

# Linux Server (Ubuntu server)

---

## Configuración de las tarjetas de red

La configuración de las tarjetas de red se almacena bajo el lenguaje de marcado para configuración YAML habitualmente en la ruta:

```
sudo nano /etc/netplan/50-cloud-init.yaml
```

Este sería un ejemplo de configuración para dos tarjetas de red:

```
network:
  version: 2
  ethernet:
    # Primera tarjeta de red
    enp0s3:
      dhcp4: no # Desactivo el DHCP por IPv4

      # Por lo tanto asigno IP estatica y net mask
      addresses: [192.168.213.200/20]

      # Asigno DNS
      nameservers:
        addresses: [1.1.1.1, 8.8.8.8]

      # Puerta de enlace/dirección del router
      routes:
        - to: default
          via: 192.168.213.1

    # Segunda tarjeta de red
    enp0s8:
      dhcp4: true # Esta va por DHCP
```

## Controlar procesos y tareas

### Procesos

Para ver la lista de procesos utilizamos el comando `$ ps`

La salida es así:

```

polkitd      663  0.0  0.3 308160 7808 ?        Ssl  06:37   0:00 /usr/lib/polkitd
root         718  0.0  0.4  18120 8704 ?        Ss   06:37   0:00 /usr/lib/syst
root         733  0.0  0.6 468960 13440 ?        Ssl  06:37   0:00 /usr/libexec/
syslog       773  0.0  0.2 222508 5888 ?        Ssl  06:37   0:00 /usr/sbin/rsy
root         793  0.0  1.1 109676 22912 ?        Ssl  06:37   0:00 /usr/bin/pyth
root         802  0.0  0.6 318296 12672 ?        Ssl  06:37   0:00 /usr/sbin/Mod
root         867  0.0  0.1   6824 2688 ?        Ss   06:37   0:00 /usr/sbin/cro
root         876  0.0  0.2   6952 4608 tty1     Ss   06:37   0:00 /bin/login -p
root         982  0.0  0.0      0  0 ?        S    06:40   0:00 [psimon]
jaime        984  0.0  0.5  20348 11520 ?        Ss   06:40   0:00 /usr/lib/syst
jaime        985  0.0  0.1  21144 3516 ?        S    06:40   0:00 (sd-pam)
jaime        993  0.0  0.2   8784 5504 tty1     S    06:40   0:00 -bash
root        1021  0.0  0.0      0  0 ?        I    06:43   0:00 [kworker/u5:0
jaime        1032  0.0  0.1   6924 3712 tty1     S+   06:47   0:00 nano
root        1033  0.0  0.2   6956 4608 tty2     Ss   06:47   0:00 /bin/login -p
root        1035  0.0  0.0      0  0 ?        I    06:47   0:00 [kworker/u6:2
jaime        1089  0.0  0.2   8652 5632 tty2     S    06:47   0:00 -bash
root        1112  0.0  0.0      0  0 ?        I    06:50   0:00 [kworker/0:0]
root        1113  0.0  0.0      0  0 ?        I    06:50   0:00 [kworker/0:3]
root        1118  0.0  0.0      0  0 ?        I    06:53   0:00 [kworker/u5:1
root        1121  0.0  0.2   6960 4736 tty3     Ss   06:53   0:00 /bin/login -p
jaime        1177  0.0  0.2   8652 5376 tty3     S    06:53   0:00 -bash
jaime        1186  0.0  0.1   6924 3712 tty3     S+   06:54   0:00 nano
jaime        1191  0.0  0.2  10884 4480 tty2     R+   06:54   0:00 ps axu
jaime@damserver:~$

```

El problema es que la lista no está ordenada ni en tiempo real por lo que no podemos saber cual es el proceso más pesado.

Para esto existe el comando `$ top`

```

top - 06:57:51 up 19*min, 3 users, load average: 0,79, 0,30, 0,13
Tasks: 111 total, 2 running, 109 sleeping, 0 stopped, 0 zombie
%Cpu(s):  0,2 us, 48,9 sy,  0,0 ni, 50,9 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
MiB Mem :  1967,9 total,  1278,5 free,   578,1 used,   257,5 buff/cache
MiB Swap:  3069,0 total,  3069,0 free,    0,0 used,  1389,7 avail Mem
PID to signal/kill [default pid = 1325]

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1325	jaime	20	0	5684	1792	1792	R	95,7	0,1	0:05.46	yes
592	root	-51	0	0	0	0	S	4,0	0,0	0:11.42	irq/18--
1198	jaime	20	0	11932	5888	3712	R	0,3	0,3	0:00.27	top
1201	root	20	0	0	0	0	I	0,3	0,0	0:01.46	kworker+
1	root	20	0	22084	13020	9308	S	0,0	0,6	0:01.41	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0,0	0,0	0:00.00	pool_wor
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker+
5	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker+
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker+
7	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker+
9	root	20	0	0	0	0	I	0,0	0,0	0:00.49	kworker+
10	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker+
11	root	20	0	0	0	0	I	0,0	0,0	0:00.00	kworker+
12	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker+
13	root	20	0	0	0	0	I	0,0	0,0	0:00.00	rcu_tas+
14	root	20	0	0	0	0	I	0,0	0,0	0:00.00	rcu_tas+
15	root	20	0	0	0	0	I	0,0	0,0	0:00.00	rcu_tas+

Desde top para matar un proceso pulsamos la letra **K** y se nos pregunta si deseamos matar por defecto el que mas cpu usa, pero podemos escribir el PID de cualquier proceso, después nos pregunta que señal mandarle (15 porfavor muerete, 9 meterle un tiro)

