

AWS
re:Invent

The world is **asynchronous**





LATAM EDA ROADSHOW - COLOMBIA

Construyendo Arquitecturas orientadas a eventos en AWS

Ana Barragán (she/her)

Solutions Architect
AWS

Agenda

Resumen sobre Serverless

Retos con los Sistemas Distribuidos

El camino hacia Arquitecturas Orientadas a Eventos

Consideraciones de diseño

Próximos pasos



Resumen sobre Serverless



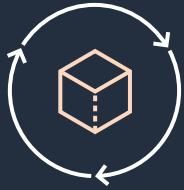
Hoy, el 80% del tiempo de los desarrolladores se usa en operaciones y mantenimiento.*

En el futuro, el **único código que escribirá será lógica de negocio.**

* Fuente: Deloitte 2019



Serverless le permite enfocarse en agregar valor al negocio



Agilidad

*Lanzar al mercado
más rápido,
Crear nuevas
funcionalidades*



Rendimiento

*Alto
rendimiento y
escalabilidad*



Costo

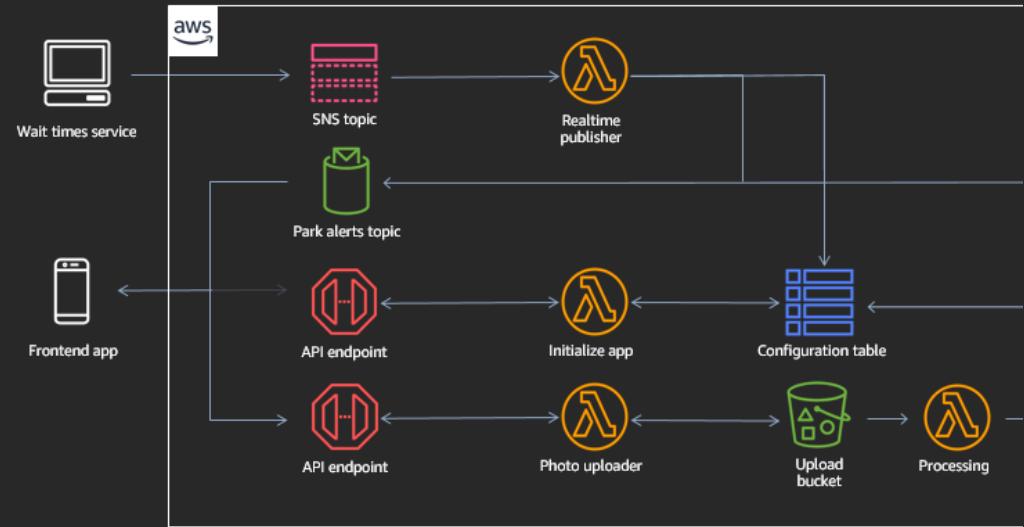
*Pagar por
valor, menor
TCO*



Seguridad

*Seguridad y
aislamiento por
diseño*

Pequeñas piezas, bajo acoplamiento



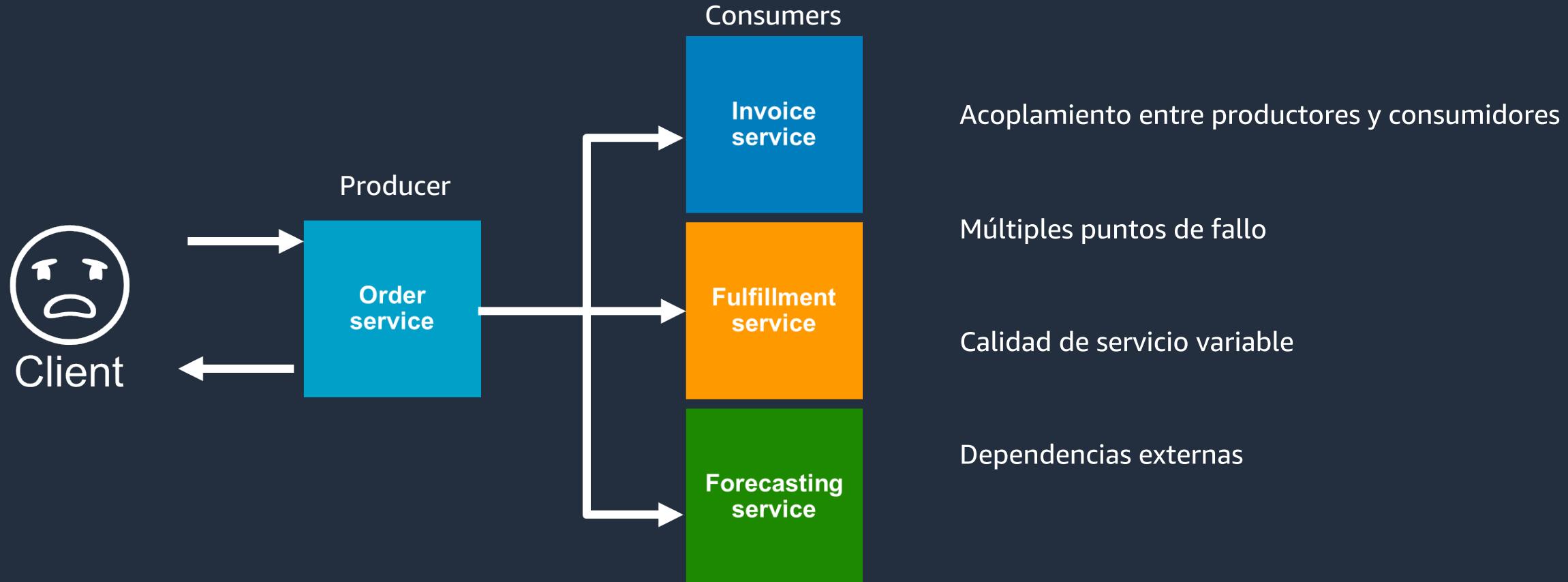
Retos con los sistemas distribuidos



Los microservicios al inicio son sencillos



Retos de las APIs síncronas a lo largo del tiempo



Acoplamiento – La palabra mágica en las integraciones



El acoplamiento es una medida de la independencia de la variabilidad entre sistemas conectados

El desacoplamiento tiene un costo, tanto en el diseño como en el tiempo de ejecución

El acoplamiento no es algo binario

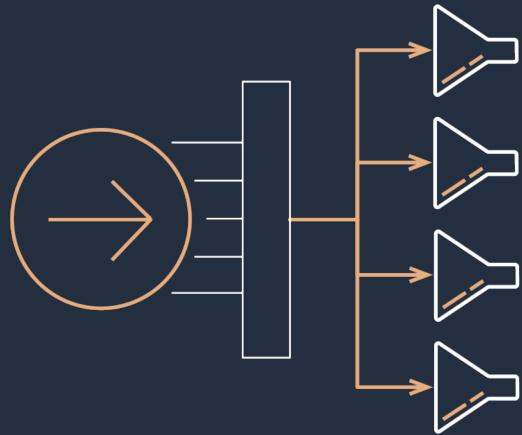
El acoplamiento no es unidimensional

Las diferentes caras del acoplamiento

Dependencia temporal:	Síncrono, asíncrono
Dependencia de ubicación:	Dirección IP, DNS
Dependencia de formato de datos:	Binario, XML, JSON, ProtoBuf, Avro
Dependencia semántica:	Nombre, Apellido, Código Postal
Dependencia de tipo de datos:	int16, int32, string, UTF-8, null, empty
Dependencia de estilo de interacción:	Mensajes, RPC, query-style (GraphQL)
Dependencia de conversación:	Paginación, caché, reintentos
Dependencia tecnológica:	Java vs. C++

