

Administrer la machine en ssh (shell sous bash)

Astuces Bash

Bash est le programme qui est exécuté lorsque vous vous connectez en ssh sur votre machine. Vous pouvez choisir un autre programme de shell comme sh, ash, etc.

```
root@kazik root# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

Vous trouverez beaucoup de sites qui traitent de ce sujet : <http://www.google.fr/search?q=bash>.

Quelques astuces :

- bash complète votre ligne de commande avec la touche tab

par exemple :

```
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 accepte les alias. Si par exemple vous tapez souvent **ls -l** vous pouvez créer un alias et le taper directement :

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

maintenant, si vous tapez, **ll** ça sera équivalent de **ls -l --color**. Vous pouvez l'ajouter sur votre fichier `.bashrc` pour que la prochaine fois que vous puissiez en profiter à nouveau :

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

Surveiller le système

Lorsqu'on se connecte sur la machine, il faut prendre réflexe de vérifier rapidement ce qu'il se passe sur votre machine. Si ce résultat visuel n'est pas le même que d'habitude, cela permet de découvrir très rapidement qu'il y a un problème.

```
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
```

la commande **w** permet de voir rapidement l'uptime, la charge de la machine et combien de personnes sont connectées sur votre machine. On voit ici qu'il est 3h40 du matin. La machine a un uptime de **99** jours. Il y a 1 seule personne connectée et la charge est de **0.00**. Bref, tout est normal.

La charge veut dire : combien de process utilisent le CPU en même temps. Vous avez la moyenne sur 1 minute, 5 minutes et 15 minutes. On dit que la charge ne devrait pas dépasser 1 sur une machine avec CPU. Mais tout dépend de ce que l'on fait tourner sur la machine.

```
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
```

Free donne l'utilisation de la RAM. On voit que la RAM est de 256 Mo. Elle est utilisée à presque 100%. Ceci est normal même s'il ne se passe rien sur la machine. Linux utilise toute la RAM pour accélérer l'exécution des process en stockant des informations diverses. Elle est donc toujours utilisée à 100%. Si un moment il y a un process qui a besoins de la RAM, Linux va effacer ces informations et va lui donner la possibilité de prendre la RAM nécessaire. D'ailleurs, on voit qu'en vrai 70Mo sont réellement utilisés par les process et le reste, c'est-à-dire 184Mo, juste par Linux.

Le swap (la ram sur disque) est de 512 Mo. Il est utilisée lorsqu'il n'y a plus de RAM (mais alors vraiment plus). Il faut savoir que swap est nettement plus lent que la RAM. Si votre machine utilise beaucoup de swap ce n'est pas bon. Vous surchargez votre disque et donc vous l'usez plus vite. De plus, l'exécution est plus lente que d'habitude. Le serveur tourne donc au ralenti. Et comme les nouvelles requêtes arrivent toujours et toujours, il y en a de plus en plus en même temps. Puis la machine plante. C'est un cas classique qui arrive à presque chaque débutant. Solution ? Reboot en hard de la machine. Consultez le guide: Comment Rebooter Mon Serveur Dédié ?

Qu'est-ce qui tourne sur la machine ?

OVH

```

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/authdaemon.ldap start
root 644 0.0 0.1 3612 408 ? S Jun26 0:00 /usr/lib/courier-imap/libexec/authlib/authdaemon.ldap start
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_mysql --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/mysql --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/mysql --basedir=/ --datadir=/var/lib/mysql

```

OVH

```
--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
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
```

On peut donc voir que la sortie **ps auxw** donne le résultat sur plusieurs colonnes.

La 1ère colonne est le **user** ou **l'utilisateur** qui fait tourner la machine.

La 2ème colonne est le **pid** du process.

La 3ème colonne est l'utilisation CPU du process.

La 4ème colonne est l'utilisation RAM du process.

L'avant dernière colonne c'est le temps d'exécution accumulé c'est à dire équivalent à l'utilisation à 100% du CPU.

La dernière colonne est le nom du process. si vous tapez **ps auxwwww** vous allez avoir toute la commande.

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

init gère le fonctionnement de toute la machine. C'est le process qui est exécuté en premier.

Il permet la gestion de 6 niveaux via **/etc/inittab**. Les niveaux qui nous intéressent sont :

0 – halt ou arrêt de la machine

1 – single ou mode en réparation (pas de réseau)

3 – fonctionnement normal

6 – reboot de la machine

Le niveau dans lequel votre machine fonctionne habituellement est 3.

