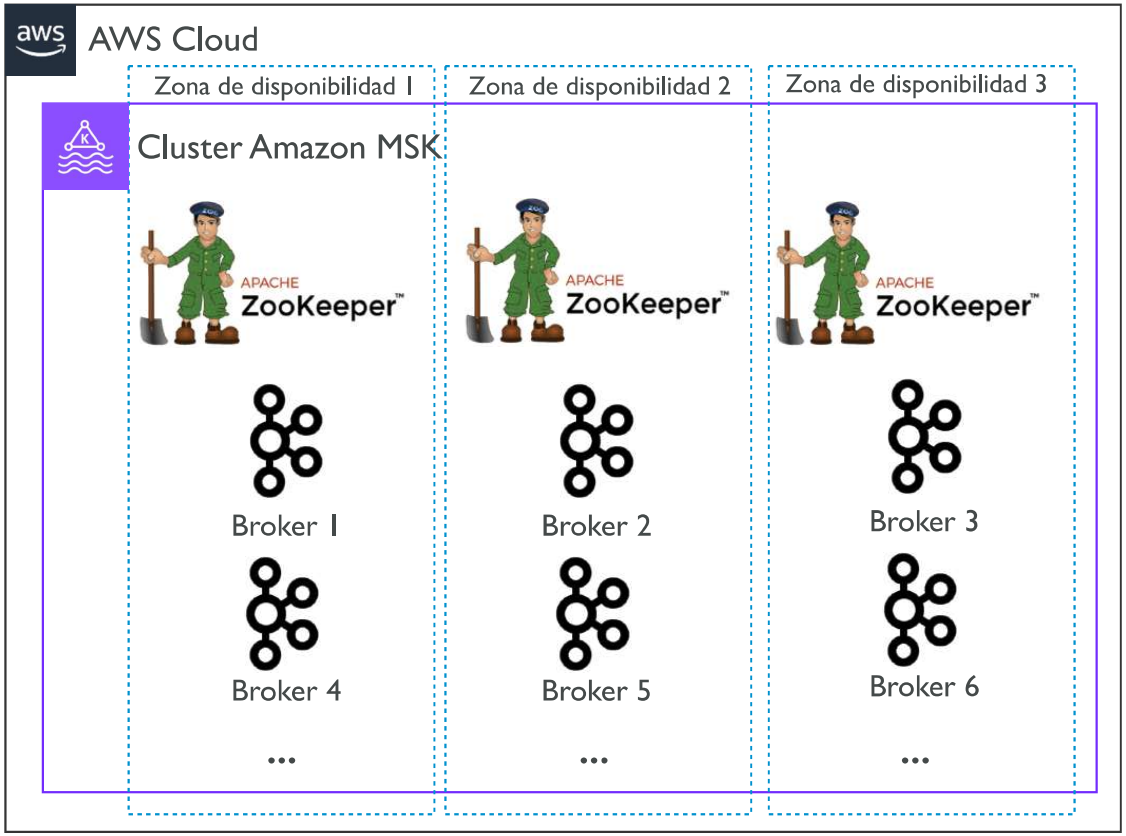


© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Configuraciones del Clúster MSK

- Elige el número de Zonas de Disponibilidad (AZ)
  - Se recomienda 3
- Selecciona la VPC y subredes
- Tipo de instancia del broker
  - Ej: kafka.m5.large
- Número de brokers por AZ
  - Puedes añadir más brokers más tarde
- Tamaño de los volúmenes EBS
  - 1GB - 16TB



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon MSK – Seguridad

- **Cifrado**:
  - Opcional en tránsito utilizando TLS entre los brokers
  - Opcional en tránsito utilizando TLS entre los clientes y brokers
  - En reposo para tus volúmenes EBS utilizando KMS
- **Seguridad de red**:
  - Autoriza grupos de seguridad específicos para tus clientes de Apache Kafka
- **Autenticación y autorización (importante):**
  - Define quién puede leer/escribir en qué topics
  - Mutual TLS (AuthN) + Kafka ACLs (AuthZ)
  - SASL/SCRAM (AuthN) + Kafka ACLs (AuthZ)
  - Control de acceso IAM (AuthN + AuthZ)