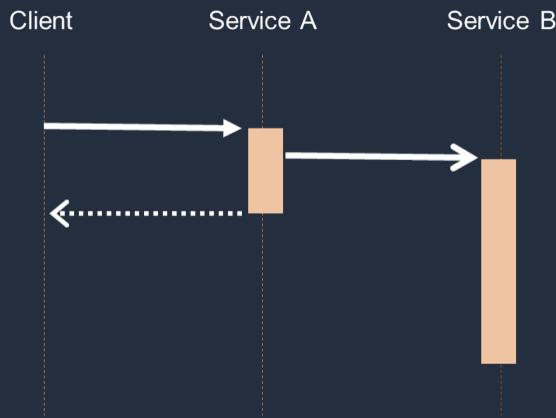


Arquitecturas Orientadas a Eventos impulsan la escalabilidad y confiabilidad



Enrutadores de Eventos

Abstraen la interacción entre productores y consumidores



Eventos Asíncronos

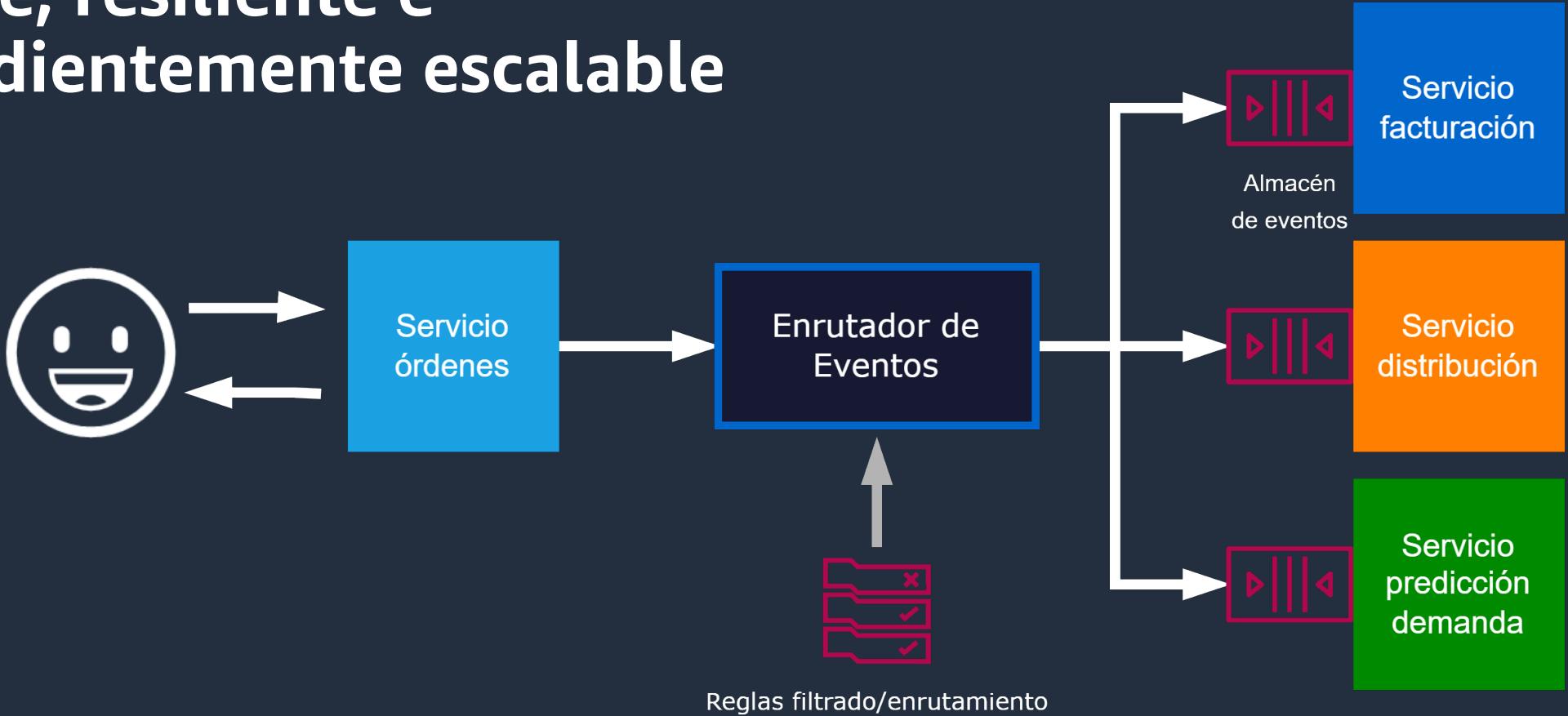
Mejora la respuesta y reduce las dependencias



Almacenes de Eventos

Acumula los mensajes hasta que los servicios son capaces de procesarlos

Confiable, resiliente e independientemente escalable



"Si su aplicación es nativa en nube, de gran escala, o distribuida, y no incluye un componente de mensajería o de eventos, entonces probablemente sea un bug".

Tim Bray, general-purpose internet-software geek

El camino hacia Arquitecturas Orientadas a Eventos

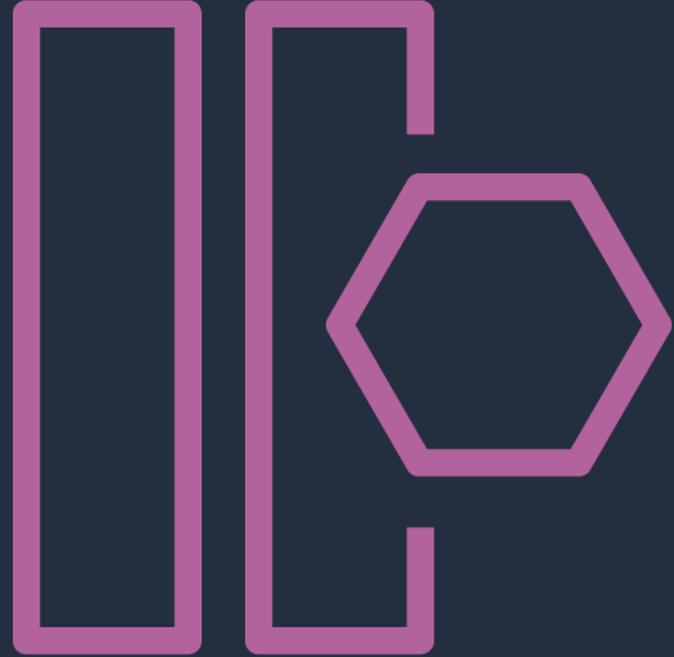


El camino hacia Arquitecturas Orientadas a Eventos

PASO 1

Comience con el dominio





evento

[e-ven-to] sustantivo

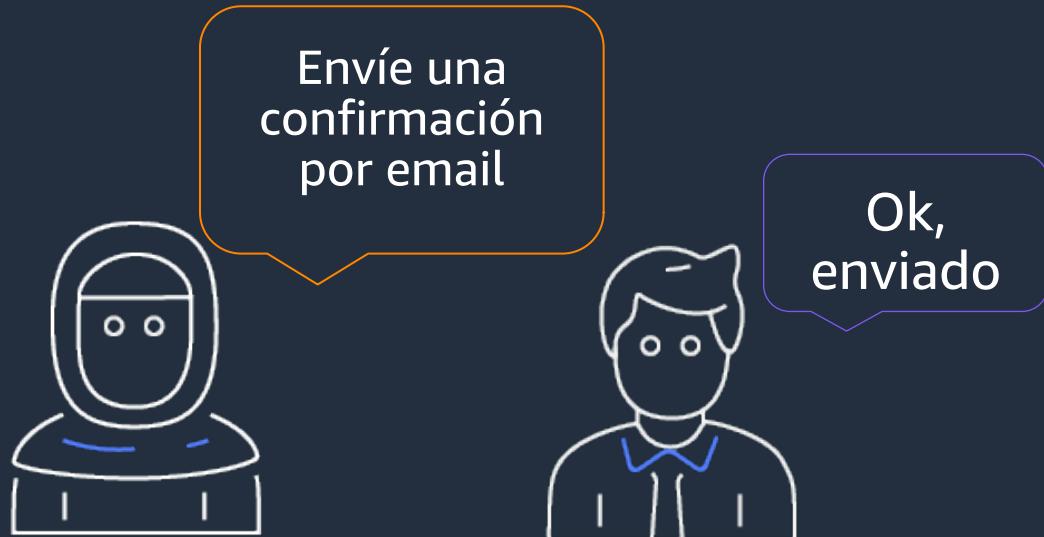
**Una señal de que el
estado del sistema
ha cambiado.**

Propiedades de los eventos

```
{  
  "source": "com.orders",  
  "detail-type": "OrderCreated",  
  "detail": {  
    "metadata": {  
      "idempotency-key": "c1b95b88"  
    },  
    "data": {  
      "order-id": "1073459984"  
    }  
  }  
}
```

- Los eventos son señales de cambios en el estado del sistema
- Los eventos ocurren en el pasado (e.g. OrdenCreada)
- Los eventos no se pueden cambiar (inmutables)
- Disminuya el acoplamiento semántico restringiendo la información a datos importantes

Los Eventos son observables, no dirigidos



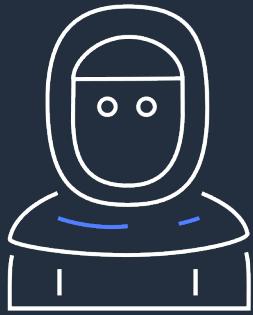
Comandos dirigidos



Eventos observables

EDA comienzan con **eventos de negocio**

OrdenCreada



OrdenRecogida



OrdenEnviada



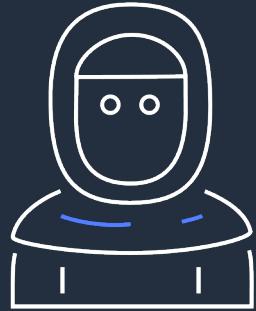
DevolucionSolicitada



DevolucionRecibida

Mapee los eventos a los **dominios de negocio** usando **atributos, no entidades**

OrdenCreada



Tienda

OrdenRecogida

OrdenEntregad
a



Distribución

DevolucionSolicitada



DevolucionRecibida



Soporte

Modelamiento orientado a eventos

- 
- A photograph of a man with dark hair and a beard, wearing glasses and a blue button-down shirt. He is looking towards the right side of the frame. Behind him is a whiteboard or wall covered with numerous colorful sticky notes in various colors like red, yellow, green, and blue, suggesting a collaborative workspace or planning session.
1. Identificar eventos de negocio, procesos, actores, etc.
 2. Agrupe los conceptos relacionados
 3. Defina los bounded contexts y sub-dominios

La documentación es importante

<https://www.eventcatalog.dev/>

The screenshot shows the EventCatalog documentation tool. At the top, there's a navigation bar with links for EventCatalog, Docs, API, Blog, Community, Discord, Demo, and GitHub. The main content area features a large heading: "Documentation tool for Event-Driven Architectures". Below it, a sub-headline states: "EventCatalog is an Open Source project that helps you document your events, services and domains." A note below says "Powered by markdown, components and plugins." Two buttons at the bottom are "Start Documenting →" and "View Demo". To the right, a large window displays the schema for an event named "AddedItemToCart". The schema includes fields for name, version, summary, producers (Basket), consumers (Data Layer), owners (dboyne), and a Mermaid diagram showing the flow from Shopping API to AddedItemToCart to Customer Portal. The Mermaid code is:

```
graph LR; Shopping_API --> AddedItemToCart; AddedItemToCart --> Customer_Portal
```

Below the schema, there's a "Details" section with a note about correlation IDs and a "Event Schema" section with JSON code:

```
1 {  
2   "$id": "https://example.com/cart-schema.json",  
3   "$schema": "https://json-schema.org/draft/2020-12/schema",  
4   "title": "AddedItemToCart",  
5   "type": "object",  
6   "properties": {
```



