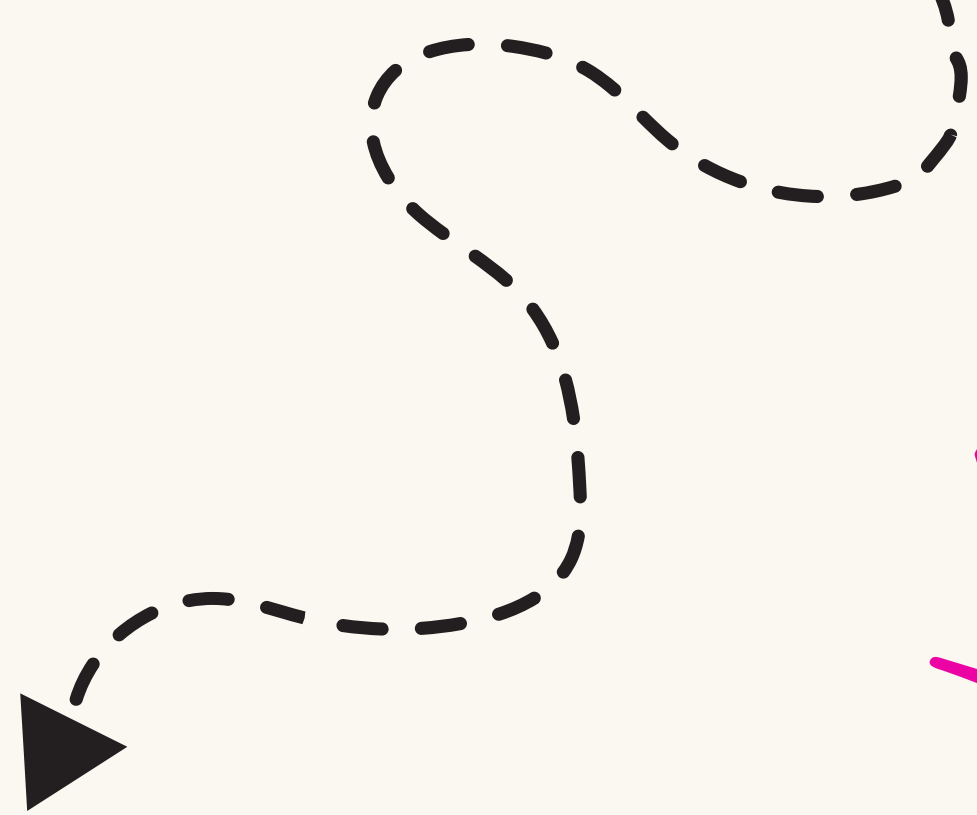




Laboratorio

# MANAGING SERVICES MONITORING



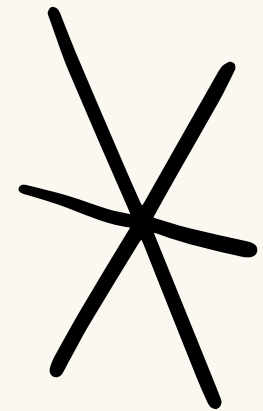
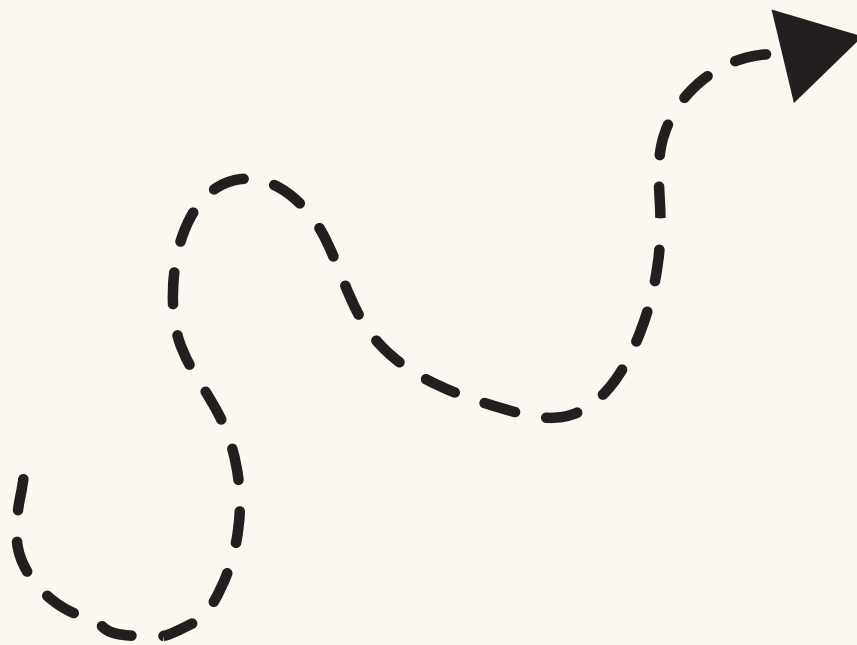
## **PARTICIPANTES:**