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon MSK – Monitoreo

- **CloudWatch Metrics:** Utiliza Amazon CloudWatch para monitorear y recolectar métricas de rendimiento



Monitoreo básico: Proporciona una visión general del rendimiento del clúster y los brokers



Monitoreo mejorado: Ofrece métricas detalladas de los brokers, permitiendo un análisis más profundo



Monitoreo a nivel de tema: Facilita el seguimiento de métricas específicas de los temas, lo que ayuda a identificar problemas a nivel granular

- **Prometheus:** Herramienta de monitoreo de código abierto que se integra con Apache Kafka para exportar métricas



JMX Exporter: Captura métricas de Java Management Extensions (JMX) para monitorear la JVM



Node Exporter: Recolecta métricas de hardware y sistema operativo, como uso de CPU y disco

- **Entrega de Logs del Broker:** Permite la entrega de logs de los brokers a varios destinos para análisis y almacenamiento.







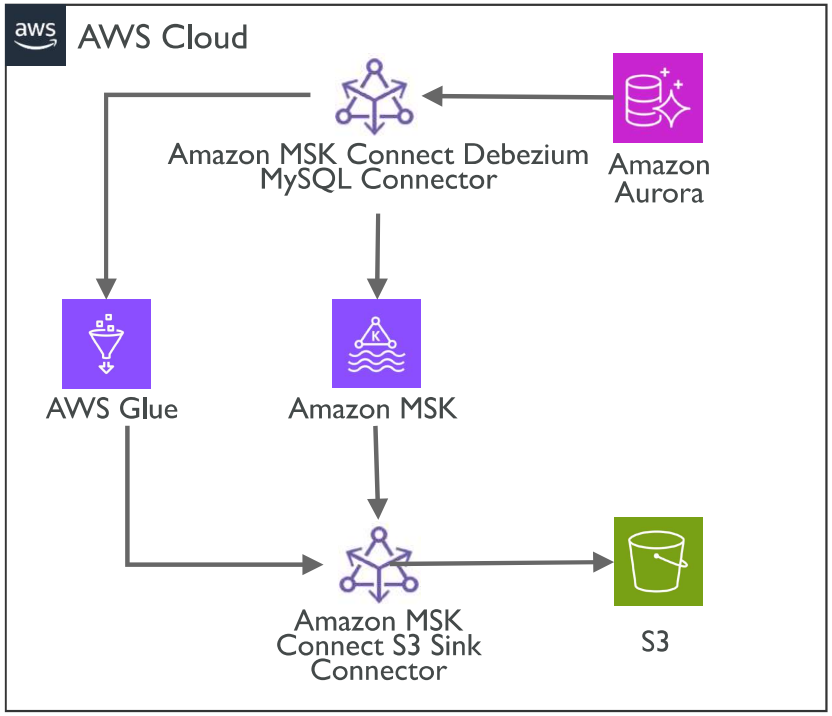
© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon MSK Connect

- Funcionalidad de Amazon MSK que **permite integrar** clústeres de Apache Kafka con múltiples sistemas de destino y origen de datos de manera eficiente

- Los trabajadores de Kafka Connect se gestionan completamente en AWS, simplificando la administración
- Escalado dinámico según la carga de trabajo para asegurar un rendimiento óptimo
- Puedes desplegar cualquier conector de Kafka Connect en MSK Connect como un plugin
  - Conectores disponibles: Amazon S3, Amazon Redshift, Amazon OpenSearch, Debezium, entre otros



