

Cuando la nube se defiende sola

Remediación automática en **AWS** con eventos y funciones Lambda

Luis Lunar (Luis Eduardo)
Noviembre 19, 2025

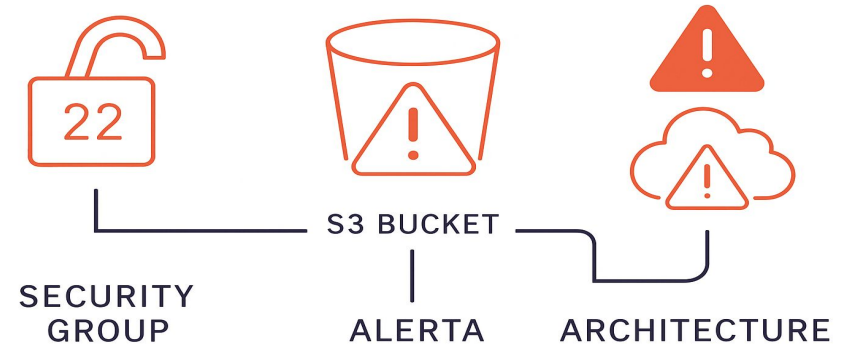


El conflicto: cuando la nube NO se defiende sola

El conflicto, cuando una mala configuración abre la puerta

- Un puerto 22 expuesto sin querer
- Un bucket S3 creado sin autorización
- Una regla insegura que nadie ve
- Un cambio que pasa “debajo del radar”

“¿Qué pasa si nadie corrige este error... y nadie se da cuenta?”



Qué vas a aprender hoy (Objetivos + Mapa de ruta)

Objetivos de aprendizaje

- Entender cómo AWS detecta acciones sensibles en tiempo real.
- Aprender a diseñar un pipeline de remediación automática.
- Conocer el rol de CloudTrail, EventBridge, Lambda, S3, SNS y CloudWatch.
- Revisar un laboratorio paso a paso sin hacer demo en vivo.
- Llevarte un modelo replicable para tu propia cuenta de AWS.



Visión global del pipeline de remediación automática

Pipeline de seguridad automatizada flujo completo:

1. CloudTrail registra la acción.
2. EventBridge filtra el evento
3. Lambda ejecuta la remediación.
4. CloudWatch Logs registra la evidencia.
5. SNS envía la alerta.
6. S3 centraliza la auditoría.

Resultado: La nube detecta, corrige y notifica sin intervención manual.

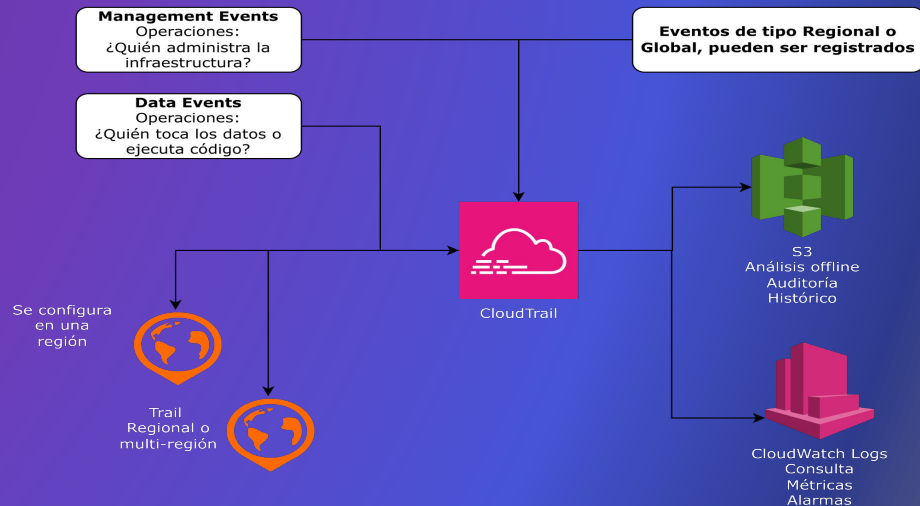


CloudTrail: ojos y memoria de la nube

CloudTrail: registro y trazabilidad total

- Registra cada acción API en tu cuenta.
- Captura *quién, cuándo, desde dónde y qué hizo*.
- Fuente primaria de eventos para seguridad y auditoría.
- Alimenta a EventBridge en tiempo real.

