

Datos Generales:

Nombre: Tomás Alfredo Villaseca Constantinescu

País: Chile

Fecha: 24/09/2023

Contacto: tomas.villaseca.c@gmail.com

Después de completar este laboratorio, podrá realizar lo siguiente:

- Crear una notificación de Amazon SNS
- Configurar una alarma de CloudWatch
- Realizar una prueba de estrés a una instancia de EC2
- Confirmar que se envió un correo electrónico de Amazon SNS
- Crear un panel de CloudWatch

Entorno del laboratorio → instancia EC2 pre-configurada Stress Test

- Stress Test → Rol IAM adjuntado que permite conectarse vía Session Manager.

El **Logging** y **Monitoring** son técnicas aplicadas para lograr un objetivo común. Trabajan juntas para ayudar a garantizar que siempre se cumplan las baselines de rendimiento y las directrices de seguridad de un sistema.

Logging = Proceso de registrar y almacenar eventos de datos como log files.

- Logs = contienen detalles de bajo nivel que pueden darle visibilidad sobre cómo funciona su aplicación o sistema en determinadas circunstancias.
- Logs ayudan a los administradores de seguridad a identificar red flags que se pasan por alto fácilmente en su sistema.

Monitoring = Proceso de analizar y recopilar datos para garantizar un rendimiento óptimo.

- Ayuda a detectar el acceso no autorizado y ayuda a alinear el uso de sus servicios con la seguridad de la organización.

En este laboratorio, creará una alarma de Amazon CloudWatch que se iniciará cuando la instancia de EC2 supere un umbral específico de utilización de CPU.

- Creará una suscripción mediante Amazon SNS que le envíe un email si se activa esta alarma.
- Iniciará sesión en la instancia EC2 y ejecutará un comando de prueba de estrés que provoque que la utilización de la CPU de la instancia EC2 alcance el 100 %.

Esta prueba simula que un actor malicioso se hace con el control de la instancia EC2 y dispara la CPU.

- Los spikes de CPU tienen varias causas posibles, una de las cuales es el malware.

Amazon CloudWatch = Servicio web que permite monitorizar y administrar varias métricas y configurar acciones de alarma basadas en los datos de dichas métricas.



Amazon
CloudWatch

- Las métricas → representan puntos de datos de sus recursos.
- Los servicios de AWS envían métricas a AWS CloudWatch.
- Crea automáticamente gráficos que muestran cómo ha cambiado el rendimiento a lo largo del tiempo.
- Alarma de Amazon CloudWatch → Realiza acciones automáticamente si el valor de su métrica ha superado o quedado por debajo de un umbral predefinido.
- Configurar alarma → Especifique si desea recibir una notificación cuando se active la alarma (integrado con Amazon SNS).

Amazon Single Notification Service (Amazon SNS) = Servicio de publicación/suscripción.



Amazon SNS

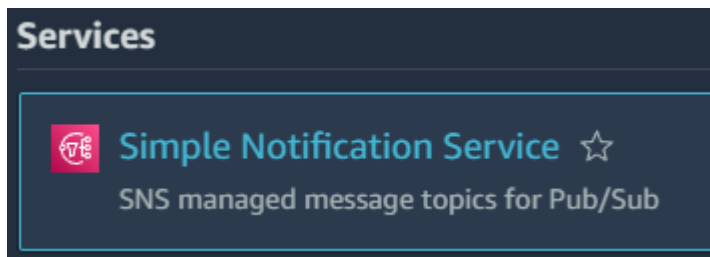
- Se utiliza para que los editores publiquen mensajes a los suscriptores.
- SNS Topic = Canal para la entrega de mensajes.
- Los suscriptores pueden ser servidores web, direcciones de correo electrónico, funciones de AWS Lambda o varias otras opciones.
- Puede utilizarse para enviar notificaciones a los usuarios finales → SMS, correo electrónico, etc.

