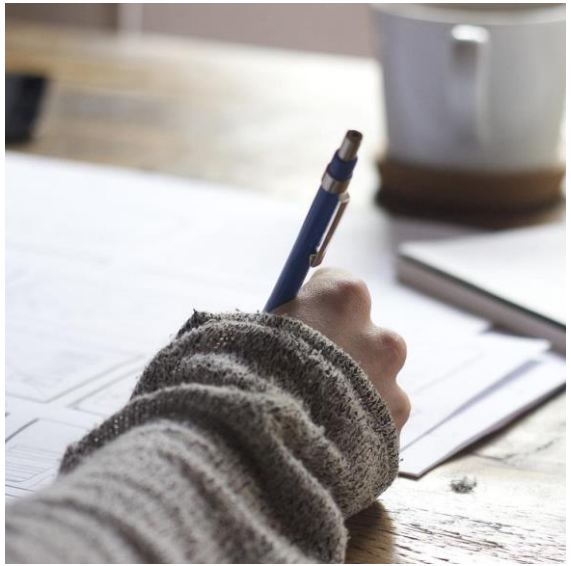
Diagnósticos en Linux

En esta presentación
Vamos a aprender sobre los comandos utilizados para diagnosticar problemas en linux



1.

Comandos Linux

Diagnósticos general

voipdo.com

» Para diagnosticar en linux debemos utilizar comandos muy específicos que arrojan información importante. Vamos a utilizar estos comandos si queremos saber el espacio en el disco, la ip que tenemos, hacer un trace route, saber la capacidad de memoria, los servicios que tenemos arriba y demás.

»Recuerde que todos los comandos son en minúscula

aname -a

»Nos informa la versión del Kernel, procesador y nombre del sistema.

```
root@jandres# uname -a
Linux dominio.com 2.6.18-194.3.1.el5 #1 SMP Thu May 13 13:09:10
EDT 2010 i686 i686 i386 GNU/Linux
```

df -h

»Nos muestra el uso del disco y sus particiones.

```
root@jandres# df -h
S.ficheros          Tamaño Usado  Disp Uso% Montado en
/dev/cciss/c0d0p1      64G   54G  9,0G  86% /
tmpfs                1013M     0 1013M   0% /dev/shm
/usr/tmpDSK           485M   15M  445M   4% /tmp
```

free

»Nos muestra el uso de la memoria

```
root@jandres# free
    total       used       free     shared    buffers  cached
Mem:  2073728    1951632     122096   0     149644    1252812
-/+ buffers/cache:     549176    1524552
Swap:      1999992        472    1999520
```

netstat -a

»Nos muestra las conexiones de red actuales

```
[root@li839-210 ~]# netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
tcp        0      0 *:ssh                       *:*                         LISTEN
tcp        0      0 *:sieve-filter              *:*                         LISTEN
tcp        0      0 li839-210.members.linod:ssh ip-198-164.tricom.net:54891 ESTABLISHED
tcp        0    320 li839-210.members.linod:ssh ip-198-164.tricom.net:55131 ESTABLISHED
tcp        0     84 li839-210.members.linod:ssh 59.63.188.32:14455          ESTABLISHED
tcp        0      0 *:ssh                       *:*                         LISTEN
udp        0      0 *:commplex-main             *:*
udp        0      0 *:4520                      *:*
udp        0      0 *:sip                       *:*
udp        0      0 *:iax                       *:*
udp        0      0 *:mgcp-callagent            *:*
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node Path
unix  8      [ ]         DGRAM                    12369  /dev/log
unix  2      [ ACC ]     STREAM     LISTENING     9481   @/com/ubuntu/upstart
unix  2      [ ]         DGRAM                    9903   @/org/kernel/udev/udevd
unix  2      [ ACC ]     STREAM     LISTENING     12602  /var/run/asterisk/asterisk.ctl
unix  2      [ ACC ]     STREAM     LISTENING     12615  /var/run/fail2ban/fail2ban.sock
unix  3      [ ]         STREAM     CONNECTED     12333
unix  3      [ ]         STREAM     CONNECTED     12332
unix  3      [ ]         DGRAM                    9907
unix  3      [ ]         DGRAM                    9906
unix  2      [ ]         DGRAM                    2880737
unix  3      [ ]         STREAM     CONNECTED     2885183
unix  2      [ ]         DGRAM                    2885033
unix  2      [ ]         DGRAM                    12627
unix  2      [ ]         DGRAM                    12472
unix  2      [ ]         DGRAM                    2885190
unix  2      [ ]         DGRAM                    12575
unix  3      [ ]         STREAM     CONNECTED     2885182
```