**“Si pasó en tu cuenta,
CloudTrail lo sabe.”**

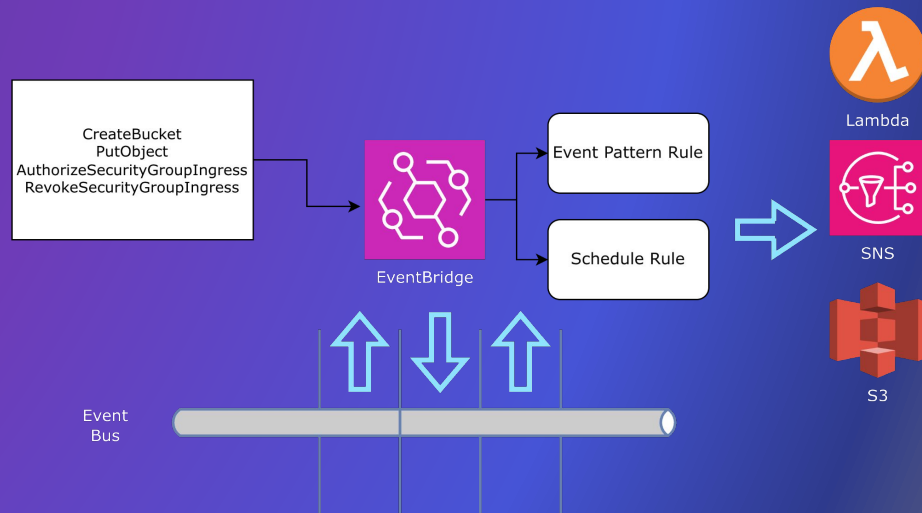


EventBridge: el motor que detecta y dispara acciones

EventBridge: detección inteligente en tiempo real

- Detecta eventos de CloudTrail en tiempo real.
- Filtra acciones sensibles mediante *Event Patterns*.
- Ejecuta automatizaciones vía Lambda.
- Permite reglas por servicio, acción, usuario, IP, región, etc.
- Base del modelo event-driven en seguridad.

“EventBridge convierte eventos en acciones.”



Lambda: el cerebro de la remediación

**Lambda toma el evento que detectó
EventBridge y aplica la corrección necesaria.**

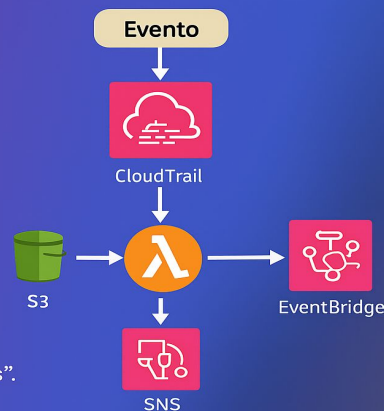
- EventBridge, no corrige nada.
- SNS, no corrige nada.
- CloudWatch Logs, no corrige nada.

**“Solo Lambda ejecuta la
remediación”**

Lambda: el cerebro de la remediación



- Ejecuta acciones correctivas automáticamente.
- Dos funciones en tu laboratorio:
 - Elimina bucket no autorizado tras un “CreateBucket”.
 - Revoca regla SSH 22 insegura tras un “AuthorizeSecurityGroupIngress”.



¿Qué tipo de remediaciones hace Lambda en tu laboratorio?

Lambda realiza dos acciones críticas:

a) **Remediación 1:** Bucket no autorizado

Ante un CreateBucket:

- Verifica si el bucket cumple las reglas esperadas.
- Si NO cumple → elimina automáticamente el bucket.
- Registra evidencia.
- Envía notificación vía SNS.

“Este es un control **preventivo** + **correctivo** inmediato.”

b) **Remediación 2:** Regla SSH 22 abierta

Ante un AuthorizeSecurityGroupIngress:

- Analiza el contenido del evento.
- Si detecta algo como 0.0.0.0/0 + puerto 22.
- Elimina la regla insegura.
- Registra evidencia en CloudWatch Logs.
- Envía alerta via SNS.

