

Lab 281

Supervisar una instancia EC2



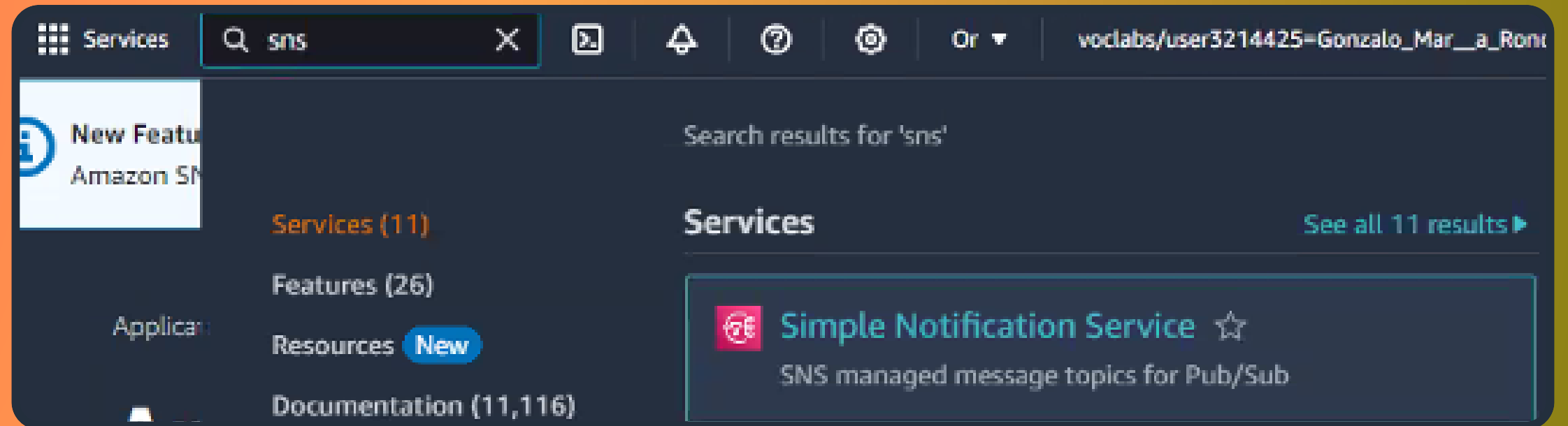
Objetivos

