

# Administrar un servidor con SSH/Bash

## Introducción

Bash es el programa que es ejecutado cuando se conecta en ssh a su servidor. Puede elegir otro programa de shell como sh, ash, etc.

```
[root@nsxxxx.net root]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

Puede encontrar más información en Google : <http://www.google.es/search?q=bash>

Para conectarse en SSH a su servidor dedicado puede consultar nuestra guía : [SshSobreServidorDedicado](#)

## Astucias Bash

**bash completa la línea de comando con la tecla [tab]**

por ejemplo:

```
[root@kazik root]# ls -l a
[tab]
[tab]
a2ps-4.13-3.i386.rpm align.ps analog apaon
alaska.mp3 anaconda-ks.cfg apaoff apare
[root@kazik root]# ls -l a
[root@kazik root]# ls -l an
[tab]
[root@kazik root]# ls -l ana
[tab]
[tab]
anaconda-ks.cfg analog
[root@kazik root]# ls -l anal
[tab]
[root@kazik root]# ls -l analog
```

**bash acepta los alias**

Si por ejemplo teclaa a menudo `ls -l` puede crear un alias y escribirlo directamente:

```
[root@kazik root]# alias ll="ls -l --color
```

Ahora si teclea **ll** será equivalente a `ls -l --color`.

**bash tiene fichero de inicio [bashrc]**

Puede añadir una línea al fichero `.bashrc` para que la próxima vez también pueda utilizarlo:

```
[root@kazik root]# cat >> .bashrc
alias ll="ls -l --color"
[ctrl-d]
[root@kazik root]#
```

### Vigilar el sistema

Cuando nos conectamos al servidor, hay que verificar rápidamente su estado. A fuerza de hacerlo, aprendemos a apreciar un resultado visual.

Si este resultado visual no es el mismo que de costumbre, sabemos rápidamente que hay un problema.

El comando **w** permite ver rápidamente el uptime, la carga del servidor y el número de personas conectadas.

```
[root@ns root]# w

3:40am up 99 days, 9:08, 1 user, load average: 0.00, 0.00, 0.00

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
root pts/0 ping.ovh.net 3:40am 0.00s 0.13s 0.03s w
```

Aquí vemos que son las 3h40 de la mañana. El servidor tiene un uptime de **99 días**.

Sólo hay una persona conectada y la carga es de **0.00**. En resumen, todo es normal.

La carga indica cuantos procesos utilizan la CPU al mismo tiempo. Dispone de la media en 1 minuto, 5 minutos y 15 minutos.

La carga no debería pasar de **1** con 1 CPU. Pero todo depende de lo que estamos ejecutando en la máquina.

```
[root@ns root]# free
total used free shared buffers cached
Mem: 254696 249608 5088 0 95688 83616
-/+ buffers/cache: 70304 184392
Swap: 522104 3808 518296
```

El comando **free** muestra la utilización de la RAM. Vemos que la RAM es de 256. Se está utilizando casi al 100%. Es normal incluso si no se ejecuta nada.

Linux utiliza toda la RAM para acelerar la ejecución de procesos almacenando informaciones diversas. Por tanto, se utiliza siempre al 100%.

Si un proceso necesita RAM, Linux va a eliminar algunas informaciones y le dará la posibilidad de tomar la RAM que necesita.

Además vemos que realmente 70 MB son realmente utilizados por los procesos y el resto, es decir, 184 MB sólo por Linux.

El swap (ram en el disco duro) es de 512. Se utiliza cuando no queda RAM.

El swap es mucho más lento que la RAM. No es buena señal que su máquina utilice mucho el swap. Está sobrecargando el disco duro y la ejecución es más lenta que habitualmente. El servidor ralentiza. Y como las solicitudes siguen llegando, el servidor acaba por colgarse. Solución: rearrancar el servidor en hard.

Consulte la guía : [RebootDeLaMaquina](#).

¿Qué se ejecuta en el servidor?

### El comando PS

Podemos ver que **ps auxw** muestra el resultado en varias columnas.

La 1ª columna es el **user** o **usuario** que ha lanzado el proceso.

La 2ª columna es el **pid** del proceso

La 3ª columna es la utilización CPU del proceso

La 4ª columna es la utilización RAM del proceso