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Amazon MSK Serverless

- MSK Serverless **elimina la necesidad de administrar** manualmente los recursos del clúster de Kafka
- Ajusta automáticamente los recursos según la demanda, asegurando un rendimiento óptimo
- Configuración sencilla, simplemente define los temas y particiones necesarios
- Utiliza AWS IAM para gestionar permisos y accesos de manera segura



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Kinesis Data Streams vs Amazon MSK

Amazon Kinesis Data Streams

- 1 MB límite de tamaño de mensaje
- Flujos de datos con fragmentos (Shards)
- División y fusión de fragmentos
- Cifrado en tránsito con TLS
- Seguridad:
  - Gestión de accesos y permisos mediante AWS IAM



Amazon MSK

- 1 MB por defecto, configurable (ej: 10 MB)
- Los datos se organizan en temas y particiones para facilitar el manejo y procesamiento
- Las particiones se añaden a temas ya existentes
- Cifrado en tránsito
- Seguridad:
  - Autenticación mutua y listas de control de acceso de Kafka

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com



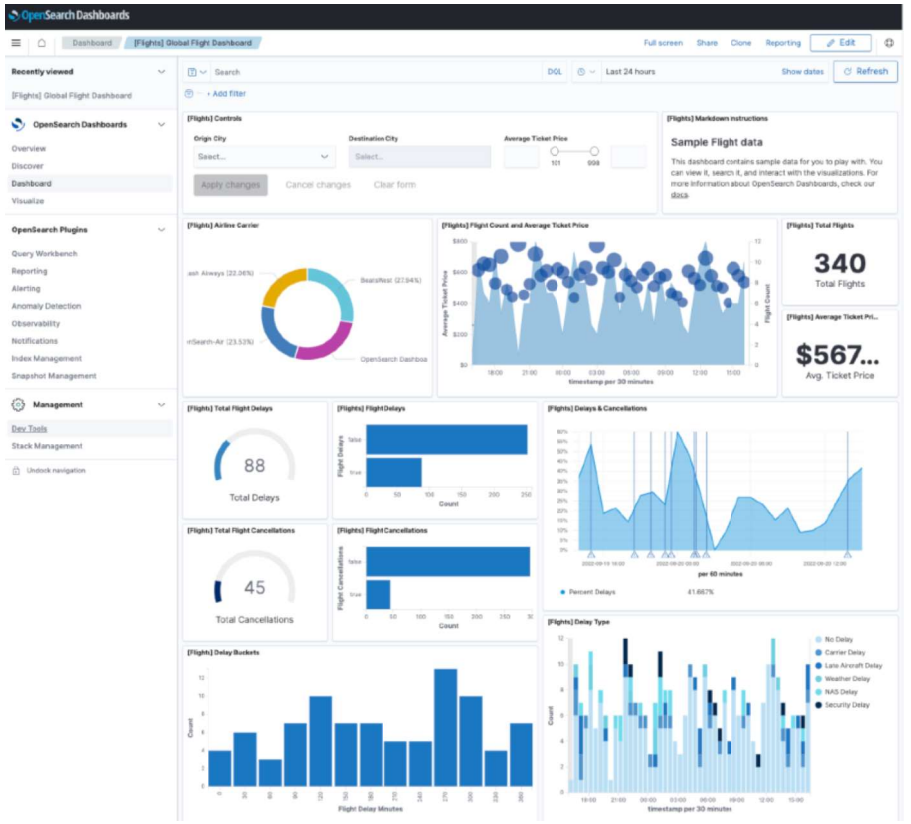
Amazon OpenSearch Service

www.blockstellart.com

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Visión general de Amazon OpenSearch

- OpenSearch es un fork de Elasticsearch (iniciado por AWS)
- Servicio gestionado que facilita la **búsqueda, visualización y análisis** de grandes volúmenes de datos en tiempo real
- Incluye **OpenSearch Dashboards**, una herramienta de visualización integrada para analizar y monitorear datos
- Casos de uso:
  - Búsqueda de texto de forma rápida
  - Monitoreo de aplicaciones
  - Gestión de seguridad e información de eventos