“Aquí Lambda te protege de un riesgo de **alto impacto**.”



¿Qué contiene el evento que recibe Lambda?

El evento que recibe Lambda es un JSON 100% igual al evento original de CloudTrail filtrado por EventBridge.

Contiene:

- Nombre del servicio: "eventSource": "s3.amazonaws.com".
- Operación: "eventName": "CreateBucket".
- Identidad del usuario: userIdentity.
- Dirección IP: sourceIPAddress.
- Recurso afectado: requestParameters.
- Hora exacta, región y cuenta.

Este JSON es suficiente para:

- Identificar el bucket creado.
- Identificar la regla de seguridad agregada.
- Decidir la acción automática.

“Lambda no **adivina** nada, simplemente lee el evento.”



Lambda necesita permisos (y por eso el rol IAM es clave)

Lambda NO puede corregir nada si no tiene permisos, debes darle un rol que permita:

Para S3:

- s3:DeleteBucket
- s3:ListBucket
- s3:GetBucketPolicy
- etc.

Para Security Groups:

- ec2:RevokeSecurityGroupIngress
- ec2:DescribeSecurityGroups

Para logs:

- logs:CreateLogGroup
- logs:PutLogEvents.

Y para SNS:

- sns:Publish.



“Lambda es la pieza que convierte un evento en una acción concreta, es donde ocurre la remediación automática. Sin Lambda, detectaríamos el problema... pero no haríamos nada al respecto.”



CloudWatch Logs, S3 y SNS: evidencias y alertas

CloudWatch Logs es donde **Lambda** deja todo registrado:

- Qué acción ejecutó.
- Qué recurso afectó.
- Qué decisión tomó.
- Qué valores encontró en el evento JSON.
- El resultado de la remediación.
- Errores, advertencias y detalles internos.

“CloudWatch Logs es donde queda la **evidencia técnica del proceso**.

Si quieres saber qué hizo Lambda exactamente, lo ves aquí.”



S3 → “El repositorio de auditoría”

S3 funciona como almacenamiento histórico. Mientras CloudWatch Logs guarda evidencia técnica temporal, S3 guarda evidencia permanente.

En el laboratorio, S3 sirve como:

- Repositorio de evidencia.
- Archivos históricos.
- Punto de control para auditoría

“CloudWatch Logs te dice qué pasó. S3 te **guarda** el registro histórico para auditoría.”



SNS → “La alerta inmediata”

SNS no guarda nada.

SNS no analiza nada.

SNS no ejecuta nada.

SNS **notifica lo que pasó.**

En el pipeline:

- Lambda remedia.
- Lambda publica en SNS.
- SNS envía un correo al equipo técnico o de seguridad.

“SNS no forma parte de la remediación, forma parte de la **comunicación**, es la voz del pipeline.”



Laboratorio – Pasos



PASO 1 — Crear bucket S3 de evidencias

PASO 1 — Crear bucket de evidencias en S3

Ir a S3 → Buckets – Clic en Create bucket

Configurar:

Bucket name: audit-evidence-`<randomkey>`

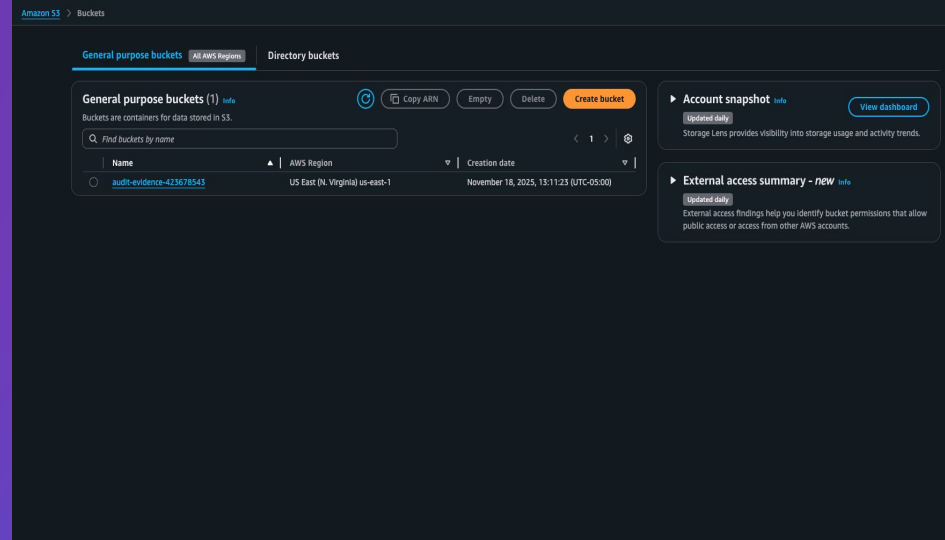
Región: us-east-1 (N. Virginia)