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

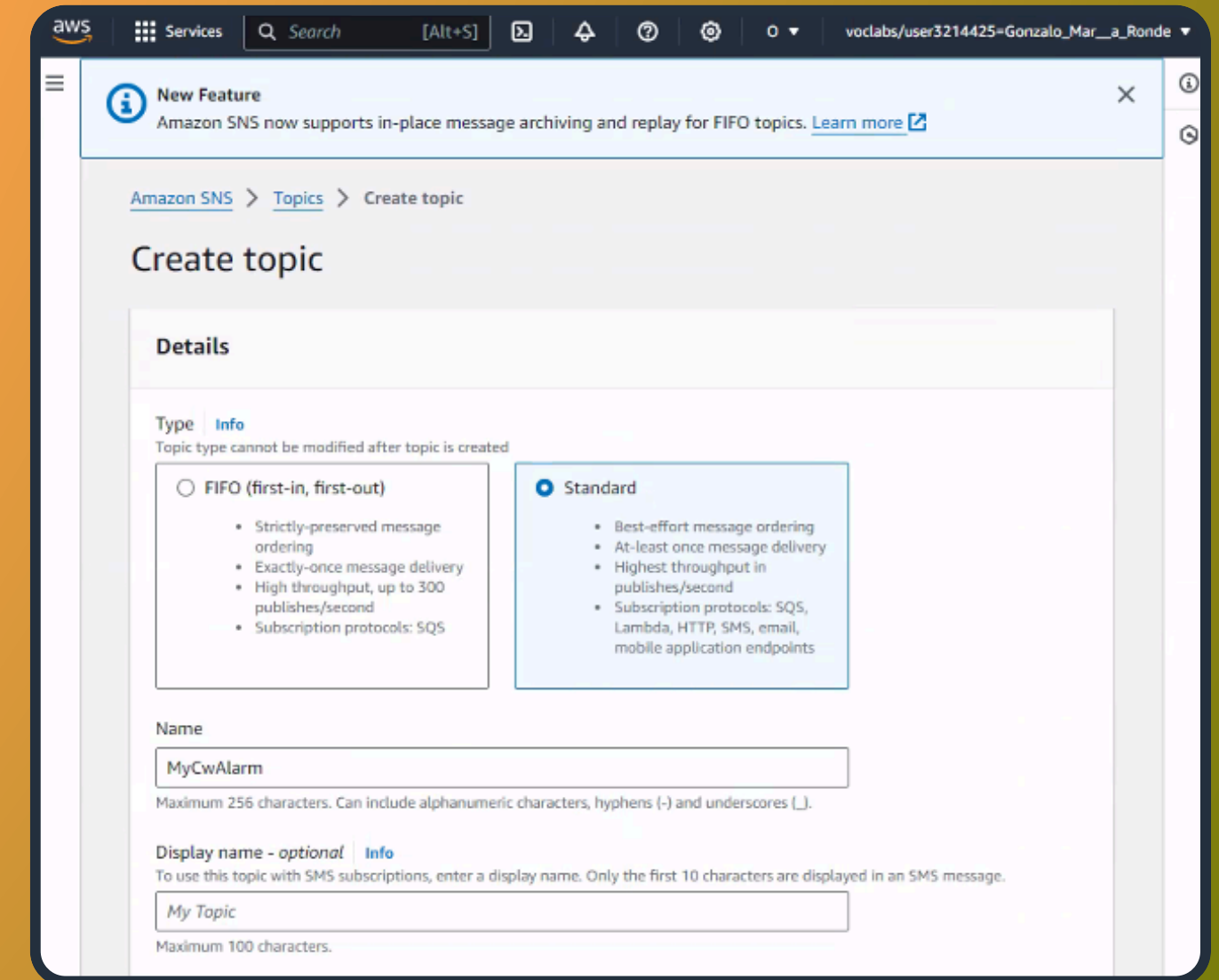
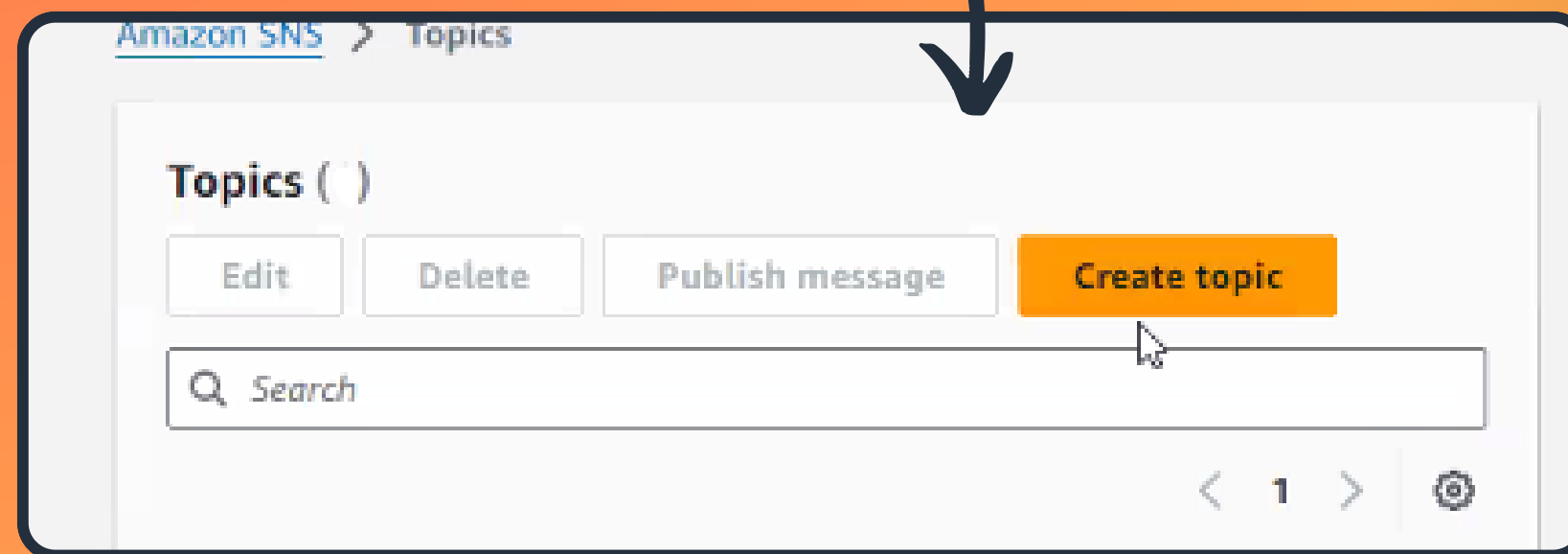
Tarea 1