```
[root@ns root]# ps auxw
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 2 0.0 0.0 0 0 ? SW Jun26 0:00 [keventd]
root 3 0.0 0.0 0 0 ? SWN Jun26 0:35 [ksoftirqd_CPU0]
root 4 0.0 0.0 0 0 ? SW Jun26 0:33 [kswapd]
root 5 0.0 0.0 0 0 ? SW Jun26 0:00 [bdflush]
root 6 0.0 0.0 0 0 ? SW Jun26 0:28 [kupdated]
root 7 0.0 0.0 0 0 ? SW< Jun26 0:00 [mdrecoveryd]
root 8 0.0 0.0 0 0 ? SW Jun26 5:50 [kjournald]
root 113 0.0 0.0 0 0 ? SW Jun26 28:29 [kjournald]
root 354 0.0 0.0 0 0 ? SW Jun26 0:00 [eth0]
root 453 0.0 0.1 1444 480 ? S Jun26 0:03 syslogd -m 0
root 458 0.0 0.1 1372 324 ? S Jun26 0:00 klogd -2
root 640 0.0 0.1 3612 408 ? S Jun26 0:00
/usr/lib/courier-imap/libexec/authlib/authdaemond.ldap start
root 644 0.0 0.1 3612 408 ? S Jun26 0:00
/usr/lib/courier-imap/libexec/authlib/authdaemond.ldap start
```

## OVH

```
root 645 0.0 0.1 3612 408 ? S Jun26 0:00
/usr/lib/courier-imap/libexec/authlib/authdaemon.ldap start
root 650 0.0 0.1 3612 408 ? S Jun26 0:00
/usr/lib/courier-imap/libexec/authlib/authdaemon.ldap start
root 651 0.0 0.1 3612 408 ? S Jun26 0:00
/usr/lib/courier-imap/libexec/authlib/authdaemon.ldap start
root 652 0.0 0.1 3612 408 ? S Jun26 0:00
/usr/lib/courier-imap/libexec/authlib/authdaemon.ldap start
root 654 0.0 0.1 1600 432 ? S Jun26 0:00
/usr/lib/courier-imap/libexec/couriertcpd -address=0
-stderrlogger=/usr/lib/courier-imap/libexec/l
root 660 0.0 0.1 1360 392 ? S Jun26 0:00
/usr/lib/courier-imap/libexec/logger imapd
root 678 0.0 0.2 1568 544 ? S Jun26 0:03 crond
daemon 717 0.0 0.1 1416 480 ? S Jun26 0:00 /usr/sbin/atd
root 743 0.0 0.5 1424 1424 ? SL Jun26 0:13 watchdog
root 749 0.0 0.0 1348 16 ? S Jun26 0:10
/usr/local/clockspeed/bin/clockspeed
root 779 0.0 0.0 1356 4 tty1 S Jun26 0:00 /sbin/mingetty tty1
root 780 0.0 0.0 1356 4 tty2 S Jun26 0:00 /sbin/mingetty tty2
root 781 0.0 0.0 1356 4 tty3 S Jun26 0:00 /sbin/mingetty tty3
root 782 0.0 0.0 1356 4 tty4 S Jun26 0:00 /sbin/mingetty tty4
root 785 0.0 0.0 1356 4 tty5 S Jun26 0:00 /sbin/mingetty tty5
root 786 0.0 0.0 1356 4 tty6 S Jun26 0:00 /sbin/mingetty tty6
root 788 0.0 0.0 1368 4 ttyS0 S Jun26 0:00 /sbin/agetty ttyS0 9600
root 515 0.0 0.3 2148 776 ? S Jun26 0:00 xinetd -stayalive -reuse
-pidfile /var/run/xinetd.pid
root 29469 0.0 0.3 2208 912 ? S< Jun27 0:08 /usr/local/etc/ncftpd/ncftpd
-q /usr/local/etc/ncftpd/general.cf /usr/local/etc/ncftpd/domain.cf
root 29470 0.0 0.2 2008 704 ? SN Jun27 0:00 /usr/local/etc/ncftpd/ncftpd
-q /usr/local/etc/ncftpd/general.cf /usr/local/etc/ncftpd/domain.cf
named 4681 0.0 0.9 11984 2448 ? S Jun27 0:00 named -u named
named 4683 0.0 0.9 11984 2448 ? S Jun27 0:08 named -u named
named 4684 0.0 0.9 11984 2448 ? S Jun27 6:52 named -u named
named 4685 0.0 0.9 11984 2448 ? S Jun27 0:03 named -u named
named 4686 0.0 0.9 11984 2448 ? S Jun27 1:11 named -u named
root 17592 0.0 0.6 4856 1600 ? S Aug31 0:06 /usr/bin/perl
/usr/libexec/webmin/miniserv.pl /etc/webmin/miniserv.conf
root 7798 0.0 1.5 8500 3844 ? S Sep16 0:01 /usr/local/apache/bin/httpd
root 8849 0.0 0.4 2260 1036 ? S Sep16 0:00 /bin/sh /usr/bin/safe_mysqld
--datadir=/var/lib/mysql --pid-file=/var/lib/mysql/ns.witbebenchmark6
mysql 8868 0.0 0.4 10584 1040 ? S Sep16 0:00 /usr/sbin/mysqld --basedir=/
--datadir=/var/lib/mysql --user=mysql --pid-file=/var/lib/mysql/ns.wi
mysql 8870 0.0 0.4 10584 1040 ? S Sep16 0:00 /usr/sbin/mysqld --basedir=/
--datadir=/var/lib/mysql --user=mysql --pid-file=/var/lib/mysql/ns.wi
mysql 8871 0.0 0.4 10584 1040 ? S Sep16 0:00 /usr/sbin/mysqld --basedir=/
--datadir=/var/lib/mysql --user=mysql --pid-file=/var/lib/mysql/ns.wi
root 20539 0.0 0.5 2980 1356 ? S Sep28 0:00 /usr/sbin/sshd
nobody 28146 0.0 1.6 8772 4084 ? S Oct03 0:00 /usr/local/apache/bin/httpd
nobody 28210 0.0 1.6 8772 4084 ? S Oct03 0:00 /usr/local/apache/bin/httpd
nobody 28340 0.0 1.6 8772 4084 ? S Oct03 0:00 /usr/local/apache/bin/httpd
```

## OVH