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Características principales de Amazon OpenSearch



Búsqueda y análisis

Proporciona un motor de búsqueda potente y herramientas de análisis para diversas aplicaciones



Escalabilidad

Permite la escalabilidad horizontal, facilitando la adición de nodos adicionales



Seguridad

Características avanzadas de seguridad como encriptación, autenticación, autorización y auditoría



Visualización

OpenSearch Dashboards ofrece una interfaz intuitiva para la visualización de datos

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

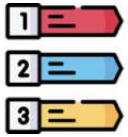
Conceptos de OpenSearch [1/2]

Documentos



Son las unidades básicas de información en OpenSearch. Se almacenan en formato JSON y contienen datos estructurados que pueden ser buscados y analizados

Índices



Son colecciones de documentos. Los índices permiten organizar, buscar y analizar los documentos almacenados

Shards (Fragmentos)



Los índices en OpenSearch se dividen en fragmentos (shards) para distribuir la carga de trabajo y mejorar el rendimiento

Clústeres



Un conjunto de uno o más nodos (servidores) que trabajan juntos para almacenar datos y proporcionar capacidades de búsqueda y análisis.

Nodos



Es una instancia de OpenSearch que forma parte de un cluster. Los nodos pueden tener diferentes roles, como nodo principal, nodo de datos o nodo de coordinación, dependiendo de su configuración y propósito

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Conceptos de OpenSearch [2/2]

Pipeline de datos



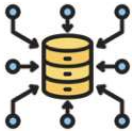
Se refiere a la secuencia de pasos y procesos que transforman y transportan datos desde su origen hasta su destino en OpenSearch

OpenSearch Dashboards



Es una herramienta de visualización de datos que permite crear y compartir dashboards interactivos

Agregaciones



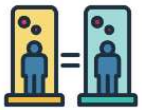
Son operaciones de análisis de datos que permiten calcular estadísticas y métricas, agrupar datos, y realizar operaciones avanzadas como análisis de series temporales

Consulta (Query)



Son solicitudes de búsqueda realizadas al cluster de OpenSearch para recuperar y analizar datos

Réplicas



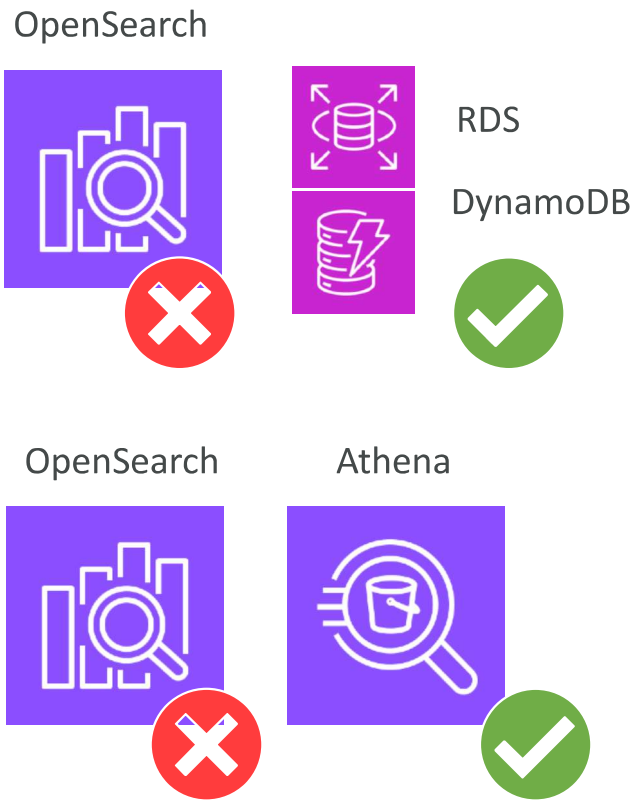
Son copias adicionales de los fragmentos de un índice. Se utilizan para asegurar la disponibilidad de los datos y mejorar la tolerancia a fallos