Block Public Access: Activado (por defecto)

Object Ownership: ACLs disabled

Ejemplo: audit-evidence-a92fbd

Nota: No cambiar encryption, versioning ni configuraciones avanzadas.



PASO 2 — Crear Trail en CloudTrail

PASO 2 — Habilitar el registro de eventos con CloudTrail

Ir a CloudTrail → Trails - Clic en Create trail
Configurar:

Trail name: AuditTrail

S3 bucket: Crear nuevo:

cloudtrail-logs-<randomkey>

Management events: Read/Write (activado)

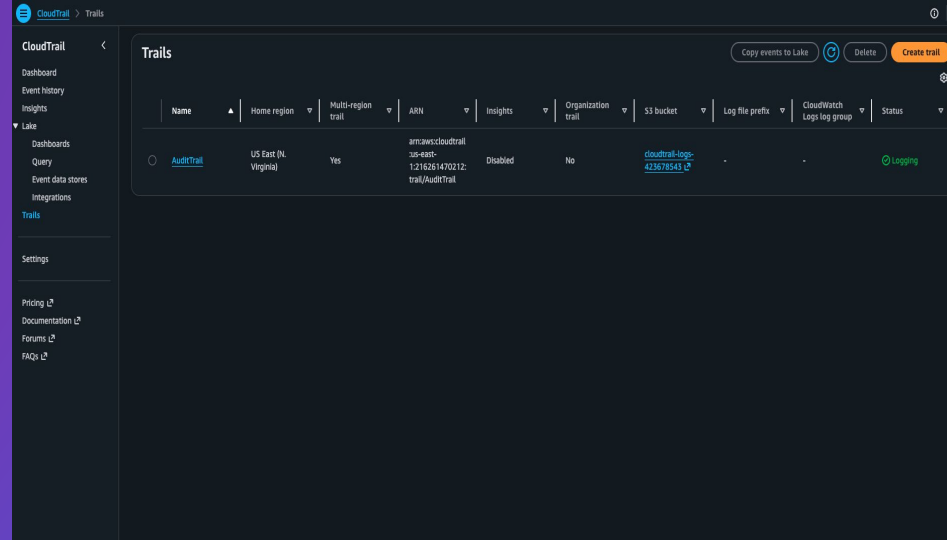
Data events: Opcional (dejar apagado para este lab)

Log file SSE-KMS Encryption: Dejar por defecto (opcional)

Clic en Create trail

Ejemplo de bucket:

cloudtrail-logs-a41dcb



PASO 3 — Crear SNS Topic + Suscripción por Email

PASO 2 — Habilitar el registro de eventos con CloudTrail

Ir a SNS (Simple Notification Service)

Configurar:

*Menú izquierdo → *Topics*

Clic en Create topic

Tipo: Standard

Nombre: AuditNotificationsTopic

Dejar el resto por defecto → Clic en Create topic

Configurar:

*Crear suscripción por correo

Dentro del Topic → *Subscriptions*

Clic en Create subscription

Protocol: Email – **Endpoint:** (tu correo real, ej: miemail@gmail.com)

Clic en Create subscription



PASO 3 — Crear SNS Topic + Suscripción por Email

Amazon SNS > Topics > Create topic

Create topic

Details

Type [info](#)
Topic type cannot be modified after topic is created.

☐ FIFO (first-in, first-out)

- Strictly-preserved message ordering
- Exactly-once message delivery
- Subscription protocols: SQS, Lambda, Data Firehose, HTTP, SMS, email, mobile application endpoints

☒ Standard

- Best-effort message ordering
- At-least-once message delivery
- Subscription protocols: SQS, Lambda, Data Firehose, HTTP, SMS, email, mobile application endpoints

Name

AuditNotificationsTopic

Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).

