AWS COMMUNITY DAY PERÚ 2026

Cuando una API externa falla

Diseñando integraciones resilientes en AWS con ECS Fargate

Diego Matienzo

Senior Software Engineer | Technical Leadership | Enterprise Integrations | AWS



TRACK	SALA	HORA
Cloud Architecture & Modernization	A1002	11:15

3 de octubre de 2026



SOBRE MÍ

Diego Matienzo

Senior Software Engineer | Technical Leadership | Enterprise Integrations | AWS

Desarrollo soluciones backend e integraciones Empresariales sobre AWS. Me enfoco en arquitectura, resiliencia y automatización de procesos.

diegomatienzo · diegomc365 · @diegomc365

Lo que veremos hoy

01

El problema

¿Qué pasa cuando fallamos a mitad del proceso?

02

Diseñar la recuperación

Estado, errores e idempotencia.

03

Ejecutarlo en AWS

ECS Fargate y arquitectura de la demo.

04

Demo

Fallo
↓
Estado
↓
Recuperación

01

El problema

¿Qué pasa cuando fallamos a mitad del proceso?

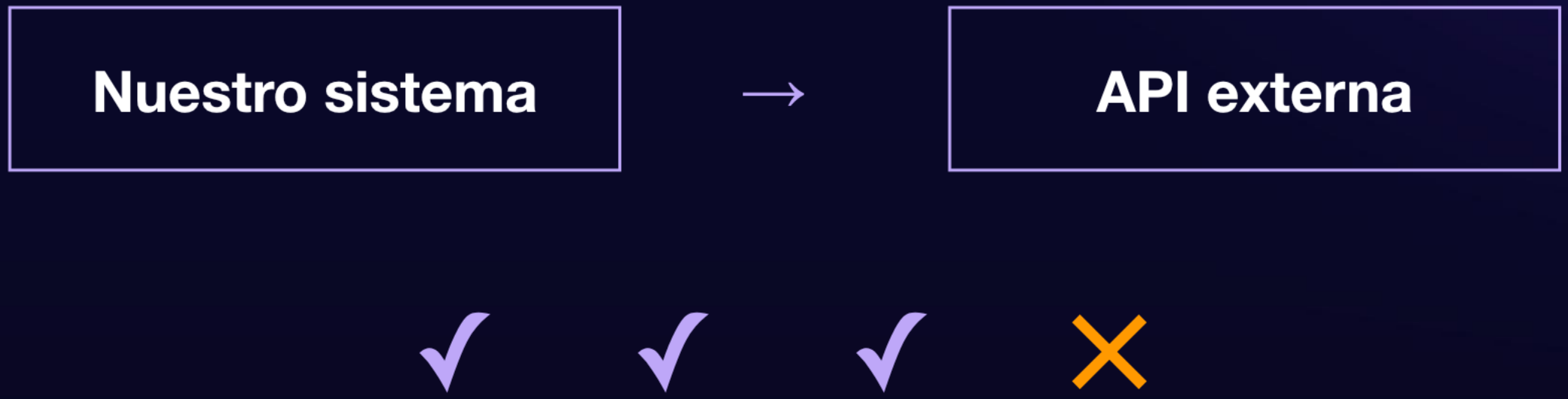


Todo falla, todo el tiempo

Werner Vogels

CTO de Amazon

Todo funciona… hasta que una dependencia falla.



ORDEN #1001


Validar orden
Crear transacción externa
Confirmar transacción
Actualizar nuestro sistema

Falló en el peor momento

Validar orden ✓

Crear transacción externa ✓

EXT-78432

Confirmar transacción ✗

Actualizar nuestro sistema ⏸

¿Qué hacemos ahora?

¿Simplemente volvemos a empezar?

INTENTO 1

Validar ✓

Crear transacción ✓

EXT-78432

Confirmar ✗

RETRY INGENUO

Validar ✓

Crear transacción ⚠️

EXT-91234

¿Acabamos de duplicar la operación?

02

Diseñar la recuperación

Estado, errores e idempotencia



¿Qué necesita recordar el sistema?

