

fase 1, analisis en tiempo real del sistema

ejecutar y analizar las herramientas, top y ver procesos activos en carga y uso de cpu, htop y uptime y free -m

ejecutar top

```
top - 13:17:53 up 12 min, 2 users, load average: 0,10, 0,05, 0,01
Tasks: 127 total, 1 running, 126 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0,3 us, 0,8 sy, 0,0 ni, 98,0 id, 0,3 wa, 0,0 hi, 0,5 si, 0,0 st
MiB Mem : 1967,8 total, 1119,3 free, 405,5 used, 595,6 buff/cache
MiB Swap: 2048,0 total, 2048,0 free, 0,0 used. 1562,3 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1265	root	20	0	0	0	0	I	2,0	0,0	0:01.17	kworker/0:1-events
1597	oscar	20	0	11916	5632	3584	R	1,0	0,3	0:00.09	top
16	root	20	0	0	0	0	S	0,3	0,0	0:00.48	ksoftirqd/0
83	root	0	-20	0	0	0	I	0,3	0,0	0:00.07	kworker/0:1H-kblockd
366	root	rt	0	289116	27136	8576	S	0,3	1,3	0:00.37	multipathd
733	root	20	0	871528	46592	27264	S	0,3	2,3	0:04.46	fail2ban-server
1507	root	20	0	0	0	0	I	0,3	0,0	0:00.02	kworker/u5:3-flush-252:0
1	root	20	0	22128	13184	9344	S	0,0	0,7	0:02.17	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_workqueue_release
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-rcu_g

ejecutar htop

```
0[|||||] 7.8% Tasks: 37, 46 thr, 90 kthr; 2 running
1[|] 2.6% Load average: 0.09 0.05 0.00
Mem[|||||] 253M/1.92G Uptime: 00:14:19
Swp[|] 0K/2.00G
```

Main	I/O	PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
		1166	oscar	20	0	15124	6840	4992	S	2.0	0.3	0:01.38	sshd: oscar@pts/0
		1	root	20	0	22128	13184	9344	S	0.0	0.7	0:02.18	/sbin/init
		309	root	19	-1	50472	16264	15240	S	0.0	0.8	0:00.38	/usr/lib/systemd/systemd-journald
		366	root	RT	0	282M	27136	8576	S	0.0	1.3	0:00.15	/sbin/multipathd -d -s
		377	root	20	0	29068	7680	4992	S	0.0	0.4	0:00.23	/usr/lib/systemd/systemd-udevd
		379	root	20	0	282M	27136	8576	S	0.0	1.3	0:00.00	/sbin/multipathd -d -s
		380	root	RT	0	282M	27136	8576	S	0.0	1.3	0:00.00	/sbin/multipathd -d -s
		381	root	RT	0	282M	27136	8576	S	0.0	1.3	0:00.00	/sbin/multipathd -d -s
		382	root	RT	0	282M	27136	8576	S	0.0	1.3	0:00.00	/sbin/multipathd -d -s
		383	root	RT	0	282M	27136	8576	S	0.7	1.3	0:00.27	/sbin/multipathd -d -s
		384	root	RT	0	282M	27136	8576	S	0.0	1.3	0:00.00	/sbin/multipathd -d -s
		578	systemd-re	20	0	21584	12800	10624	S	0.0	0.6	0:00.25	/usr/lib/systemd/systemd-resolved
		588	systemd-ti	20	0	91020	7808	6912	S	0.0	0.4	0:00.11	/usr/lib/systemd/systemd-timesyncd
		669	systemd-ti	20	0	91020	7808	6912	S	0.0	0.4	0:00.00	/usr/lib/systemd/systemd-timesyncd
		699	systemd-ne	20	0	18996	9344	8320	S	0.0	0.5	0:00.10	/usr/lib/systemd/systemd-networkd
		730	messagebus	20	0	9796	5376	4480	S	0.0	0.3	0:00.23	@dbus-daemon --system --address=systemd: --nofork --nopidfi
		733	root	20	0	851M	46592	27264	S	0.0	2.3	0:00.76	/usr/bin/python3 /usr/bin/fail2ban-server -xf start
		735	bind	20	0	501M	19328	14208	S	0.0	1.0	0:00.05	/usr/sbin/named -f -u bind
		738	polkitd	20	0	300M	7808	7040	S	0.0	0.4	0:00.10	/usr/lib/polkit-1/polkitd --no-debug
		755	root	20	0	18208	8832	7808	S	0.0	0.4	0:00.16	/usr/lib/systemd/systemd-logind
		765	root	20	0	457M	13312	11264	S	0.0	0.7	0:00.11	/usr/libexec/udisks2/udisksd
		773	root	20	0	457M	13312	11264	S	0.0	0.7	0:00.04	/usr/libexec/udisks2/udisksd

1Help F2Setup F3Search F4Filter F5Free F6SortBy F7Nice F8Nice F9Kill F10Quit

uptime

```
oscar@srv-base-oscarhernandez:~$ uptime
13:20:53 up 15 min, 2 users, load average: 0,12, 0,07, 0,01
oscar@srv-base-oscarhernandez:~$
```

free -m

