LABORATORIO DE MONITOREO Y AUDITORÍA EN AWS

Servicios utilizados: AWS CloudTrail y AWS CloudWatch

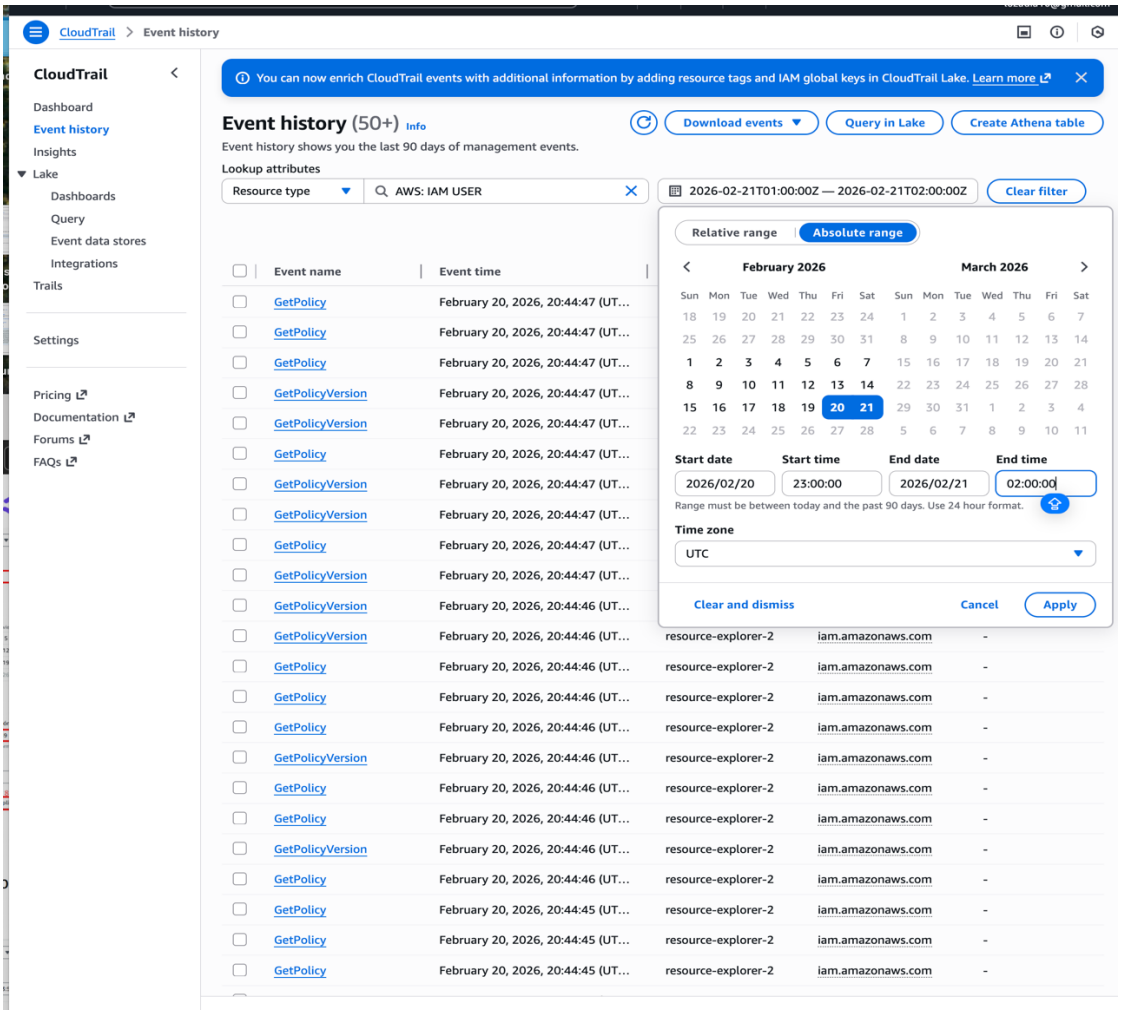
---

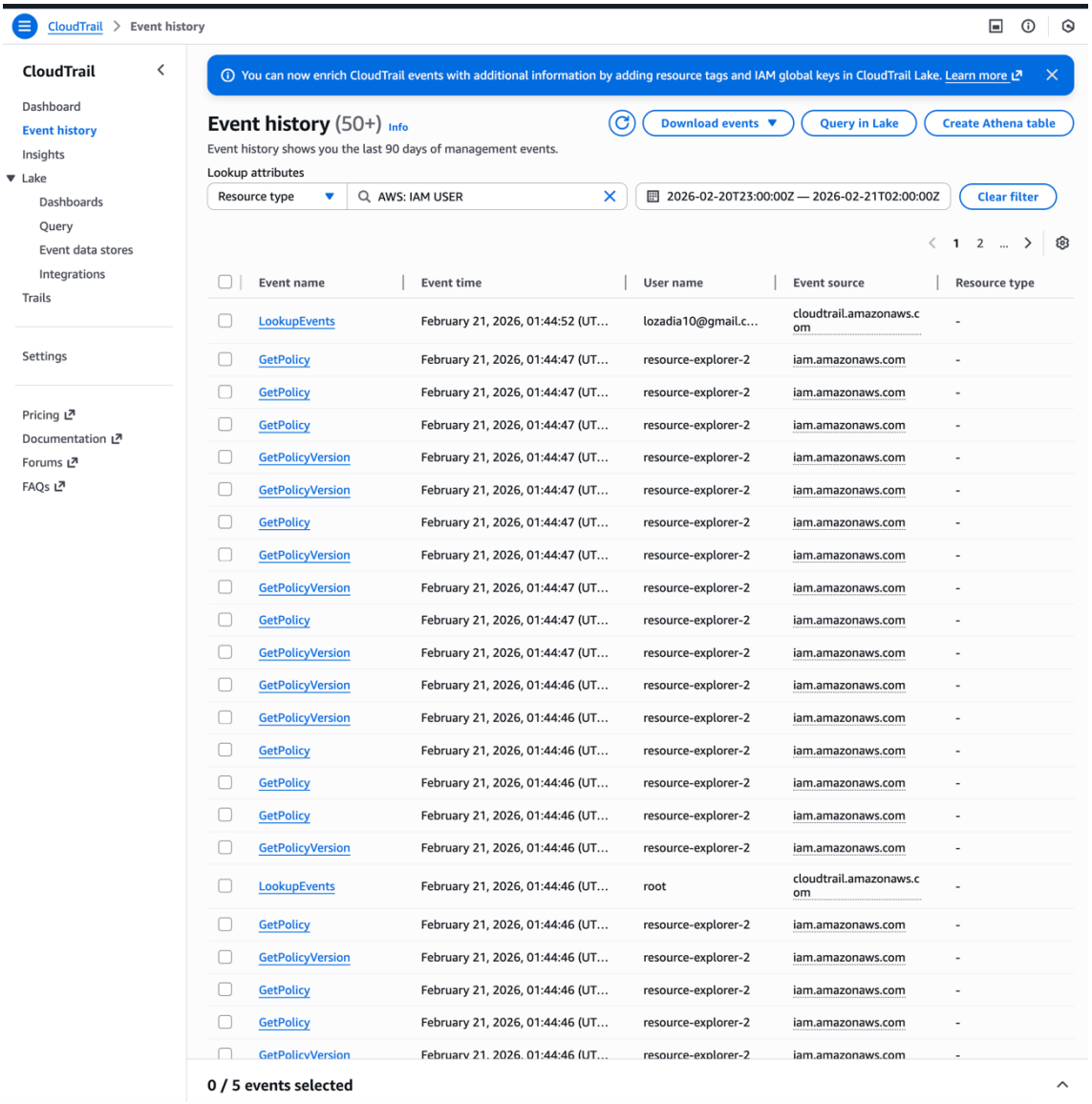
1. Introducción

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).