Display name - optional [info](#)
To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message.

AuditNotificationsTopic

Maximum 100 characters.

► **Encryption - optional**
Amazon SNS provides in-transit encryption by default. Enabling server-side encryption adds at-rest encryption to your topic.

► **Access policy - optional** [info](#)
This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic.

► **Data protection policy - optional** [info](#)
This policy defines which sensitive data to monitor and to prevent from being exchanged via your topic.

Amazon SNS > Subscriptions > Create subscription

Create subscription

Details

Topic ARN

arn:aws:sns:us-east-1:123456789012:audit-notifications

Protocol

The type of endpoint to subscribe to.

Email

Endpoint

An email address that can receive notifications from Amazon SNS.

luanarg@example.com

ⓘ After your subscription is created, you must confirm it. [info](#)

► **Subscription filter policy - optional** [info](#)
This policy filters the messages that a subscriber receives.

► **Redrive policy (dead-letter queue) - optional** [info](#)
Send undeliverable messages to a dead-letter queue.

Cancel Create subscription



PASO 4 — Crear la función Lambda de remediación – Parte 1

PASO 2 — Habilitar el registro de eventos con CloudTrail

Ir a Lambda → Create function

Configurar:

Name: ProcessAuditEvent

Runtime: Python 3.12

Architecture: x86_64 (o ARM si lo prefieres)

Execution role:

Usar rol existente:

ProcessAuditEventExecutionRole

Verificar permisos del rol

El rol debe incluir (previamente creado):

- AWSLambdaBasicExecutionRole
- AmazonS3FullAccess
- AmazonEC2FullAccess
- AmazonSNSFullAccess
- CloudWatchLogsFullAccess

Configurar variables de entorno

EVIDENCE_BUCKET = audit-evidence-`<id>`

SNS_TOPIC_ARN =

arn:aws:sns:us-east-1:`<id>`:AuditNotificationsTopic

Finalmente **pegar el código** compartido por la comunidad.



PASO 4 — Crear la función Lambda de remediación – Parte 2

Lambda

Functions

Create function

Create function Info

Choose one of the following options to create your function.

Author from scratch

Start with a simple Hello World example.

Use a blueprint

Build a Lambda application from sample code and configuration presets for common use cases.

Container image

Select a container image to deploy for your function.

Basic information

Function name

Enter a name that describes the purpose of your function.

ProcessAuditEvent

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Runtime

Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Python 3.12

Architecture

Choose the instruction set architecture you want for your function code.

arm64

x86_64

Permissions

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

► Change default execution role

► Additional configurations

Use additional configurations to set up networking, security, and governance for your function. These settings help secure and customize your Lambda function deployment.

