



# 11º

## Lab - AWS re/Start Adm Servicios Monitoreo





# Comprobando el estado del Servicio **httpd**

A continuación, se tratarán los siguientes temas:

- Verificar el estado del servicio `httpd` para asegurarse de que se está ejecutando y que puede realizar una conexión `http` con la dirección IP del host local.
- También aprenderá a monitorear la instancia EC2 de Amazon Linux 2
- Utilizar el comando `top` de Linux
- Utilizar AWS CloudWatch

Nota: **httpd** es el servicio para el servido *http Apache* (servidor web donde se ejecutan distintos sitios web conocidos)

# Tarea 01

---



Para comprobar el estado del servicio *httpd*, nos es útil el comando **systemctl status**

Notamos que inicialmente está inactivo

```
[ec2-user@ip-10-0-10-85 ~]$ 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: inactive (dead)
     Docs: man:httpd.service(8)
[ec2-user@ip-10-0-10-85 ~]$
```

Para activarlo usamos el comando **systemctl start**

```
[ec2-user@ip-10-0-10-85 ~]$ sudo systemctl start httpd.service
[ec2-user@ip-10-0-10-85 ~]$ 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 Sun 2023-12-31 03:18:42 UTC; 12s ago
     Docs: man:httpd.service(8)
  Main PID: 2599 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes se
served/sec: 0 B/sec"
    CGroup: /system.slice/httpd.service
            └─2599 /usr/sbin/httpd -DFOREGROUND
              └─2600 /usr/sbin/httpd -DFOREGROUND
                └─2602 /usr/sbin/httpd -DFOREGROUND
                  └─2607 /usr/sbin/httpd -DFOREGROUND
                    └─2609 /usr/sbin/httpd -DFOREGROUND
                      └─2614 /usr/sbin/httpd -DFOREGROUND

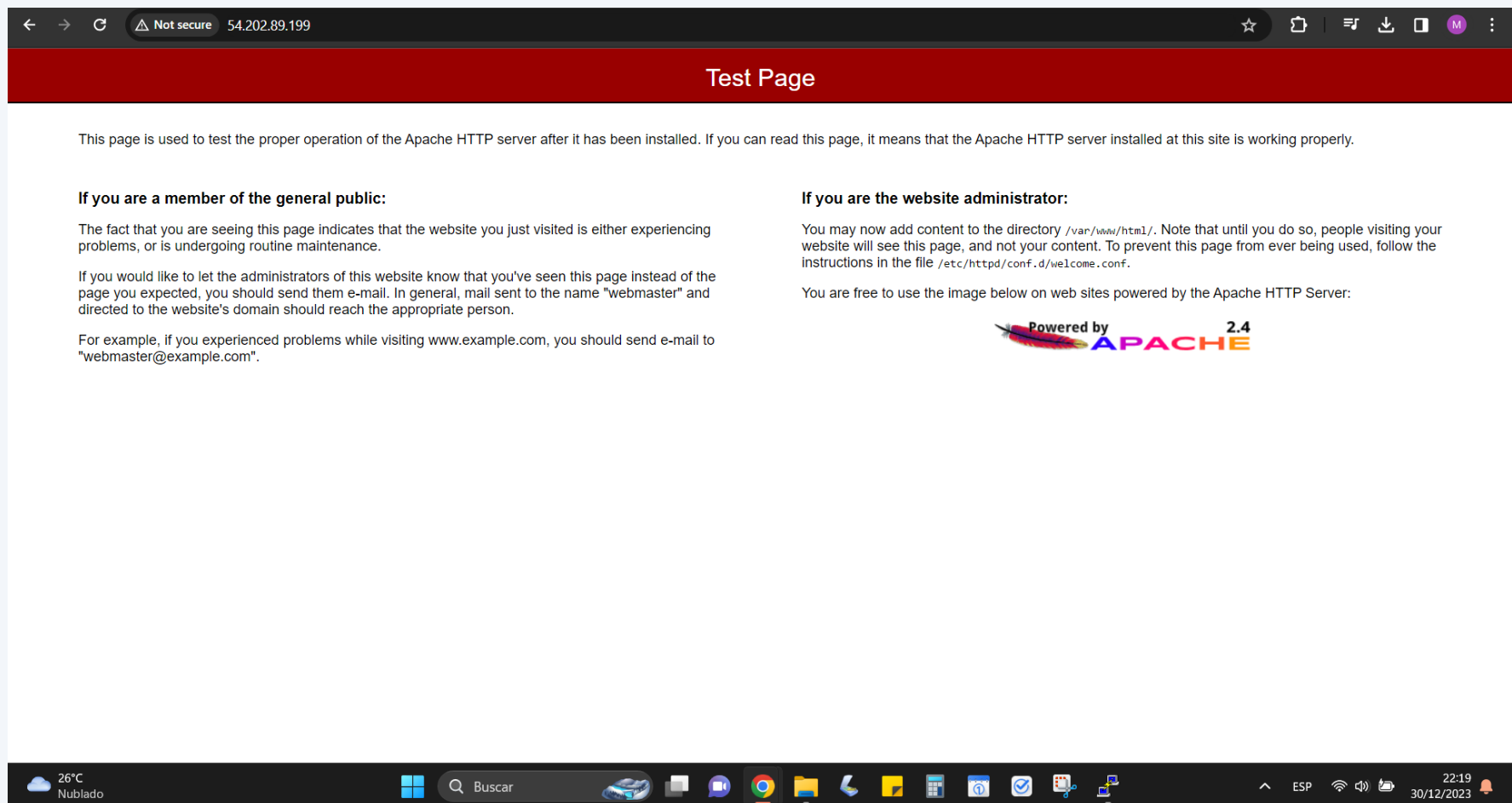
Dec 31 03:18:42 ip-10-0-10-85.us-west-2.compute.internal systemd[1]: Starting...
Dec 31 03:18:42 ip-10-0-10-85.us-west-2.compute.internal systemd[1]: Started ...
Hint: Some lines were ellipsized, use -l to show in full.
```