El camino hacia Arquitecturas Orientadas a Eventos

PASO 1

Comience con el dominio



PASO 2

Coordine eventos

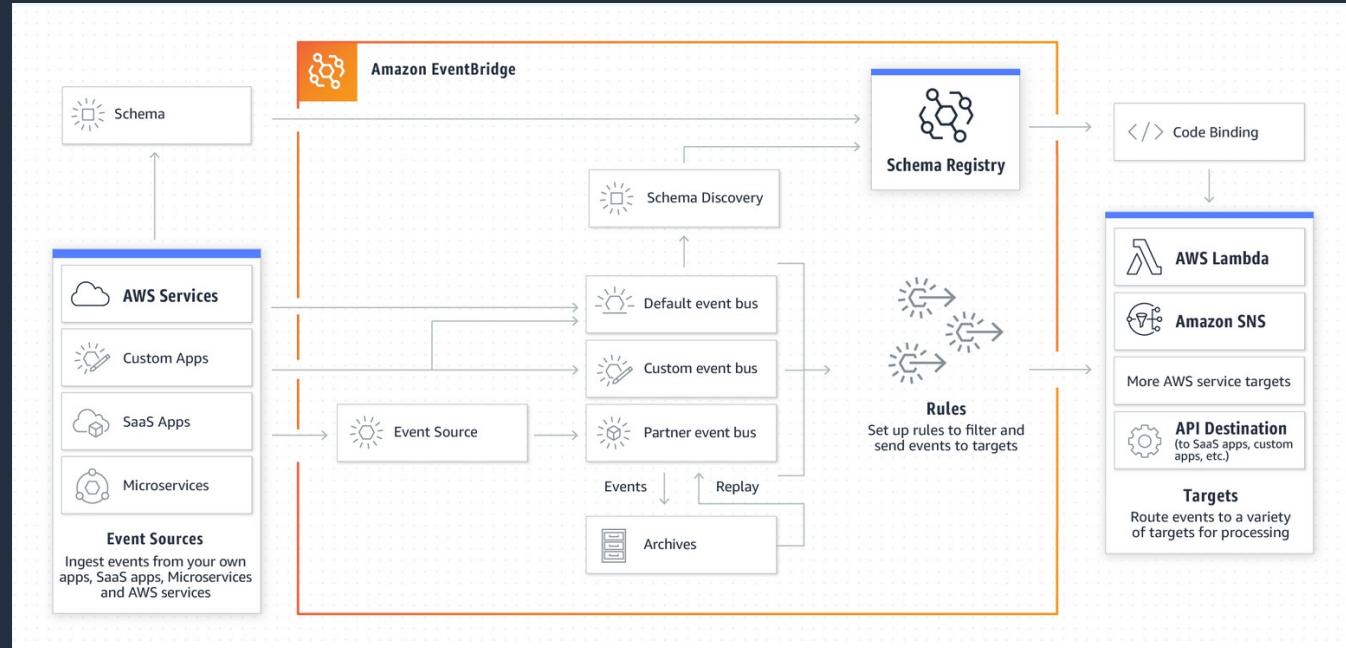
Establezca una **Coreografía** de eventos *entre dominios* usando suscripciones





Amazon EventBridge

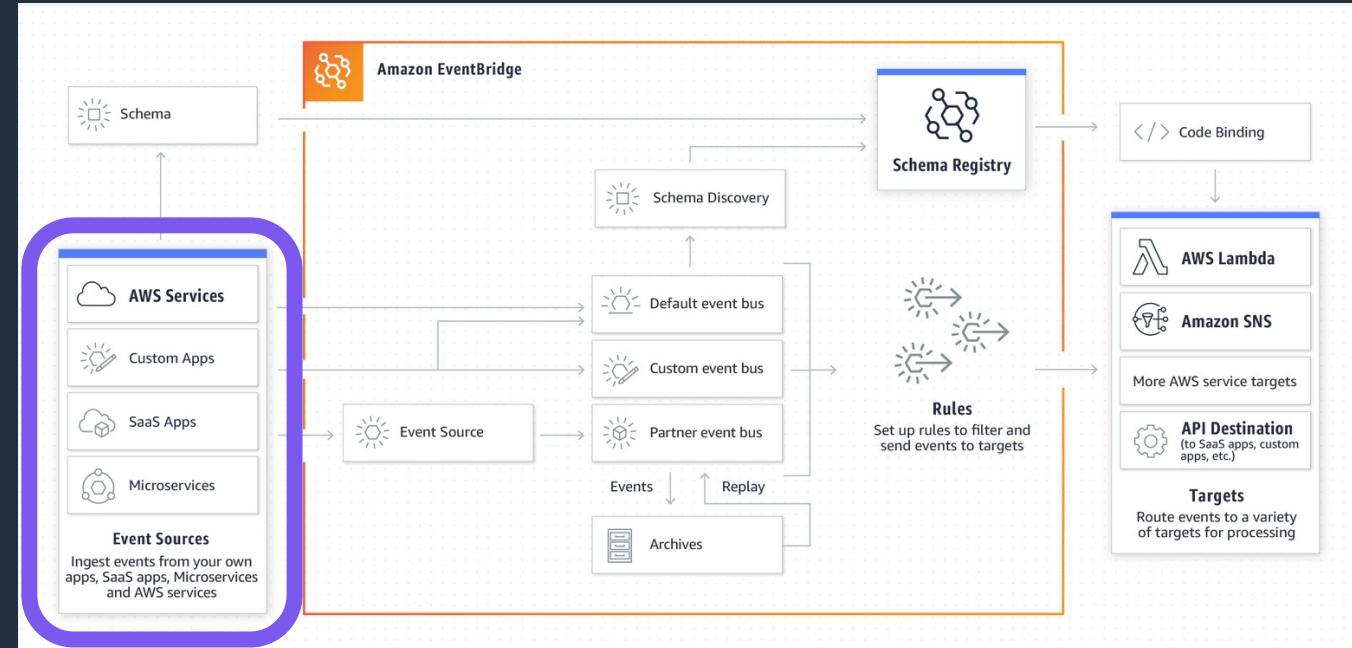
Amazon EventBridge es un servicio de bus de eventos simple, flexible, completamente administrado, de pago por uso, que facilita la ingestión y procesamiento de datos desde servicios de AWS, sus propias aplicaciones, y aplicaciones SaaS.





Amazon EventBridge – Fuente de eventos

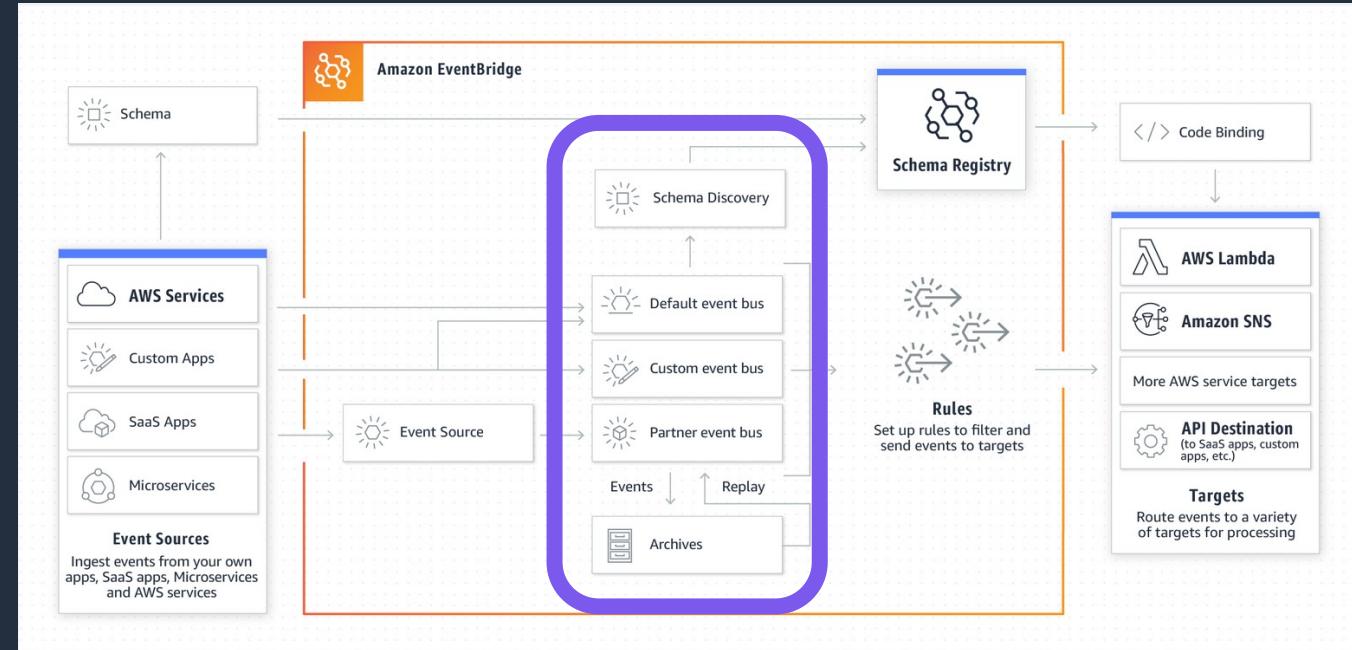
Amazon EventBridge es un servicio de bus de eventos simple, flexible, completamente administrado, de pago por uso, que facilita la ingestión y procesamiento de datos desde servicios de AWS, sus propias aplicaciones, y aplicaciones SaaS.