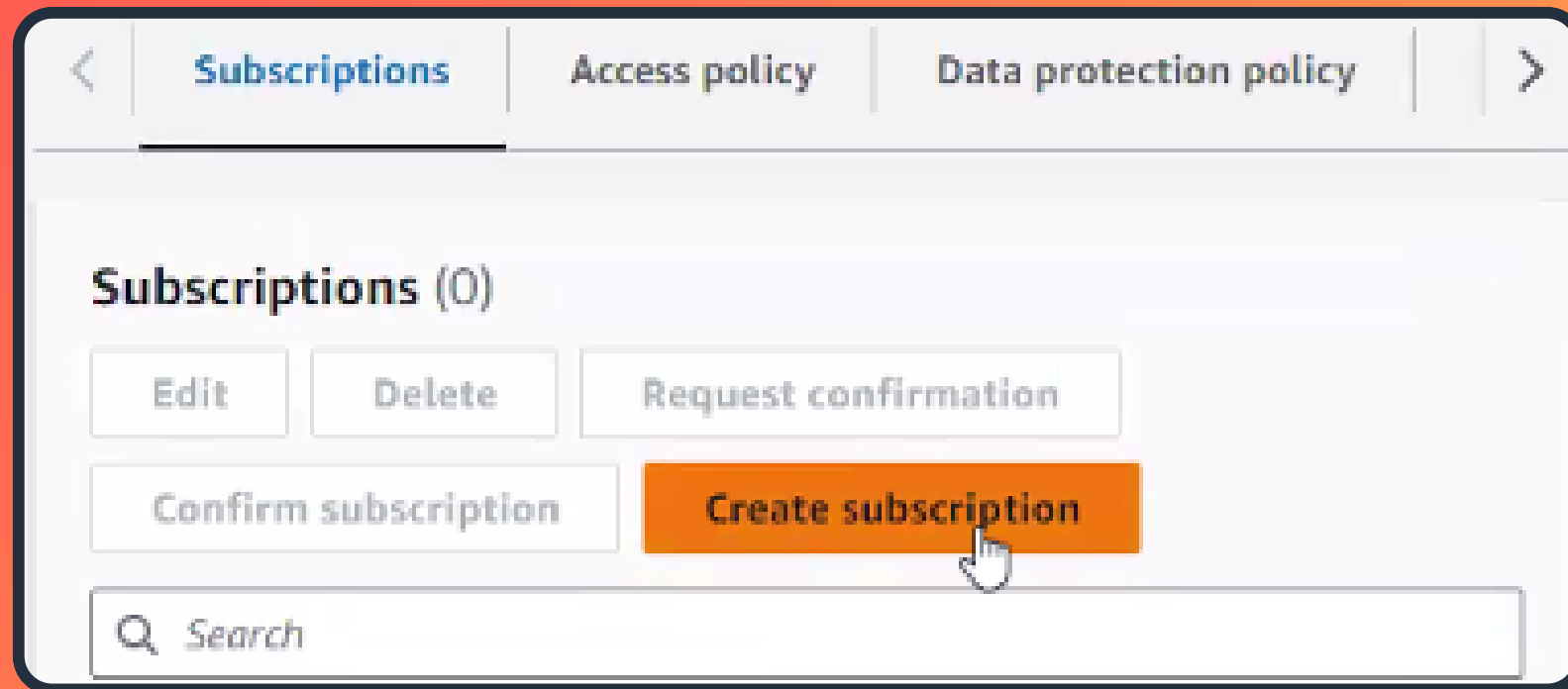
Configurar Amazon SNS

Entramos a Simple Notification Service



Nos vamos a Topics y creamos uno nuevo





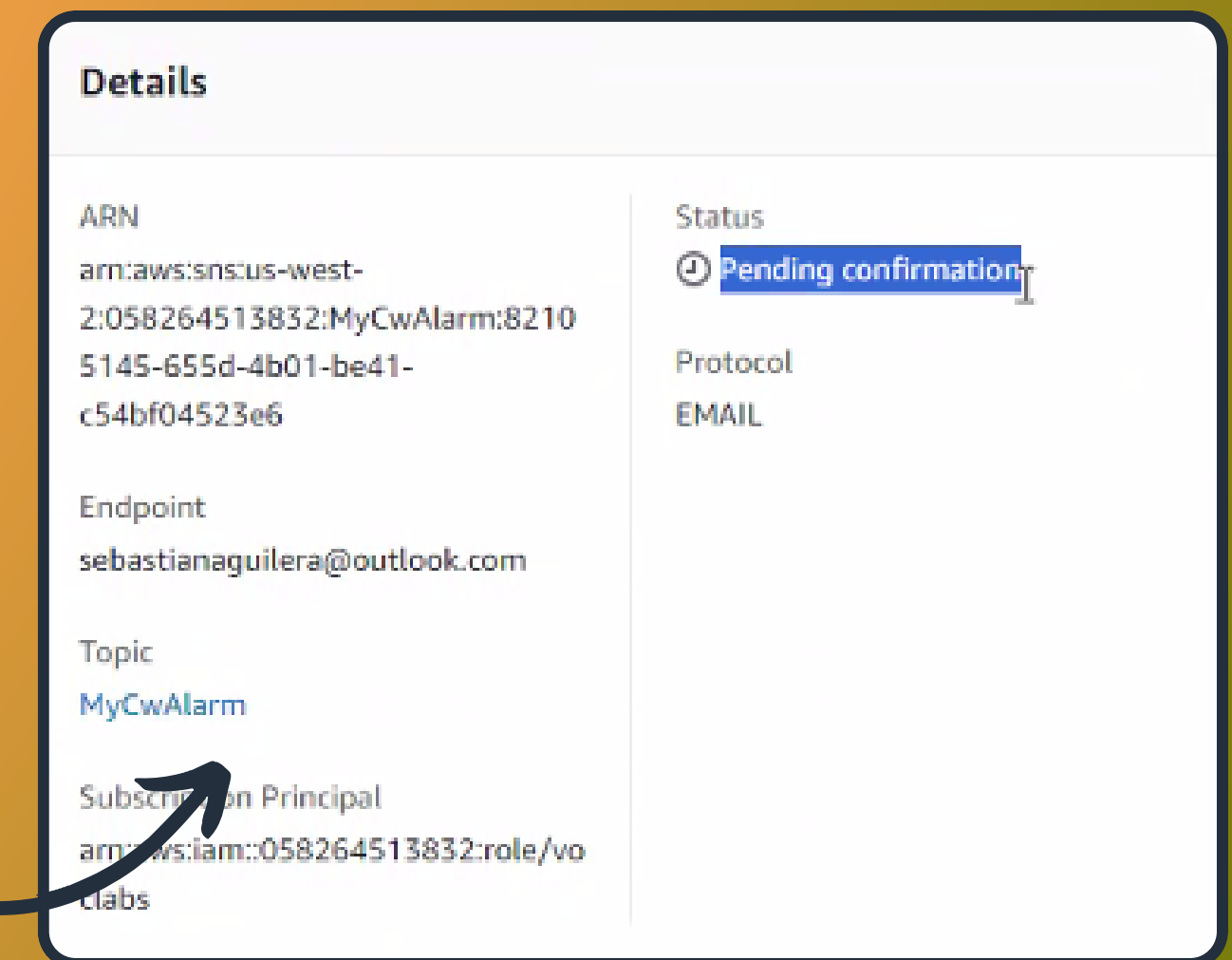
Nos vamos a la tab de Subscriptions y creamos una nueva suscripción

A screenshot of the 'Create subscription' form in the AWS IAM console. The breadcrumb navigation is 'Amazon SNS > Subscriptions > Create subscription'. The form has a 'Details' section with three fields: 'Topic ARN' (containing 'arn:aws:sns:us-west-2:058264513832:MyCwAlarm'), 'Protocol' (a dropdown menu with 'Email' selected), and 'Endpoint' (containing 'sebastianaguilera@outlook.com'). A blue information bar at the bottom states: 'After your subscription is created, you must confirm it. Info'.

Ingresamos los datos



Confirmamos la suscripción en el mail ingresado



AWS Notification - Subscription Confirmation



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

To: You



Wed 2024-05-15 08:44

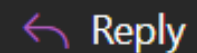
You have chosen to subscribe to the topic:

arn:aws:sns:us-west-2:058264513832: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](#)



Reply



Forward



Esto hará que nos lleguen mensajes a nuestra bandeja de entrada ante cualquier eventualidad

Confirmamos la subscripción a las notificaciones en nuestro e-mail

Subscription: 82105145-655d-4b01-be41-c54bf04523e6

Edit

Delete

Details

ARN

arn:aws:sns:us-west-2:058264513832:MyCwAlarm:82105145-655d-4b01-be41-c54bf04523e6