```
nobody 28600 0.0 1.6 8772 4084 ? S Oct03 0:00 /usr/local/apache/bin/httpd
nobody 28756 0.0 1.6 8772 4084 ? S Oct03 0:00 /usr/local/apache/bin/httpd
nobody 28992 0.0 1.6 8772 4084 ? S Oct03 0:00 /usr/local/apache/bin/httpd
nobody 28994 0.0 1.6 8772 4084 ? S Oct03 0:00 /usr/local/apache/bin/httpd
nobody 29123 0.0 1.6 8704 4076 ? S Oct03 0:00 /usr/local/apache/bin/httpd
nobody 29384 0.0 1.6 8772 4084 ? S Oct03 0:00 /usr/local/apache/bin/httpd
nobody 29908 0.0 1.5 8704 4048 ? S 01:58 0:00 /usr/local/apache/bin/httpd
nobody 30037 0.0 1.6 8704 4076 ? S 02:28 0:00 /usr/local/apache/bin/httpd
root 30108 0.0 0.3 2208 948 ? S< 02:44 0:00 /usr/local/etc/ncftpd/ncftpd
-q /usr/local/etc/ncftpd/general.cf /usr/local/etc/ncftpd/domain.cf
nobody 30300 0.0 1.5 8704 4048 ? S 03:28 0:00 /usr/local/apache/bin/httpd
root 30355 0.0 0.8 3792 2048 ? S 03:40 0:00 sshd: root@pts/0
root 30358 0.0 0.5 2452 1292 pts/0 S 03:40 0:00 -bash
root 30421 0.0 0.2 2652 764 pts/0 R 03:44 0:00 ps auxw
```

Antes de la última columna está el tiempo de ejecución acumulado, es decir, equivalente a la utilización 100% de la máquina.

La última columna es el nombre del proceso. Si ejecuta **ps auxwww** obtendrá el nombre con la ruta completa.

### El proceso INIT

El proceso **init** administra el funcionamiento de toda la máquina.