Amazon EventBridge – Bus de eventos

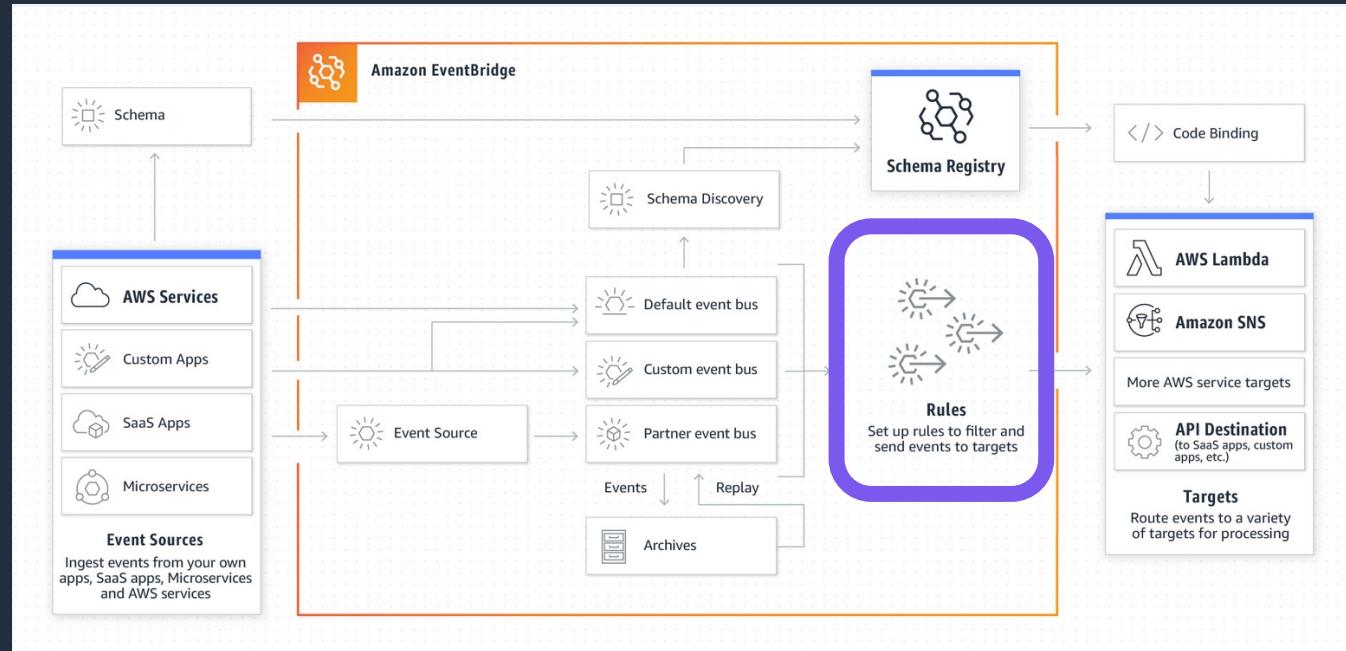
Amazon EventBridge es un servicio de bus de eventos simple, flexible, completamente administrado, de pago por uso, que facilita la ingestión y procesamiento de datos desde servicios de AWS, sus propias aplicaciones, y aplicaciones SaaS.





Amazon EventBridge - reglas

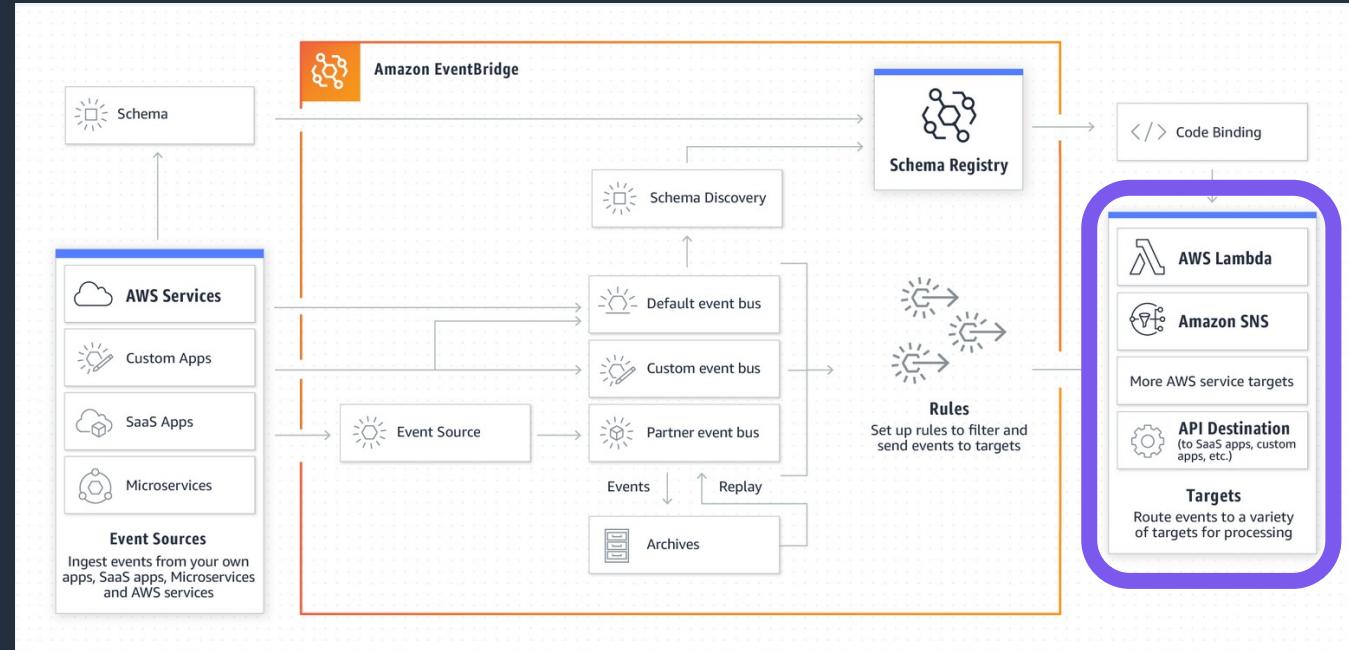
Amazon EventBridge es un servicio de bus de eventos simple, flexible, completamente administrado, de pago por uso, que facilita la ingestión y procesamiento de datos desde servicios de AWS, sus propias aplicaciones, y aplicaciones SaaS.





Amazon EventBridge - Objetivos

Amazon EventBridge es un servicio de bus de eventos simple, flexible, completamente administrado, de pago por uso, que facilita la ingesta y procesamiento de datos desde servicios de AWS, sus propias aplicaciones, y aplicaciones SaaS.



EventBridge: reglas de enrutamiento basadas en contenido

EventBridge: Evento de ejemplo

```
{  
  "source": "com.orders",  
  "detail-type": "OrderCreated",  
  "detail": {  
    "metadata": {  
    },  
    "data": {  
      "order-id": "1073459984",  
      "created-at": "2021-11-26T16:05:09-04:00",  
      "price": 24.62  
      "currency": "AU",  
    }  
  }  
}
```



EventBridge: reglas de enrutamiento basadas en contenido

EventBridge: Evento de ejemplo

```
{  
  "source": "com.orders",  
  "detail-type": "OrderCreated",  
  "detail": {  
    "metadata": {  
    },  
    "data": {  
      "order-id": "1073459984",  
      "created-at": "2021-11-26T16:05:09-04:00",  
      "price": 24.62  
      "currency": "AU",  
    }  
  }  
}
```

EventBridge: regla de ejemplo

```
{  
  "detail": {  
    "data": {  
      "currency": ["AU", "NZ"]  
    }  
  }  
}
```



EventBridge: reglas de enrutamiento basadas en contenido

EventBridge: Evento de ejemplo

```
{  
  "source": "com.orders",  
  "detail-type": "OrderCreated",  
  "detail": {  
    "metadata": {},  
    "data": {  
      "order-id": "1073459984",  
      "created-at": "2021-11-26T16:05:09-04:00",  
      "price": 24.62  
      "currency": "AU",  
    }  
  }  
}
```

EventBridge: regla de ejemplo

```
{  
  "detail": {  
    "data": {  
      "currency": ["AU", "NZ"]  
    }  
  }  
}
```



EventBridge: reglas de enrutamiento basadas en contenido

EventBridge: Evento de ejemplo

```
{  
  "source": "com.orders",  
  "detail-type": "OrderCreated",  
  "detail": {  
    "metadata": {},  
    "data": {  
      "order-id": "1073459984",  
      "created-at": "2021-11-26T16:05:09-04:00",  
      "price": 24.62  
      "currency": "AU",  
    }  
  }  
}
```

EventBridge: regla de ejemplo

```
{  
  "detail": {  
    "data": {  
      "currency": ["AU", "NZ"]  
    }  
  }  
}
```



Match!

