Hardening SSH en AWS EC2 con Ubuntu

Marcos García Rodríguez
25 de septiembre de 2025
2º DAW

1. Introducción teórica

El SSH (Secure Shell) es una herramienta que usamos para conectarnos a un servidor de forma remota y segura. Gracias a ella podemos administrar el sistema, ejecutar comandos o transferir archivos sin tener que estar físicamente frente al equipo. SSH cifra toda la comunicación, lo que evita que alguien pueda ver o alterar los datos que enviamos o recibimos.

Aun así, dejar la configuración por defecto puede ser peligroso.

- Acceso root directo: Permitir que el usuario root se conecte por SSH es arriesgado, ya que si alguien logra adivinar la contraseña, tendrá control total del servidor. Es mejor desactivar ese acceso y usar un usuario normal con permisos de sudo.

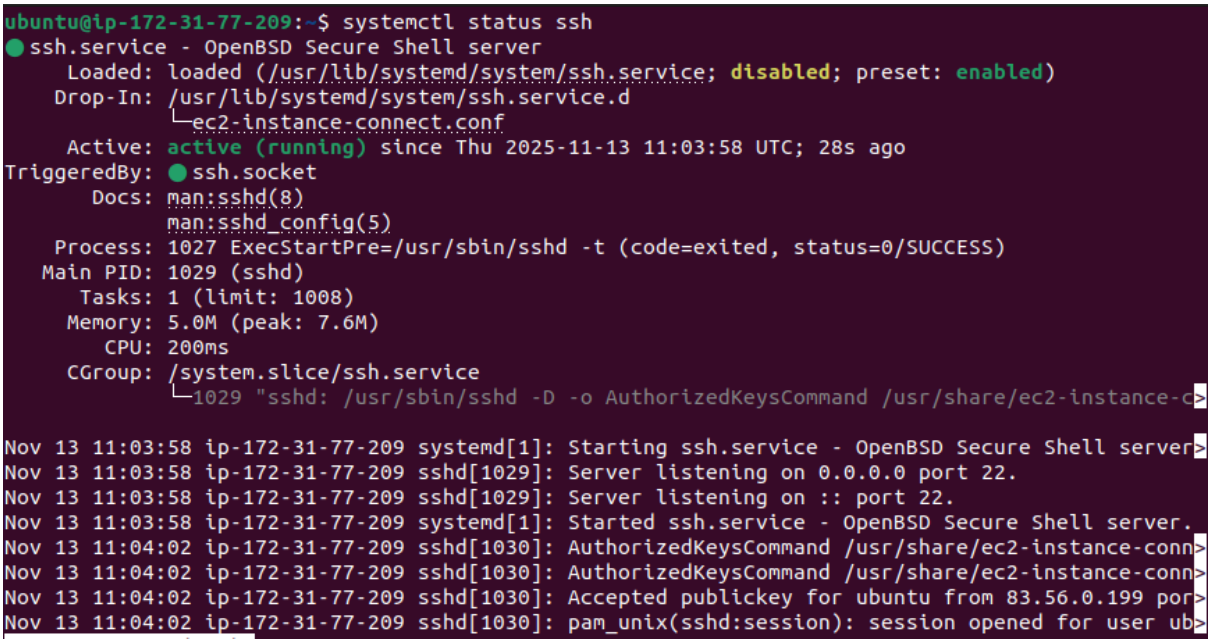
- Contraseñas débiles: Si usamos contraseñas fáciles o repetidas, los atacantes pueden adivinarlas con programas automáticos. Lo más seguro es usar claves públicas y privadas o contraseñas largas y complejas.

- Sesiones inactivas: Si dejamos una sesión SSH abierta y sin uso, alguien podría aprovecharla si tiene acceso al equipo. Conviene configurar un tiempo de cierre automático cuando no haya actividad.

En resumen, SSH es una herramienta clave para la administración remota, pero solo será realmente segura si se configura bien y se aplican buenas prácticas básicas de seguridad. Esto ayuda a proteger el servidor y a mantener los datos a salvo.

2. Preparación del entorno

Accedemos a la instancia y comprobamos que SSH está activo.