Status

✔ Confirmed

Protocol

EMAIL

Endpoint

sebastianaguilera@outlook.com

Topic

MyCwAlarm

Subscription Principal

arn:aws:iam::058264513832:role/voclabs

Regresamos a Suscripciones y podemos ver que el Estado ahora es Confirmado

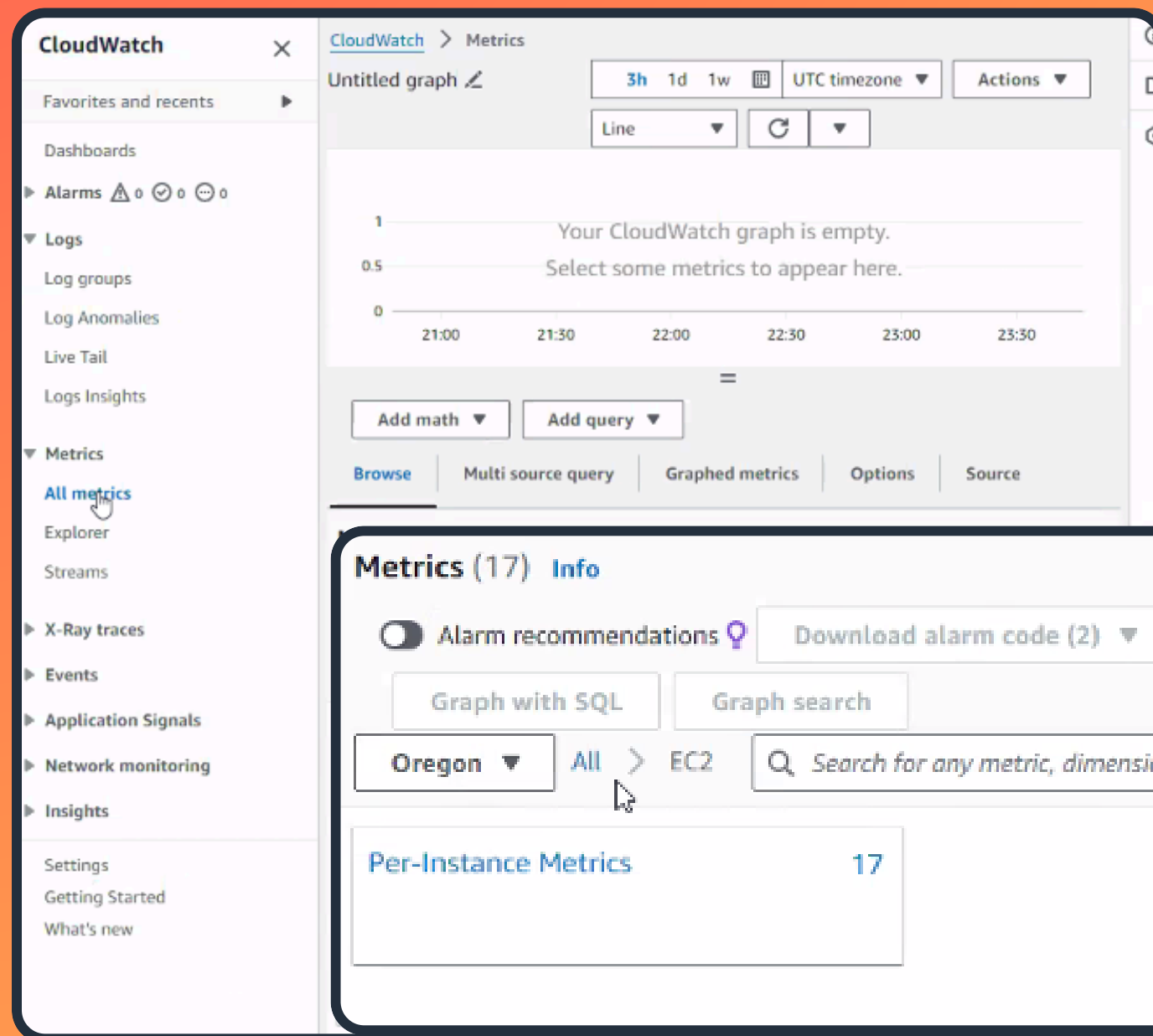


Tarea 2

Crear una alarma de
CloudWatch

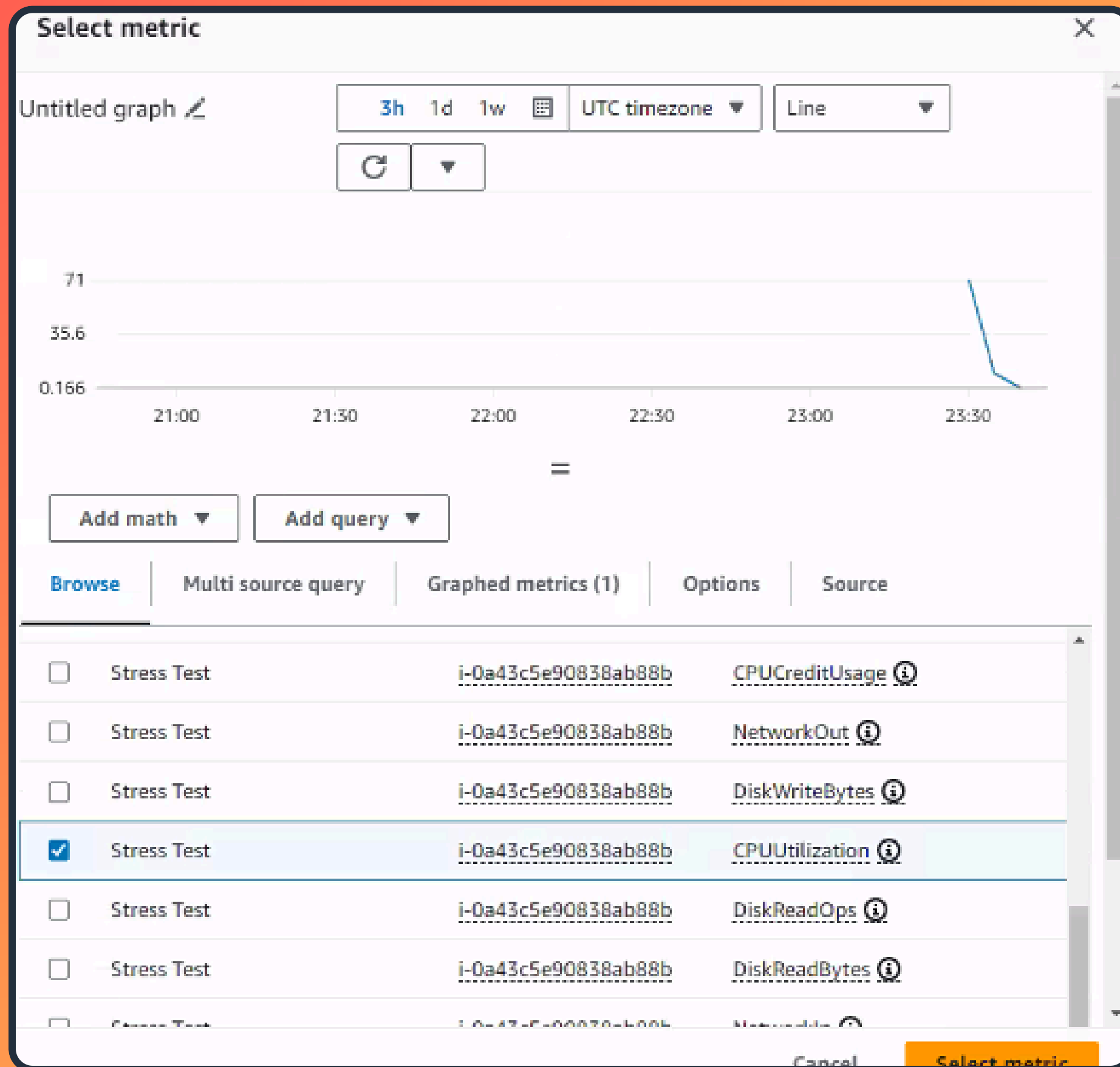


Entramos al servicio de CloudWatch



Ingresamos en “All metrics” y vemos todas las métricas por instancia

CloudWatch es un servicio de supervisión y observabilidad, ofrece datos e información útil para supervisar sus aplicaciones, responder a los cambios en el rendimiento de todo el sistema y optimizar el uso de recursos.

