

LABORATORIO DE MONITOREO Y AUDITORÍA EN AWS

Servicios utilizados: AWS CloudTrail y AWS CloudWatch

1. Introducción

En este laboratorio, mi objetivo fue implementar una capa de observabilidad en mi infraestructura de AWS. Para ello, configuré **CloudTrail** con el fin de registrar toda la actividad de la cuenta y **CloudWatch** para monitorear el rendimiento de una instancia EC2, estableciendo alarmas automáticas ante niveles críticos de consumo de CPU.

2. AWS CloudTrail

Iniciamos el proceso explorando el **Historial de eventos** de CloudTrail, lo cual nos permitió visualizar las últimas acciones realizadas en la cuenta (quién creó qué recurso y en qué momento).

The screenshot shows the AWS CloudTrail console with the 'Event history' tab selected. On the left, there's a navigation sidebar with options like Dashboard, Event history, Insights, Lake (which is expanded to show Dashboards, Query, Event data stores, Integrations, and Trails), Settings, Pricing, Documentation, Forums, and FAQs. The main area has a heading 'Event history (50+)' with a 'Info' link. Below it, a message says 'Event history shows you the last 90 days of management events.' There's a 'Lookup attributes' section with a dropdown for 'Resource type' set to 'AWS: IAM USER' and a search bar. To the right, there are buttons for 'Download events', 'Query in Lake', 'Create Athena table', and a 'Clear filter' button. A large table lists events with columns for 'Event name' and 'Event time'. Most entries are 'GetPolicy' or 'GetPolicyVersion' calls from February 20, 2026, at various times. At the bottom of the table, there are buttons for 'Clear and dismiss', 'Cancel', and 'Apply'. Above the table, there's a date range selector with 'Relative range' and 'Absolute range' tabs, a calendar for selecting dates, and input fields for 'Start date' (2026/02/20), 'Start time' (23:00:00), 'End date' (2026/02/21), and 'End time' (02:00:00). A note says 'Range must be between today and the past 90 days. Use 24 hour format.' Below the table, there's a dropdown for 'Time zone' set to 'UTC'. The overall interface is clean and modern, typical of AWS services.

The screenshot shows the AWS CloudTrail Event history interface. On the left, there's a navigation sidebar with options like Dashboard, Event history (which is selected), Insights, Lake, and Settings. The main area has a header "Event history (50+)" with a "Info" link. Below it, a message says "Event history shows you the last 90 days of management events." There are buttons for "Download events", "Query in Lake", and "Create Athena table". A search bar filters results by "Resource type" (set to "AWS: IAM USER") and a date range ("2026-02-20T23:00:00Z — 2026-02-21T02:00:00Z"). A "Clear filter" button is also present. The main content is a table with columns: "Event name", "Event time", "User name", "Event source", and "Resource type". The table lists numerous entries for IAM API calls like "LookupEvents", "GetPolicy", and "GetPolicyVersion" made by "resource-explorer-2" and "root" users. At the bottom, it says "0 / 5 events selected".

2.1. Creación de un Trail Personalizado

Debido a que el historial por defecto solo guarda 90 días, se procedió a crear un **Trail nuevo** para almacenar los registros de forma permanente en un bucket de S3:

- **Nombre del Trail:** Definimos un nombre único para identificar el rastreo.
- **Almacenamiento:** Configuramos un nuevo bucket de S3 para depositar los archivos de log.
- **Eventos de gestión:** Aseguramos que se registraran todas las operaciones de lectura y escritura de la API.

Step 1
 Choose trail attributes
 Step 2
 Choose log events
Step 3
 Review and create

Choose trail attributes

General details
A trail created in the console is a multi-region trail. [Learn more](#)

Trail name
Enter a display name for your trail.

3-128 characters. Only letters, numbers, periods, underscores, and dashes are allowed.

Enable for all accounts in my organization
To review accounts in your organization, open AWS Organizations. [See all accounts](#)

Storage location [Info](#)
 Create new S3 bucket
Create a bucket to store logs for the trail.
 Use existing S3 bucket
Choose an existing bucket to store logs for this trail.

Trail log bucket and folder
Enter a new S3 bucket name and folder (prefix) to store your logs. Bucket names must be globally unique.