Comprobamos el estado del firewall, si está desactivado, lo habilitamos.

```
ubuntu@ip-172-31-77-209:~$ sudo ufw status
Status: inactive
ubuntu@ip-172-31-77-209:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
ubuntu@ip-172-31-77-209:~$ sudo ufw status
Status: active
ubuntu@ip-172-31-77-209:~$
```

Agregamos el puerto 22/tcp

```
ubuntu@ip-172-31-77-209:~$ sudo ufw allow 22/tcp
Rule added
Rule added (v6)
```

3. Incrementar la seguridad del servicio SSH

Antes de comenzar, hacemos una copia de seguridad del archivo de configuración.

```
ubuntu@ip-172-31-77-209:~$ sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak
```

- Desactivar el inicio de sesión directo del usuario root.

```
PermitRootLogin no
DenyUsers root
```

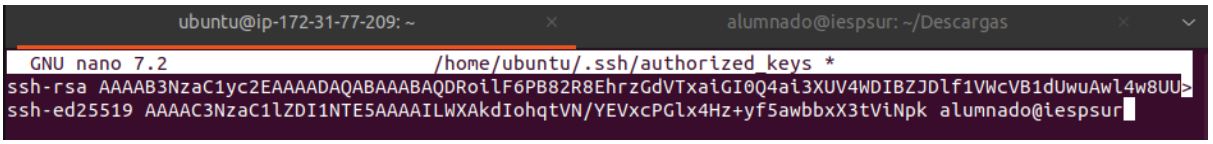
- Deshabilitar la autenticación mediante contraseña y habilitar la autenticación por clave pública.

```
PubkeyAuthentication yes
PasswordAuthentication no
ChallengeResponseAuthentication no
```

- Generar y registrar claves SSH seguras desde el cliente local.

```
alumnado@iespsur:~/Descargas$ ssh-keygen -t ed25519 -C alumnado@iespsur
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/alumnado/.ssh/id_ed25519): clavePractica3
clavePractica3 already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in clavePractica3
Your public key has been saved in clavePractica3.pub
The key fingerprint is:
SHA256:ThoJhW0Jskr2Vb4w3apw2OYaECYd6KSbt4OqhFPwHgQ alumnado@iespsur
The key's randomart image is:
+--[ED25519 256]--+
|E.o .+.o         |
|.+ +..B .        |
|*=+ .= o .       |
|**o +.o.o        |
|.+++ +ooS        |
|+oo.= .=         |
|o+.o o. .        |
|o.o o            |
|=  o             |
+----[SHA256]-----+
```

Copiamos la clave con un CAT y la pegamos en el archivo /.ssh/authorized_keys de la instancia.



- Limitar el número de intentos fallidos y el tiempo de espera en el inicio de sesión.
  Modificamos el archivo para que quede de la siguiente forma:

```
LoginGraceTime 30m
PermitRootLogin no
DenyUsers root
#StrictModes yes
MaxAuthTries 5
MaxSessions 10
```

- Restringir los usuarios autorizados a conectarse mediante SSH.

```
# Authentication:
AllowUsers ubuntu alumnado
```

- Desactivar reenvíos y el uso de X11 para reducir la superficie de ataque.
  Ajustamos para que quede de la siguiente forma:

```
#AllowAgentForwarding yes
AllowTcpForwarding no
#GatewayPorts no
X11Forwarding no

#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
PrintMotd no
#PrintLastLog yes
#TCPKeepAlive yes
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
#UseDNS no
#PidFile /run/sshd.pid
#MaxStartups 10:30:100
PermitTunnel no
#ChrootDirectory none
#VersionAddendum none
```

- Cambiar el puerto SSH por defecto y ajustar las reglas del firewall.
  Cambiamos al puerto 2222

```
#
Port 2222
```

  Agregamos el nuevo puerto a las reglas del firewall

```
ubuntu@ip-172-31-77-209:~$ sudo ufw allow 2222/tcp
Rule added
Rule added (v6)
ubuntu@ip-172-31-77-209:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW IN    Anywhere
2222/tcp                   ALLOW IN    Anywhere
22/tcp (v6)                ALLOW IN    Anywhere (v6)
2222/tcp (v6)              ALLOW IN    Anywhere (v6)
```