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Anti-patrones de Amazon OpenSearch

- **OLTP (Procesamiento de Transacciones en Línea)**
  - Sin transacciones: OpenSearch no está diseñado para manejar transacciones ACID completas
  - RDS o DynamoDB es mejor: Para aplicaciones que requieren transacciones completas
- **Consultas ad-hoc de Datos**
  - Athena es mejor: OpenSearch no está optimizado para consultas ad-hoc de datos
- **Recordar que OpenSearch es principalmente para búsqueda y análisis**
  - OpenSearch está diseñado específicamente para proporcionar capacidades de búsqueda rápida y análisis de datos en tiempo real



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

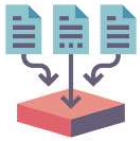
Amazon OpenSearch Serverless

- **Escalado automático bajo demanda**



  - Permite que el servicio escale automáticamente según las necesidades de carga de trabajo, sin intervención manual

- **Funciona con "colecciones" en lugar de dominios aprovisionados**



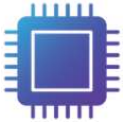
  - Las colecciones pueden ser de tipo "búsqueda" o "series temporales"
  - Facilita la organización y el acceso a los datos de manera más eficiente

- **Siempre encriptado con tu clave KMS**



  - Define quién puede acceder a los datos
  - Todos los datos almacenados están encriptados
  - Permite establecer políticas de seguridad a través de múltiples colecciones

- **Capacidad medida en Unidades de Cómputo de OpenSearch (OCUs)**



  - Puedes establecer un límite superior para controlar costos
  - El límite inferior es siempre 2 OCUs para indexación y 2 OCUs para búsqueda, asegurando un mínimo de capacidad

© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

BLOCKSTELLART

Preparación del examen de certificación

www.blockstellart.com

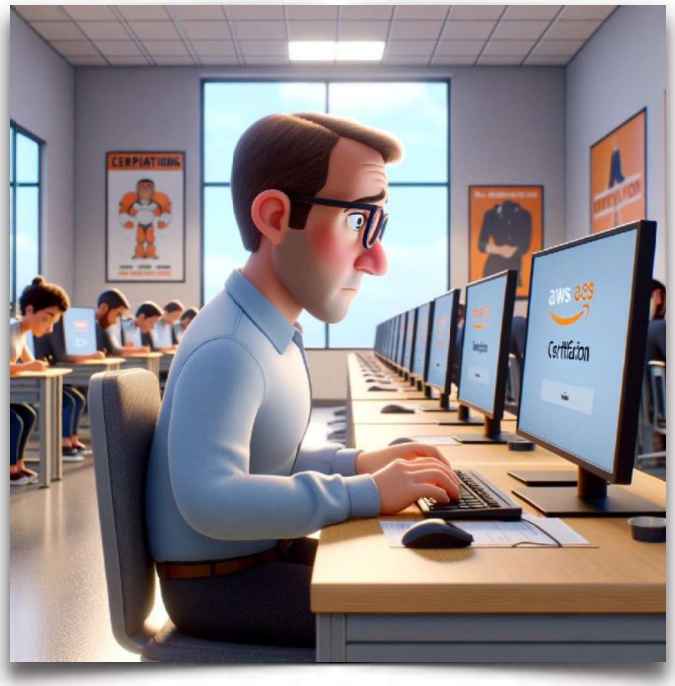
Todos los derechos reservados © Joan Amengual www.blockstellart.com

Tómate tu tiempo leyendo las preguntas

Busca palabras clave sobre los requisitos

Una empresa de comercio electrónico desea **optimizar sus recomendaciones de productos** *para los usuarios, utilizando un* **modelo de aprendizaje automático** *que se ajusta continuamente con datos de interacción. Aunque el volumen de datos es generalmente estable, se observan* **picos significativos durante eventos de ventas** *masivas como el Black Friday.*