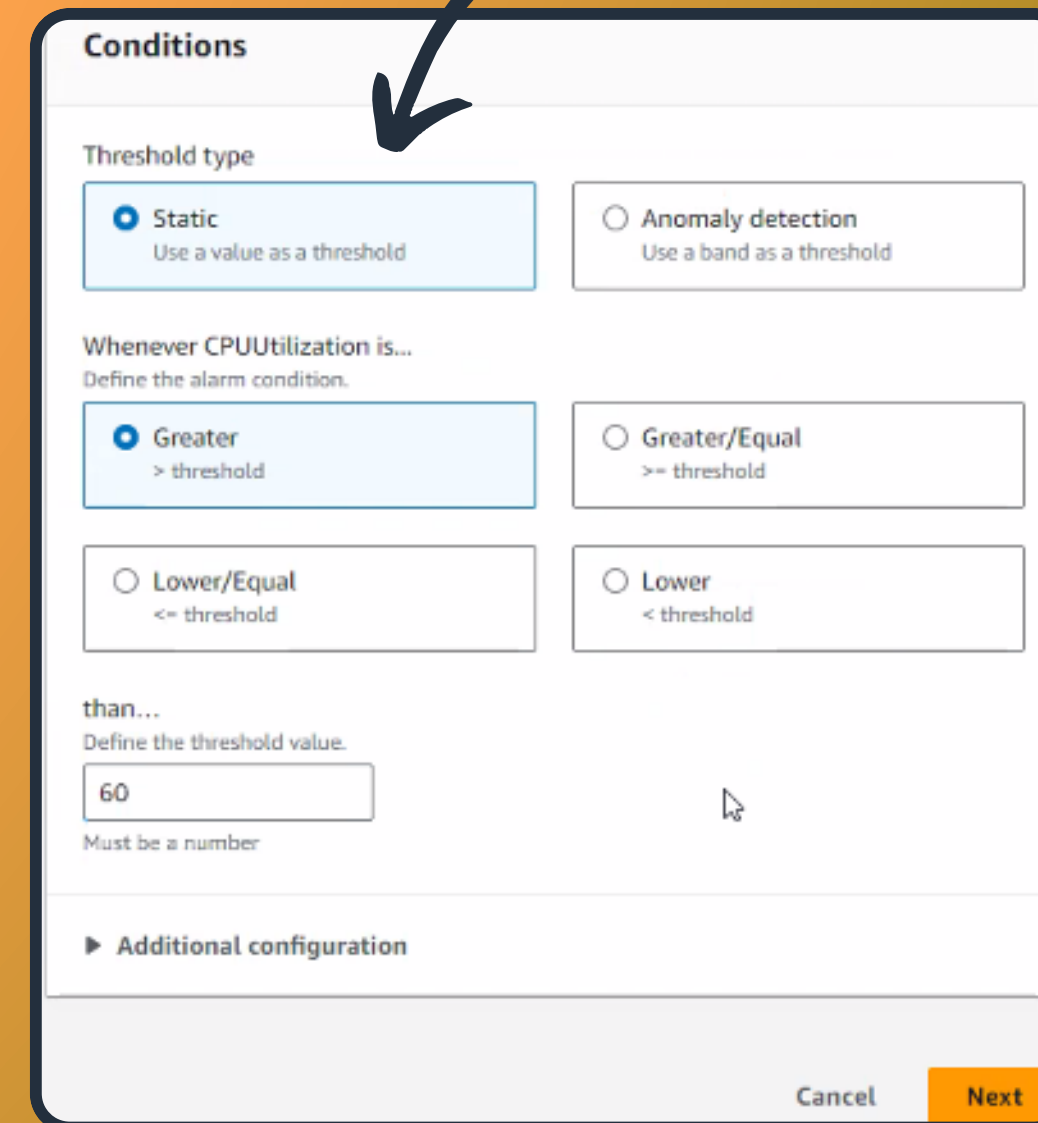
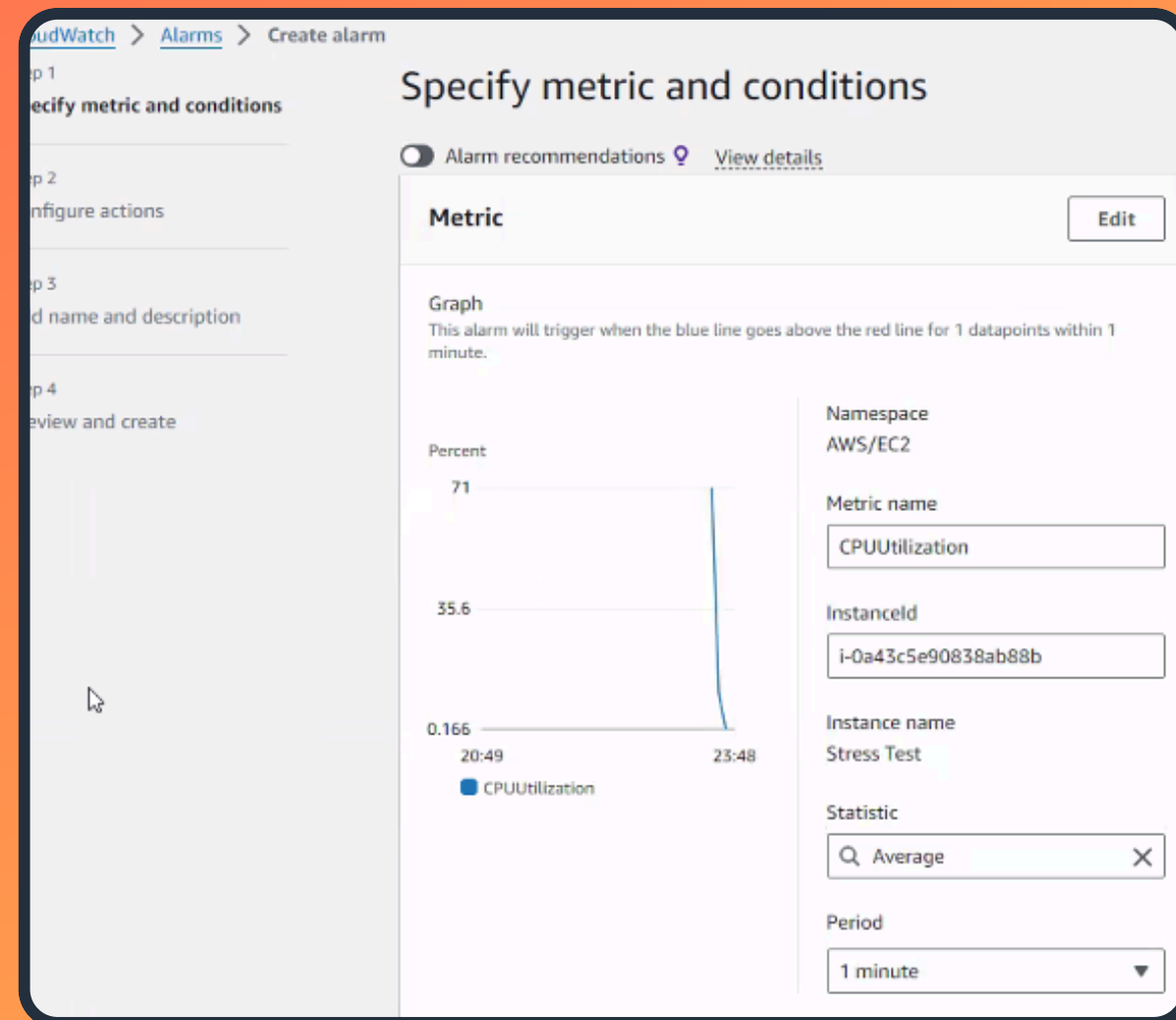


Seleccionamos
CPUUtilization

Seleccionamos en todas las alarmas



En Specify metric and conditions establecemos las siguientes configuraciones y seleccionemos Next



Step 1
[Specify metric and conditions](#)

Step 2
Configure actions

Step 3
Add name and description

Step 4
Preview and create

Configure actions

Notification

Alarm state trigger
Define the alarm state that will trigger this action. [Remove](#)

☒ **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.

☒ Select an existing SNS topic

☐ Create new topic

☐ Use topic ARN to notify other accounts

Send a notification to...