Tarea 1: Configurar Amazon SNS

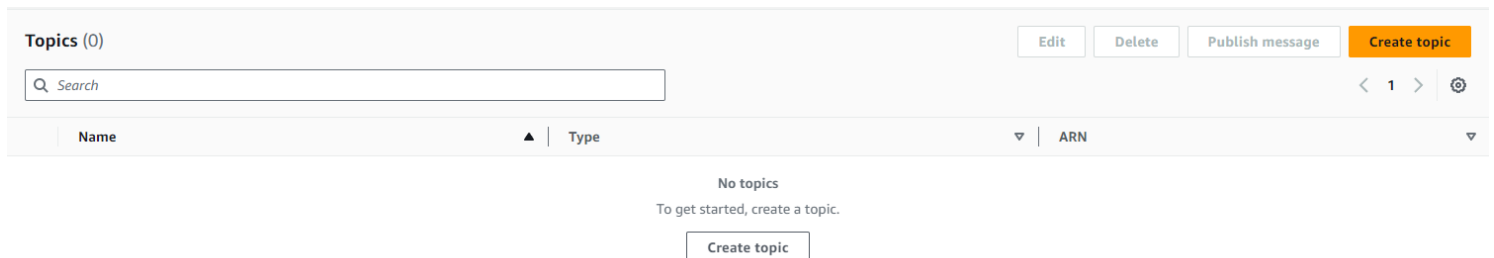
En esta tarea, creará un SNS topic y luego se suscribe a él con una dirección de correo electrónico.

Amazon SNS es un servicio de mensajería totalmente administrado para la comunicación tanto de aplicación a aplicación (A2A) como de aplicación a persona (A2P).

Paso 1: AWS Management Console → Search → Amazon SNS



Paso 2: Amazon SNS → Panel de navegación → Topics → Create topic



Paso 3: Create topic → Details

- Type = Standard
- Name = MyCwAlarm

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
- High throughput, up to 300 publishes/second
- Subscription protocols: SQS

☒ Standard

- Best-effort message ordering
- At-least once message delivery
- Highest throughput in publishes/second
- Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

Name

MyCwAlarm

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

Paso 4: Topics → MyCwAlarm → Subscriptions → Create subscription

MyCwAlarm

EditDeletePublish message

Details

Name

MyCwAlarm

ARN

arn:aws:sns:us-west-2:242475850820:MyCwAlarm

Type

Standard

Display name

-

Topic owner

242475850820

Subscriptions

Access policyData protection policyDelivery policy (HTTP/S)Delivery status loggingEncryptionTagsIntegrations

Subscriptions (0)

EditDeleteRequest confirmationConfirm subscriptionCreate subscription

Search

< 1 > ⚙

ID

Endpoint

Status

Protocol

No subscriptions found

You don't have any subscriptions to this topic.

Paso 5: Create subscriptions → Details

- Topic ARN → Default
- Protocol → Email
- Endpoint → Ingresar correo electrónico

Create subscription

Details

Topic ARN

arn:aws:sns:us-west-2:242475850820:MyCwAlarm

×

Protocol

The type of endpoint to subscribe

Email

Endpoint

An email address that can receive notifications from Amazon SNS.

tomas.villaseca.c@gmail.com

Paso 6: Status → Pending Confirmation

- Recibirá un correo electrónico → AWS Notification – Subscription Confirmation
- Abrir el correo y confirmar suscripción.
- Status → Confirmed

Subscription: 8d330df9-2f2e-4143-b026-6ca6bd18f975

Details

ARN

arn:aws:sns:us-west-2:242475850820:MyCwAlarm:8d330df9-2f2e-4143-b026-6ca6bd18f975

Endpoint

tomas.villaseca.c@gmail.com

Topic

[MyCwAlarm](#)

Subscription Principal

arn:aws:iam::242475850820:role/voclabs

Status

⌚ Pending confirmation

Protocol

EMAIL

AWS Notification - Subscription Confirmation Recibidos x

AWS Notifications <no-reply@sns.amazonaws.com>

para mí ▼

🌐 inglés ▼ > español ▼ [Traducir mensaje](#)

You have chosen to subscribe to the topic:

arn:aws:sns:us-west-2:242475850820:MyCwAlarm

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):

[Confirm subscription](#)

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to [sns-opt-out](#)



Simple Notification Service

Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:

arn:aws:sns:us-west-2:242475850820:MyCwAlarm:8d330df9-2f2e-4143-b026-6ca6bd18f975

If it was not your intention to subscribe, [click here to unsubscribe](#).

En esta tarea, se creó un SNS topic y se creó una suscripción para el topic utilizando una dirección de correo electrónico.