```

top - 07:01:29 up 23 min,  3 users,  load average: 0,41, 0,41, 0,22
Tasks: 113 total,   2 running, 111 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0,0 us, 53,0 sy,  0,0 ni, 46,5 id,  0,0 wa,  0,0 hi,  0,5 si,  0,0 st
MiB Mem :  1967,9 total,  1296,8 free,   559,7 used,   257,6 buff/cache
MiB Swap:  3069,0 total,  3069,0 free,    0,0 used.  1408,1 avail Mem
Send pid 1334 signal [15/sigterm]
  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
  1334 jaime      20   0   5684   1792   1792  R   99,7    0,1    0:06,76  yes

```

Hay distintas versiones de **top** como **htop** y eso con mas colorines.

### Comandos interesantes

- \$ **yes** : Imprime tol rato **y** usando toda la cpu
- \$ **kill -9 PID** : Matar a la fuerza
- \$ **killall nombreprograma** : mata todas las instancias de un programa

### Tareas

Para ver la lista de tareas \$ **jobs** lo que saca:

```

jaime@damserver:~$ jobs
[1]-  Stopped                  yes
[2]+  Stopped                  sleep 600
jaime@damserver:~$ _

```

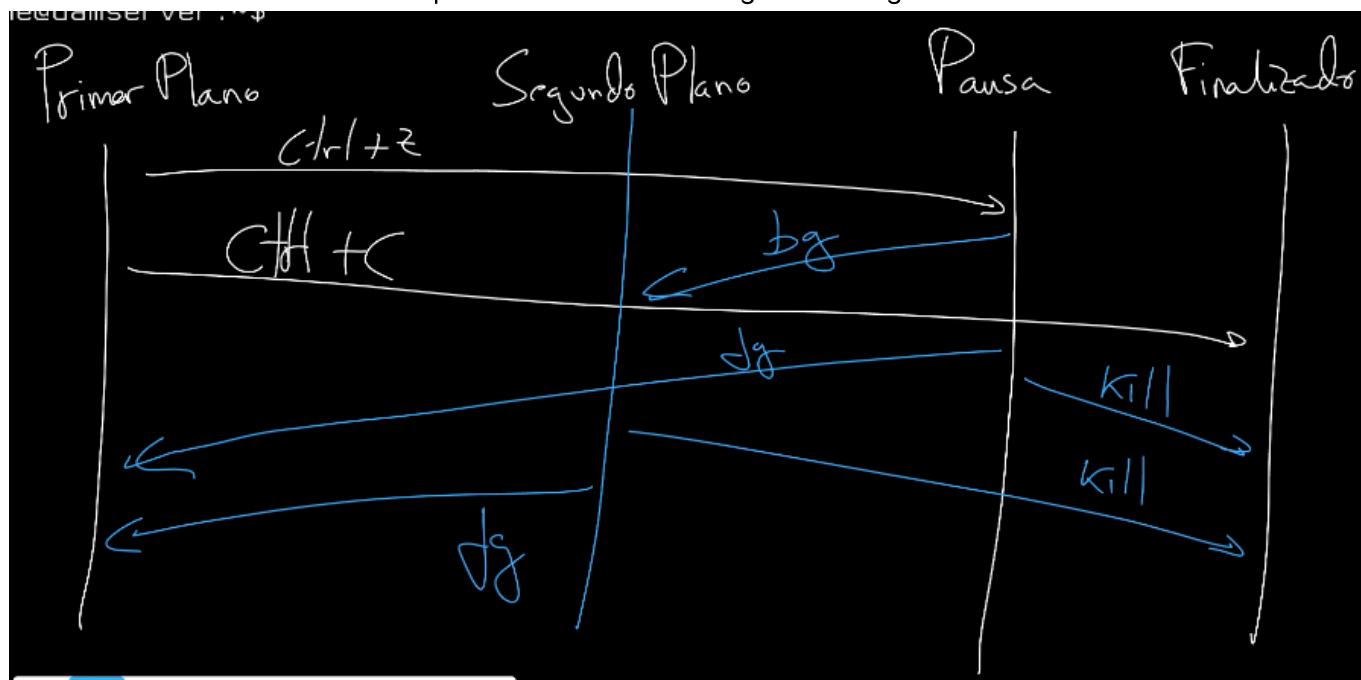
Cada tarea tiene un numero asignado al cual podremos referirnos despues para algunas operaciones.

### Las operaciones con tareas mas interesantes son:

- Pasar a segundo plano una tarea \$ **bg num**
- Traer a primer plano (o reanudarlo si estaba stopped) \$ **fg num**
- Detener una tarea 'ctrl + Z'
- Reanudarla con \$ **fg num** o \$ **bg num**
- Arrancar en segundo plano directamente: \$ **COMANDO &**
- Matar el proceso de una tarea: \$ **kill %ID**

Si una tarea está en segundo plano puede no dejar escribir en la terminal, en ese caso deberíamos cambiar de terminal y **matar** el proceso.

Para mover las tareas sirve de esquema de lo anterior la siguiente imagen:



## ☁ Memoria RAM

Podemos monitorizar directamente la memoria con el comando `free`.

## 🛠 Software de monitorización util

- `ps`
- `top`: Monitor de recursos en RT por defecto.
- `htop`: Monitor de recursos en RT muy bonito.
- `nmon`: Monitor de recursos en RT modular.
- `dstat`: Monitor de recursos en tiempo real.
- `iptraf-ng`: Analizador de red tipo Wireshark.