```
root 1 0.0 0.1 1376 452 ? S Jun26 0:14 init [3]
```

Es el proceso que se ejecuta en primer lugar. Permite la gestión de 6 niveles vía **/etc/inittab**. Los niveles que nos interesan son:

```
0 - halt o parada de la máquina
1 - single o modo en reparación (sin red)
2 a 5 - funcionamiento normal
6 - reboot de la máquina
```

El nivel en el que su servidor funciona habitualmente es el 3.

Es aconsejable no manipular estos ficheros. Si los modifica, se arriesga a no arrancar correctamente. En ciertos sistemas (Red Hat, Fedora) La red funciona únicamente en el nivel 3.

Los ficheros de arranque de cada nivel se encuentran en **/etc/rc.d/**. Para el nivel 3 es **/etc/rc.d/rc3.d/**.

Puede ver los procesos que arranca (comenzando por **S**) y los que finaliza (comenzando por **K**).

## OVH

```
[root@ns root]# cd /etc/rc.d/rc3.d/
[root@ns rc3.d]# ls -l
total 0
lrwxrwxrwx 1 root root 20 mai 15 14:22 K15postgresql ->
../init.d/postgresql
lrwxrwxrwx 1 root root 18 mai 15 14:23 K45arpwatch -> ../init.d/arpwatch
lrwxrwxrwx 1 root root 15 mai 15 14:22 K45named -> ../init.d/named
lrwxrwxrwx 1 root root 16 mai 15 14:22 K65identd -> ../init.d/identd
lrwxrwxrwx 1 root root 14 mai 15 12:32 K74nscd -> ../init.d/nscd
lrwxrwxrwx 1 root root 14 mai 15 14:22 K74ntpd -> ../init.d/ntpd
lrwxrwxrwx 1 root root 17 mai 15 14:21 S10network -> ../init.d/network
lrwxrwxrwx 1 root root 14 mai 15 14:22 S11sshd -> ../init.d/sshd
lrwxrwxrwx 1 root root 16 mai 15 14:20 S12syslog -> ../init.d/syslog
lrwxrwxrwx 1 root root 22 mai 15 12:54 S13named -> /etc/rc.d/init.d/named
lrwxrwxrwx 1 root root 18 mai 15 14:20 S17keytable -> ../init.d/keytable
lrwxrwxrwx 1 root root 16 mai 15 14:21 S20random -> ../init.d/random
lrwxrwxrwx 1 root root 16 mai 15 14:22 S56xinetd -> ../init.d/xinetd
lrwxrwxrwx 1 root root 22 mai 15 12:45 S80qmail -> /etc/rc.d/init.d/qmail
lrwxrwxrwx 1 root root 22 mai 15 12:43 S85httpd -> /etc/rc.d/init.d/httpd
lrwxrwxrwx 1 root root 22 mai 15 12:54 S85imapd -> /etc/rc.d/init.d/imapd
lrwxrwxrwx 1 root root 15 mai 15 14:22 S90crond -> ../init.d/crond
lrwxrwxrwx 1 root root 15 mai 15 12:33 S90mysql -> ../init.d/mysql
lrwxrwxrwx 1 root root 13 mai 15 14:20 S95atd -> ../init.d/atd
lrwxrwxrwx 1 root root 25 mai 15 12:33 S95watchdog ->
/etc/rc.d/init.d/watchdog
lrwxrwxrwx 1 root root 11 mai 15 14:21 S99local -> ../rc.local
lrwxrwxrwx 1 root root 18 août 31 19:26 S99webmin -> /etc/init.d/webmin
```

Están en orden: inicialización de la red (network), sshd, syslog, named, xinetd, qmail, httpd, imapd, mysql y webmin.

El resto son administrativos (crond, atd, local, etc)

### Particiones de disco y SWAP

El proceso que administra el intercambio (SWAP) es el **kswapd**. Si consume demasiada CPU diremos que la máquina utiliza demasiado el swap.

```
root 4 0.0 0.0 0 0 ? SW Jun26 0:33 [kswapd]
```

Los procesos KJOURNALD son los procesos que administran las particiones de disco.

