
Preguntas Frecuentes - FAQ

Implantación de Aplicaciones Web

José Juan Sánchez Hernández

Curso 2023/2024

Índice

1 Preguntas Frecuentes - FAQ	1
1.1 Error al conectar por SSH	1
1.1.1 Solución	1
1.2 Cómo ejecutar procesos en segundo plano de forma permanente	2
1.3 Cómo comprobar si un proceso está en ejecución	2
1.4 Cómo matar un proceso en Linux	2
2 Licencia	4

Índice de figuras

Índice de cuadros

1 Preguntas Frecuentes - FAQ

1.1 Error al conectar por SSH

```
1 @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
2 @      WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!      @
3 @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
4 IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
5 Someone could be eavesdropping on you right now (man-in-the-middle attack)!
6 It is also possible that a host key has just been changed.
7 The fingerprint for the ECDSA key sent by the remote host is
8 SHA256:Rcpw9NAYK0erfK1wOUaTXFAqnqN2gmUqe5EbrokL9YU.
9 Please contact your system administrator.
10 Add correct host key in /Users/josejuan/.ssh/known_hosts to get rid of this
    message.
11 Offending ECDSA key in /Users/josejuan/.ssh/known_hosts:36
12 ECDSA host key for 192.168.0.169 has changed and you have requested strict
    checking.
13 Host key verification failed.
```

1.1.1 Solución

Este mensaje de error indica que el `fingerprint` de la clave ECDSA (Elliptic Curve Digital Signature Algorithm) enviado por el host remoto, en este caso el host `192.168.0.169`, no coincide con el `fingerprint` que actualmente está almacenado en el archivo `~/.ssh/known_hosts` de nuestro equipo. El archivo `~/.ssh/known_hosts` contiene los valores de los `fingerprints` de las máquinas a las que nos hemos conectado alguna vez por SSH.

Este error puede suceder porque el host remoto al que queremos conectarnos ha modificado su clave ECDSA o porque la dirección IP pública del host remoto ahora está asignada a otro host diferente.

Para solucionar este problema, puede hacer los siguientes pasos:

1. Edita el archivo `~/.ssh/known_hosts` de tu máquina local tu editor de texto favorito.
2. **Busca la línea** que hace referencia a la dirección IP del host remoto al que quieres conectarte y **elimínala** del archivo. Por ejemplo, en nuestro caso buscaríamos la línea que contiene la dirección IP `192.168.0.169` y la eliminaríamos.

```
1 192.168.0.169 ecdsa-sha2-nistp256 Rcpw9NAYK0erfK1wOUaTXFAqnqN2gmUqe5EbrokL9YU
```

3. Guarda los cambios y cierra el editor de texto.

Una vez hecho esto, cuando nos conectemos por SSH al host remoto se añadirá una nueva línea en el archivo `~/.ssh/known_hosts` con el nuevo valor del `fingerprint` de la clave ECDSA.

1.2 Cómo ejecutar procesos en segundo plano de forma permanente

Tiene que tener en cuenta que si ha iniciado el proceso en segundo plano con el carácter ampersand (&) al final del comando, el proceso finalizará cuando se cierre la sesión SSH.

Para evitar que el proceso finalice la sesión SSH podemos iniciarlo con alguna de las siguientes utilidades: `nohup`, `screen` o `tmux`.

- `nohup`: Esta utilidad ejecuta el comando que recibe como parámetro y hace que ignore las señales `SIGHUP`, que son las señales que se envían a un proceso cuando el terminal que los controla se cierra.

Ejemplo:

```
1 nohup goaccess /var/log/apache2/access.log -o /var/www/html/stats/index.html --log-format=COMBINED --real-time-html &
```

- `screen`: Esta utilidad permite iniciar una sesión y tener varios terminales virtuales en ella. Los procesos que se ejecutan en estos terminales virtuales no finalizarán al cerrar el terminal virtual.

Ejemplo:

```
1 screen -dmL goaccess /var/log/apache2/access.log -o /var/www/html/stats/index.html --log-format=COMBINED --real-time-html
```

Las opciones que se han utilizado en el ejemplo anterior son: `-d` inicia una sesión y automáticamente se desconecta de ella, `-m` fuerza la creación de una nueva sesión y `-L` habilita el log.

- `tmux`: Es una alternativa más reciente que `screen`. Realiza la misma funcionalidad que `screen`.

1.3 Cómo comprobar si un proceso está en ejecución

Podemos comprobar que el proceso se está ejecutando de forma correcta haciendo un listado de todos los procesos que están en ejecución con `ps aux` y buscando en la lista el nombre del proceso con `grep mysqld`.

```
1 ps aux | grep mysqld
```

1.4 Cómo matar un proceso en Linux

- `kill`: Para matar un proceso a partir del nombre del proceso.

Ejemplo: Matar el proceso `mysqld`

```
1 pkill mysqld
```

- `kill`: Para matar un proceso a partir del PID (Process Identifier).

Ejemplo: Matar el proceso con PID igual a 1815.

```
1 kill -9 1815
```

2 Licencia

Esta página forma parte del curso Implantación de Aplicaciones Web de José Juan Sánchez Hernández y su contenido se distribuye bajo una licencia Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.