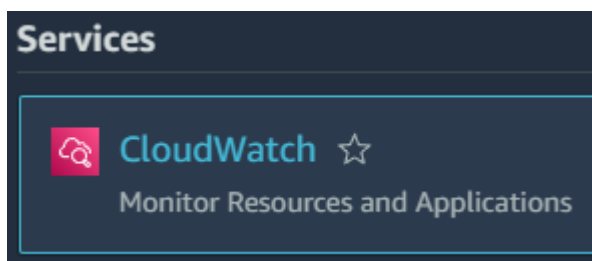
- Este topic puede enviar alertas a la dirección de correo electrónico asociada a la suscripción a Amazon SNS.

Tarea 2: Crear una alarma de CloudWatch

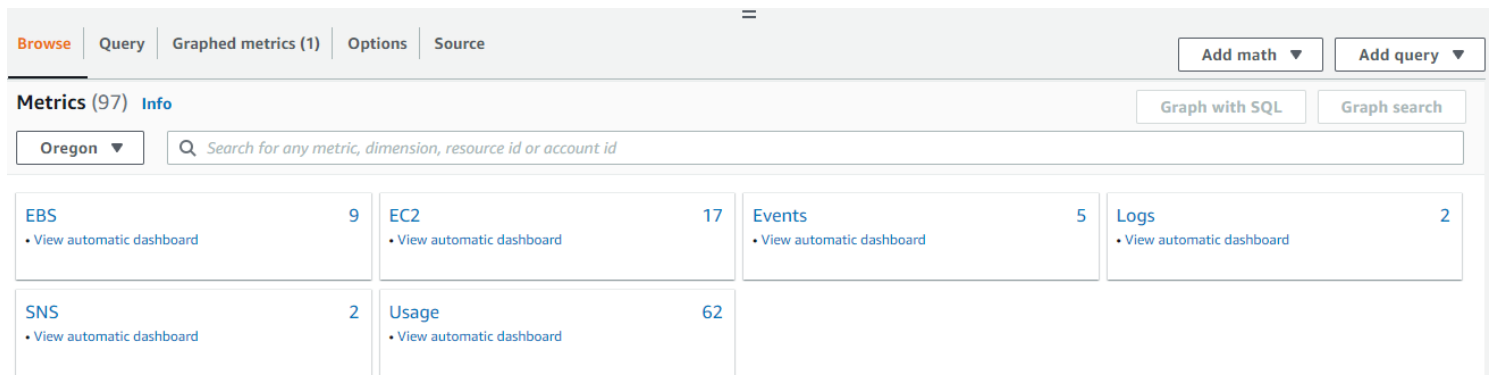
En esta tarea, verá algunas métricas y logs almacenados en CloudWatch.

Crearé una alarma de CloudWatch para iniciar y enviar un correo electrónico a su SNS topic si la instancia EC2 Stress Test aumenta a más del 60% de utilización de la CPU.

Paso 1: AWS Management Console → Search → CloudWatch



Paso 2: CloudWatch → Panel de navegación → Metrics → All metrics



Paso 3: Metrics → EC2 → Per-instance Metrics

- Seleccionar la casilla “CPUUtilization” para la EC2 Stress Test.

| <input checked="" type="checkbox"/> | Instance name 1/1 ▲ | InstanceId ▼ | Metric name |
|-------------------------------------|---------------------|---------------------|----------------|
| <input checked="" type="checkbox"/> | Stress Test | i-024490b8c028d7095 | CPUUtilization |

Paso 4: Panel de navegación → Alarms → All alarms

Alarms (0) ☐ Hide Auto Scaling alarms

| Name | State | Last state update | Conditions | Actions |
|---|-------|-------------------|------------|---------|
| No alarms | | | | |
| No alarms to display | | | | |
| Read more about Alarms | | | | |
| <input type="button" value="Create alarm"/> | | | | |

Paso 5: Alarms → Create alarm

- Select metric → EC2 → Per-Instance Metrics
- Seleccionar la casilla “CPUUtilization” para la EC2 Stress Test

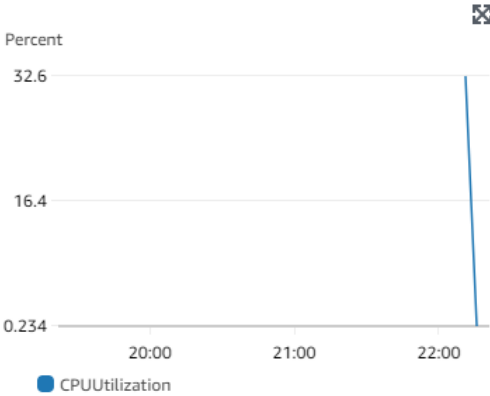
| | | | |
|-------------------------------------|-------------|---------------------|----------------|
| <input checked="" type="checkbox"/> | Stress Test | i-024490b8c028d7095 | CPUUtilization |
|-------------------------------------|-------------|---------------------|----------------|