Cancel

Create function

Info

Tutorials

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

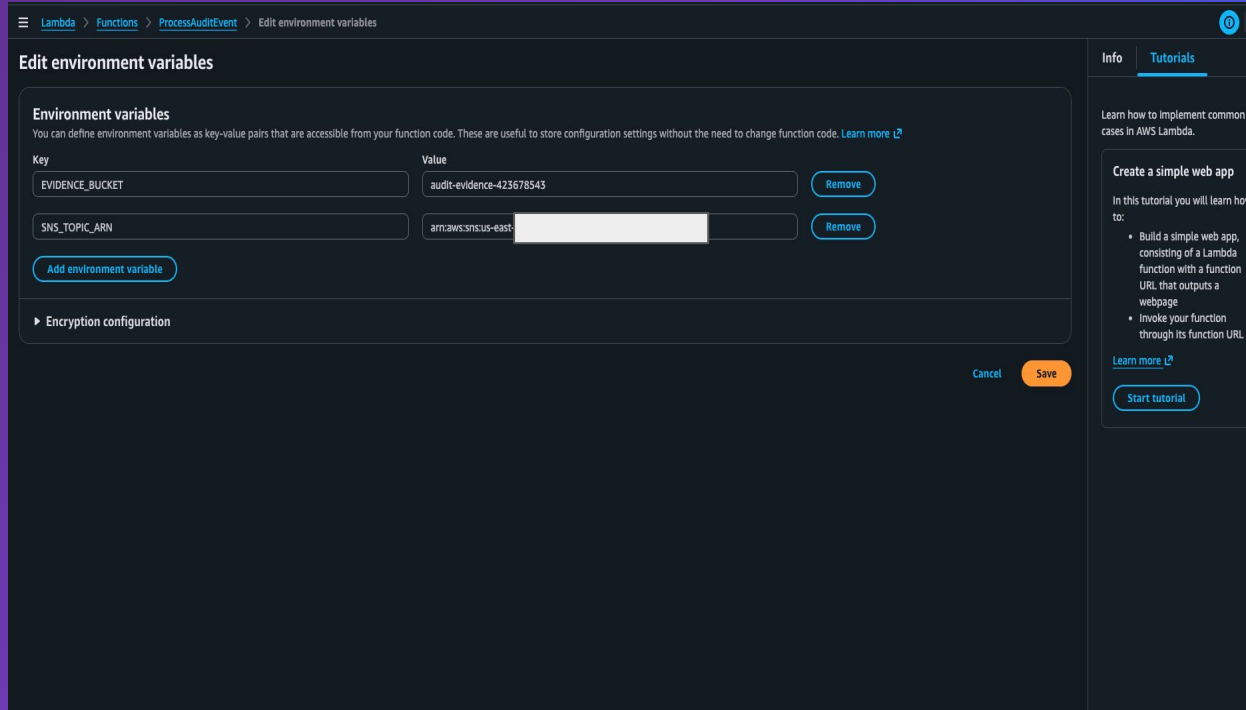
In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

[Learn more](#)

Start tutorial

PASO 4 — Crear la función Lambda de remediación – Parte 3



The screenshot shows the AWS Lambda console interface for editing environment variables. The breadcrumb trail at the top reads: **Lambda** > **Functions** > **ProcessAuditEvent** > **Edit environment variables**. The main heading is **Edit environment variables**. Below it, the **Environment variables** section explains that these are key-value pairs accessible from function code, with a [Learn more](#) link. A table lists two variables: **EVIDENCE_BUCKET** with value **audit-evidence-423678543**, and **SNS_TOPIC_ARN** with value **arn:aws:sns:us-east-1:123456789012:my-topic**. Each variable has a **Remove** button. An **Add environment variable** button is at the bottom left of the table. Below the table is an **Encryption configuration** section with a right-pointing arrow. At the bottom right are **Cancel** and **Save** buttons. On the right sidebar, the **Tutorials** tab is active, showing a section titled **Create a simple web app** with a description and two bullet points: 'Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage' and 'Invoke your function through its function URL'. There is a [Learn more](#) link and a **Start tutorial** button.

Edit environment variables

Environment variables
You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more](#)

Key	Value	
EVIDENCE_BUCKET	audit-evidence-423678543	Remove
SNS_TOPIC_ARN	arn:aws:sns:us-east-1:123456789012:my-topic	Remove

[Add environment variable](#)

► Encryption configuration

[Cancel](#) [Save](#)

Info **Tutorials**

Learn how to implement common cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

[Learn more](#)

[Start tutorial](#)



PASO 5 — Crear reglas en EventBridge – Parte 1

PASO 5 — Configurar reglas EventBridge para activar Lambda

Regla: CreateBucket

Detecta cuando se crea un bucket sin autorización.

- **Rule name:** DetectCreateBucket
- **Service:** S3
- **Event type:** CreateBucket
- **Target:** ProcessAuditEvent
- **Event Pattern JSON:**

```
{
  "source": ["aws.s3"],
  "detail-type": ["AWS API Call via
CloudTrail"],
  "detail": {
    "eventSource": ["s3.amazonaws.com"],
    "eventName": ["CreateBucket"]
  }
}
```



PASO 5 — Crear reglas en EventBridge – Parte 2

PASO 5 — Configurar reglas EventBridge para activar Lambda

Regla: CreateBucket

Detecta cuando se crea un bucket sin autorización.