ps -aux

»Nos muestra los procesos corriendo

```
root@jandres# ps -aux
Warning: bad syntax, perhaps a bogus '-'?
See /usr/share/doc/procps-3.2.7/FAQ
USER PID %CPU %MEM VSZ   RSS TTY STAT START   TIME COMMAND
root   1  0.0  0.0   2152   612 ?  Ss   Jul15   0:52 init [3]
root   2  0.0  0.0      0     0 ?     S<   Jul15   0:17 [migration/0]
root   3  0.0  0.0      0     0 ?     SN   Jul15   0:00 [ksoftirqd/0]
root   4  0.0  0.0      0     0 ?     S<   Jul15   0:00 [watchdog/0]
root   5  0.0  0.0      0     0 ?     S<   Jul15   0:25 [migration/1]
root   6  0.0  0.0      0     0 ?     SN   Jul15   0:00 [ksoftirqd/1]
....
....
....
```

ifconfig

»Nos muestra los ips asignados a nuestras tarjetas de red, ya sea DHCP o estática

```
[adminuser@localhost ~]$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.30  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::a00:27ff:feb0:a29f  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:b0:a2:9f  txqueuelen 1000  (Ethernet)
        RX packets 756392  bytes 1110263646 (1.0 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 384557  bytes 26150557 (24.9 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 0  (Local Loopback)
        RX packets 4  bytes 340 (340.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 340 (340.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.122.1  netmask 255.255.255.0  broadcast 192.168.122.255
        ether 00:00:00:00:00:00  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
```

ping

»Es el comando más sencillo para saber si tenemos configurada nuestra red como deseamos. En este ejemplo estoy dando ping a google para verificar que tengo acceso a internet y tengo mis DNS configurados correctamente

```
[adminuser@localhost ~]$ ping google.com
PING google.com (172.217.3.142) 56(84) bytes of data.
64 bytes from yyz08s13-in-f142.1e100.net (172.217.3.142): icmp_seq=1 ttl=55 time=59.0 m
s
64 bytes from yyz08s13-in-f142.1e100.net (172.217.3.142): icmp_seq=2 ttl=55 time=38.9 m
s
64 bytes from yyz08s13-in-f142.1e100.net (172.217.3.142): icmp_seq=3 ttl=55 time=40.4 m
s
64 bytes from yyz08s13-in-f142.1e100.net (172.217.3.142): icmp_seq=4 ttl=55 time=39.6 m
s
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 38.994/44.507/59.009/8.389 ms
[adminuser@localhost ~]$
```

traceroute

»Este comando nos permite ver los saltos que se necesitan para llegar a un destino

```
[adminuser@localhost ~]$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  gateway (192.168.1.1)  0.410 ms  0.348 ms  0.336 ms
 2  * * *
 3  172.17.116.253 (172.17.116.253)  10.727 ms  10.718 ms  10.685 ms
 4  172.17.128.33 (172.17.128.33)  11.111 ms  10.865 ms  10.810 ms
 5  172.17.254.229 (172.17.254.229)  49.695 ms  51.373 ms 172.17.255.101 (172.17.255.10
1)  40.586 ms
 6  * 72.14.216.225 (72.14.216.225)  44.199 ms  46.187 ms
 7  108.170.249.17 (108.170.249.17)  43.835 ms *  48.050 ms
 8  108.170.228.27 (108.170.228.27)  35.912 ms 108.170.229.153 (108.170.229.153)  43.02
2 ms 108.170.228.43 (108.170.228.43)  41.638 ms
 9  google-public-dns-a.google.com (8.8.8.8)  49.099 ms  46.560 ms  45.483 ms
[adminuser@localhost ~]$
```

route

»Este comando nos permite ver las rutas utilizadas para llegar a un punto. En el caso de Asterisk, generalmente utilizamos rutas estáticas para llegar al gateway del troncal SIP de nuestro proveedor. Route nos muestra si la salida está configurada correctamente.

```
[adminuser@localhost ~]$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         gateway         0.0.0.0         UG    100    0        0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U     100    0        0 eth0
192.168.122.0   0.0.0.0         255.255.255.0   U     0      0        0 virbr0
[adminuser@localhost ~]$
```

“

Ahora vamos a probar cada uno de estos comandos de diagnósticos. Iniciemos

Stevenson Canó

¿ALGUNA PREGUNTA?

Llama al:

- 809-473-7879
- Ing.cano@canoconsulting.net

voipdo.com