Paso 6: Select metric → Specify metric and conditions → Metric

- Metric name = CPUUtilization
- Instaceld = Default
- Statistic = Average
- Period = 1 minute

Metric

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



Namespace
AWS/EC2

Metric name

Instaceld

Instance name
Stress Test

Statistic

Period

Paso 7: Select metric → Specify metric and conditions → Conditions

- Threshold type = Static
- Whenever CPUUtilization is... (Define the alarm condition) → Greater (> threshold)
- than... (Define the threshold value) → 60

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.

60

Must be a number

Paso 8: Select metric → Specify metric and conditions → Notification

- Alarm state trigger = In alarm
- Select an SNS topic = Select an existing SNS topic
- Send a notification to... → MyCwAlarm

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.

Remove

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

☒ Select an existing SNS topic

☐ Create new topic

☐ Use topic ARN to notify other accounts

Send a notification to...

Q MyCwAlarm X

Only email lists for this account are available.

Email (endpoints)
tomas.villasaca.c@gmail.com - [View in SNS Console](#)

Add notification

Paso 9: Select metric → Specify metric and conditions → Name and description

- Alarm name = LabCPUUtilizationAlarm
- Alarm description = Cloudwatch alarm for Stress Test EC2 instance CPUUtilization

Name and description

Alarm name

LabCPUUtilizationAlarm

Alarm description - optional [View formatting guidelines](#)

Edit

Preview

Cloudwatch alarm for Stress Test EC2 instance CPUUtilization

Up to 1024 characters (60/1024)

Cancel

Previous

Create alarm

Alarms (1/1)

☐ Hide Auto Scaling alarms

Clear selection



Create composite alarm

Search

Any state

Any type

Any actions ...

| <input checked="" type="checkbox"/> | Name | State | Last state update | Conditions | Actions |
|-------------------------------------|--|-------------------|---------------------|--|-----------------|
| <input checked="" type="checkbox"/> | LabCPUUtilizationAlarm | Insufficient data | 2023-09-24 22:26:14 | CPUUtilization > 60 for 1 datapoints within 1 minute | Actions enabled |

En esta tarea visualizó algunas métricas de Amazon EC2 en CloudWatch. También creó una alarma de CloudWatch que inicia un estado de alarma cuando el umbral de utilización de la CPU supera el 60%.

Tarea 3: Probar la alarma de Cloudwatch

En esta tarea, iniciará sesión en la instancia EC2 Stress Test y ejecutará un comando que estresa la carga de la CPU al 100%. Este aumento en la utilización de la CPU activa la alarma de CloudWatch, lo que provoca que Amazon SNS envíe una notificación por correo electrónico a la dirección de correo electrónico asociada con el SNS topic.

Paso 1: AWS Management Console → EC2 → Instances → Stress Test → Connect

- Session Manager → Connect

Session ID: user2741130=Tom__sVillaseca-05db053cc306e5d0f

Instance ID: i-024490b8c028d7095

```
sh-4.2$
```

Paso 2: Incrementar la carga de la CPU de la instancia ingresando el siguiente comando:

```
sudo stress --cpu 10 -v --timeout 400s
```