Logs will be stored in aws-cloudtrail-logs-654654478122-8412331d/AWSLogs/654654478122

Log file SSE-KMS encryption [Info](#)
 Enabled

Customer managed AWS KMS key
 New
 Existing

AWS KMS alias

▼ Additional settings

Log file validation [Info](#)
 Enabled

SNS notification delivery [Info](#)
 Enabled

CloudWatch Logs - optional
Configure CloudWatch Logs to monitor your trail logs and notify you when specific activity occurs. Standard CloudWatch and CloudWatch Logs charges apply. [Learn more](#)

CloudWatch Logs [Info](#)
 Enabled

► Policy document

Tags - optional [Info](#)
You can add one or more tags to help you manage and organize your resources, including trails.

3. Monitoreo y Alarmas con AWS CloudWatch

Una vez asegurada la auditoría, nos enfocamos en el monitoreo preventivo de la instancia EC2.

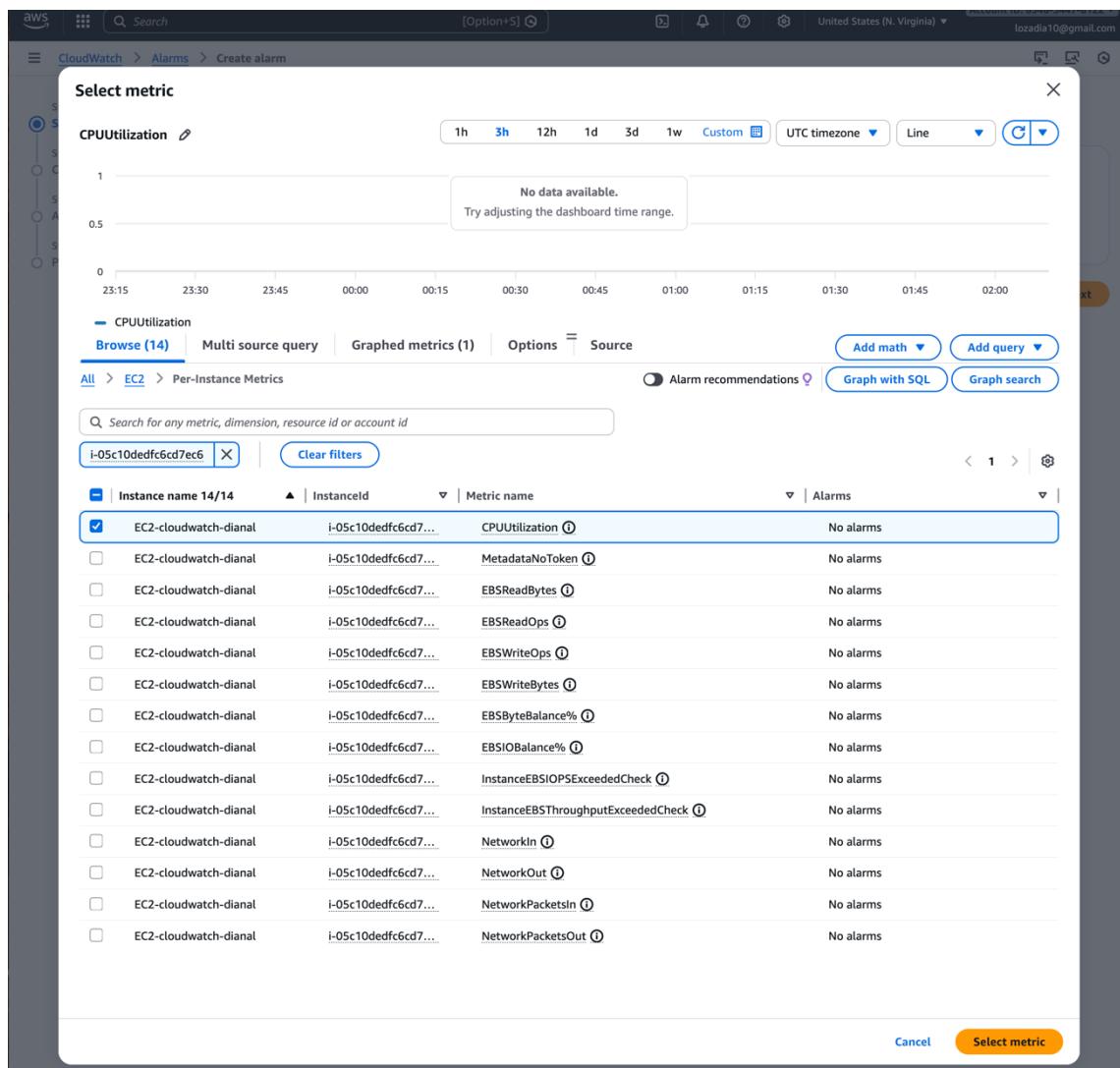
The screenshot shows the AWS EC2 Instances page for instance `i-05c10dedfc6cd7ec6`. The left sidebar navigation includes:

- EC2** (selected)
- Dashboard
- AWS Global View
- Events
- Instances** (selected)
 - Instances
 - Instance Types
 - Launch Templates
 - Spot Requests
 - Savings Plans
 - Reserved Instances
 - Dedicated Hosts
 - Capacity Reservations
 - Capacity Manager [New](#)
- Images**
 - AMIs
 - AMI Catalog
- Elastic Block Store**
 - Volumes
 - Snapshots
 - Lifecycle Manager
- Network & Security**
 - Security Groups
 - Elastic IPs
 - Placement Groups
 - Key Pairs
 - Network Interfaces
- Load Balancing**
 - Load Balancers
 - Target Groups
 - Trust Stores
- Auto Scaling**
 - Auto Scaling Groups
- Settings

3.1. Creación de Alarma de CPU

Configuré una alarma para que me notificara si la instancia superaba un umbral de uso de CPU:

- Creé una alarma de CloudWatch basada en la métrica `CPUUtilization`. Establecí un umbral crítico para recibir una notificación automática si el servidor experimenta una carga de trabajo inusual.



- **Notificación (SNS):** Creé un nuevo tema de SNS para recibir alertas por correo electrónico cuando la alarma cambiara al estado ALARM.

CloudWatch > Alarms > Create alarm

Configure actions

Step 1: Specify metric and conditions
Step 2: Configure actions (selected)
Step 3: Add alarm details
Step 4: Preview and create

Notification

Alarm state trigger
Define the alarm state that will trigger this action.

In alarm
The metric or expression is outside of the defined threshold.

OK
The metric or expression is within the defined threshold.

Insufficient data
The alarm has just started or not enough data is available.

Send a notification to the following SNS topic
Define the SNS (Simple Notification Service) topic that will receive the notification.

Create new topic
 Select an existing SNS topic
 Use topic ARN to notify other accounts

Create a new topic...
The topic name must be unique.
Dianal

SNS topic names can contain only alphanumeric characters, hyphens (-) and underscores (_).

Email endpoints that will receive the notification...
Add a comma-separated list of email addresses. Each address will be added as a subscription to the topic above.

lozadia10@gmail.com

user1@example.com, user2@example.com

Create topic
Add notification

Lambda action
Add Lambda action

Auto Scaling action
Add Auto Scaling action

EC2 action
Add EC2 action

Systems Manager action [Learn more](#)
This action will create an Incident or OpsItem in Systems Manager when the alarm is **In alarm** state.

Add Systems Manager action



Simple Notification Service

Subscription removed!

Your subscription, arn:aws:sns:us-east-1:654654478122:Dianal:5478f00c-bbf7-425a-a8a2-ae37f3668a9c, has been deleted.

CloudWatch > Alarms > Create alarm

Alarm recommendations available
Turn on Recommendations to pre-populate the wizard with the recommended alarms.

Step 1 **Specify metric and conditions** Step 2 Configure actions Step 3 Add alarm details Step 4 Preview and create

Specify metric and conditions

Metric

Graph
This alarm will trigger when the blue line goes above the red line for 1 datapoints within 5 minutes.