- ELISA GAMARRA, GABRIEL PORLEY
- BENJAMIN SABAÑO, NACHO SUÁREZ
- SANTIAGO BURGUEÑO



# Objetivos

- Verificar el estado del servicio httpd
- Realizar una conexión http con la dirección IP del host local
- Detener la conexión httpd

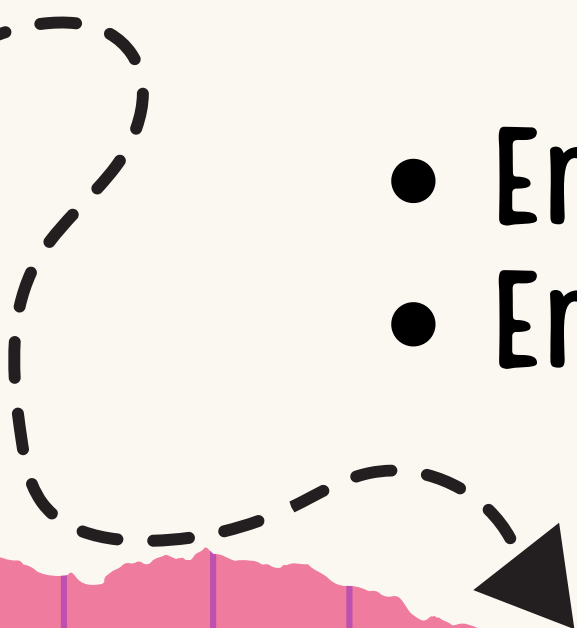




## **TAREA 1:**



## **CONECTARSE A LA INSTANCIA EC2**

- Esperaremos a que la instancia esté cargada y nos conectaremos a la misma utilizando SSH.
  - En Windows: usaremos PuTTY
  - En Linux: con el comando ssh
- 

# CONEXIÓN CON LA INSTANCIA

```
ec2-user@ip-10-0-10-144:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
#_      Amazon Linux 2  
###_      AL2 End of Life is 2025-06-30.  
#####\      A newer version of Amazon Linux is available!  
#####|      Amazon Linux 2023, GA and supported until 2028-03-15.  
#####/      https://aws.amazon.com/linux/amazon-linux-2023/  
V~' '->  
/      No packages needed for security; 2 packages available  
./_      Run "sudo yum update" to apply all updates.  
/_/      [ec2-user@ip-10-0-10-144 ~]$
```

## **TAREA 2:**

## **VER EL ESTADO DEL SERVICIO HTTPD**

- HTTPD es el servicio para el servidor http de Apache, el cual se encuentra instalado en la instancia.
- Comprobaremos el estado del servicio httpd utilizando el comando `systemctl`:
- `sudo systemctl status httpd.service`