```
root 8 0.0 0.0 0 0 ? SW Jun26 5:51 [kjournald]
```

```
root 113 0.0 0.0 0 0 ? SW Jun26 28:29 [kjournald]
```

Podemos mostrar las particiones con el comando **df** :

```
[root@ns root]# df
Filesystem 1k-blocks Used Available Use% Mounted on
/dev/hda1 2016016 794996 1118608 42% /
/dev/hda2 37033060 848540 34303312 3% /home
none 127348 0 127348 0% /dev/shm
```

Aquí podemos ver que las particiones :

- / y **/home** son un disco **ide** (hd) montado en primario (hda).
- **/dev/shm** es la ram que puede ser utilizada en disco (a la inversa del swap).

```
[root@ns root]# mount
/dev/hda1 on / type ext3 (rw)
none on /proc type proc (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/hda2 on /home type ext3 (rw,usrquota)
none on /dev/shm type tmpfs (rw)
```

La partición **/dev/hda1** está montada en **ext3** en **rw** (read and write). ext3 es la versión anotada de ext2, un sistema de particionamiento muy utilizado bajo Linux.

La anotación consiste en escribir un diario en el que el sistema guarda informaciones sobre las operaciones que debe efectuar. En caso de reboot en hard, el sistema no necesita verificar todo el disco para buscar los errores. Basta con ejecutar de nuevo el diario.<

### Otros ejemplos de particiones

Un ejemplo de **df** con un disco **scsien raid5 por soft** con una tarjeta adaptec :

```
[root@ftp root]# df
Filesystem 1k-blocks Used Available Use% Mounted on
/dev/hda1 2016016 890216 1023388 47% /
/dev/hda2 12484192 381124 11468904 4% /home
none 257804 0 257804 0% /dev/shm
/dev/md0 141113720 46991780 86953764 36% /ftp
```

```
[root@ftp root]# cat /proc/mdstat
Personalities : [linear] [raid0] [raid1] [raid5]
read_ahead 1024 sectors
md0 : active raid5 sde1 [4] sdd1[3] sdc1[1] sdb1[0] sda1[2]
143363584 blocks level 5, 64k chunk, algorithm 0 [5/5] [UUUUU]

unused devices: <none>
```

Otro ejemplo en **raid-1 hard scsi** con una tarjeta mylex :

```
[root@sql root]# df
Filesystem lk-blocks Used Available Use% Mounted on
/dev/rd/c0d0p1 2015808 1395020 518392 73% /
/dev/rd/c0d0p2 15364496 814668 13769336 6% /home

[root@sql root]# cat /proc/rd/c0/current_status
DAC960 RAID Driver Version 2.4.11 of 11 October 2001
Copyright 1998-2001 by Leonard N. Zubkoff <lnz@dandelion.com>
Configuring Mylex DAC960PRL PCI RAID Controller

Firmware Version: 4.08-0-37, Channels: 1, Memory Size: 4MB
PCI Bus: 1, Device: 10, Function: 1, I/O Address: Unassigned
PCI Address: 0xF6AFE000 mapped at 0xE0800000, IRQ Channel: 6
Controller Queue Depth: 124, Maximum Blocks per Command: 128
Driver Queue Depth: 123, Scatter/Gather Limit: 33 of 33 Segments
Stripe Size: 64KB, Segment Size: 8KB, BIOS Geometry: 128/32

Physical Devices:
0:0 Vendor: IBM Model: DNES-318350W Revision: SA30
Serial Number: AKFJ0356
Disk Status: Online, 35840000 blocks

0:4 Vendor: IBM Model: DNES-318350W Revision: SA30
Serial Number: AKFJ5804
Disk Status: Online, 35840000 blocks

Logical Drives:
/dev/rd/c0d0: RAID-1, Online, 35831808 blocks, Write Thru
No Rebuild or Consistency Check in Progress
```

Otro ejemplo en **nfs** (network file system) :

```
[root@xxx /root]# df
Filesystem lk-blocks Used Available Use% Mounted on
/dev/hda1 2016016 1001556 912048 52% /
```

## OVH

```
/dev/hda5 17496684 11502724 5105168 69% /var/log
xx.xx.xx.xx:/parti
184320000 175846984 8473016 95% /home
```

### Procesos de mensajes

Si vigilamos los procesos :