ESTADO

¿Terminó o falló?

PASO ACTUAL

¿Dónde quedó?

INTENTO

¿Cuántas veces procesamos ?

ID EXTERNO

¿Qué ya ocurrió afuera?

Del problema al modelo técnico

Estado → **status**

Paso actual → **currentStep**

Intentos → **attemptCount**

Transacción → **externalTransactionId**

Orden #1001

status

FAILED_RETRYABLE

currentStep

CONFIRM_EXTERNAL_TRANSACTION

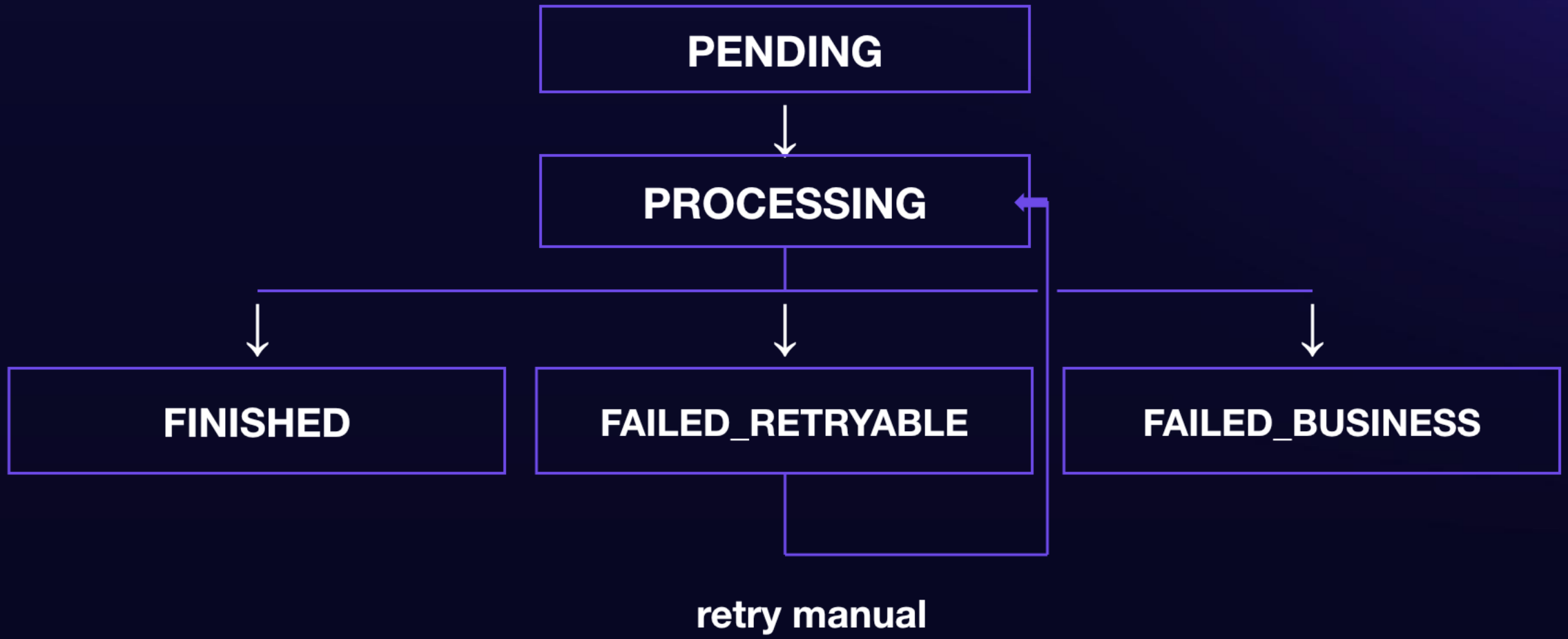
attemptCount

1

externalTransactionId

EXT-78432

Una operación no es solo “OK” o “ERROR”