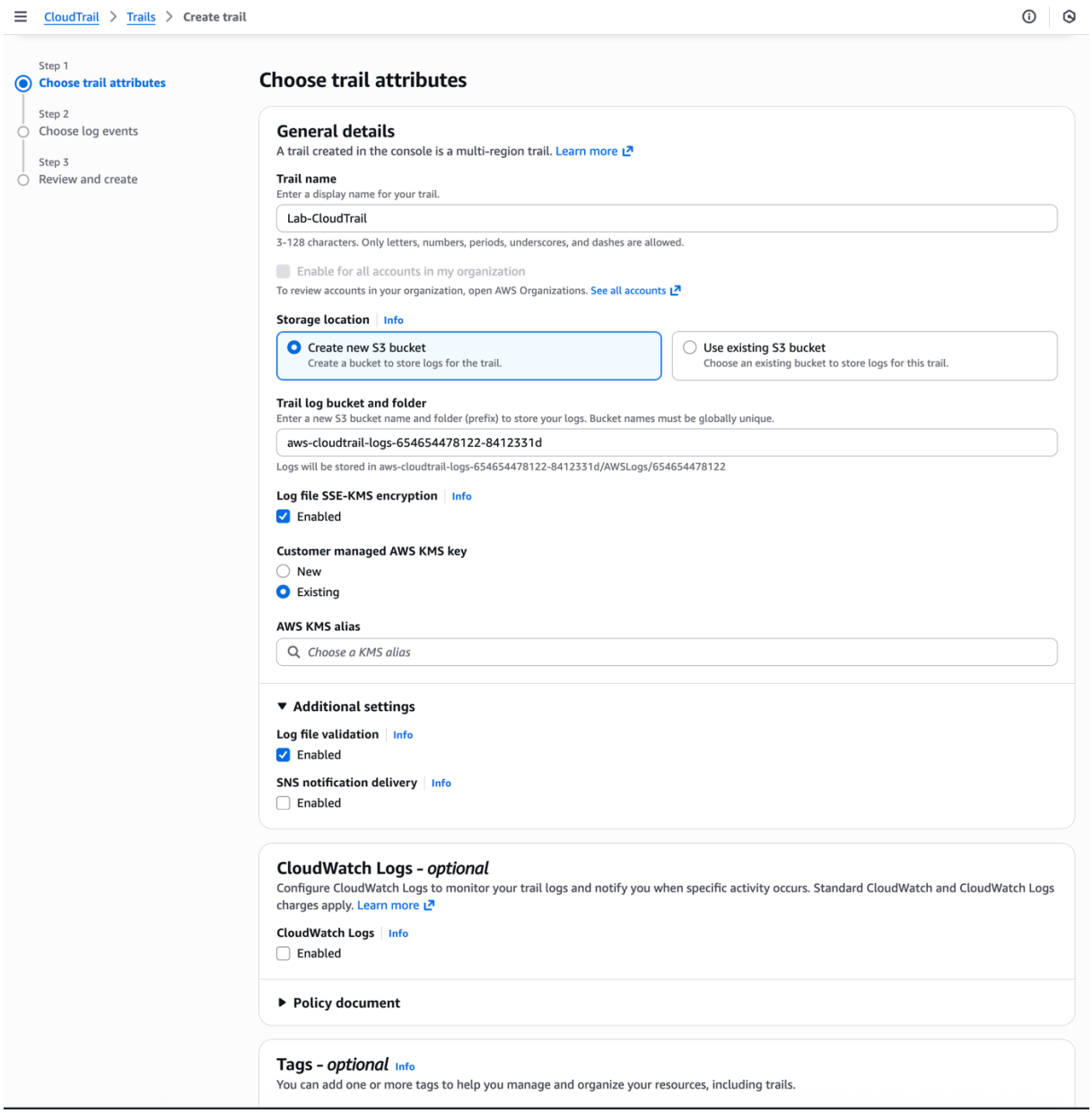




2.1. Creación de un Trail Personalizado

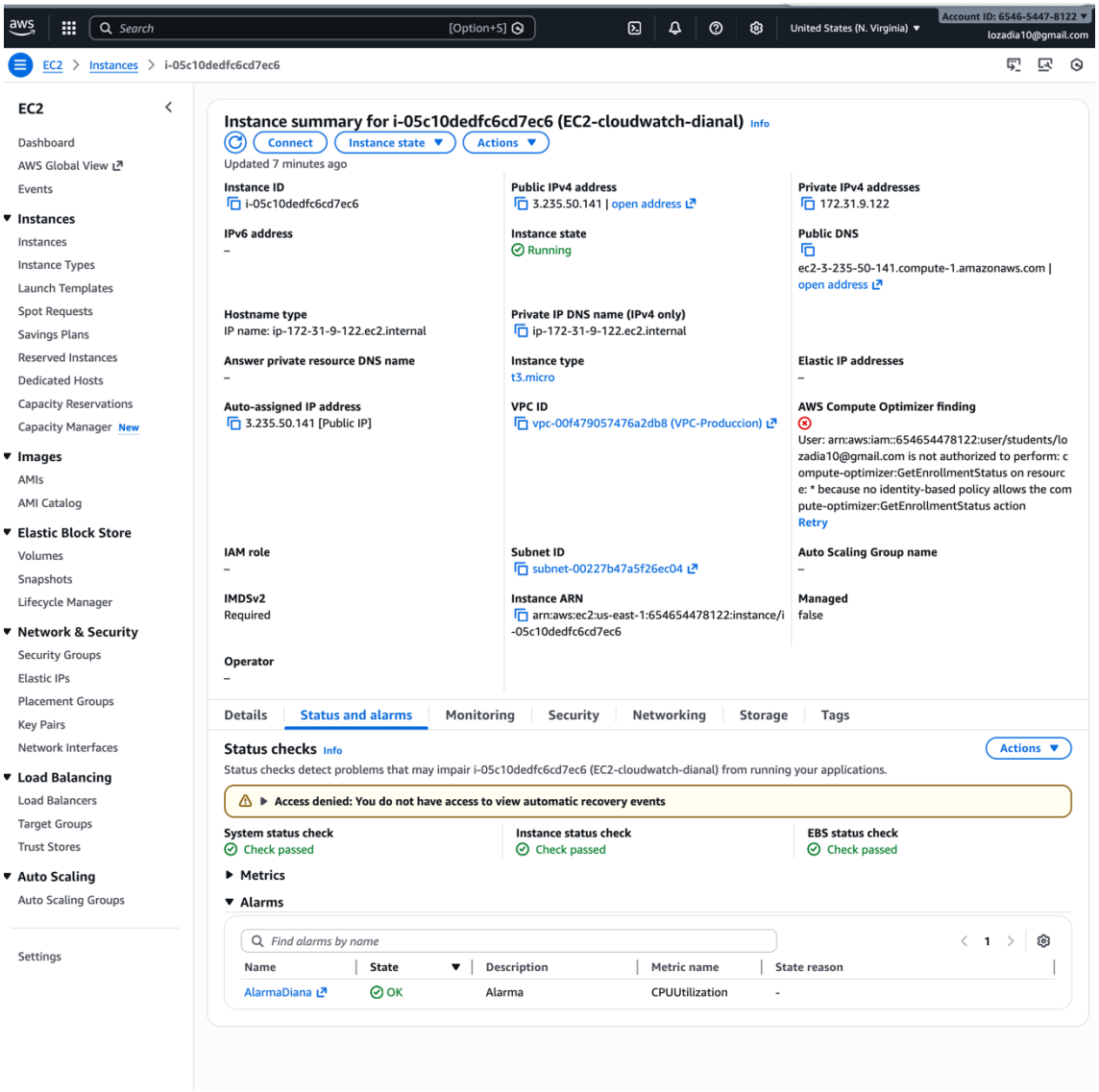
Debido a que el historial por defecto solo guarda 90 días, se procedió a crear un **Trail nuevo** para almacenar los registros de forma permanente en un bucket de S3:

- **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.



3. Monitoreo y Alarmas con AWS CloudWatch

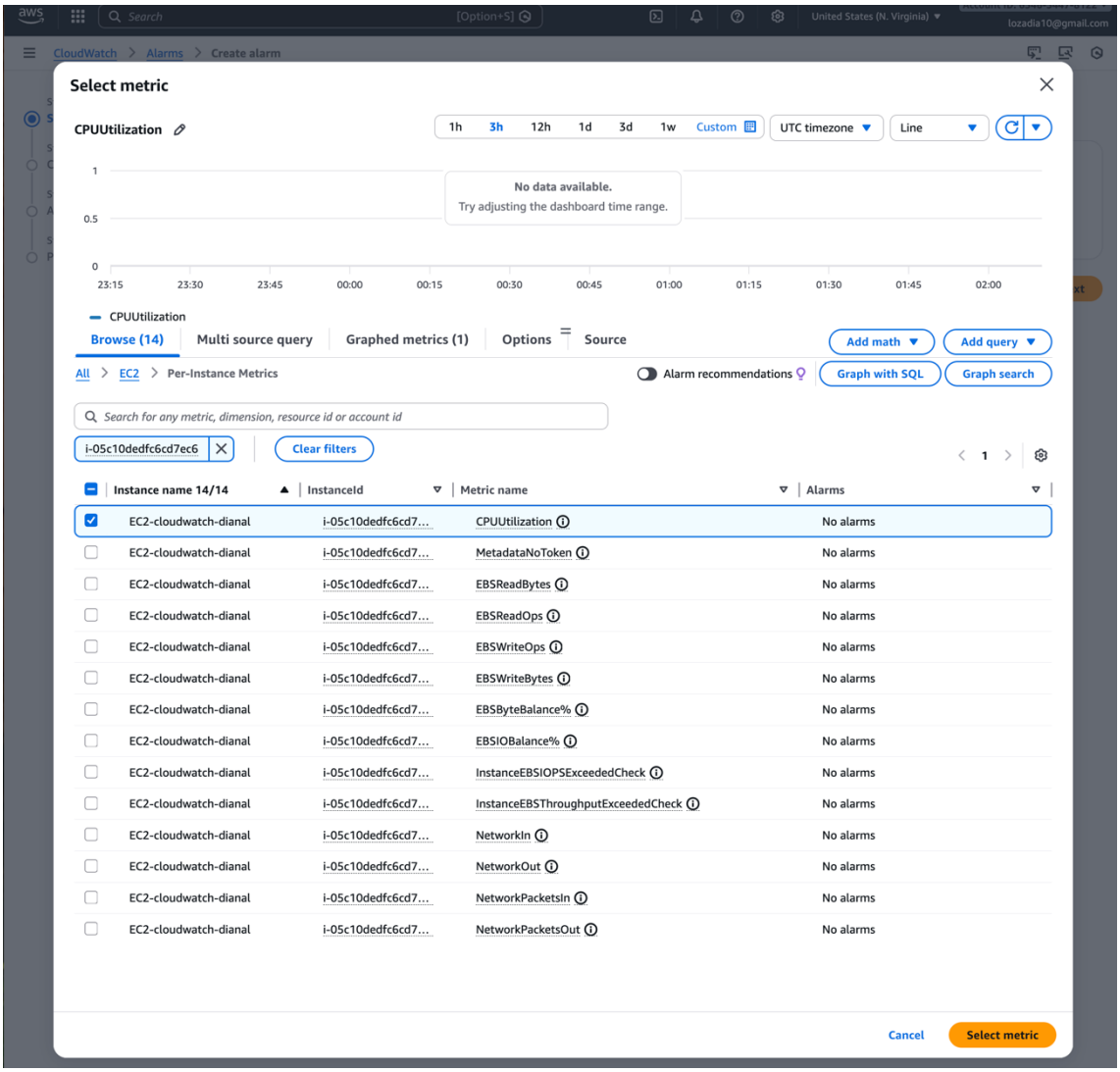
Una vez asegurada la auditoría, nos enfocamos en el monitoreo preventivo de la instancia EC2.