Only topics belonging to this account are listed here. All persons and applications subscribed to the selected topic will receive notifications.

Email (endpoints)
sebastianaguilera@outlook.com - [View in SNS Console](#)

[Add notification](#)

En la pagina **Configure actions** cambiamos algunas opciones...

[CloudWatch](#) > [Alarms](#) > Create alarm

Step 1
[Specify metric and conditions](#)

Step 2
[Configure actions](#)

Step 3
Add name and description

Step 4
Preview and create

Add name and description

Name and description

Alarm name

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

[Edit](#) [Preview](#)

CloudWatch alarm for Stress Test EC2 instance CPUUtilization

Up to 1024 characters (60/1024)

i Markdown formatting is only applied when viewing your alarm in the console. The description will remain in plain text in the alarm notifications.

...tales como estado de alarma, el tema de SNS, el destino de la notificacion, nombre y descripcion de la misma.

Tarea 3

Probar la alarma de
CloudWatch

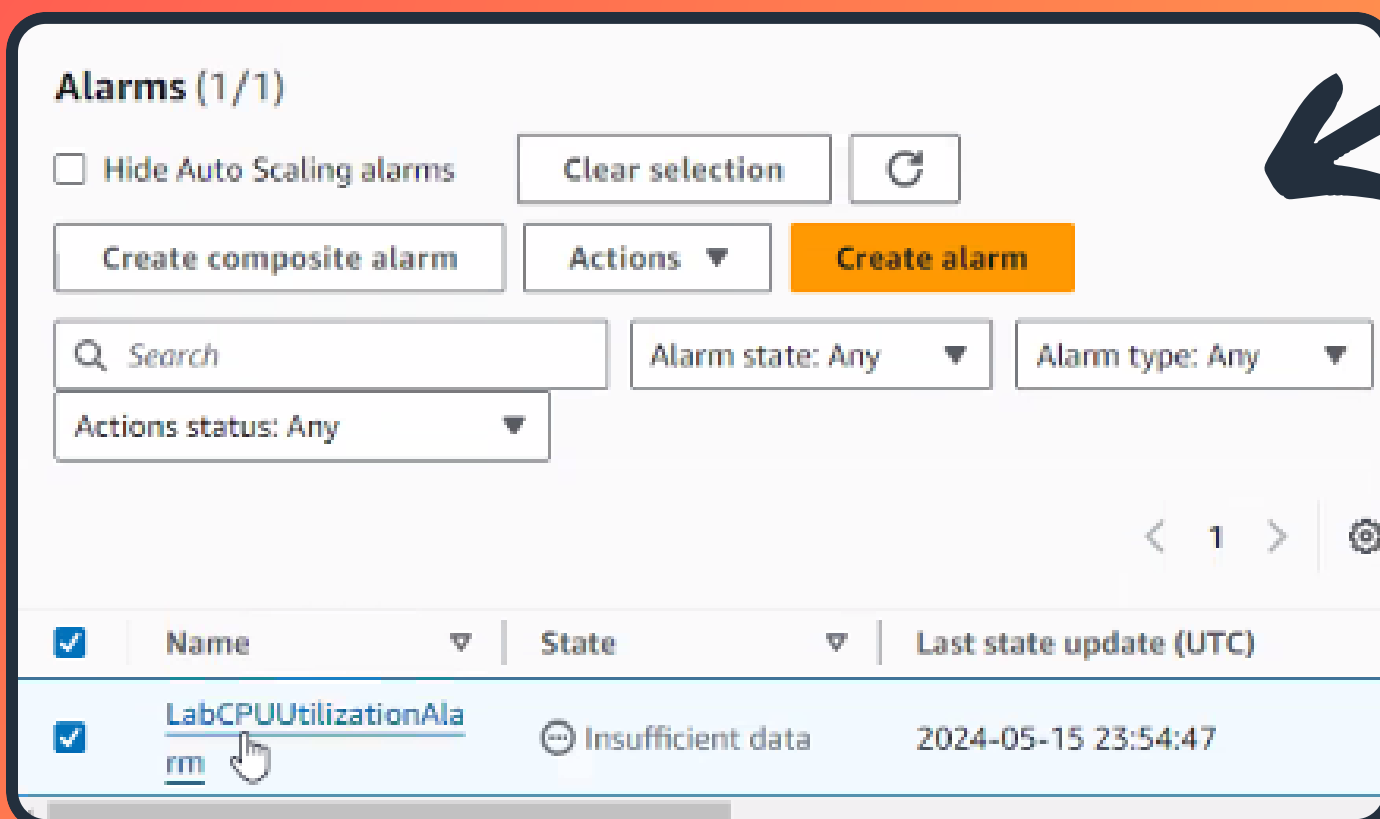
Ejecutamos el comando stress para aumentar la carga de trabajo de la instancia.

```
an-4.2$ sudo stress --cpu 10 -v --timeout 400s
stress: info: [6415] dispatching hogs: 10 cpu, 0 io, 0 vm, 0 hdd
stress: debug: [6415] using backoff sleep of 30000us
stress: debug: [6415] setting timeout to 400s
stress: debug: [6415] --> hogcpu worker 10 [6416] forked
stress: debug: [6415] using backoff sleep of 27000us
stress: debug: [6415] setting timeout to 400s
stress: debug: [6415] --> hogcpu worker 9 [6417] forked
stress: debug: [6415] using backoff sleep of 24000us
stress: debug: [6415] setting timeout to 400s
stress: debug: [6415] --> hogcpu worker 8 [6418] forked
stress: debug: [6415] using backoff sleep of 21000us
stress: debug: [6415] setting timeout to 400s
stress: debug: [6415] --> hogcpu worker 7 [6419] forked
stress: debug: [6415] using backoff sleep of 18000us
stress: debug: [6415] setting timeout to 400s
stress: debug: [6415] --> hogcpu worker 6 [6420] forked
stress: debug: [6415] using backoff sleep of 15000us
stress: debug: [6415] setting timeout to 400s
stress: debug: [6415] --> hogcpu worker 5 [6421] forked
stress: debug: [6415] using backoff sleep of 12000us
stress: debug: [6415] setting timeout to 400s
stress: debug: [6415] --> hogcpu worker 4 [6422] forked
stress: debug: [6415] using backoff sleep of 9000us
stress: debug: [6415] setting timeout to 400s
stress: debug: [6415] --> hogcpu worker 3 [6423] forked
stress: debug: [6415] using backoff sleep of 6000us
stress: debug: [6415] setting timeout to 400s
stress: debug: [6415] --> hogcpu worker 2 [6424] forked
stress: debug: [6415] using backoff sleep of 3000us
stress: debug: [6415] setting timeout to 400s
stress: debug: [6415] --> hogcpu worker 1 [6425] forked
```