- **Rule name:** DetectInsecureSSH
- **Service:** EC2
- **Event:** Reglas de SG modificadas
- **Target:** ProcessAuditEvent
- **Event Pattern JSON:**

```
{
  "source": ["aws.ec2"],
  "detail": {
    "eventSource":
    ["ec2.amazonaws.com"],
    "eventName": [
      "AuthorizeSecurityGroupIngress",
      "RevokeSecurityGroupIngress",
      "ModifySecurityGroupRules"
    ]
  }
}
```



PASO 5 — Crear reglas en EventBridge – Parte 1

Amazon EventBridge > Rules > DetectCreateBucket > Edit rule

Amazon EventBridge <

Dashboard

▼ Developer resources

Learn

Sandbox

Quick starts

▼ Buses

Event buses

[Rules](#) **Updated**

Global endpoints

Archives

Replays

▼ Pipes

Pipes

▼ Scheduler

Schedules

Schedule groups

▼ Integration

Partner event sources

API destinations

Connections

▼ Schema registry

Schemas

Event pattern [Info](#)

Creation method

☐ Use schema
Use an Amazon EventBridge schema to generate the event pattern.

☒ Use pattern form
Use a template provided by EventBridge to create an event pattern.

☐ Custom pattern (JSON editor)
Write an event pattern in JSON.

Event source

AWS service or EventBridge partner as source

AWS services

AWS service

The name of the AWS service as the event source

S3 (Simple Storage Service)

Event type

The type of events as the source of the matching pattern

Bucket-Level API Call via CloudTrail

All events that are delivered via CloudTrail have AWS API Call via CloudTrail as the value for detail-type. Events from API actions that start with the keywords List, Get, or Describe are not processed by EventBridge, with the exception of events from the following STS actions: GetFederationToken and GetSessionToken. Data events (for example, for Amazon S3 object level events, DynamoDB, and AWS Lambda) must have trails configured to receive those events.

[Learn more.](#)

Event pattern

Event pattern, or filter to match the events

```
1 {
2   "source": ["aws.s3"],
3   "detail-type": ["AWS API Call via CloudTrail"],
4   "detail": {
5     "eventSource": ["s3.amazonaws.com"],
6     "eventName": ["createBucket"]
7   }
8 }
```