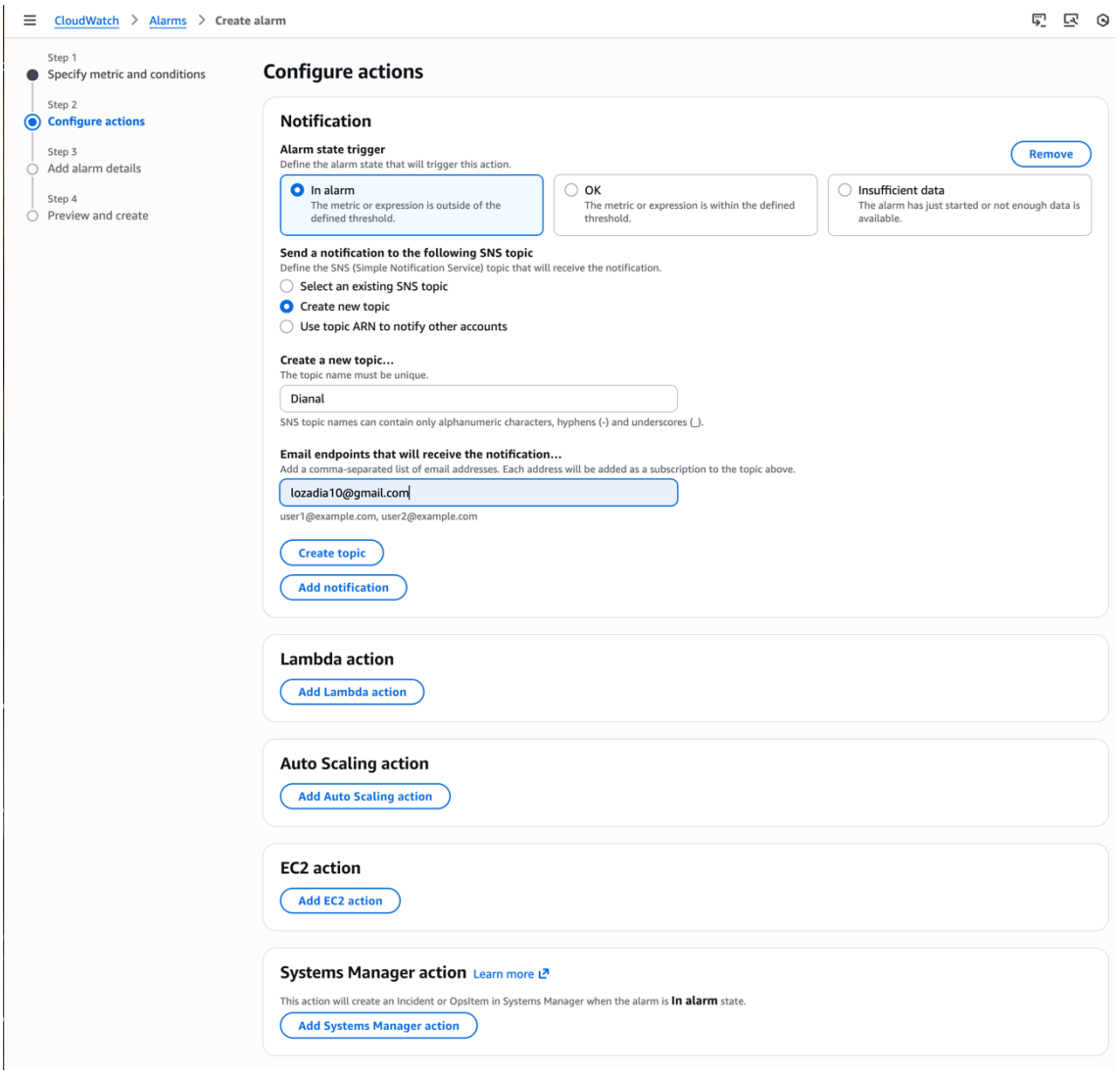
3.1. Creación de Alarma de CPU

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.