Session ID: user2741130=Tom__sVillaseca-05db053cc306e5d0f

Instance ID: i-024490b8c028d7095

```
sh-4.2$ sudo stress --cpu 10 -v --timeout 400s
stress: info: [6453] dispatching hogs: 10 cpu, 0 io, 0 vm, 0 hdd
stress: debug: [6453] using backoff sleep of 30000us
stress: debug: [6453] setting timeout to 400s
stress: debug: [6453] --> hogcpu worker 10 [6454] forked
stress: debug: [6453] using backoff sleep of 27000us
stress: debug: [6453] setting timeout to 400s
stress: debug: [6453] --> hogcpu worker 9 [6455] forked
stress: debug: [6453] using backoff sleep of 24000us
stress: debug: [6453] setting timeout to 400s
stress: debug: [6453] --> hogcpu worker 8 [6456] forked
stress: debug: [6453] using backoff sleep of 21000us
stress: debug: [6453] setting timeout to 400s
stress: debug: [6453] --> hogcpu worker 7 [6457] forked
stress: debug: [6453] using backoff sleep of 18000us
stress: debug: [6453] setting timeout to 400s
stress: debug: [6453] --> hogcpu worker 6 [6458] forked
stress: debug: [6453] using backoff sleep of 15000us
stress: debug: [6453] setting timeout to 400s
stress: debug: [6453] --> hogcpu worker 5 [6459] forked
stress: debug: [6453] using backoff sleep of 12000us
stress: debug: [6453] setting timeout to 400s
stress: debug: [6453] --> hogcpu worker 4 [6460] forked
stress: debug: [6453] using backoff sleep of 9000us
stress: debug: [6453] setting timeout to 400s
stress: debug: [6453] --> hogcpu worker 3 [6461] forked
stress: debug: [6453] using backoff sleep of 6000us
stress: debug: [6453] setting timeout to 400s
stress: debug: [6453] --> hogcpu worker 2 [6462] forked
stress: debug: [6453] using backoff sleep of 3000us
stress: debug: [6453] setting timeout to 400s
stress: debug: [6453] --> hogcpu worker 1 [6463] forked
```

Paso 3: Abrir una nueva terminal de la instancia EC2 Stress Test

- Ingresar el comando **top**
- top =

Session ID: user2741130=Tom__sVillaseca-0894d83e08595293b


Instance ID: i-024490b8c028d7095

```
top - 22:29:11 up 17 min,  0 users,  load average: 5.28, 1.41, 0.53
Tasks:  99 total,  11 running,  51 sleeping,   0 stopped,   0 zombie
%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
KiB Mem :  993500 total,  411912 free,  104096 used,  477492 buff/cache
KiB Swap:   0 total,   0 free,   0 used.  746128 avail Mem
```

| PID | USER | PR | NI | VIRT | RES | SHR | S | %CPU | %MEM | TIME+ | COMMAND |
|------|------|----|----|--------|-------|-------|---|------|------|---------|-----------------|
| 6454 | root | 20 | 0 | 7580 | 92 | 0 | R | 10.0 | 0.0 | 0:04.37 | stress |
| 6456 | root | 20 | 0 | 7580 | 92 | 0 | R | 10.0 | 0.0 | 0:04.37 | stress |
| 6457 | root | 20 | 0 | 7580 | 92 | 0 | R | 10.0 | 0.0 | 0:04.37 | stress |
| 6458 | root | 20 | 0 | 7580 | 92 | 0 | R | 10.0 | 0.0 | 0:04.37 | stress |
| 6459 | root | 20 | 0 | 7580 | 92 | 0 | R | 10.0 | 0.0 | 0:04.37 | stress |
| 6460 | root | 20 | 0 | 7580 | 92 | 0 | R | 10.0 | 0.0 | 0:04.37 | stress |
| 6461 | root | 20 | 0 | 7580 | 92 | 0 | R | 10.0 | 0.0 | 0:04.37 | stress |
| 6463 | root | 20 | 0 | 7580 | 92 | 0 | R | 10.0 | 0.0 | 0:04.37 | stress |
| 6455 | root | 20 | 0 | 7580 | 92 | 0 | R | 9.7 | 0.0 | 0:04.37 | stress |
| 6462 | root | 20 | 0 | 7580 | 92 | 0 | R | 9.7 | 0.0 | 0:04.37 | stress |
| 6464 | root | 20 | 0 | 720000 | 21304 | 11260 | S | 0.3 | 2.1 | 0:00.05 | ssm-session-wor |
| 1 | root | 20 | 0 | 123624 | 5600 | 3940 | S | 0.0 | 0.6 | 0:02.25 | systemd |
| 2 | root | 20 | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | kthreadd |