Namespace
AWS/EC2

Metric name
CPUUtilization

InstanceId
i-05c10dedfc6cd7ec6

Instance name
EC2-cloudwatch-dianal

Statistic
 Average

Period
5 minutes

Conditions

Threshold type

Static
Use a value as a threshold

Anomaly detection
Use a band as a threshold

Whenever CPUUtilization is...
Define the alarm condition.

Greater
> threshold

Greater/Equal
>= threshold

Lower/Equal
<= threshold

Lower
< threshold

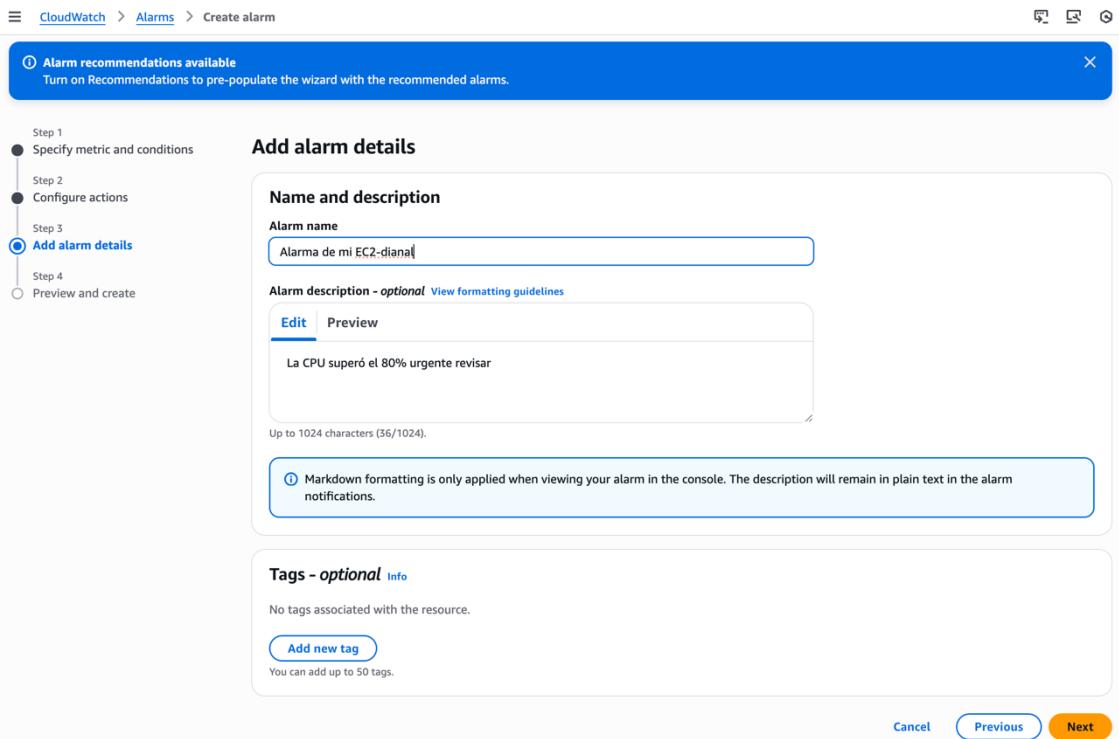
than...
Define the threshold value.

80

Must be a number.

► Additional configuration

[Cancel](#) [Next](#)



4. Prueba de Estrés y Validación

Para verificar que mi configuración funcionaba, procedí a estresar el servidor manualmente.

- Instalación de herramientas:** Me conecté a la EC2 e instalé la utilidad stress.
- Ejecución del test:** Ejecuté el comando `stress --cpu $(nproc) --timeout 300 &` para elevar artificialmente el consumo de recursos.

```
ec2-user@ip-172-31-9-122:~$ -bash: stress: command not found
sudo dnf install stress-ng -y
Last metadata expiration check: 0:00:37 ago on Sat Feb 21 02:35:50 2026.
Dependencies resolved.
=====
Package          Architecture Version       Repository      Size
=====
Installing:
stress-ng        x86_64      0.15.05-1.amzn2023   amazonlinux   2.3 M
Installing dependencies:
Judy             x86_64      1.0.5-25.amzn2023.0.3  amazonlinux   153 k
libbsd           x86_64      0.10.0-7.amzn2023.0.2  amazonlinux   109 k
lksctp-tools     x86_64      1.0.18-9.amzn2023.0.3  amazonlinux   92 k
=====
Transaction Summary
=====
Install 4 Packages

Total download size: 2.7 M
Installed size: 9.7 M
Downloading Packages:
(1/4): libbsd-0.10.0-7.amzn2023.0.2.x86_64.rpm          2.5 MB/s | 109 kB   00:00
(2/4): lksctp-tools-1.0.18-9.amzn2023.0.3.x86_64.rpm      1.9 MB/s |  92 kB   00:00
(3/4): Judy-1.0.5-25.amzn2023.0.3.x86_64.rpm            2.7 MB/s | 153 kB   00:00
```