Agregamos también en AWS

- Reiniciar el servicio y comprobar el acceso seguro.
  Finalmente, reiniciamos el servicio con el comando
  systemctl restart ssh
  Y comprobamos su estado.

```
ubuntu@ip-172-31-77-209:~$ sudo systemctl restart ssh
ubuntu@ip-172-31-77-209:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
    Drop-In: /usr/lib/systemd/system/ssh.service.d
             └─ec2-instance-connect.conf
     Active: active (running) since Thu 2025-11-13 11:27:40 UTC; 8s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 1581 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 1583 (sshd)
      Tasks: 1 (limit: 1008)
     Memory: 1.2M (peak: 1.4M)
        CPU: 24ms
     CGroup: /system.slice/ssh.service
             └─1583 "sshd: /usr/sbin/sshd -D -o AuthorizedKeysCommand /usr/share/ec2-instance-connect/eic_run_a>

Nov 13 11:27:40 ip-172-31-77-209 systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Nov 13 11:27:40 ip-172-31-77-209 sshd[1583]: Server listening on 0.0.0.0 port 22.
Nov 13 11:27:40 ip-172-31-77-209 sshd[1583]: Server listening on :: port 22.
Nov 13 11:27:40 ip-172-31-77-209 systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
lines 1-20/20 (END)
```

4. Auditoría y comprobación

- Verificar la configuración activa del servicio SSH.

sudo sshd -T | grep -E "port|permitrootlogin|passwordauthentication|pubkeyauthentication"

```
ubuntu@ip-172-31-77-209:~$ sudo sshd -T | grep -E "port|permitrootlogin|passwordauthentication|pubkeyauthentication"
port 2222
permitrootlogin no
pubkeyauthentication yes
passwordauthentication no
gatewayports no
```

- Probar la conexión desde un cliente remoto utilizando clave pública.

```
alumnado@iespsur:~/Descargas$ ssh -i "daweb2025.pem" -p 2222 ubuntu@ec2-34-204-189-105.compute-1.amazonaws.com
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Thu Nov 13 11:52:09 UTC 2025

  System load:  0.0               Temperature:           -273.1 C
  Usage of /:   26.4% of 6.71GB   Processes:             110
  Memory usage: 24%               Users logged in:       0
  Swap usage:   0%                IPv4 address for ens5: 172.31.77.209


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Thu Nov 13 11:46:34 2025 from 83.56.0.199
```

- Revisar los logs de acceso y posibles intentos fallidos.

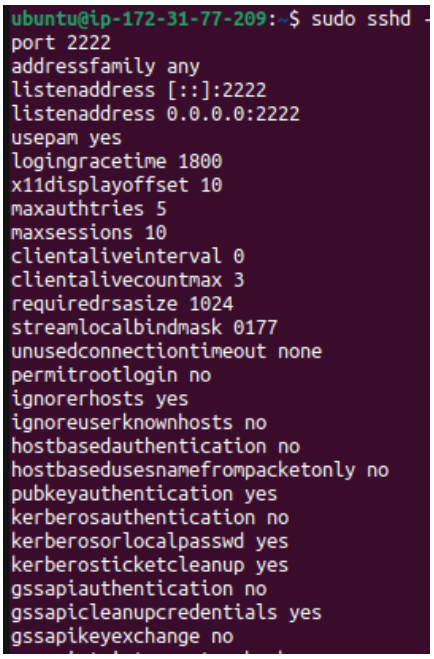
```
ubuntu@ip-172-31-77-209:~$ sudo journalctl -u ssh
Nov 13 11:03:58 ip-172-31-77-209 systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Nov 13 11:03:58 ip-172-31-77-209 sshd[1029]: Server listening on 0.0.0.0 port 22.
Nov 13 11:03:58 ip-172-31-77-209 sshd[1029]: Server listening on :: port 22.
```

```
ubuntu@ip-172-31-77-209:~$ sudo cat /var/log/auth.log | grep "sshd"
2025-11-13T11:03:58.253441+00:00 ip-172-31-77-209 sshd[1029]: Server listening on 0.0.0.0 port 22.
2025-11-13T11:03:58.253555+00:00 ip-172-31-77-209 sshd[1029]: Server listening on :: port 22.
2025-11-13T11:04:02.112004+00:00 ip-172-31-77-209 sshd[1030]: AuthorizedKeysCommand /usr/share/ec2-instance-connect/eic_run_authorized_
keys ubuntu SHA256:b5XZpZ9JsLD1wn/DrO+C+nTnHs89mPIJgOY6BUstKs0 failed, status 22
```