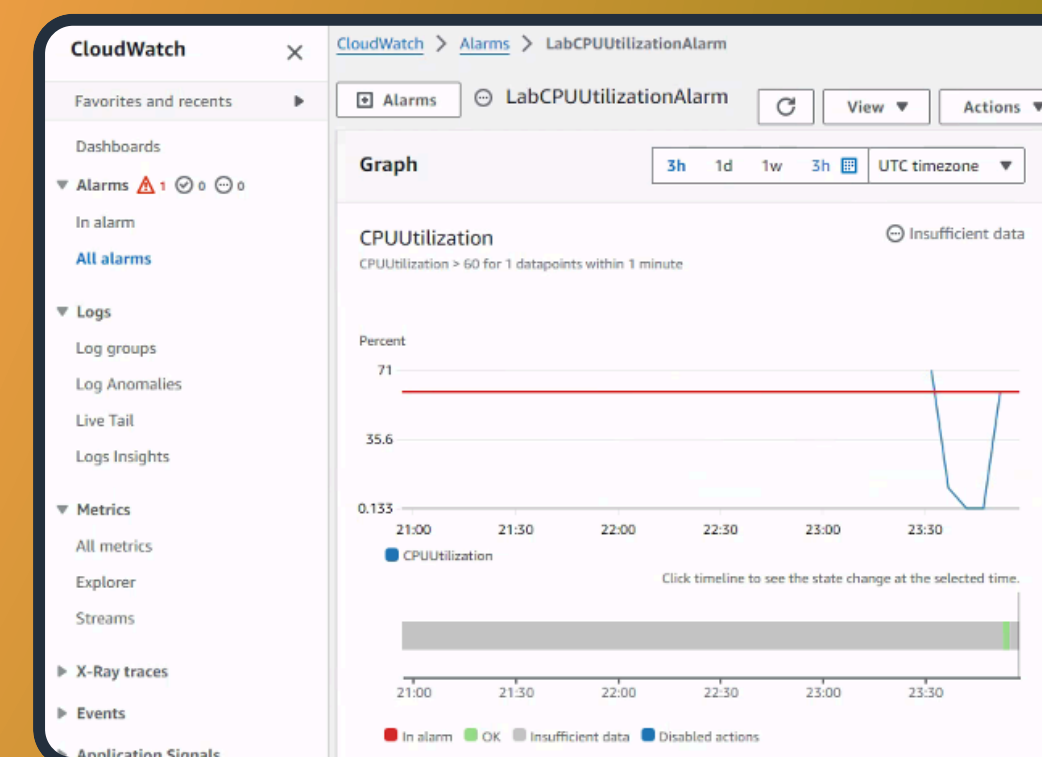
Ejecutamos el comando top para ver los procesos y la utilización de recursos

```
top 20:04:00 up 15 min, 0 users, load average: 0.00, 1.00, 0.00
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
Mem: 993492 total, 427492 free, 106564 used, 459436 buff/cache
Swap: 0 total, 0 free, 0 used, 741796 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 6416 root        20   0   7580     96      0 R   10.0   0.0   0:05.85 stress
 6417 root        20   0   7580     96      0 R   10.0   0.0   0:05.85 stress
 6418 root        20   0   7580     96      0 R   10.0   0.0   0:05.85 stress
 6419 root        20   0   7580     96      0 R   10.0   0.0   0:05.85 stress
 6420 root        20   0   7580     96      0 R   10.0   0.0   0:05.85 stress
 6421 root        20   0   7580     96      0 R   10.0   0.0   0:05.85 stress
 6422 root        20   0   7580     96      0 R   10.0   0.0   0:05.85 stress
 6423 root        20   0   7580     96      0 R   10.0   0.0   0:05.85 stress
 6424 root        20   0   7580     96      0 R   10.0   0.0   0:05.85 stress
 6425 root        20   0   7580     96      0 R   10.0   0.0   0:05.85 stress
    1 root        20   0 123620   5500   3840 S    0.0   0.6   0:01.76 systemd
    2 root        20   0      0      0      0 S    0.0   0.0   0:00.00 kthreadd
    4 root         0 -20      0      0      0 I    0.0   0.0   0:00.00 kworker/0:0H
    5 root        20   0      0      0      0 I    0.0   0.0   0:00.15 kworker/u30:0
    6 root         0 -20      0      0      0 I    0.0   0.0   0:00.00 mm_percpu_wq
    7 root        20   0      0      0      0 S    0.0   0.0   0:00.08 ksoftirqd/0
    8 root        20   0      0      0      0 I    0.0   0.0   0:00.30 rcu_sched
    9 root        20   0      0      0      0 I    0.0   0.0   0:00.00 rcu_bh
   10 root        rt   0      0      0      0 S    0.0   0.0   0:00.00 migration/0
   11 root        rt   0      0      0      0 S    0.0   0.0   0:00.00 watchdog/0
   12 root        20   0      0      0      0 S    0.0   0.0   0:00.00 cpuhp/0
   14 root        20   0      0      0      0 S    0.0   0.0   0:00.00 kdevtmpfs
   15 root         0 -20      0      0      0 I    0.0   0.0   0:00.00 netns
  192 root        20   0      0      0      0 S    0.0   0.0   0:00.00 khungtaskd
  193 root        20   0      0      0      0 S    0.0   0.0   0:00.00 oom_reaper
  194 root         0 -20      0      0      0 I    0.0   0.0   0:00.00 writeback
  196 root        20   0      0      0      0 S    0.0   0.0   0:00.00 kcompactd0
  197 root       25   5      0      0      0 S    0.0   0.0   0:00.00 ksm
  198 root       39  19      0      0      0 S    0.0   0.0   0:00.00 khugepaged
  199 root         0 -20      0      0      0 I    0.0   0.0   0:00.00 crypto
  200 root         0 -20      0      0      0 I    0.0   0.0   0:00.00 kintegrityd
  202 root         0 -20      0      0      0 I    0.0   0.0   0:00.00 kblockd
  544 root        20   0      0      0      0 S    0.0   0.0   0:00.00 xen-balloon
  555 root         0 -20      0      0      0 I    0.0   0.0   0:00.00 md
  558 root         0 -20      0      0      0 I    0.0   0.0   0:00.00 edac-poller
  563 root         0 -20      0      0      0 I    0.0   0.0   0:00.00 watchdogd
  704 root        20   0      0      0      0 S    0.0   0.0   0:00.01 kauditd
  710 root        20   0      0      0      0 S    0.0   0.0   0:00.13 kswapd0
  800 root         0 -20      0      0      0 I    0.0   0.0   0:00.00 xfsalloc
  801 root         0 -20      0      0      0 I    0.0   0.0   0:00.00 xfs_mru_cache
  856 root         0 -20      0      0      0 I    0.0   0.0   0:00.00 kthrotld
  867 root        20   0      0      0      0 S    0.0   0.0   0:00.00 xenbus
  868 root        20   0      0      0      0 S    0.0   0.0   0:00.01 xenwatch
```

Volvemos a la consola de CloudWatch y entramos en alarmas



Al superar el 60% del uso de CPU se activara la alarma e enviara un mail al destino

ALARM: "LabCPUUtilizationAlarm" in US West (Oregon)

AWS Notifications <no-reply@sns.amazonaws.com>
To: You
Wed 2024-05-15 08:57 PM

You are receiving this email because your Amazon CloudWatch Alarm "LabCPUUtilizationAlarm" in the US West (Oregon) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [60.459034917106614 (15/05/24 23:52:00)] was greater than the threshold (60.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Wednesday 15 May, 2024 23:57:47 UTC".