3. Resultados: Pude observar en el panel de CloudWatch cómo la métrica de CPU comenzó a subir hasta cruzar el umbral definido, activando así la alerta.

```
%Cpu(s):100.0 us, 0.0 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 916.8 total, 385.2 free, 172.9 used, 358.7 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 605.6 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
26287 ec2-user 20 0 61200 5788 3872 R 100.0 0.6 2:52.23 stress-ng-cpu
26286 ec2-user 20 0 61200 5788 3868 R 99.7 0.6 2:52.23 stress-ng-cpu
  1 root 20 0 172420 17340 10796 S 0.0 1.8 0:01.01 systemd
  2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthread
  3 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_gp
  4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_par_gp
  5 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 slub_flushwq
  6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 netns
  8 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:0H-events_highpri
 10 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm_percpu_wq
 11 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_tasks_kthread
 12 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_tasks_rude_kthread
 13 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_tasks_trace_kthread
 14 root 20 0 0 0 0 S 0.0 0.0 0:00.09 ksoftirqd/0
 15 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_premempt
 16 root rt 0 0 0 0 S 0.0 0.0 0:00.01 migration/0
 18 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/0
 19 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/1
 20 root rt 0 0 0 0 S 0.0 0.0 0:00.03 migration/1
 21 root 20 0 0 0 0 S 0.0 0.0 0:00.09 ksoftirqd/1
 23 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/1:0H-events_highpri
 26 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kdevtmpfs
 27 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 inet_frag_wq
 28 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kauditd
 29 root 20 0 0 0 0 S 0.0 0.0 0:00.00 khungtaskd
 30 root 20 0 0 0 0 S 0.0 0.0 0:00.00 oom_reaper
 32 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 writeback
 33 root 20 0 0 0 0 S 0.0 0.0 0:00.07 kcompactd0
 35 root 39 19 0 0 0 S 0.0 0.0 0:00.00 khugepaged
 36 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 cryptd
 37 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kintegrityd
 38 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kblockd
Connection to ec2-3-235-50-141.compute-1.amazonaws.com closed by remote host.punt_bio
Connection to ec2-3-235-50-141.compute-1.amazonaws.com closed.
diana@iMac-de-Dimar Downloads %
```

| Name | State | Actions | Last state update (UTC) |
|--|----------|-------------------------|-------------------------|
| mi alarma de mi EC2 - Jose Manuel | In alarm | Actions enabled | 2026-02-21 02:57:35 |
| AlarmaDiana | In alarm | Actions enabled Warning | 2026-02-21 02:57:07 |
| Alarma para monitorar el uso de la CPU | OK | Actions enabled | 2026-02-21 02:56:47 |
| El uso de la CPU supero el 80% (Ojo p3) - Andres Altamar | OK | Actions enabled | 2026-02-21 02:46:25 |
| Mi Alarma de mi EC2 - Johan | OK | Actions enabled | 2026-02-21 02:43:17 |
| Alarma ec2 AndresOrtiz | OK | Actions enabled | 2026-02-21 02:35:20 |

5. Conclusiones

- **CloudTrail** es mi herramienta de seguridad principal, ya que me permite reconstruir eventos en caso de incidentes o cambios no autorizados.
- Con **CloudWatch**, aprendí que no necesito estar vigilando la consola manualmente; las alarmas me permiten actuar de forma proactiva ante fallos de rendimiento.
- La combinación de ambos servicios me otorga un control total sobre la salud y la seguridad de mi infraestructura en la nube.