# ESTADO DEL HTTPD

```
ec2-user@ip-10-0-10-144:~  
[ec2-user@ip-10-0-10-144 ~]$ sudo systemctl start httpd.service  
[ec2-user@ip-10-0-10-144 ~]$ sudo systemctl status httpd.service  
● httpd.service - The Apache HTTP Server  
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor prese  
t: disabled)  
   Active: active (running) since Fri 2024-04-26 21:30:37 UTC; 16s ago  
     Docs: man:httpd.service(8)  
  Main PID: 2594 (httpd)  
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes se  
rved/sec:  0 B/sec"  
    CGroup: /system.slice/httpd.service  
            └─2594 /usr/sbin/httpd -DFOREGROUND  
              └─2595 /usr/sbin/httpd -DFOREGROUND  
                └─2597 /usr/sbin/httpd -DFOREGROUND  
                  └─2602 /usr/sbin/httpd -DFOREGROUND  
                    └─2604 /usr/sbin/httpd -DFOREGROUND  
                      └─2609 /usr/sbin/httpd -DFOREGROUND  
  
Apr 26 21:30:37 ip-10-0-10-144.us-west-2.compute.internal systemd[1]: Startin...  
Apr 26 21:30:37 ip-10-0-10-144.us-west-2.compute.internal systemd[1]: Started...  
Hint: Some lines were ellipsized, use -l to show in full.  
[ec2-user@ip-10-0-10-144 ~]$
```