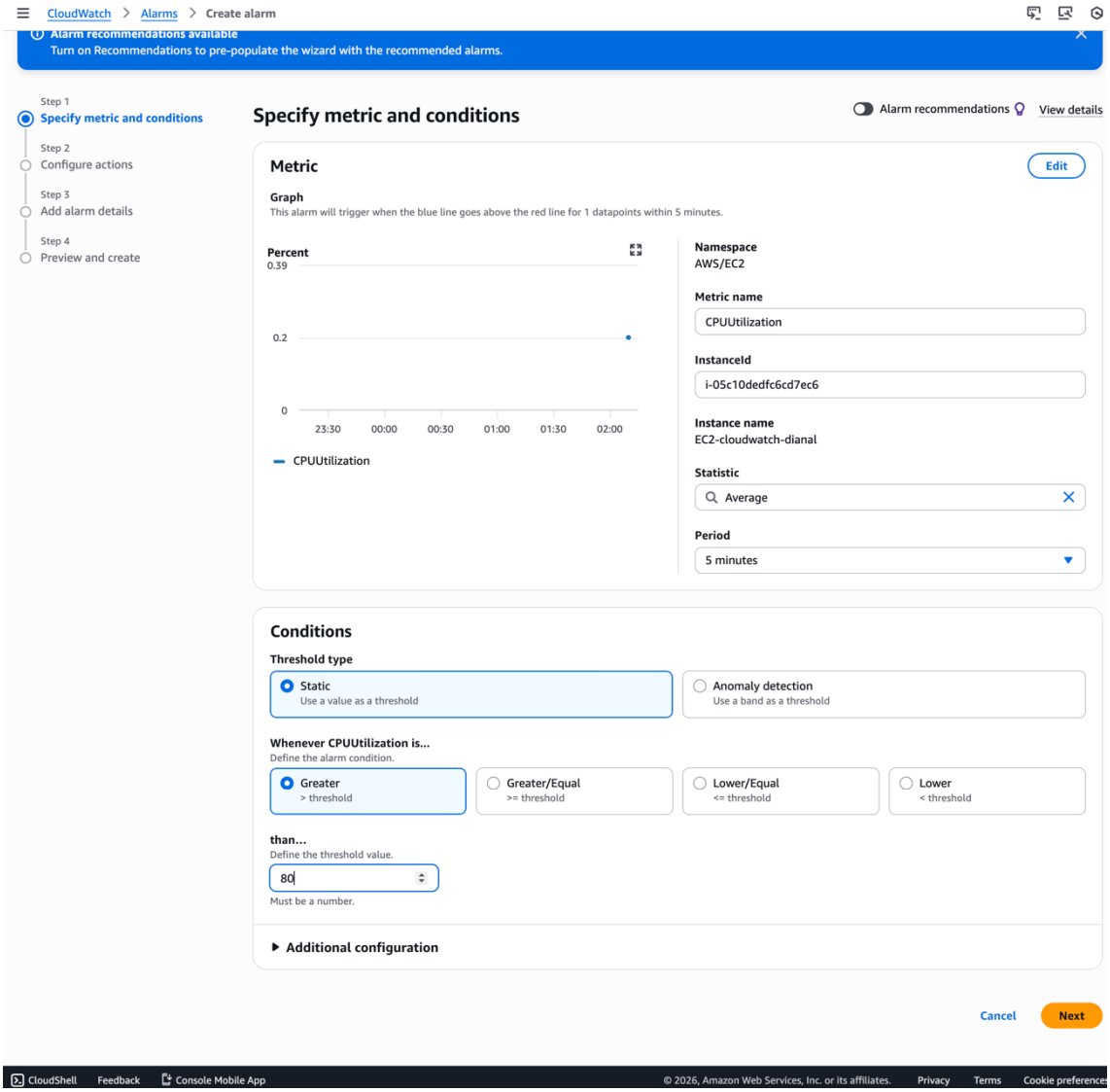


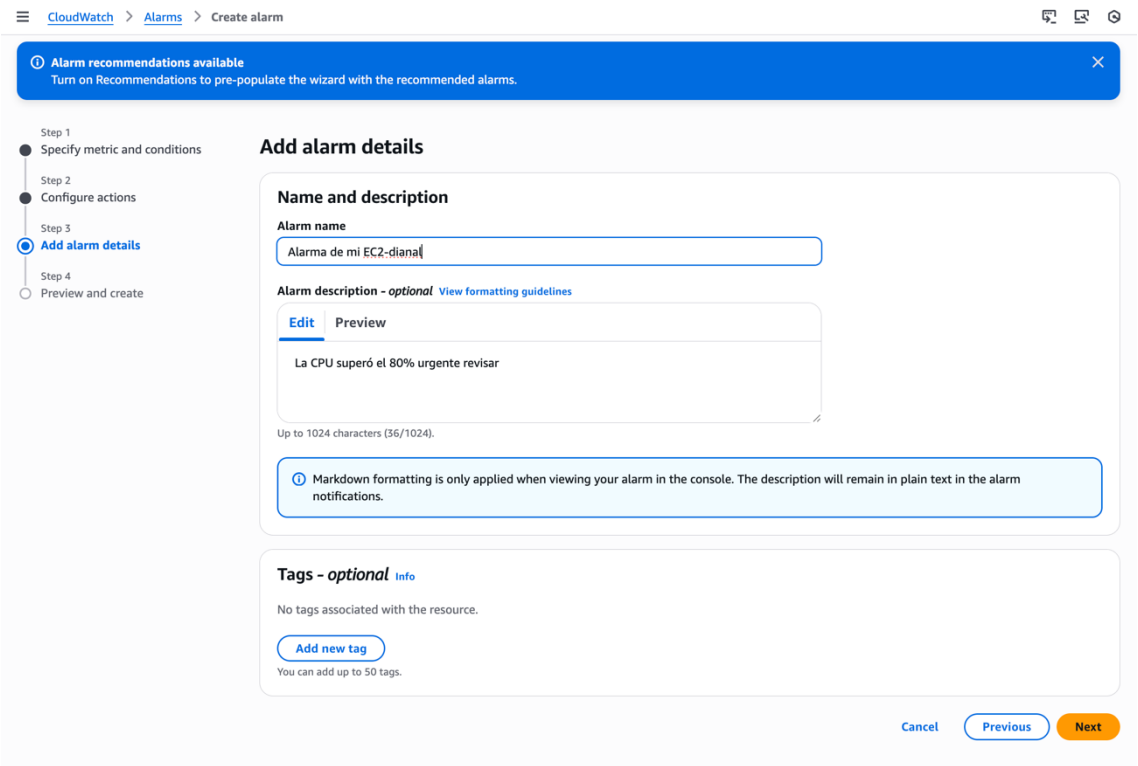
aws

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
Alarm recommendations
View details
Metric
Edit
Graph
This alarm will trigger when the blue line goes above the red line for 1 datapoints within 5 minutes.
Percent
0.39
0.2
0
23:30
00:00
00:30
01:00
01:30
02:00
CPUUtilization
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
CloudShell
Feedback