Copy

Test pattern

Edit pattern

Event Type Specification 1

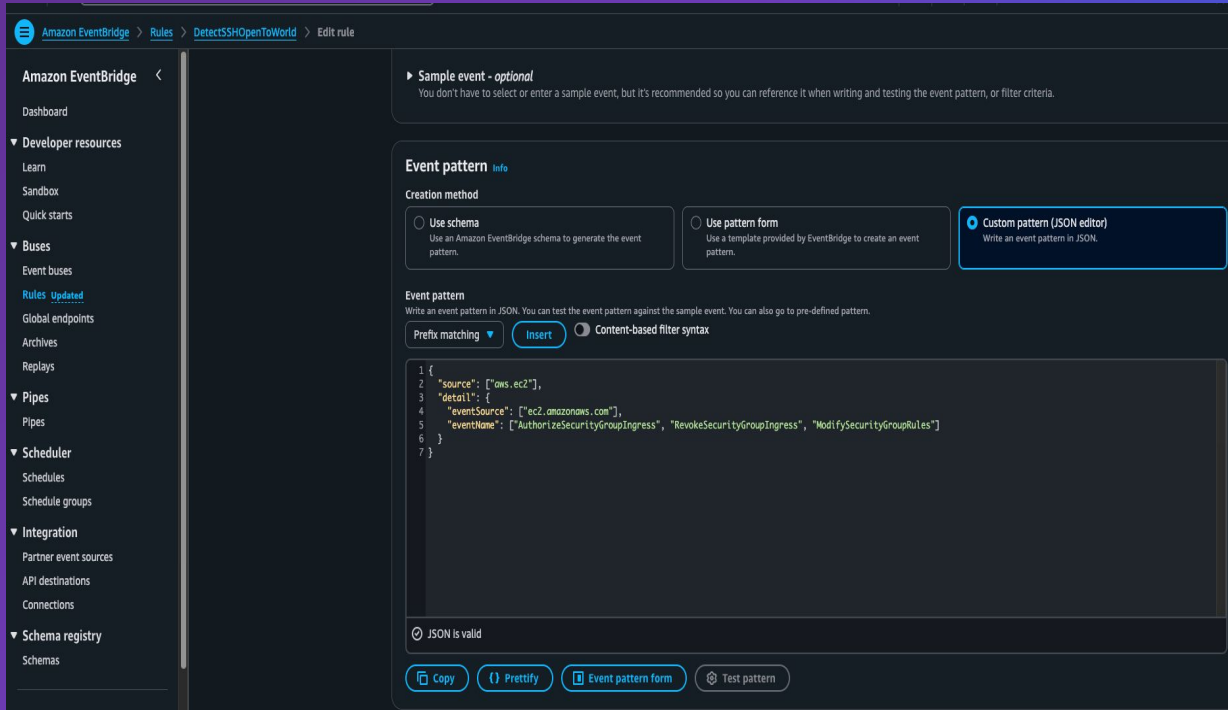
☐ Any operation

☒ Specific operation(s)

Specific operation(s)

CreateBucket

PASO 5 — Crear reglas en EventBridge – Parte 2



The screenshot shows the Amazon EventBridge console interface for editing a rule named 'DetectSSHOpenToWorld'. The left sidebar contains navigation links for Dashboard, Developer resources, Buses, Pipes, Scheduler, Integration, and Schema registry. The main content area is titled 'Edit rule' and includes a breadcrumb trail: Amazon EventBridge > Rules > DetectSSHOpenToWorld > Edit rule.

Sample event - optional
You don't have to select or enter a sample event, but it's recommended so you can reference it when writing and testing the event pattern, or filter criteria.

Event pattern info

Creation method

- ☐ Use schema
Use an Amazon EventBridge schema to generate the event pattern.
- ☐ Use pattern form
Use a template provided by EventBridge to create an event pattern.
- ☒ Custom pattern (JSON editor)
Write an event pattern in JSON.

Event pattern
Write an event pattern in JSON. You can test the event pattern against the sample event. You can also go to pre-defined pattern.

Prefix matching ☒ Insert ☐ Content-based filter syntax

```
1 {
2   "source": ["aws.ec2"],
3   "detail": {
4     "eventSource": ["ec2.amazonaws.com"],
5     "eventName": ["AuthorizeSecurityGroupIngress", "RevokeSecurityGroupIngress", "ModifySecurityGroupRules"]
6   }
7 }
```

☒ JSON is valid



PASO 6 — Pruebas Reales del Pipeline

Escenario S3: Bucket no autorizado

Crear un bucket temporal (cualquier nombre).
Esperar 5–10 segundos mientras EventBridge procesa el evento.

Confirmar que el bucket se eliminó automáticamente.

Revisar CloudWatch Logs → buscar log de Lambda:

- Acción detectada: CreateBucket
- Bucket marcado como no autorizado
- Acción realizada: DeleteBucket

Verificar correo SNS confirmando la remediación.

Escenario Security Group: SSH inseguro

Crear o seleccionar un Security Group de prueba.
Agregar la regla:

- Port 22
- Source: 0.0.0.0/0

Esperar 5–10 s: Lambda debe eliminar automáticamente la regla insegura.

Validar en CloudWatch Logs:

- Evento: AuthorizeSecurityGroupIngress
- Evaluación de riesgo
- Remediación ejecutada: RevokeSecurityGroupIngress



PASO FINAL — Limpieza del entorno

Limpieza del laboratorio

- Eliminar Lambda
- Eliminar reglas EventBridge
- Eliminar SG y buckets de prueba
- Eliminar SNS Topic si ya no se usa



Referencias y Documentación Oficial

AWS — Servicios utilizados en el laboratorio

CloudTrail (Registro y auditoría)

<https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-user-guide.html>

EventBridge (Reglas y automatización)

<https://docs.aws.amazon.com/eventbridge/latest/userguide/what-is-amazon-eventbridge.html>

AWS Lambda (Remediación automática)

<https://docs.aws.amazon.com/lambda/latest/dg/welcome.html>

Amazon S3 (Evidencias y almacenamiento)

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html>

Amazon SNS (Notificaciones por correo)

<https://docs.aws.amazon.com/sns/latest/dg/welcome.html>