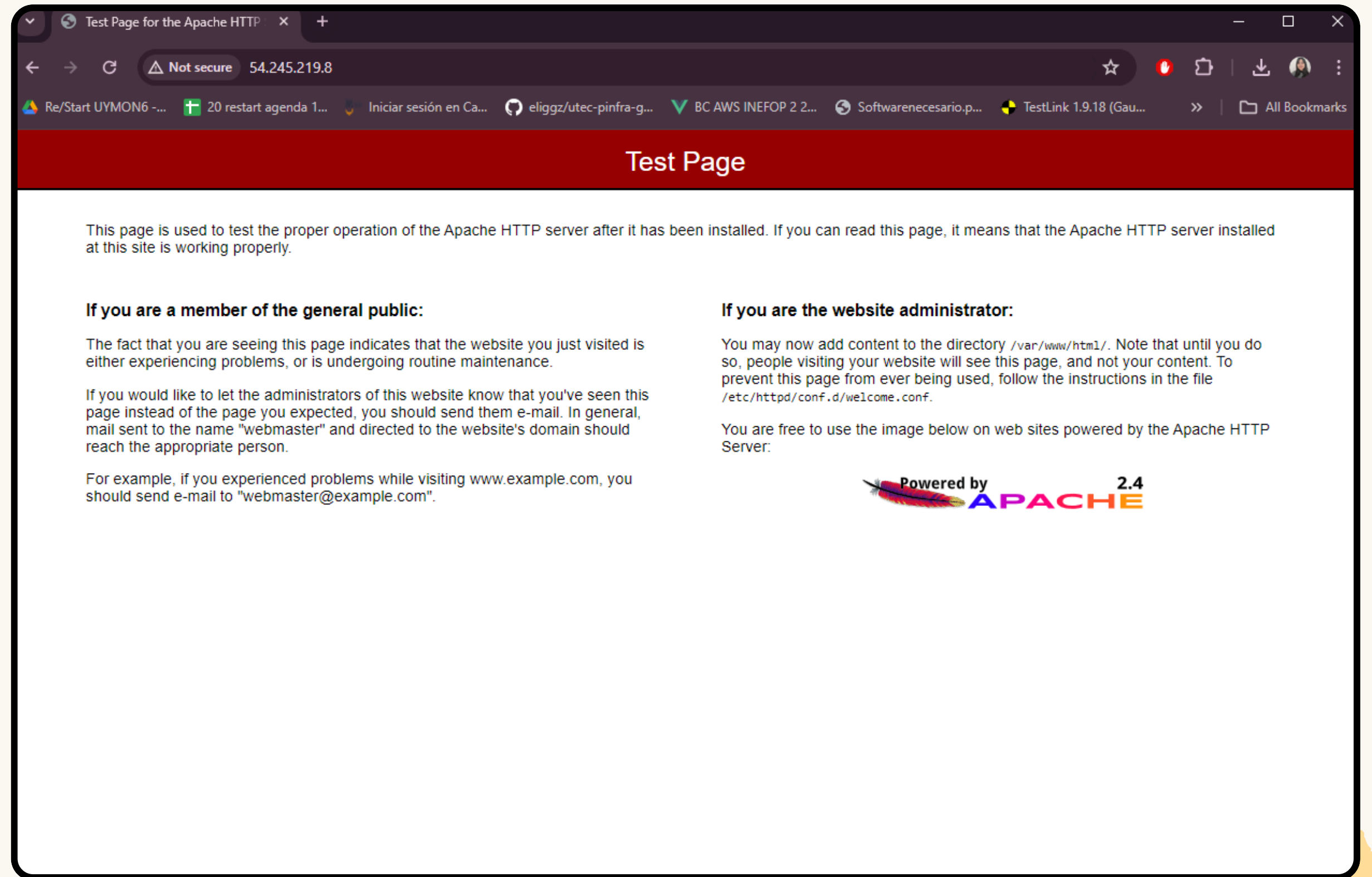
## **TAREA 3:**

## **MONITOREAR LA INSTANCIA EC2**

- En este ejercicio utilizaremos ciertos comandos de Linux para monitorear la instancia de EC2.
- Entraremos a la consola de AWS y nos logearemos dentro de CloudWatch para ver como este servicio puede proveernos con mucha información para monitorear la instancia.

# PÁGINA WEB

**Una vez que el  
httpd se esté  
ejecutando  
debemos verificar  
que funciona  
correctamente  
abriendo una  
pestaña nueva,  
colocando la IP y  
realizando la  
conexión a través  
de http**





# DETENER

Para detener el servicio, ingresamos el comando `sudo systemctl stop httpd.service`

```
ec2-user@ip-10-0-10-144:~  
[ec2-user@ip-10-0-10-144 ~]$ sudo systemctl stop httpd.service  
[ec2-user@ip-10-0-10-144 ~]$ sudo systemctl status httpd.service  
● httpd.service - The Apache HTTP Server  
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)  
   Active: inactive (dead)  
     Docs: man:httpd.service(8)  
  
Apr 26 21:30:37 ip-10-0-10-144.us-west-2.compute.internal systemd[1]: Starting The Apache HTTP Server:  
Apr 26 21:30:37 ip-10-0-10-144.us-west-2.compute.internal systemd[1]: Started The Apache HTTP Server:  
Apr 26 21:33:41 ip-10-0-10-144.us-west-2.compute.internal systemd[1]: Stopping The Apache HTTP Server:  
Apr 26 21:33:42 ip-10-0-10-144.us-west-2.compute.internal systemd[1]: Stopped The Apache HTTP Server:  
Hint: Some lines were ellipsized, use -l to show in full.  