20+ servicios de AWS y APIs externas como destino



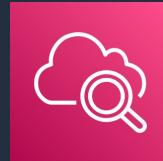
AWS Lambda



AWS Step Functions



Amazon Kinesis Streams



Amazon CloudWatch



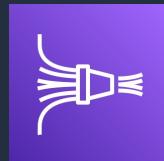
AWS CodePipeline



Amazon ECS



Amazon SNS



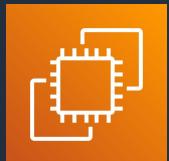
Amazon Kinesis Firehose



AWS Systems Manager



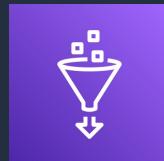
AWS CodeBuild



Amazon EC2



Amazon SQS



AWS Glue



AWS Incident Manager



Amazon Sagemaker



AWS Batch



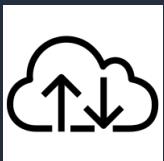
Amazon API Gateway



Amazon Redshift



Amazon Inspector

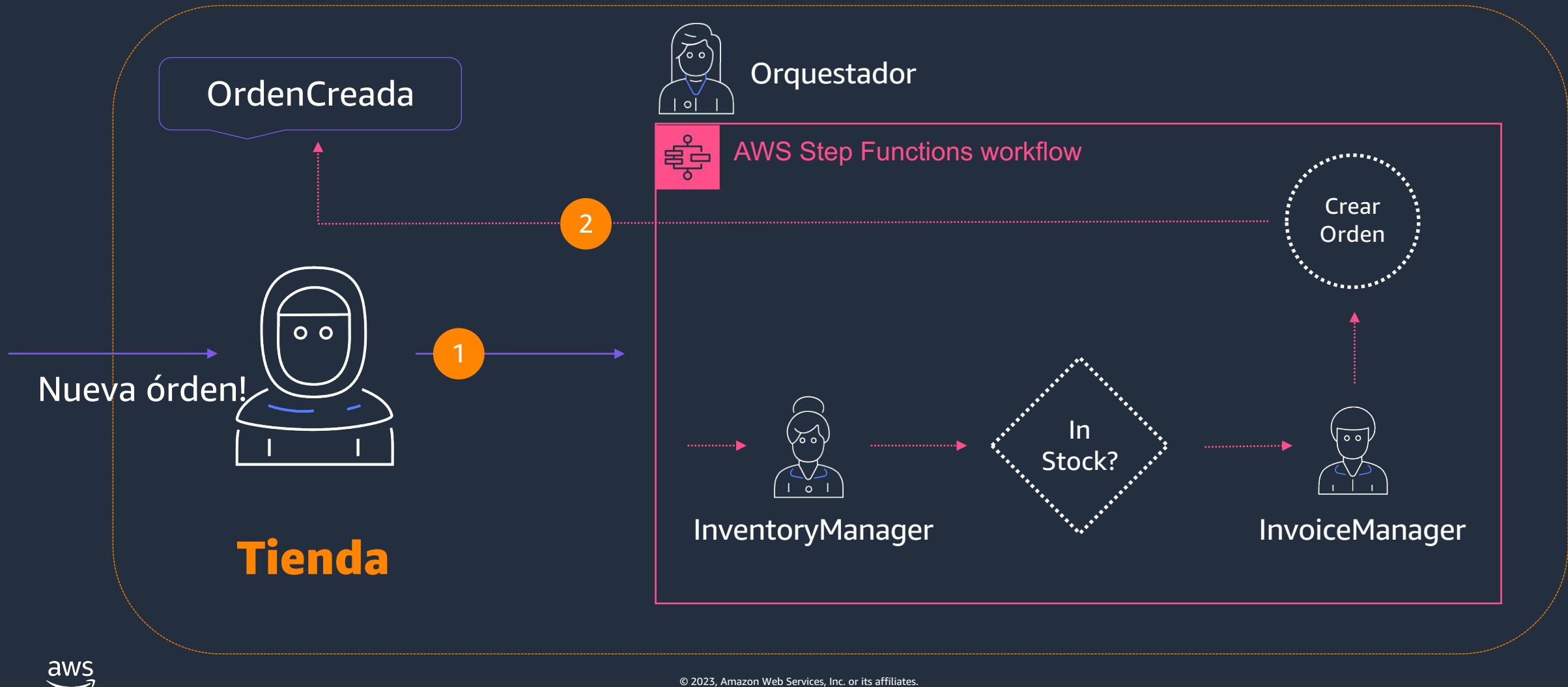


API Destinations

Orqueste un proceso de negocio *dentro de un dominio*, y que resulte en la publicación de un evento



Orqueste un proceso de negocio *dentro de un dominio*, y que resulte en la publicación de un evento





AWS Step Functions



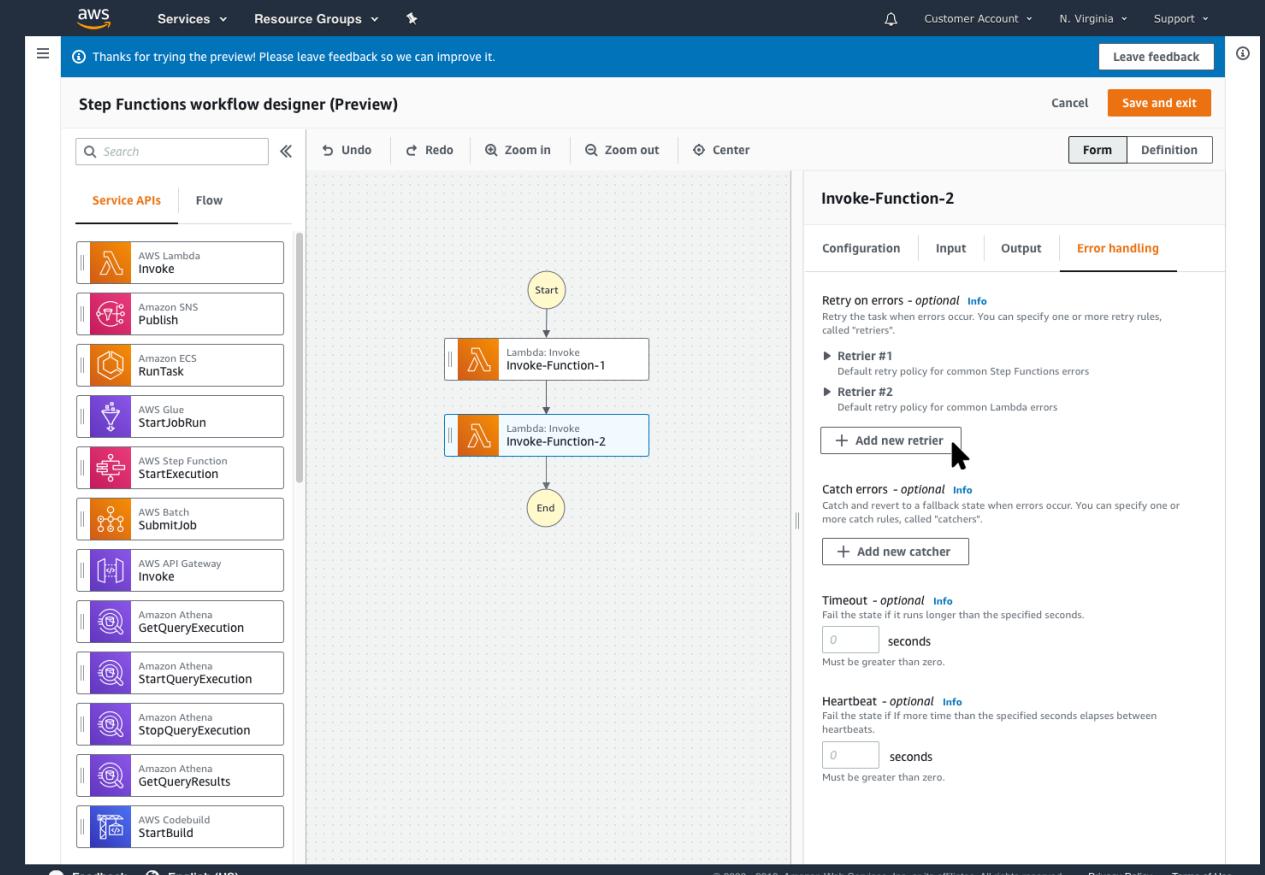
Los **workflows** que usted crea con Step Functions son llamados **máquinas de estados**, y **cada paso** de su workflow es llamado un **estado**.



Cuando ejecuta su máquina de estados, **cada movimiento** de un estado al siguiente es llamada una **transición de estado**.



Usted puede **reusar componentes**, editar fácilmente la secuencia de pasos o cambie el código invocado por las tareas a medida que las necesidades cambien.



AWS STEP FUNCTIONS WORKFLOW STUDIO

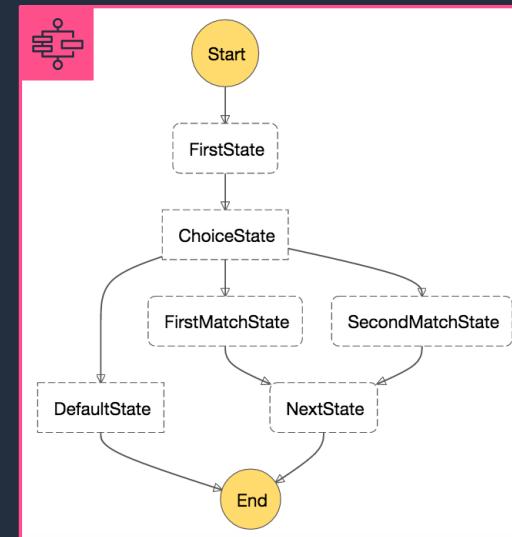




Defina

- Arrastre con Workflow Studio
- JSON (Amazon States Language)
- CDK (TypeScript, JavaScript, Python, Java, C#)
- SDK para ciencia de datos (Python)

Visualice



Ejecute y monitoree

The top screenshot shows the AWS Lambda console for a state machine named 'New_Order'. It displays a state transition graph with nodes like 'Start', 'FirstOrder', 'RegionChoice', 'CreateOrder', 'OpenOrder', 'ProcessOrder', 'DatabaseError', 'UnserveRegion', and 'End'. The graph shows various execution paths and states. To the right, there is an 'Execution Details' panel showing the execution status as 'Succeeded', the state machine ARN, execution ID, start and close times, and step details.

The bottom screenshot shows CloudWatch Logs for the log group '/aws/states/Tmp212262019-Logs'. It lists log events with timestamps and messages. One event at 2019-11-26 17:32:13 shows an 'ExecutionStarted' event with details about the role ARN and input parameters.