Console Mobile App
© 2026, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



4. Prueba de Estrés y Validación

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

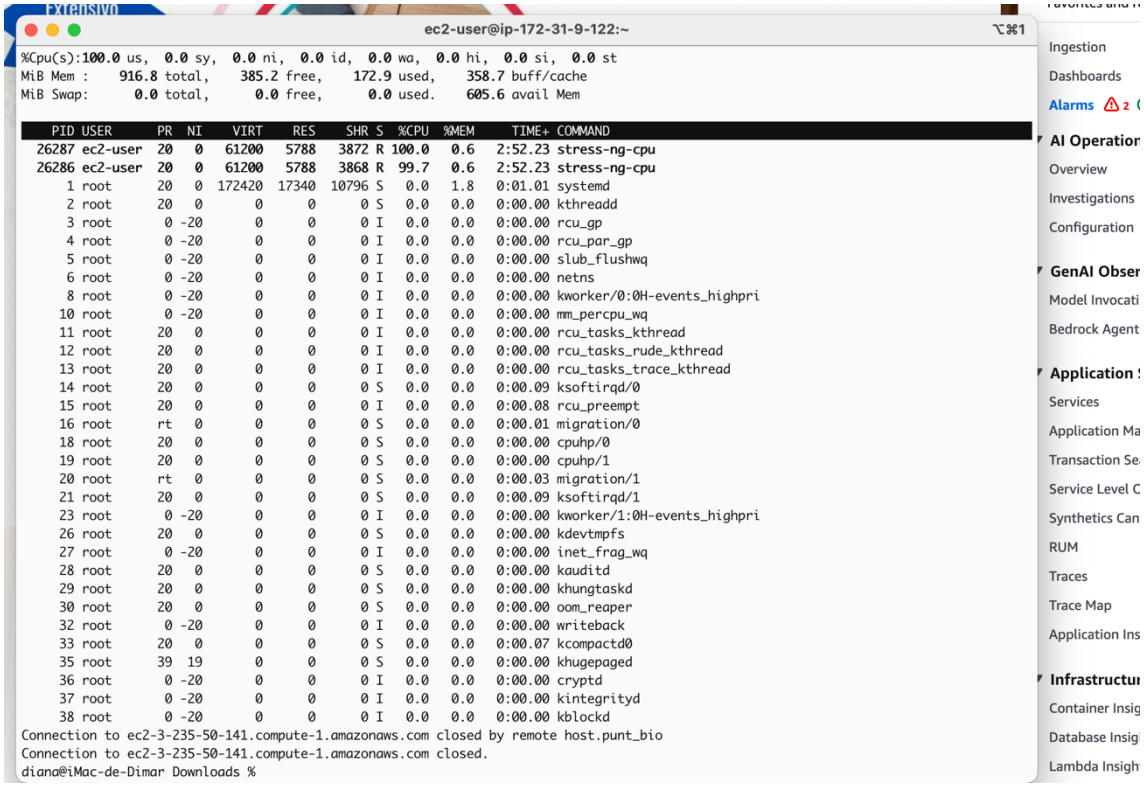
1. **Instalación de herramientas:** Me conecté a la EC2 e instalé la utilidad stress.
2. **Ejecución del test:** Ejecuté el comando stress --cpu $(nproc) --timeout 300 & para elevar artificialmente el consumo de recursos.