[ec2-user@ip-10-0-10-144 ~]$
```

**TOP**

Mostramos la  
lista de procesos  
en ejecución  
ingresando el  
comando top

```
ec2-user@ip-10-0-10-144:~  
top - 21:35:09 up 8 min,  1 user,  load average: 0.00, 0.06, 0.06  
Tasks:  87 total,   1 running,  47 sleeping,   0 stopped,   0 zombie  
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni, 99.8 id,  0.0 wa,  0.0 hi,  0.0 si,  0.2 st  
KiB Mem :  966808 total,  440016 free,   77140 used,  449652 buff/cache  
KiB Swap:   0 total,   0 free,   0 used.  747316 avail Mem  
  
  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND  
    1 root        20   0 123616   5560  3952  S   0.0   0.6   0:00.99 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   0:00.04 kworker/u4:0  
    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   0:00.01 ksoftirqd/0  
    8 root        20   0     0     0     0  I   0.0   0.0   0:00.06 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  
   13 root        20   0     0     0     0  S   0.0   0.0   0:00.01 cpuhp/1  
   14 root        rt    0     0     0     0  S   0.0   0.0   0:00.00 watchdog/1  
   15 root        rt    0     0     0     0  S   0.0   0.0   0:00.22 migration/1  
   16 root        20   0     0     0     0  S   0.0   0.0   0:00.02 ksoftirqd/1  
   18 root         0 -20     0     0     0  I   0.0   0.0   0:00.00 kworker/1:0H  
   20 root        20   0     0     0     0  S   0.0   0.0   0:00.00 kdevtmpfs
```