Paso 4: CloudWatch → Panel de navegación → Alarms → LabCPUUtilizationAlarm

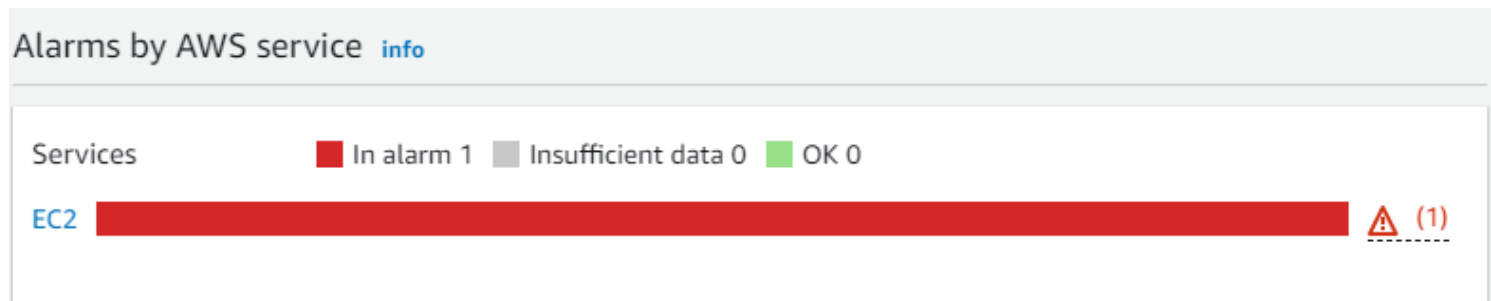
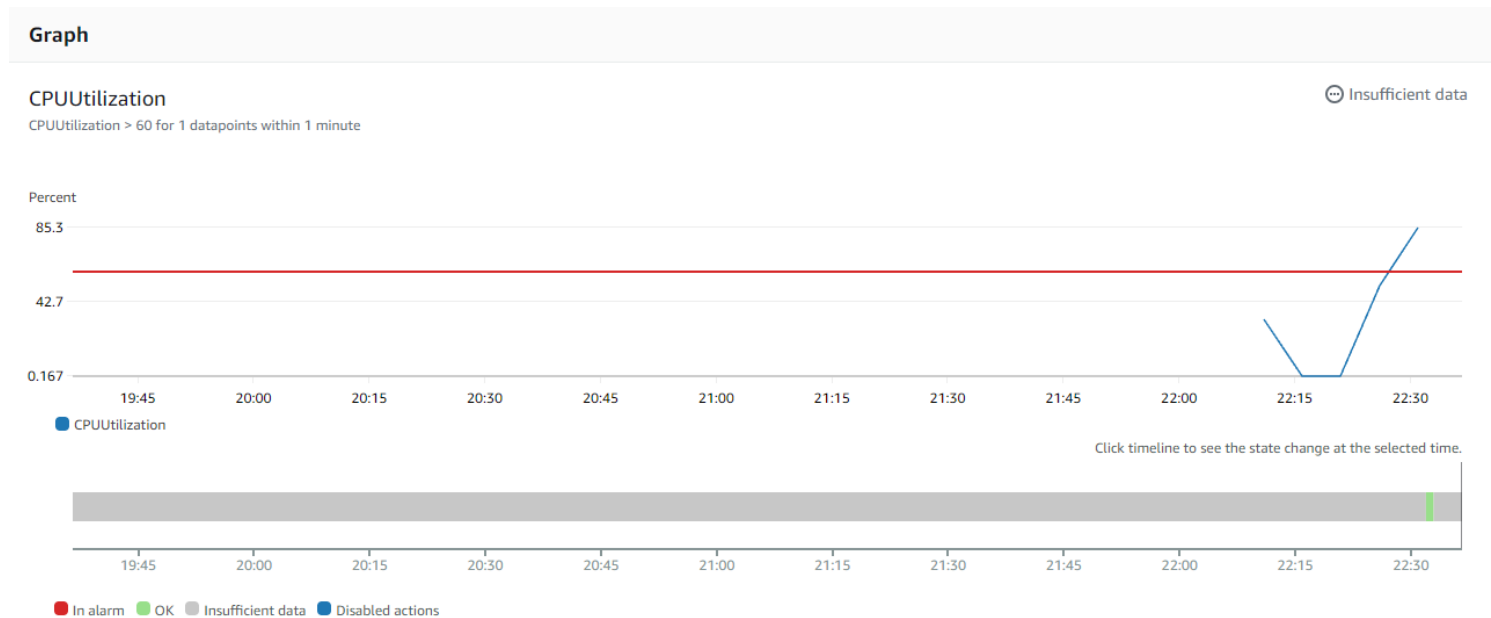
Alarms (1/1)