No todos los errores merecen retry

↻ ERROR TEMPORAL

API no disponible

Timeout

HTTP 5xx

Puede tener sentido reintentar

⊘ ERROR FUNCIONAL

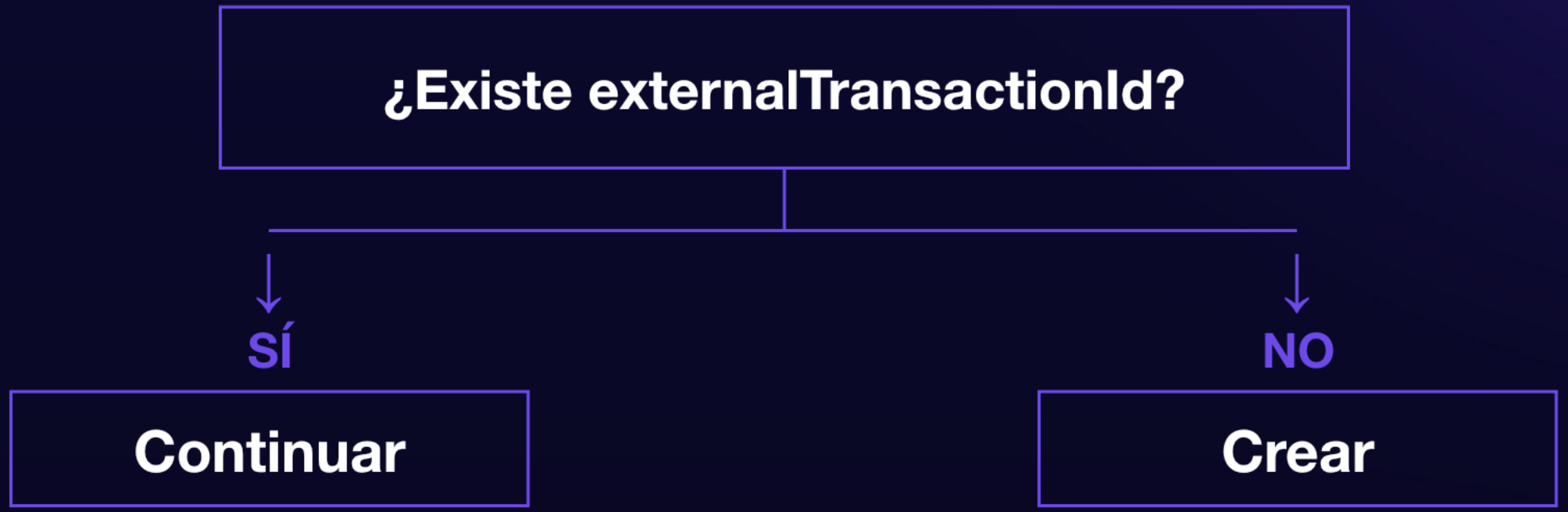
Orden cerrada

Datos inválidos

HTTP 422

Repetir no cambia el resultado

No repitas lo que ya ocurrió



EXT-78432 ya existe → no volver a crear

Idempotencia

Recuperar, no reiniciar

INTENTO 1

VALIDATE ✓

CREATE ✓ **EXT-78432**

CONFIRM ✗

INTENTO 2

CREATE ↱ ya realizado

CONFIRM ✓

UPDATE ✓

FINISH ✓

Resume > Restart

La resiliencia ocurre en dos capas

RESILIENCIA DE APLICACIÓN

Estado del proceso · Paso actual · Retry controlado
Idempotencia · Clasificación de errores

+

RESILIENCIA DE INFRAESTRUCTURA

Contenedores saludables · Reemplazo de tasks
Orquestación · Logs

03

Ejecutarlo en AWS

ECS Fargate y arquitectura de la demo



¿Qué hacen ECS y Fargate aquí?