# SCRIPT

Ejecutamos el script  
stress.sh que simula  
una carga de trabajo  
pesada en la instancia  
EC2. con el comando  
./stress.sh & top

ec2-user@ip-10-0-10-144:~

```
[ec2-user@ip-10-0-10-144 ~]$ ./stress.sh & top
```

```
[1] 2663
```

```
stress: info: [2665] dispatching hogs: 8 cpu, 4 io, 2 vm, 0 hdd
```

```
top - 21:37:04 up 10 min, 1 user, load average: 2.15, 0.50, 0.20
```

```
Tasks: 102 total, 15 running, 49 sleeping, 0 stopped, 0 zombie
```

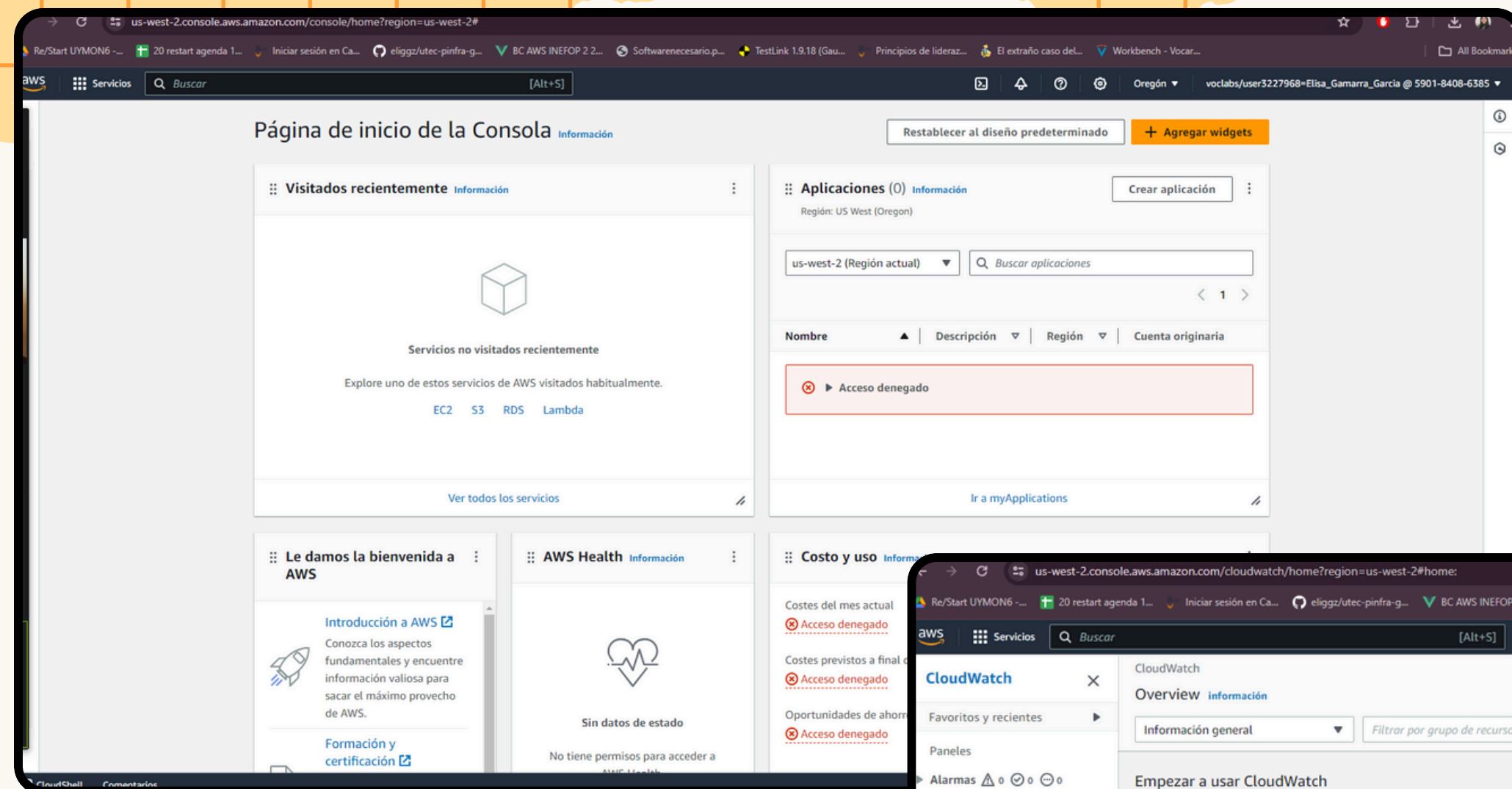
```
%Cpu(s): 58.1 us, 35.9 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.0 si, 6.0 st
```

```
KiB Mem : 966808 total, 401104 free, 116048 used, 449656 buff/cache
```