# Tarea 01

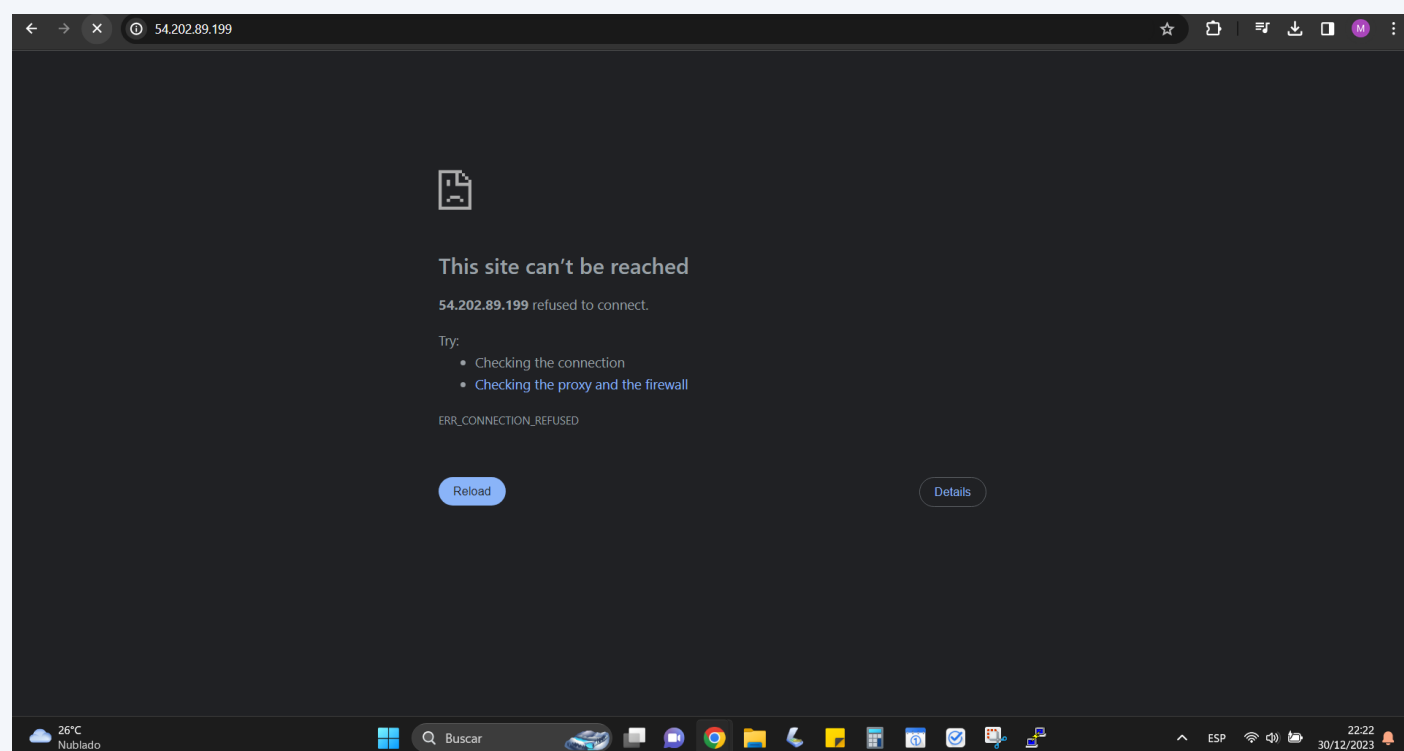
---



Notamos que el servicio ya está desplegado



Para detener el servicio, usamos **systemctl stop**  
Después de ello, el servidor se cayó