- Comprobar la exposición del puerto SSH mediante herramientas de auditoría como nmap.

```
ubuntu@ip-172-31-77-209:~$ netstat -ltn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:2222            0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.54:53           0.0.0.0:*               LISTEN
tcp6       0      0 :::2222                 :::*                    LISTEN
```

- Analizar las sugerencias del comando sshd -T para validar la configuración.



5. Reflexión final y entregable

Medidas de seguridad aplicadas y su justificación técnica

- Desactivación del inicio de sesión root:
  Se configuró `PermitRootLogin no` para impedir accesos directos con el usuario root, reduciendo la superficie de ataque en la cuenta con privilegios máximos, minimizando riesgos de intrusión.
- Autenticación por clave pública y deshabilitar contraseña:
  Con `PasswordAuthentication no` y `PubkeyAuthentication yes`, se evita el acceso con contraseñas vulnerables y se asegura la autenticación robusta mediante llaves criptográficas.
- Limitación de intentos fallidos y tiempo de espera:
  Los Parámetros `MaxAuthTries 3` y `LoginGraceTime 30` previenen ataques de fuerza bruta al limitar el número de intentos y tiempo para autenticarse.
- Restricción de usuarios autorizados:
  Solo el usuario específico autorizado puede conectarse (`AllowUsers usuario`), reduciendo riesgos por accesos no autorizados.
- Desactivación de reenvíos y X11:
  Al poner `AllowTcpForwarding no` y `X11Forwarding no` se disminuye la superficie de ataque indirecta que puede ser usada para saltar entre redes o explotar túneles.
- Cambio del puerto SSH a uno no estándar (ej. 2222):
  Reduce notablemente escaneos automatizados y ataques masivos dirigidos al puerto estándar 22, haciendo los intentos de acceso más difíciles.

- Configuración y control de reglas de firewall/grupo de seguridad AWS:
  Solo el puerto SSH configurado permite tráfico desde IPs autorizadas, restringiendo acceso externo.

Estas medidas mejoran la autenticidad, confidencialidad y disponibilidad del servicio SSH, alineadas con las mejores prácticas y recomendaciones reconocidas en AWS y sistemas Linux.

Pruebas realizadas para verificar el acceso seguro

- Estado y parámetros activos del servicio SSH comprobados con `sudo systemctl status sshd` y `sshd -T`.
- Conexión exitosa por SSH desde cliente remoto usando autenticación por clave pública y puerto personalizado.
- Monitoreo en tiempo real de logs (`/var/log/auth.log`) para detectar accesos y bloqueos de intentos no autorizados.
- Escaneo con `nmap` para asegurar que solo el puerto definido para SSH está abierto y accesible.
- Verificación de configuración acorde a la política esperada por medio de herramientas integradas (`sshd -T`).

Tres mejoras adicionales para un entorno de producción real

1. Implementación de autenticación multifactor (MFA):
   Añade una capa extra de seguridad para validar identidades más allá de la clave SSH, fundamental contra accesos comprometidos.
2. Uso de AWS Session Manager o bastión (jump host) sin abrir puertos SSH públicos:
   Facilita acceso seguro sin exponer puertos a Internet, evitando ataques externos y permitiendo control granular mediante IAM.
3. Sistema de monitorización con alertas automatizadas y rotación periódica de claves SSH:
   Mejora la detección temprana de amenazas y reduce el riesgo que supone la permanencia de claves comprometidas en el sistema.