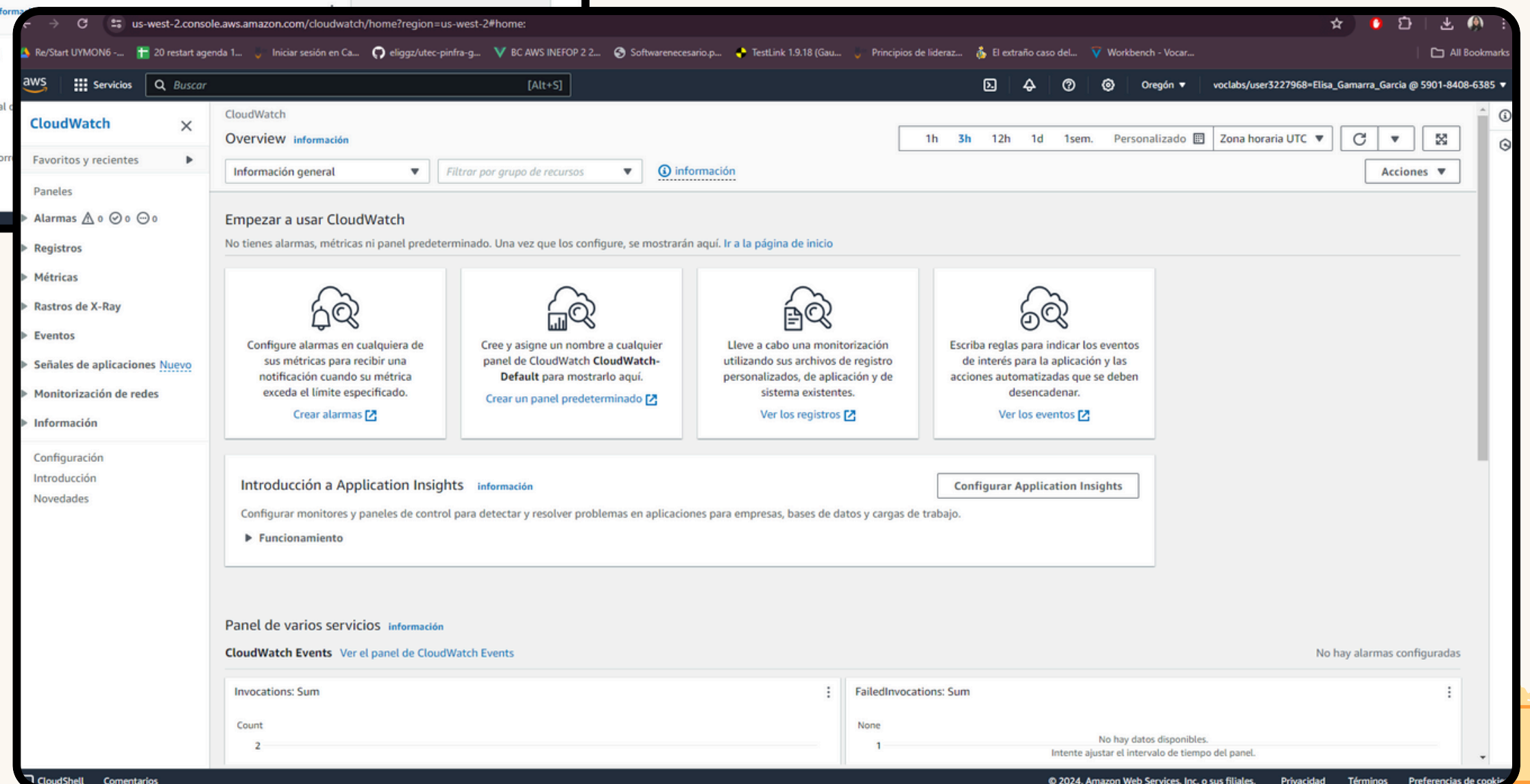
```
KiB Swap: 0 total, 0 free, 0 used. 708408 avail Mem
```