 Search

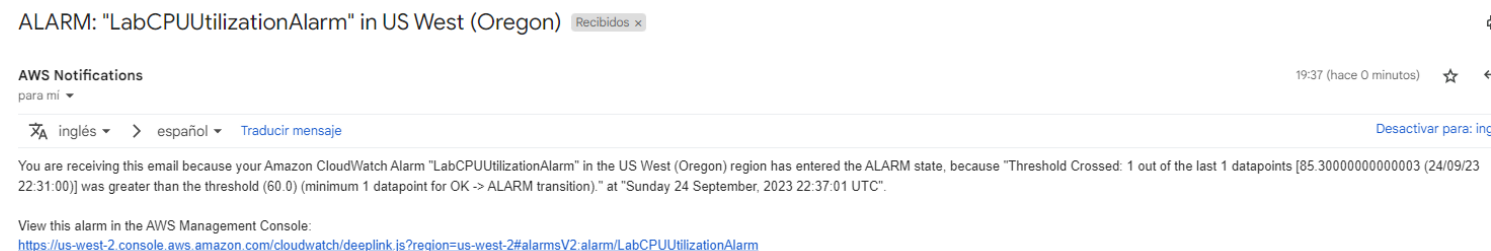
| <input checked="" type="checkbox"/> | Name | |
|-------------------------------------|--|--|
| <input checked="" type="checkbox"/> | LabCPUUtilizationAlarm | |

Paso 5: LabCPUUtilizationAlarm → Monitorear el gráfico.

- Alarm Status → In alarm (Estado de alarma)
- Se puede evidenciar en el gráfico que CPUUtilization superó el umbral de 60%.



Paso 6: Revisar correo electrónico para verificar que notificación de alarma fue enviada correctamente.



En esta tarea, ejecutó un comando para cargar la instancia EC2 al 100% durante 400 segundos. Este aumento en la utilización de la CPU activó la alarma para que pase al estado “En alarma”. Se confirmó el spike en la utilización de la CPU viendo el gráfico de CloudWatch. También recibió una notificación por correo electrónico alertándole del estado “En alarma” de la EC2.

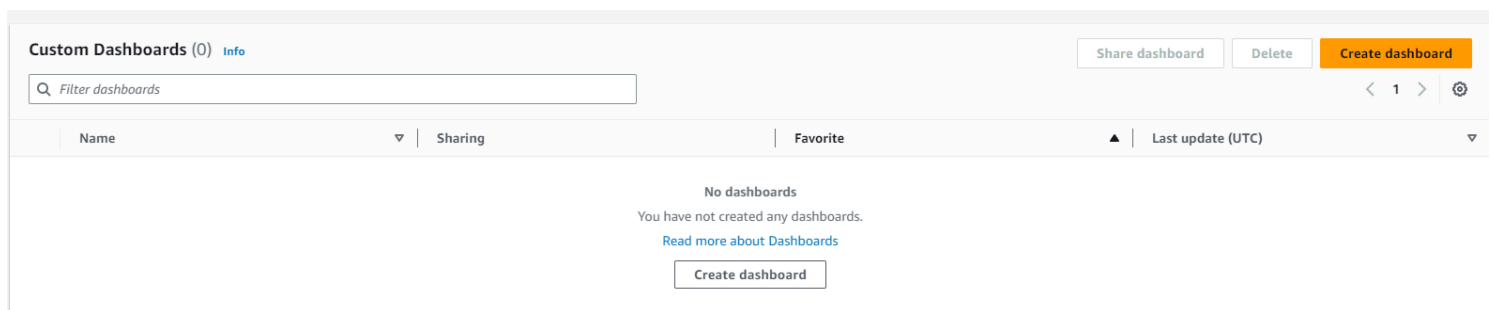
Tarea 4: Crear un panel de CloudWatch

En esta tarea, creará un panel de CloudWatch utilizando las mismas métricas de CPUUtilization que ha utilizado a lo largo de este laboratorio.

CloudWatch Dashboards = Páginas de inicio personalizables en la consola de CloudWatch que puede utilizar para supervisar sus recursos en una única vista.

- Puede monitorizar recursos que están repartidos por diferentes regiones.
- Puede utilizar los dashboards para crear vistas personalizadas de las métricas y alarmas de sus recursos de AWS.

Paso 1: CloudWatch → Panel de navegación → Dashboards → Create Dashboard



Paso 2: Create Dashboard → Configurar

- Dashboard name = LabEC2Dashboard

Create new dashboard ×

Dashboard name

LabEC2Dashboard

Valid characters in dashboard names include "0-9A-Za-z-_".