View this alarm in the AWS Management Console:
<https://us-west-2.console.aws.amazon.com/cloudwatch/deeplink.js?region=us-west-2#alarmsV2:alarm/LabCPUUtilizationAlarm>

Alarm Details:

- Name: LabCPUUtilizationAlarm
- Description: CloudWatch alarm for Stress Test EC2 instance CPUUtilization
- State Change: INSUFFICIENT_DATA -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [60.459034917106614 (15/05/24 23:52:00)] was greater than the threshold (60.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Wednesday 15 May, 2024 23:57:47 UTC
- AWS Account: 058264513832
- Alarm Arn: arn:aws:cloudwatch:us-west-2:058264513832:alarm:LabCPUUtilizationAlarm

Threshold:

- The alarm is in the ALARM state when the metric is GreaterThanThreshold 60.0 for at least 1 of the last 1 period(s) of 60 seconds.

Tarea 4

Crear un tablero en
CloudWatch

Custom Dashboards (0) [Info](#)

[Share dashboard](#) [Delete](#) [Create dashboard](#)

Dashboard name

LabEC2Dashboard

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

Creamos un nuevo Dashboard con nombre "LabEC2Dashboard"

Add widget ×

Data sources types - new

- ☒ Cloudwatch
- ☐ Other content types
- ☐ Create data sources

Widget Configuration

Data type

Metrics Logs Alarms

Widget type

- ☒ **Line**
Compare metrics over time
- ☐ Data table
Compare metrics values over time in a table
- ☐ 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
- ☐ Explorer
A single widget with multiple tag-based graphs

Cancel [Next](#)

Seleccionamos un gráfico de líneas y lo creamos

Creamos un widget para que sea más sencillo visualizar las métricas

Add metric graph

Untitled graph ☒ Persist time range 3h 1d 1w UTC timezone Line

Percent

71

35.6

0.133

21:00 21:30 22:00 22:30 23:00 23:30

CPUUtilization

=

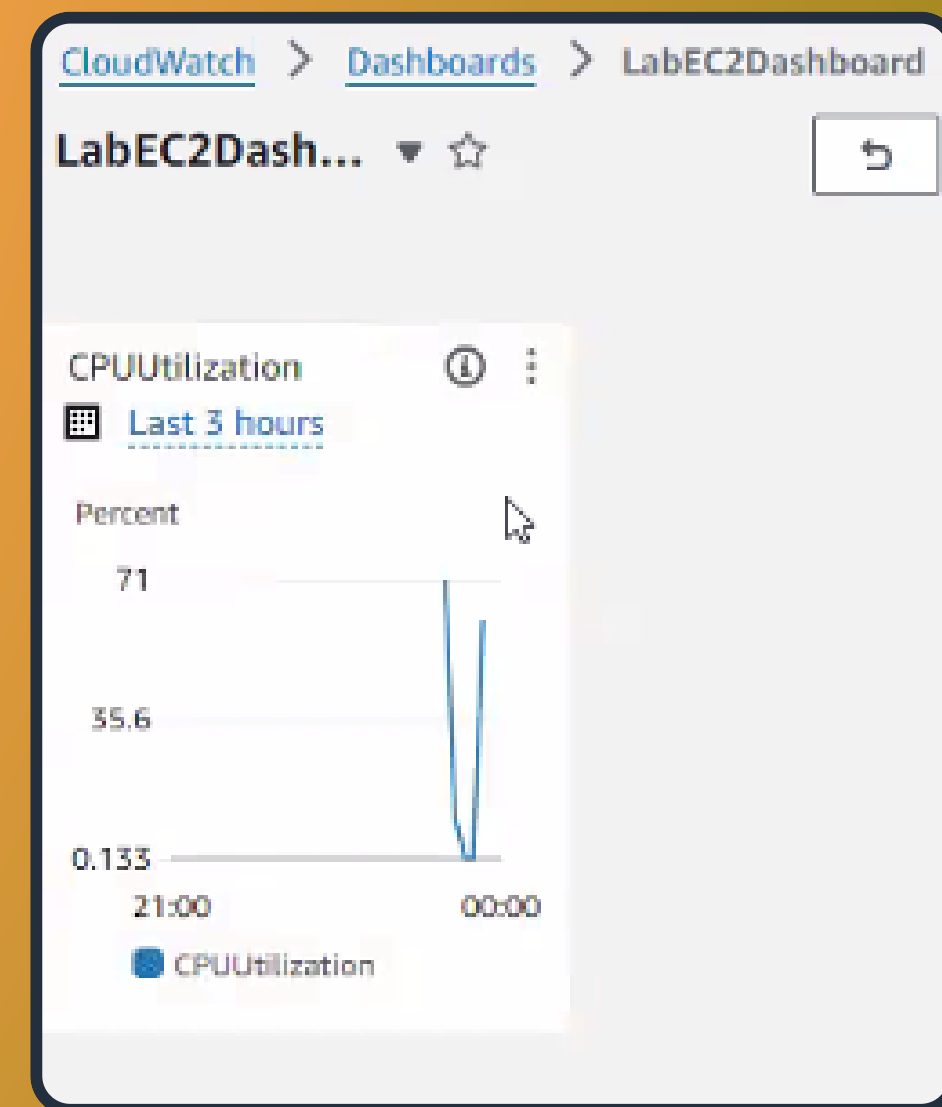
Add math Add query

Browse Multi source query Graphed metrics (1) Options Source

Oregon All > EC2 > Per-Instance Metrics < 1 >

	Instance name 17/17	Instanceld	Metric name
<input type="checkbox"/>	Stress Test	i-0a43c5e90838ab88b	CPUCreditBalance
<input type="checkbox"/>	Stress Test	i-0a43c5e90838ab88b	CPUCreditUsage
<input type="checkbox"/>	Stress Test	i-0a43c5e90838ab88b	CPUSurplusCreditBalance
<input type="checkbox"/>	Stress Test	i-0a43c5e90838ab88b	CPUSurplusCreditsCharged
<input checked="" type="checkbox"/>	Stress Test	i-0a43c5e90838ab88b	CPUUtilization

Cancel Create widget



¡Muchas gracias!

- Ignacio Suárez
- Gonzalo Rondeau
- Sebastián Aguilera
- Joel Umpierrez
- Sabrina Magnani
- Agustín Rodríguez