Amazon ECS

Orquesta nuestros containers

Tasks · Services · Health

↓

AWS Fargate

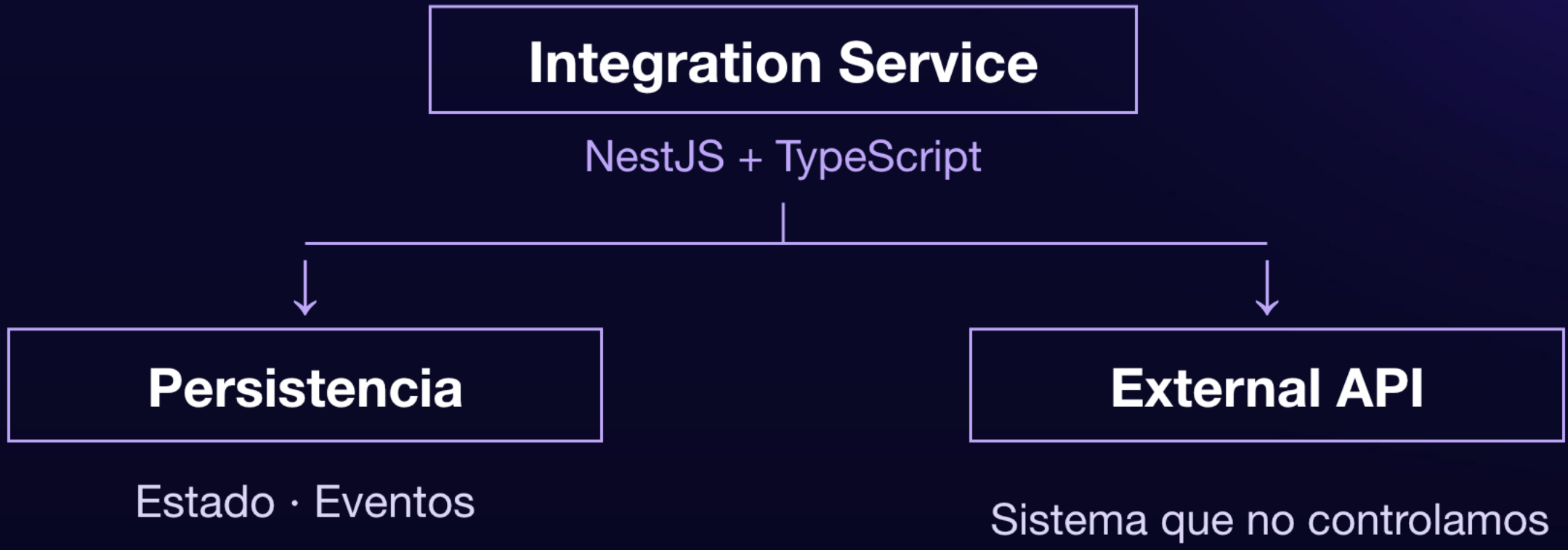
Ejecuta las tasks sin que administremos servidores