Step Functions: tipos de integración

Integraciones Optimizadas

Creadas para simplificar el uso de 17 servicios de AWS

Patrones de Integración soportados:

- Request Response
- Esperar un callback (.waitForTaskToken)
- Ejecutar un Job (.sync)

Integraciones AWS SDK

Llame más de 200 servicios de AWS directamente (9000+ acciones de API)

Patrones de Integración soportados:

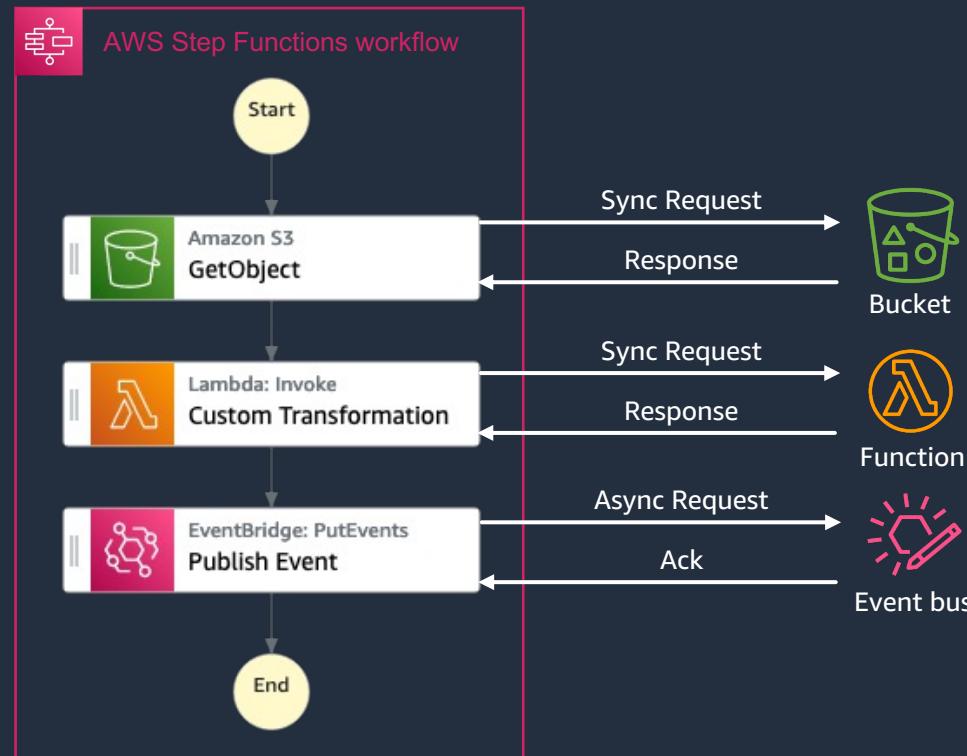
- Request Response
- Esperar un callback (.waitForTaskToken)





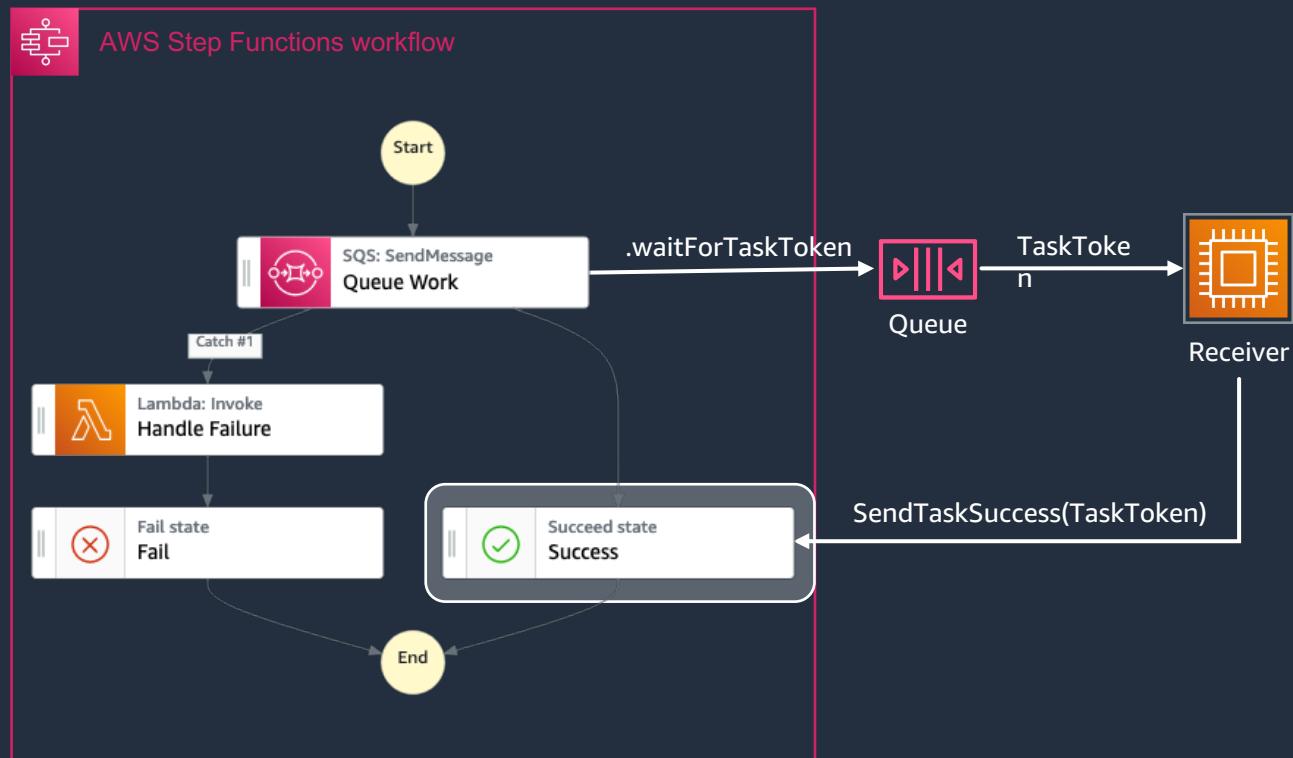
Step Functions: Request Response

Patrón de integración



Step Functions: Esperar por un Callback (.waitForTaskToken)

Patrón de integración



AWS Lambda Amazon SQS



Amazon ECS/Fargate



Amazon SNS



AWS Step Functions



Amazon EventBridge



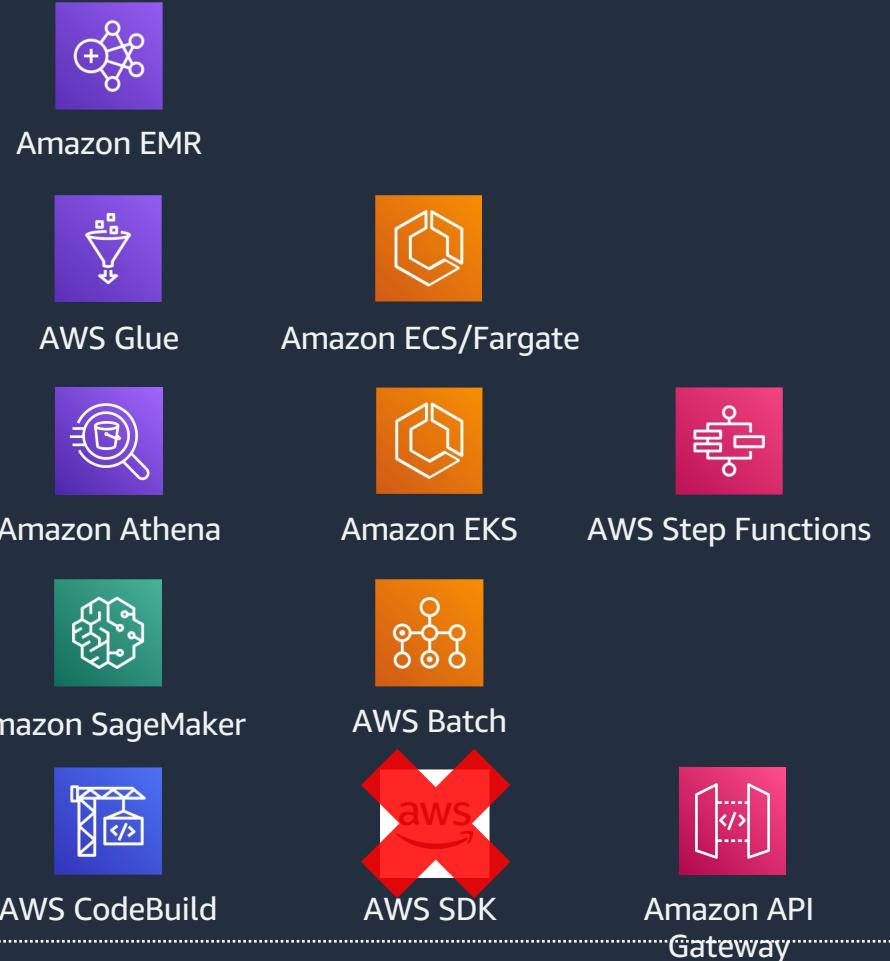
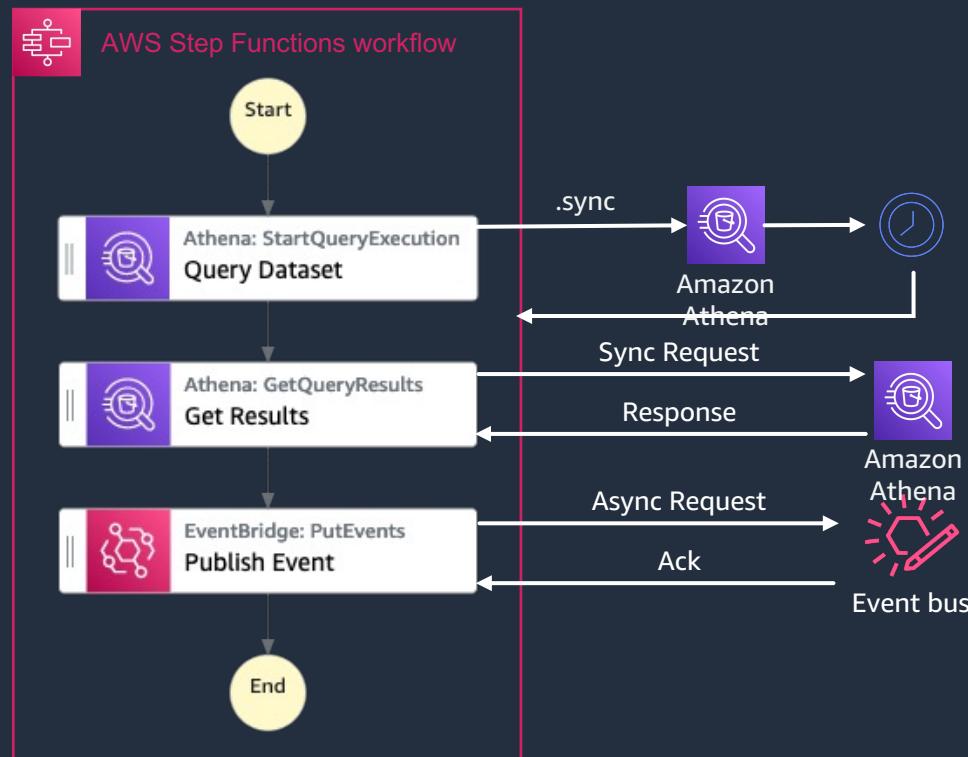
AWS SDK