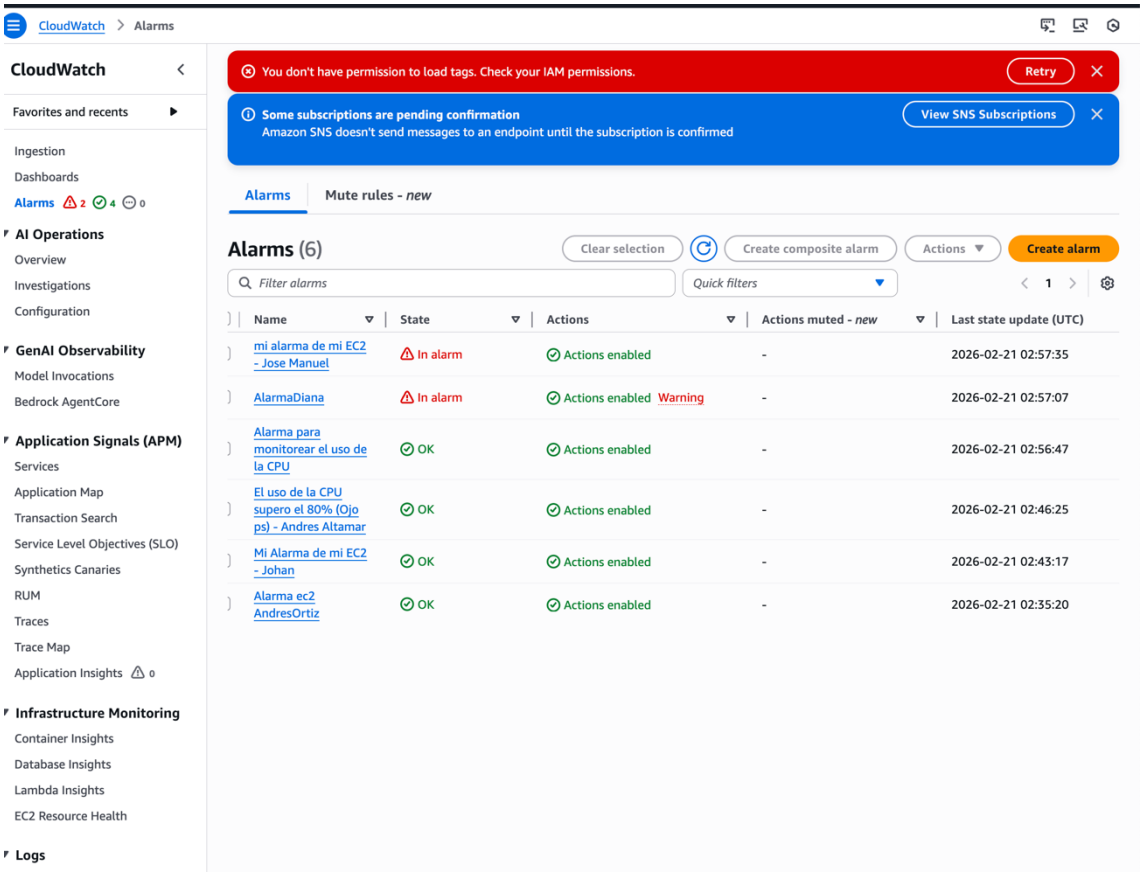
```
ec2-user@ip-172-31-9-122:~                                                    ⌥⌘1
[ec2-user@ip-172-31-9-122 ~]$ -bash: stress: command not found
sudo dnf install stress-ng -y
Last metadata expiration check: 0:00:37 ago on Sat Feb 21 02:35:50 2026.
Dependencies resolved.
================================================================================
 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.





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.