¿Qué arquitectura de sistema garantizaría una **adaptación eficiente** *y* **confiable** *a estos cambios dinámicos en la carga de datos?*



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Mantén el ritmo

- Tienes **130 minutos** y **65 preguntas**, eso es solo 2 minutos por pregunta!
- Recomendaciones:
  - 🧘Trata de **no estresarte**... eso es suficiente tiempo para leer y entender cada pregunta
  - 🤔 Asegúrate de entender bien cada alternativa antes de tomar una decisión
  - 🗑️ Reduce tus opciones **descartando las respuestas claramente erróneas**
  - 🚩 Si no estás seguro, **marca la pregunta para revisarla más tarde** y avanza
  - ⏳Si terminas antes de tiempo, utiliza los minutos restantes para revisar las preguntas marcadas y asegurarte de que no dejaste ninguna respuesta en blanco



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com



¡Enhorabuena!

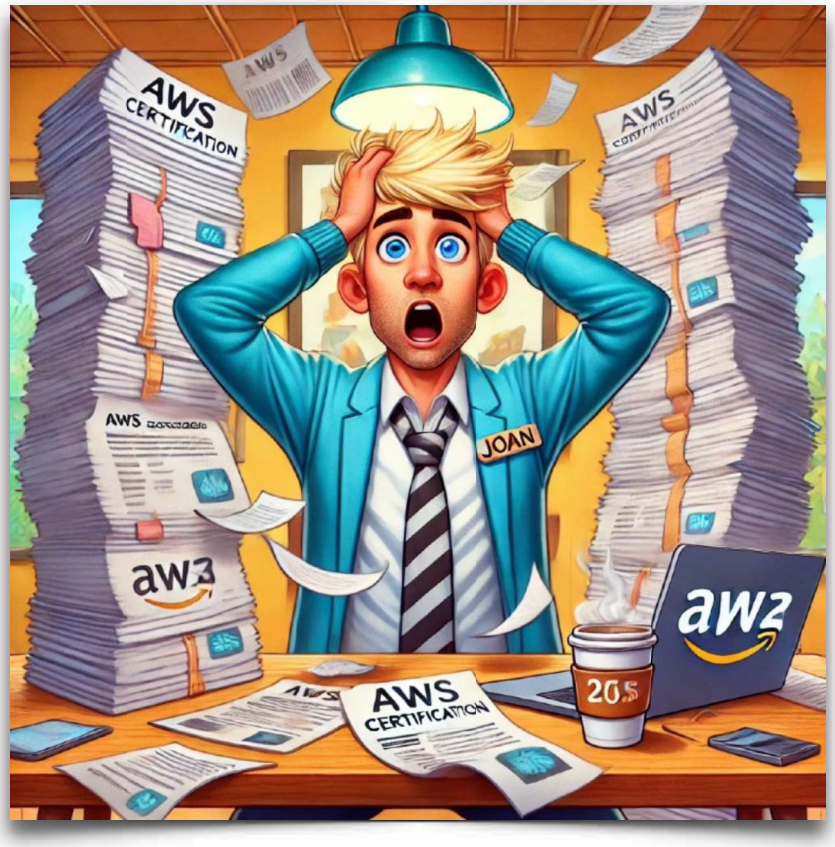
www.blockstellart.com

Todos los derechos reservados © Joan Amengual www.blockstellart.com

Roadmap de certificaciones de AWS

- Las dudas de siempre…
  - ¿Qué certificación hacer primero?
  - ¿Cuál es la certificación más adecuada para mi rol actual?
  - Y muchas más…

- No existe un único camino de certificación, mi recomendación es:
  - Escoge tu camino de certificación según un rol (Recurso)



© Joan Amengual

Todos los derechos reservados © Joan Amengual www.blockstellart.com