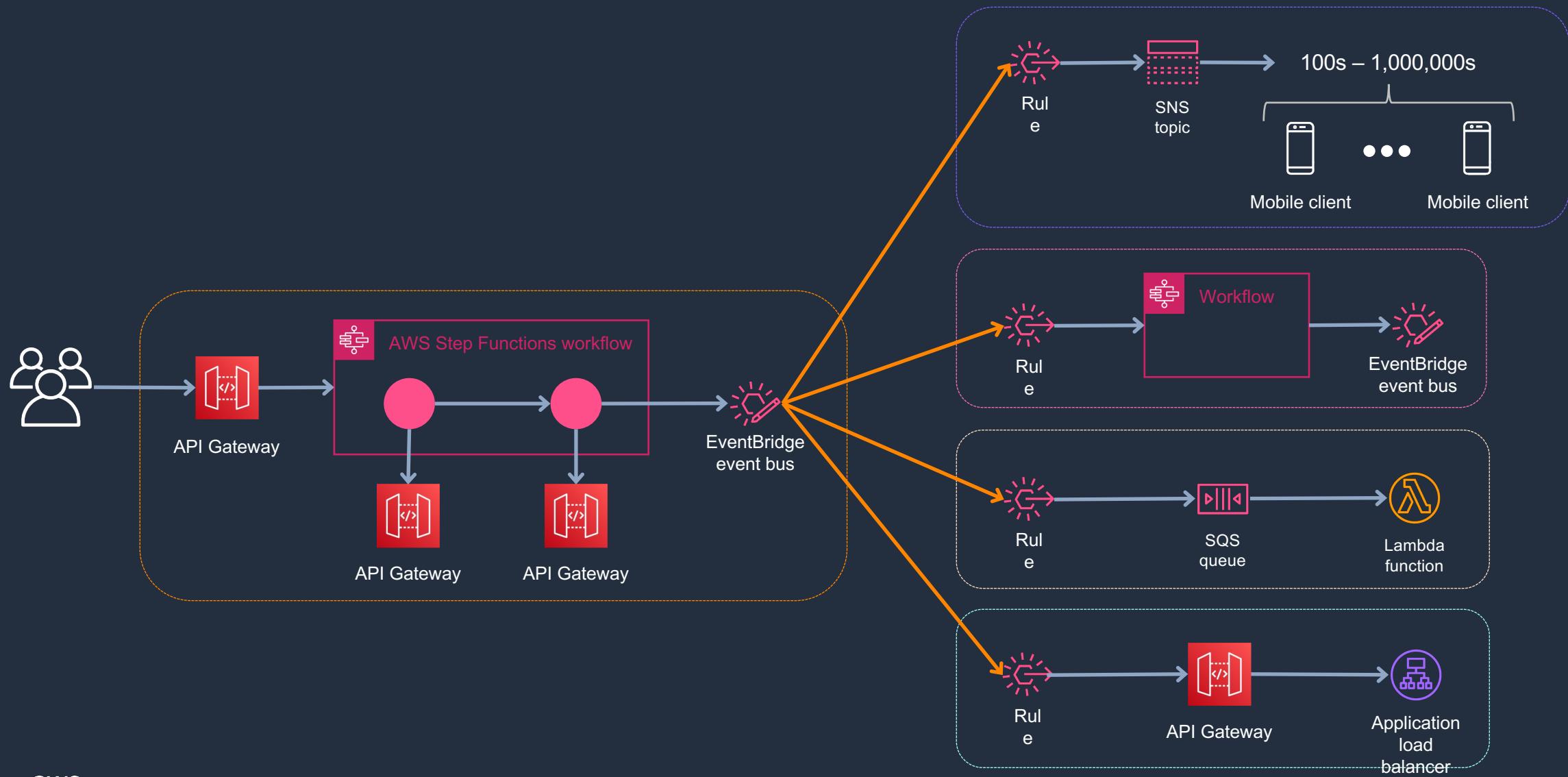
Amazon API
Gateway

Step Functions: Ejecute un Job (.sync)

Patrón de integración



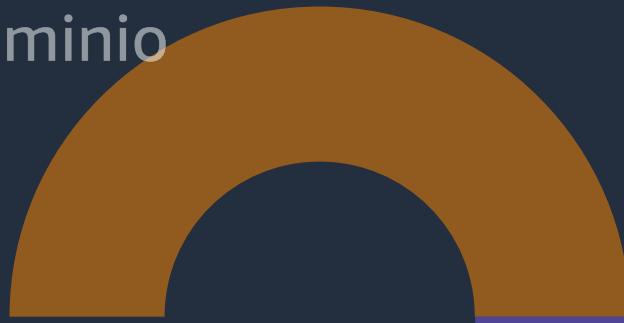
Mejor juntos: Orquestación + Coreografía



El camino hacia Arquitecturas Orientadas a Eventos

PASO 1

Comience con el
dominio



PASO 3

Seleccione un almacén de eventos

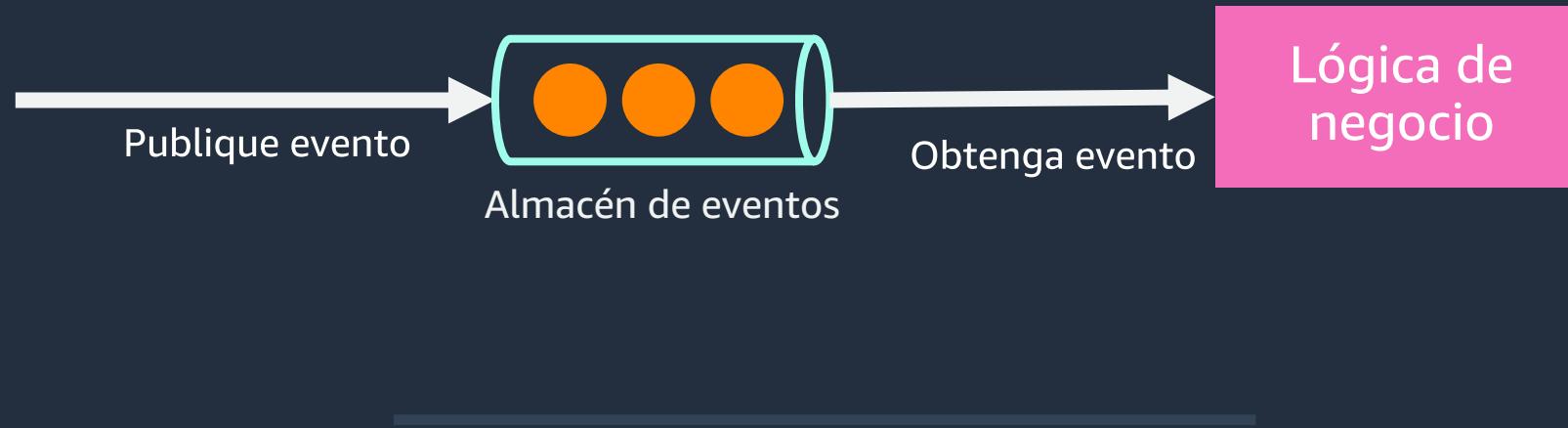


PASO 2

Coordine eventos



Mejore la resiliencia y escalabilidad con almacenes de eventos



Acumule los mensajes hasta que
los servicios estén disponibles
para procesarlos

Los almacenes de eventos manejan mensajes y streams de datos

Procesamiento de mensajes



- El **mensaje individual** es la unidad de trabajo
- Cómputo/procesamiento por mensaje
- La ocurrencia de los mensajes es variada
- Los mensajes son borrados luego de ser consumidos
- No es necesario tener rastro de la posición
- Funcionalidad DLQ incluída

Procesamiento de Streams



- El **stream de mensajes** es la unidad de trabajo
- Cómputo complejo sobre muchos mensajes
- Flujo constante de mensajes
- Los mensajes están disponibles luego de su consumo hasta su expiración
- Cada **consumidor necesita rastrear la posición actual en el stream**
- Funcionalidad DLQ no incluída



Amazon Simple Queue Service (SQS)

QUÉ ES

Servicio de encolamiento de mensajes simple, flexible, completamente manejado, para el intercambio confiable y continuo de cualquier volumen de mensajes desde cualquier lugar

(Standard y FIFO)

CASO DE USO

Construya micro servicios desacoplados, altamente escalables, sistemas distribuidos, y aplicaciones serverless en la nube

CAPACIDADES

ASOMBROSAS

Escalabilidad casi infinita sin pre-provisionar capacidad



Event Source



25B mensajes por hora





Amazon Kinesis

QUÉ ES

Le permite **ingestar**, **acumular** y **procesar** en **tiempo real** datos en streaming, para que usted pueda tener **insights** en **segundos** o minutos en vez de en horas o días.

CASO DE USO

Ingeste datos en tiempo real tales como video, audio, logs de aplicación, flujos de clics de un sitio web, y datos de telemetría de IoT para machine learning y analítica.

CAPACIDADES ASOMBROSAS

Puede manejar cualquier cantidad de datos en streaming y procesar datos provenientes de cientos de miles de Fuentes, con latencias muy bajas.