Cancel **Create dashboard**

- Create dashboard → Line → Metrics

Add widget

Line
Compare metrics over time

75 % **Number**
Instantly see the latest value for a metric

Gauge
See the latest value of a metric within a range

Stacked area
Compare the total over time

Bar
Compare categories of data

Pie
Show percentage or proportional data

Aa **Text**
Free text with markdown formatting

Custom widget
Code widgets using Lambda and more

Alarm status
Instantly see the status of your alarms in a grid view

Logs table
Explore results from Logs Insights

Explorer
A single widget with multiple tag-based graphs

From which data source would you like to create the widget?

☒ **Metrics**
Add widget based on Metrics and configure your widget on the next step.

☐ **Logs**
Add widget based on query results from CloudWatch Logs Insights.

- EC2 → Per-instance Metrics → Seleccionar “Stress Test” y “CPUUtilization”.

Add metric graph

Untitled graph ↗

☒ Persist time range 1h 3h 12h 1d 3d 1w Custom UTC Line

Percent

85.3

42.7

0.167

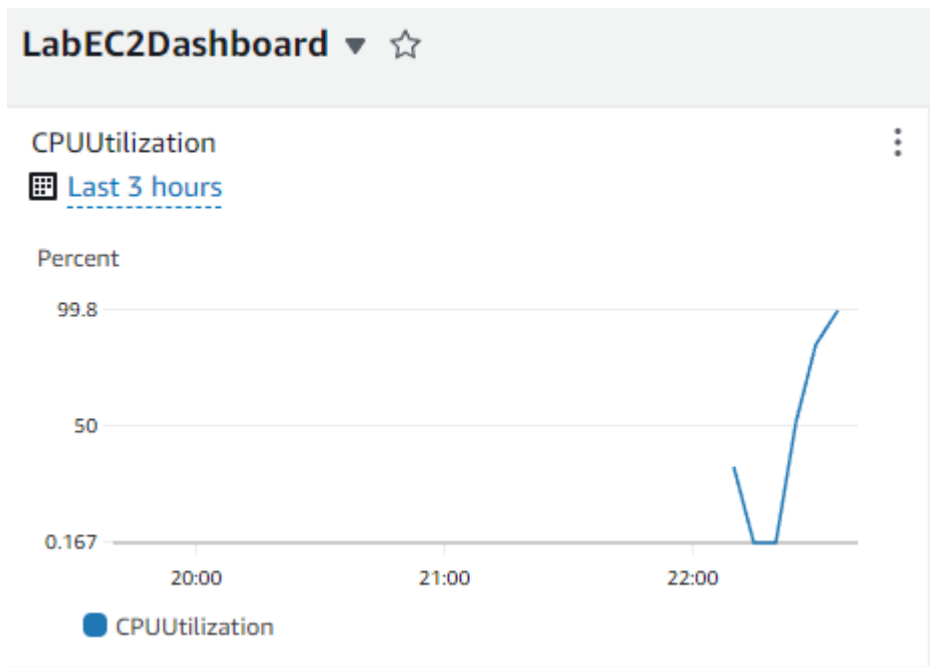
19:45 20:00 20:15 20:30 20:45 21:00 21:15 21:30 21:45 22:00 22:15 22:30

● CPUUtilization

Browse Query Graphed metrics (1) Options Source

| | | |
|-------------------------------------|---------------------|----------------|
| <input type="checkbox"/> | i-024490b8c028d7095 | NetworkIn |
| <input type="checkbox"/> | i-024490b8c028d7095 | DiskReadOps |
| <input checked="" type="checkbox"/> | i-024490b8c028d7095 | CPUUtilization |

- Create Widget → Save dashboard



Se creó un acceso directo rápido para ver la métrica **CPUUtilization** de la instancia EC2 Stress Test.

En este laboratorio, creó una alarma de CloudWatch que se activaba cuando la instancia Stress Test superaba un umbral específico de utilización de la CPU. Creó una suscripción mediante Amazon SNS que le enviaba un correo electrónico si se activaba esta alarma. Inició sesión en la instancia EC2 y ejecutó un comando de prueba de estrés que elevó la instancia EC2 al 100% de utilización de la CPU.

Esta prueba simulaba lo que podría ocurrir si un actor malicioso se hiciera con el control de una instancia EC2 y disparara la utilización de la CPU. Los spikes de CPU tienen varias causas posibles, una de las cuales es el malware.

Laboratorio Completado