```
root 453 0.0 0.1 1444 480 ? S Jun26 0:03 syslogd -m 0
root 458 0.0 0.1 1372 324 ? S Jun26 0:00 klogd -2
```

Son los procesos de **syslog** los que se ocupan de registrar todos los mensajes que el sistema envía. Se registran normalmente en **/var/log**

```
[root@ns log]# tail -f /var/log/messages
Oct 2 04:02:00 ns syslogd 1.4.1: restart.
Oct 2 04:02:00 ns syslogd 1.4.1: restart.
Oct 3 04:02:00 ns syslogd 1.4.1: restart.
Oct 3 04:02:00 ns syslogd 1.4.1: restart.
Oct 3 09:30:23 ns telnetd[25601]: ttloop: peer died: EOF
Oct 3 09:30:23 ns telnetd[25602]: ttloop: peer died: EOF
Oct 4 03:40:20 ns sshd[30355]: Accepted keyboard-interactive/pam for root
from 213.186.33.13 port 44298 ssh2
Oct 4 03:40:20 ns sshd(pam_unix)[30358]: session opened for user root by
root(uid=0)
Oct 4 04:02:00 ns syslogd 1.4.1: restart.
Oct 4 04:02:00 ns syslogd 1.4.1: restart.
```

En **/var/log/messages** encontrará los mensajes que envían los principales procesos, como telnet, sshd, named etc.

Con **tail -f** puede leer un fichero y ver los nuevos registros.

¿ Quién escucha qué ?

A continuación podemos ver todos los procesos que administran los servicios del servidor: apache, qmail, sshd, named que podemos recuperar con **netstat -tanpu**

```
[root@ns root]# netstat -tanpu
Conexiones Internet activas
Proto Recv-Q Send-Q Adresse locale Adresse distante Etat PID/Program name
tcp 0 0 0.0.0.0:3306 0.0.0.0:* LISTEN 8868/mysqld
tcp 0 0 0.0.0.0:143 0.0.0.0:* LISTEN 654/couriertcpd
tcp 0 0 0.0.0.0:80 0.0.0.0:* LISTEN 7798/httpd
tcp 0 0 0.0.0.0:10000 0.0.0.0:* LISTEN 17592/perl
tcp 0 0 213.186.36.76:53 0.0.0.0:* LISTEN 4681/named
tcp 0 0 127.0.0.1:53 0.0.0.0:* LISTEN 4681/named
tcp 0 0 0.0.0.0:21 0.0.0.0:* LISTEN 29469/ncftpd
tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN 20539/sshd
tcp 0 0 0.0.0.0:23 0.0.0.0:* LISTEN 515/xinetd
tcp 0 0 127.0.0.1:953 0.0.0.0:* LISTEN 4681/named
tcp 0 0 213.186.36.76:33206 65.57.234.3:6667 ESTABLISHED -
tcp 0 0 213.186.36.76:22 213.186.33.13:44298 ESTABLISHED 30355/0
tcp 0 0 213.186.36.76:80 81.88.96.9:39804 TIME_WAIT -
udp 0 0 0.0.0.0:10000 0.0.0.0:* 17592/perl
udp 0 0 0.0.0.0:32804 0.0.0.0:* 4681/named
udp 0 0 213.186.36.76:53 0.0.0.0:* 4681/named
udp 0 0 127.0.0.1:53 0.0.0.0:* 4681/named
udp 0 0 0.0.0.0:56125 0.0.0.0:* -
```

Podemos ver por ejemplo que hay un proceso que escucha en el puerto **0.0.0.0:80** y que se llama **httpd** con pid **7798**. Verificamos:

```
[root@ns root]# ps auxw | grep 7798
root 7798 0.0 1.5 8500 3844 ? S Sep16 0:01 /usr/local/apache/bin/httpd
```

El puerto 80 es la web (Apache). ¿Funciona?