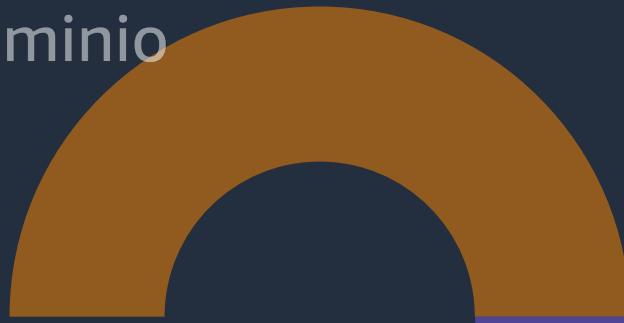
Event Source



El camino hacia Arquitecturas Orientadas a Eventos

PASO 1

Comience con el
dominio



PASO 3

Seleccione un almacén
de eventos



PASO 2

Coordine eventos



PASO 4

Estructure los eventos

Eventos simples vs. eventos descriptivos

Order 123 was created at 10:47 a.m. by customer 456



Eventos simples

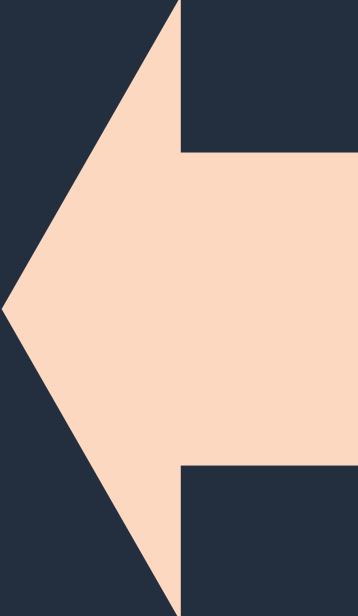
Order 123 was created at 10:47 a.m. by customer 456. The current status is Open, the total was \$237.51, the items were ...



Descripción completa

Consideraciones con eventos simples

Order 123 was
created by
customer 456



What are the
details for
order 123?

Consideraciones con eventos descriptivos

```
{  
  "source": "com.orders",  
  "detail-type": "OrderCreated",  
  "detail": {  
    "metadata": {  
      "idempotency-key": "c1b95b88"  
    },  
    "data": {  
      "order-id": "1073459984",  
      "status": "Open",  
      "total": "237.51"  
    }  
  }  
}
```

- El esquema de eventos debe soportar versiones anteriores
- El costo de mantener los valores puede incrementar con el tiempo



Consideraciones con eventos descriptivos

```
{  
  "source": "com.orders",  
  "detail-type": "OrderCreated",  
  "detail": {  
    "metadata": {  
      "idempotency-key": "c1b95b88"  
    },  
    "data": {  
      "order-id": "1073459984",  
      "status": "Open",  
      "total": "237.51"  
    }  
  }  
}
```

- El esquema de eventos debe soportar versiones anteriores
- El costo de mantener los valores puede incrementar con el tiempo



Consideraciones de diseño



Semántica de entrega

Entrega al menos una vez

Enviados pueden ser enviados a 1 destino más de una vez. Incluya lógica para detectar eventos duplicados rastreando los eventos que ya fueron procesados (**idempotencia**).

- Amazon EventBridge
- Amazon SNS Standard
- Amazon SQS Standard
- Amazon Kinesis

Entrega Exactamente 1 vez*

Especifique un identificador para que sea usado por el servicio AWS para **deduplicación**.

- Amazon MQ
- Amazon MSK
- Amazon SNS FIFO
- Amazon SQS FIFO

* *Por fuera del manejo de errores de la aplicación*



Unicidad del Evento

Idempotencia

La operación arrojará los mismos resultados sin importar si es ejecutada una o múltiples veces (EIP).

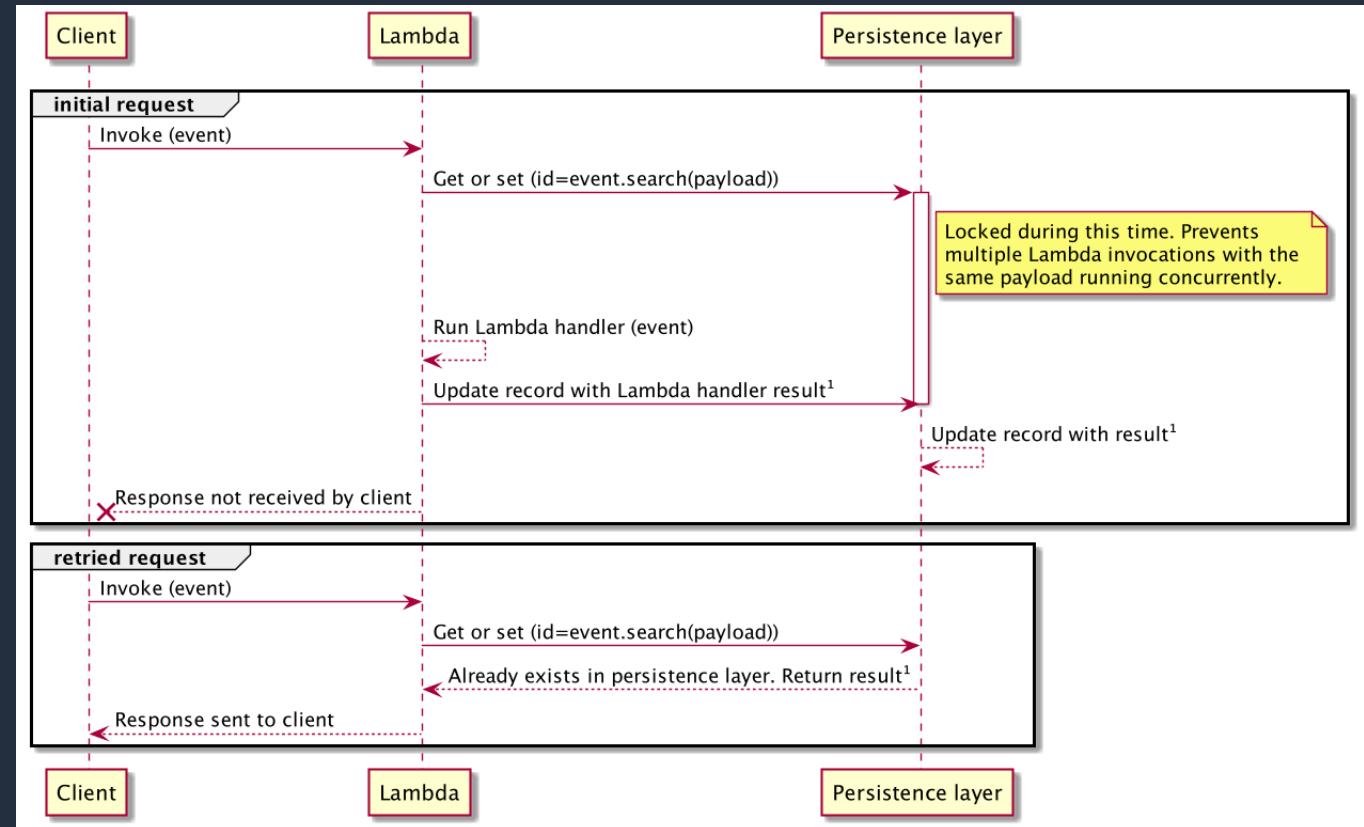
Llave de Idempotencia

Asignada al mensaje por el emisor para simplificar la deduplicación por parte del receptor.

```
{  
  "source": "com.orders",  
  "detail-type": "OrderCreated",  
  "detail": {  
    "metadata": {  
      "idempotency-key": "c1b95b88"  
    },  
    "data": {  
      "order-id": "1073459984"  
    }  
}
```

Estrategias para deduplicación de Eventos

- Maneje el estado de idempotencia en un repositorio externo de datos
- Especifique una Ventana de tiempo para deduplicación
- Librería de Idempotencia para Python en Lambda Powertools
- Haga que los reintentos se manejen de forma segura con APIs idempotentes (*AWS Builders' Library*)



(*Lambda Powertools Python Idempotency*, <https://bit.ly/3iWKu7z>)

Semántica de ordenamiento*

En desorden

Los Eventos pueden ser entregados en desorden.

- Amazon EventBridge
- Amazon SNS Standard
- Amazon SQS Standard

Ordenados

Los Eventos son entregados en orden dentro de una partición, grupo de mensajes, etc (no hay orden global).

- Amazon MQ
- Amazon MSK
- Amazon Kinesis
- Amazon SNS FIFO
- Amazon SQS FIFO

** El manejo de eventos desordenados es específico de la aplicación. Si la aplicación no puede ser diseñada para manejar eventos en desorden, considere mejor orquestación.*



Próximos pasos



Programe una reunión de seguimiento con el equipo de Serverless

- Mejores prácticas en Arquitecturas Orientadas a Eventos

<https://aws.amazon.com/event-driven-architecture/>

- Curso: Architecting Serverless Solutions

<https://www.aws.training/Details/eLearning?id=42594>

- Arquitecturas de referencia

<https://github.com/aws-samples/aws-serverless-ecommerce-platform>

- Workshop sobre cómo construir Arquitecturas Orientadas a Eventos en AWS

<http://event-driven-architecture.workshop.aws>





Thank you!

Ana Barragán

anambarr@amazon.com



Error handling

- Políticas de reintentos personalizados:
- De forma predeterminada: los eventos son reintentos de hasta 24 horas o 185 veces
- Ahora puedes configurar:
- MaximumEventAgeInSeconds: entre 60 y 86400 segundos (86400 o 24 horas por defecto)
- MaximumRetryAttempts: entre 0 y 185 (185 es el valor predeterminado)



When either condition is met, the event fails. It's then either dropped, which generates an increase to the FailedInvocations CloudWatch metric, or sent to a configured DLQ.