Il est fortement conseillé de ne pas jouer avec ces fichiers. Si vous les modifiez, votre machine risque de ne pas booter correctement. Le réseau fonctionne uniquement dans le niveau 3.

Les fichiers de démarrage dans chaque niveau se trouvent dans **/etc/rc.d/** et donc pour le niveau 3 c'est **/etc/rc.d/rc3.d/**. Vous pouvez y voir ce qui démarre (commençant avec un **S**) et ce qui est tué (commençant avec un **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
```

On retrouve donc dans l'ordre: démarrage du réseau (network), sshd, syslog, named, xinetd, qmail, httpd, imapd, mysql et webmin.

Les autres démarrages sont administratifs (crond, atd, local, etc)

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

le process qui gère le **swap**. S'il prend trop de CPU ça veut dire que votre machine swap beaucoup (trop ?).

```
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
```

Ce sont les process qui gèrent les journaux de vos partitions des disques.

```
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
```

On peut voir que vous avez la partition / et /home sont un disque **ide** (hd) monté en primaire (hda). /dev/shm est la ram qui peut être utilisée en disque (l'inverse du 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 partition **/dev/hda1** est montée en **ext3** en **rw** (read and write). ext3 est la version journalisée d'ext2, un système de partitionnement très utilisé sous Linux. La journalisation consiste à écrire un journal dans lequel le système met les informations sur les opérations à effectuer. En cas de reboot en hard le système n'a pas besoin de vérifier tout le disque à la recherche d'erreurs. Il lui suffit de rejouer le journal.

Un exemple de **df** avec un disque en **scsi** :

- en **raid-5 soft scsi** avec une carte 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 sde14 sdd13 sdc11 sdb10 sda12
143363584 blocks level 5, 64k chunk, algorithm 0 5/5 UUUUU
unused devices:

```

- en **raid-1 hard scsi** avec une carte mylex

```

root@sql root# df
Filesystem 1k-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
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

```

OVH

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

- en **nfs** (network file system)

```
root@xxx /root# df
Filesystem 1k-blocks Used Available Use% Mounted on
/dev/hda1 2016016 1001556 912048 52% /
/dev/hda5 17496684 11502724 5105168 69% /var/log
xx.xx.xx.xx:/parti
184320000 175846984 8473016 95% /home

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
```

Ce sont les process de **syslog** qui s'occupent d'enregistrer tous les messages que le système envoie. Il enregistre habituellement dans **/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 telnetd25601: tloop: peer died: EOF
Oct 3 09:30:23 ns telnetd25602: tloop: peer died: EOF
Oct 4 03:40:20 ns sshd30355: 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.
```

dans **/var/log/messages** vous pouvez trouver les messages que les principaux process envoient comme telnet, sshd, named etc. Avec **tail-f**, vous pouvez lire un fichier et voir les nouveaux enregistrements.

Qui écoute quoi ?

Ensuite on peut voir tous les process qui gèrent les serveurs : apache, qmail, sshd, named qu'on peut récupérer avec **netstat -tanpu**.

```
root@ns root# netstat -tanpu
Connexions Internet actives (serveurs et établies)
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 0.0.0.0:80 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:* -
```

On voit par exemple qu'il y a un process qui écoute le port **0.0.0.0:80** qui s'appelle **httpd** avec un pid **7798**.
On vérifie :

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

Port 80 c'est le web. Est-ce qu'il fonctionne ?

```
root@ns root# telnet localhost 80
Trying 127.0.0.1...
Connected to localhost.
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.

On va le tuer alors :

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

et il ne fonctionne plus ?

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

il est vraiment mort. On va le redémarrer :

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

et verifier quelques informations:

```
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.
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.

Il est à nouveau opérationnel.

Administration en temps réel

Si vous voulez avoir **ps auxw** en permanence, vous pouvez utiliser la commande **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
```

Cliquez sur **shift-m** pour faire un tri par utilisation de la RAM. Si votre machine plante parce que ça swap, lancez top et attendez que ça plante. Vous aurez sur votre écran le dernier top et donc vous saurez quel process fait tout planter. Voilà vous savez désormais comment voir ce qu'il se passe sur votre machine, comment voir les process, les tuer et les redémarrer. Plus qu'à charger la machine.