Compute administrado · Sin administrar EC2

↓

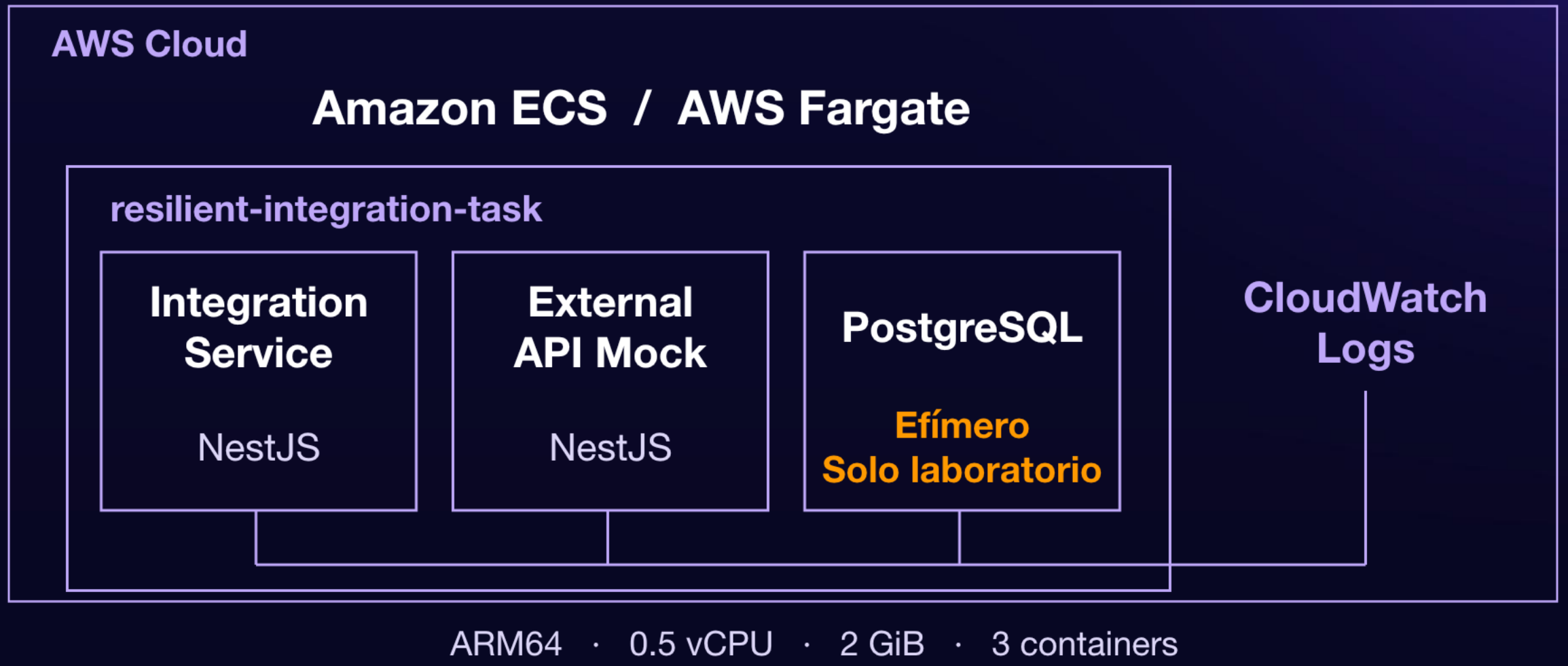
Containers

Arquitectura de la integración



NestJS · PostgreSQL · Containers

Así ejecutamos la demo en AWS



ARM64 · 0.5 vCPU · 2 GiB · 3 containers

04

Demo

Fallo -> Estado -> Recuperación



DEMO EN VIVO

Demo en vivo

Una orden, un fallo
y dos comandos.

```
npm run demo:start
```

```
npm run demo:retry
```

Fallar → entender → continuar

¿Qué harías ahora?

Transacción existente:

EXT-78432

¿Volvemos a crearla?

Lo que realmente ocurrió

INTENTO 1

VALIDATE ✓

CREATE ✓ **EXT-78432**

CONFIRM ✗

INTENTO 2

CREATE ↱ no se repite

CONFIRM ✓

UPDATE ✓

FINISH ✓

Mismo proceso. Mismo ID externo.
Sin duplicación.

Tres ideas para llevarte

1

Guarda suficiente estado

Debes saber dónde quedó el proceso.

2

Retry no significa reiniciar

No repitas efectos que ya ocurrieron.

3

Aplicación + infraestructura

ECS/Fargate complementa la resiliencia del proceso.

> Queremos asegurarnos de que el resultado de la llamada ocurra una sola vez, incluso si necesitamos realizarla varias veces.

Malcolm Featonby

Amazon Builder’s Library – Traducción de un fragmento

Recursos

Para seguir aprendiendo

aws-community-day-resilient-integration-demo

https://github.com/diegomc365/aws-community-day-resilient-integration-demo

CÓDIGO

Escanea para descargar



¿Preguntas?



Comparte tu experiencia

Sube tus fotos del día y etiqueta a la comunidad

Instagram	@aws.community.day.peru
LinkedIn	AWS Community Day Perú
TikTok	@aws.community.day.peru
X	@AWS_ComDay_Peru

awscommunityday.pe

AWS COMMUNITY DAY PERÚ 2026

¡Gracias!

Diego Matienzo

Senior Software Engineer – Technical Leadership - AWS

diegomatienzo

diegomc365

diegomc365

ESCANEA Y CONECTEMOS