```
[root@ns root]# telnet localhost 80
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^['.
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Sat, 04 Oct 2003 02:29:15 GMT
Server: Apache/1.3.28 (Unix) mod_gzip/1.3.19.1a PHP/4.3.3 mod_ssl/2.8.15
OpenSSL/0.9.6i
Content-Location: index.html.en
Vary: negotiate,accept-language,accept-charset
TCN: choice
Last-Modified: Thu, 15 May 2003 10:34:33 GMT
```

```
ETag: "18446-a71-3ec36d39;3ec37100"  
Accept-Ranges: bytes  
Content-Length: 2673  
Connection: close  
Content-Type: text/html  
Content-Language: en  
Expires: Sat, 04 Oct 2003 02:29:15 GMT
```

Connection closed by foreign host.

Parece ser que sí. Si la finalizamos:

```
[root@ns root]# kill 7798  
[root@ns root]# ps auxw | grep 7798  
root@ns root# ps auxw | grep httpd
```

¿Ya no funciona?

```
root@ns root# telnet localhost 80  
Trying 127.0.0.1...  
telnet: connect to address 127.0.0.1: Connection refused
```

Efectivamente, está realmente terminada. Vamos a reiniciarla:

```
[root@ns root]# /etc/rc.d/init.d/httpd start  
Démarrage de httpd: [ OK ]
```

y comprobar algunas informaciones:

```
[root@ns root]# ps auxw | grep httpd | grep root  
root 30899 0.4 1.4 8472 3820 ? S 04:31 0:00 /usr/local/apache/bin/httpd
```

```
[root@ns root]# netstat -tanpu | grep ":80"  
tcp 0 0 0.0.0.0:80 0.0.0.0:* LISTEN 30899/httpd
```

```
[root@ns root]# telnet localhost 80  
Trying 127.0.0.1...  
Connected to localhost.  
Escape character is '^['.
```

**HEAD / HTTP/1.0**

```

HTTP/1.1 200 OK
Date: Sat, 04 Oct 2003 02:32:18 GMT
Server: Apache/1.3.28 (Unix) mod_gzip/1.3.19.1a PHP/4.3.3 mod_ssl/2.8.15
OpenSSL/0.9.6i
Content-Location: index.html.en
Vary: negotiate,accept-language,accept-charset
TCN: choice
Last-Modified: Thu, 15 May 2003 10:34:33 GMT
ETag: "18446-a71-3ec36d39;3ec37100"
Accept-Ranges: bytes
Content-Length: 2673
Connection: close
Content-Type: text/html
Content-Language: en
Expires: Sat, 04 Oct 2003 02:32:18 GMT
Connection closed by foreign host.

```

Todo correcto. El servicio web (httpd) es operacional de nuevo.

Administración en tiempo real

Si desea tener **ps auxw** permanentemente, puede utilizar el comando **top**

```

[root@ns root]# top
4:34am up 99 days, 10:03, 1 user, load average: 0,03, 0,07, 0,02
65 processes: 64 sleeping, 1 running, 0 zombie, 0 stopped
CPU states: 0,1% user, 0,5% system, 0,0% nice, 99,2% idle
Mem: 254696K av, 245660K used, 9036K free, 0K shrd, 95316K buff
Swap: 522104K av, 3808K used, 518296K free 79948K cached

```

```

PID USER PRI NI SIZE RSS SHARE STAT %CPU %MEM TIME COMMAND
30937 root 17 0 1048 1048 824 R 0,5 0,4 0:00 top
30355 root 12 0 2064 2064 1692 S 0,1 0,8 0:00 sshd
1 root 8 0 484 452 424 S 0,0 0,1 0:14 init
2 root 9 0 0 0 0 SW 0,0 0,0 0:00 keventd
3 root 19 19 0 0 0 SWN 0,0 0,0 0:35 ksoftirqd_CPU0

```

Pulse la tecla **May-M** para ordenar por uso de RAM.

Por ejemplo, si su máquina se cuelga porque utiliza demasiado el swap, lance top y espere a que se cuelgue. Verá en pantalla el último top y así sabrá qué proceso hace que todo se cuelgue.

## OVH

Puede usar el Real Time Monitoring para realizar un seguimiento en tiempo real de todos estos parámetros.

Más información en nuestra guía : [RealTimeMonitoring](#)

Más información

: [SshSobreServidorDedicado](#) :: Cómo conectarse al SSH en un servidor dedicado

: [RebootDeLaMaquina](#) :: Cómo reinicializar mi servidor

: [RealTimeMonitoring](#) :: Descripción e instalación de la herramienta RTM