## Tarea 02

---



# Monitoreo de una Instancia de EC2

Podemos ver cómo cuánto están consumiendo los procesos que se están ejecutando en la máquina virtual, con el comando **top**

```
top - 03:30:19 up 18 min, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 87 total, 1 running, 47 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 966816 total, 426088 free, 76964 used, 463764 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 747348 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2683	ec2-user	20	0	168812	4328	3812	R	0.3	0.4	0:00.04	top
1	root	20	0	123624	5528	3880	S	0.0	0.6	0:01.30	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:+
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu +
7	root	20	0	0	0	0	S	0.0	0.0	0:00.02	ksoftirqd/0
8	root	20	0	0	0	0	I	0.0	0.0	0:00.05	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	0:00.20	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:+
20	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs

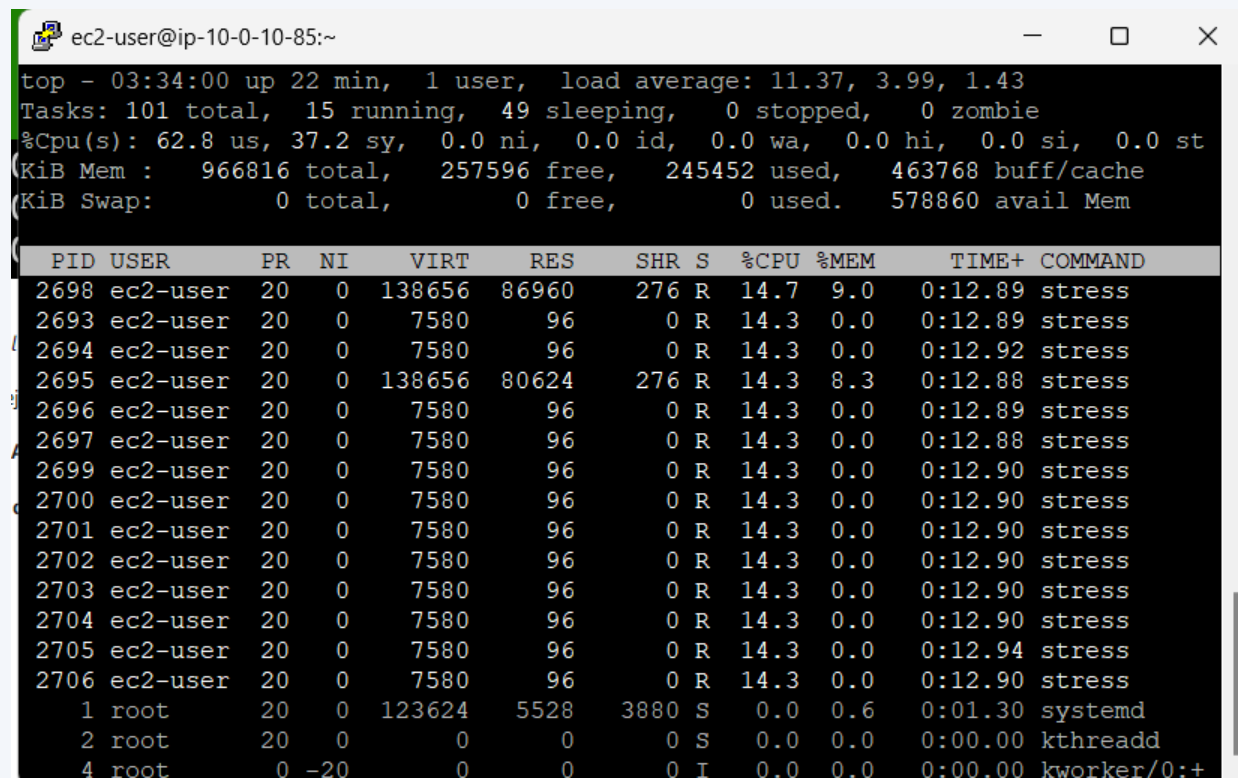
Recuerda salir con **q+ENTER**



# Tarea 02



Podemos simular *una carga de trabajo pesada* con el comando **./stress.sh & top**



Donde notamos que el porcentaje de uso supera el 60%. Este pico también lo podemos notar en el servicio de **AWS CloudWatch**. El cual nos permite monitorear en tiempo real nuestras aplicaciones, asimismo, podemos establecer alarmas o *triggers* de eventos