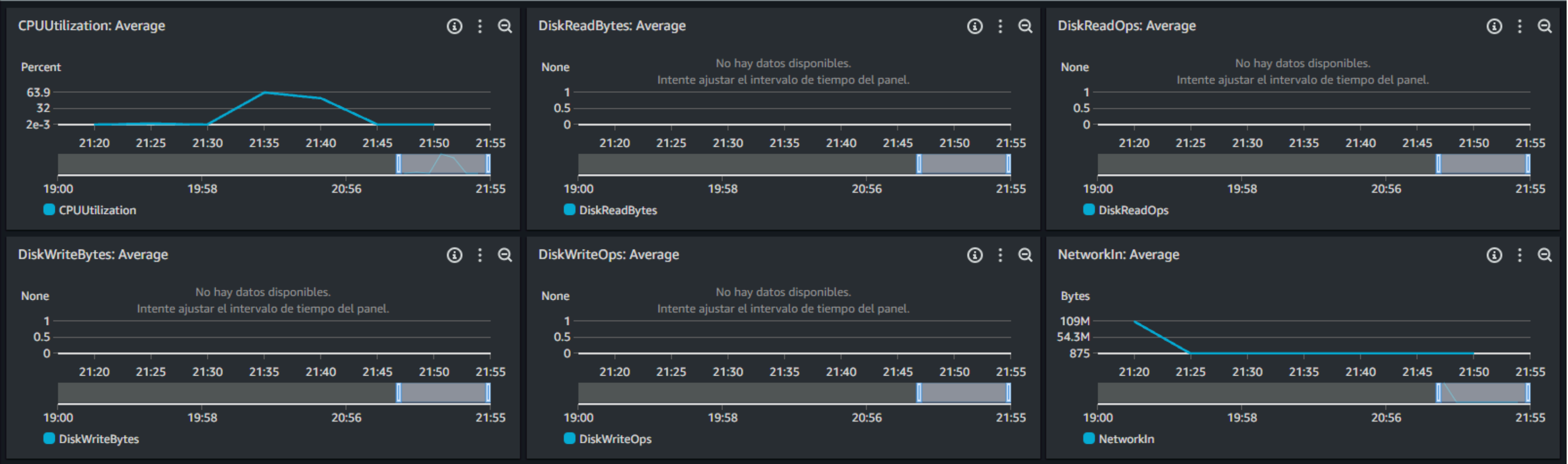
| PID  | USER     | PR | NI  | VIRT   | RES   | SHR  | S | %CPU | %MEM | TIME+   | COMMAND      |
|------|----------|----|-----|--------|-------|------|---|------|------|---------|--------------|
| 2667 | ec2-user | 20 | 0   | 7580   | 100   | 0    | R | 13.6 | 0.0  | 0:01.15 | stress       |
| 2668 | ec2-user | 20 | 0   | 138656 | 18852 | 276  | R | 13.6 | 1.9  | 0:01.15 | stress       |
| 2669 | ec2-user | 20 | 0   | 7580   | 100   | 0    | R | 13.6 | 0.0  | 0:01.16 | stress       |
| 2670 | ec2-user | 20 | 0   | 7580   | 100   | 0    | R | 13.6 | 0.0  | 0:01.15 | stress       |
| 2673 | ec2-user | 20 | 0   | 7580   | 100   | 0    | R | 13.6 | 0.0  | 0:01.16 | stress       |
| 2677 | ec2-user | 20 | 0   | 7580   | 100   | 0    | R | 13.6 | 0.0  | 0:01.16 | stress       |
| 2679 | ec2-user | 20 | 0   | 7580   | 100   | 0    | R | 13.6 | 0.0  | 0:01.16 | stress       |
| 2666 | ec2-user | 20 | 0   | 7580   | 100   | 0    | R | 13.3 | 0.0  | 0:01.15 | stress       |
| 2671 | ec2-user | 20 | 0   | 138656 | 18324 | 276  | R | 13.3 | 1.9  | 0:01.15 | stress       |
| 2672 | ec2-user | 20 | 0   | 7580   | 100   | 0    | R | 13.3 | 0.0  | 0:01.15 | stress       |
| 2674 | ec2-user | 20 | 0   | 7580   | 100   | 0    | R | 13.3 | 0.0  | 0:01.15 | stress       |
| 2675 | ec2-user | 20 | 0   | 7580   | 100   | 0    | R | 13.3 | 0.0  | 0:01.15 | stress       |
| 2676 | ec2-user | 20 | 0   | 7580   | 100   | 0    | R | 13.3 | 0.0  | 0:01.15 | stress       |
| 2678 | ec2-user | 20 | 0   | 7580   | 100   | 0    | R | 13.3 | 0.0  | 0:01.26 | stress       |
| 2664 | ec2-user | 20 | 0   | 168912 | 4404  | 3764 | R | 0.3  | 0.5  | 0:00.01 | top          |
| 1    | root     | 20 | 0   | 123616 | 5560  | 3952 | S | 0.0  | 0.6  | 0:00.99 | 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  | 0:00.04 | kworker/u4:0 |
| 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  | 0:00.01 | ksoftirqd/0  |
| 8    | root     | 20 | 0   | 0      | 0     | 0    | I | 0.0  | 0.0  | 0:00.07 | 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      |
| 13   | root     | 20 | 0   | 0      | 0     | 0    | S | 0.0  | 0.0  | 0:00.01 | cpuhp/1      |
| 14   | root     | rt | 0   | 0      | 0     | 0    | S | 0.0  | 0.0  | 0:00.00 | watchdog/1   |
| 15   | root     | rt | 0   | 0      | 0     | 0    | S | 0.0  | 0.0  | 0:00.22 | migration/1  |
| 16   | root     | 20 | 0   | 0      | 0     | 0    | S | 0.0  | 0.0  | 0:00.02 | ksoftirqd/1  |
| 18   | root     | 0  | -20 | 0      | 0     | 0    | I | 0.0  | 0.0  | 0:00.00 | kworker/1:0H |

Hacemos click en el botón AWS.  
Esto mostrara la AWS Management Console



En la barra de búsqueda de la parte superior ingresamos "CloudWatch" y hacemos clic en el enlace de CloudWatch







# CONCLUSIONES



- Completamos con éxito los objetivos del laboratorio.
- Aprendimos a monitorear instancias de EC2, ya sea desde los comandos de Linux como así de CloudWatch.

# ¡MUCHAS GRACIAS!