```
oscar@srv-base-oscarhernandez:~$ free -m
```

	total	used	free	shared	buff/cache	available
Mem:	1967	402	1120	3	597	1565
Swap:	2047	0	2047			

identificar el proceso con mayor consumo de CPU haciendo uso de top y p para ordenarlos

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
85	root	20	0	0	0	0	I	1,3	0,0	0:09.25	kworker/1:3-events

identificar el proceso con mayor uso de memoria con el comando top y pulsando m

```
733 root      20    0  871528  46592  27264 S   0,7   2,3   0:07.19 fail2ban-server
```

identificar tipo que lleva el sistema encendido y carga promedio

```
oscar@srv-base-oscarhernandez:~$ uptime
13:28:47 up 23 min,  2 users,  load average: 0,34, 0,12, 0,03
oscar@srv-base-oscarhernandez:~$
```

lleva encendido 23 minutos hay dos usuario, que son el root y ssh porque estoy conectado con la reminal de ubuntu y la carga del sistem en el ultimo minuto es de 0.34, en los ultimos 5 minutos es de 0.12 y en los ultimos 15 minutos es de 0.03

fase 2, gestion activa de procesos y prioridades

hacemos la prueba abriendo un terminal que no afecte a nada

```
oscar@srv-base-oscarhernandez:~$ sleep 300 &
[1] 1645
```

y terminamos el proceso con pkil

```
oscar@srv-base-oscarhernandez:~$ pkill sleep
[1]+  Terminated                  sleep 300
```

Cambiar la prioridad de un proceso en ejecución con renice, abrimos un proceso nuevo para que no afecte a al sistema con sleep 300 &, miramos el pid con ps aux | grep sleep y cogemos el pid que es 1164, y cambiamos la prioridad del proceso con sudo renice -n 10 -p 1164 por ultimo miramos si se ha cambiado la prioridad con ps -o pid,ni,comm -p 1164

```
oscar@srv-base-oscarhernandez:~$ sleep 300 &
[1] 1164
oscar@srv-base-oscarhernandez:~$ ps aux | grep sleep
oscar      1164  0.0  0.0  5684  1920 pts/0    S   14:14   0:00 sleep 300
oscar      1169  0.0  0.1  6544  2304 pts/0    S+  14:15   0:00 grep --color=auto sleep
oscar@srv-base-oscarhernandez:~$ sudo renice -n 10 -p 1164
[sudo] password for oscar:
1164 (process ID) old priority 0, new priority 10
oscar@srv-base-oscarhernandez:~$ ps -o pid,ni,comm -p 1164
  PID  NI COMMAND
  1164  10 sleep
```

lanzar un proceso en segundo plano y enviarlo a primer plano con fg

```
oscar@srv-base-oscarhernandez:~$ sleep 60 &
[1] 1195
oscar@srv-base-oscarhernandez:~$ jobs
[1]+  Running                  sleep 60 &
oscar@srv-base-oscarhernandez:~$ fg %1
sleep 60
```

Usar nice para iniciar un proceso con prioridad baja (por ejemplo, una copia pesada con cp). creamos dos archivos y usamos nice para hacer un proceso prioridad baja de copia de esos dos archivos

```
oscar@srv-base-oscarhernandez:~$ mkdir -p ~/origen
fallocate -l 100M ~/origen/archivo1
fallocate -l 100M ~/origen/archivo2
oscar@srv-base-oscarhernandez:~$ nice -n 19 cp -r ~/origen ~/destino
oscar@srv-base-oscarhernandez:~$
```

fase 3, monitorización y registro de uso de recursos

Usar el comando vmstat y guardar su salida en un archivo /srv/logs/vmstat.log.
primero creamos el archivo y comprobamos que funciona

```
oscar@srv-base-oscarhernandez:~$ cat /srv/logs/vmstat.log
procs -----memory----- --swap-- -----io----- -system-- -----cpu-----
 r b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st gu
 1 0     0 1516720 21432 265328    0    0   135   226 1144    0  0  1 99  0  0  0
```

en el archo de configuración añadimos

```
* /5 * * * * top -b -n 1 | sudo tee -a /srv/logs/top.log > /dev/null
```

por ultimo comprobar si funcionna correctamente

```
oscar@srv-base-oscarhernandez:~$ sudo tail /srv/logs/top.log
1246 root      20  0      0      0      0  0 I   0,0  0,0  0:00.14 kworker+
1251 root      20  0      0      0      0  0 I   0,0  0,0  0:00.00 kworker+
1326 root      20  0      0      0      0  0 I   0,0  0,0  0:00.00 kworker+
1366 root      20  0      0      0      0  0 I   0,0  0,0  0:00.02 kworker+
1382 root      20  0      0      0      0  0 I   0,0  0,0  0:00.06 kworker+
1437 root      20  0    9536   3728   3200 S   0,0  0,2  0:00.00 cron
1438 root      20  0    9532   3984   3328 S   0,0  0,2  0:00.00 cron
1439 root      20  0    2800   1664   1664 S   0,0  0,1  0:00.00 sh
1442 oscar      20  0    2800   1792   1792 S   0,0  0,1  0:00.00 sh
1446 root      20  0      0      0      0  0 I   0,0  0,0  0:00.00 kworker+
```

fase 4, simulación de sobrecarga simulado

Instalar el paquete stress o stress-ng.

```
oscar@srv-base-oscarhernandez:~$ stress --version
stress 1.0.7
oscar@srv-base-oscarhernandez:~$
```

Ejecutar una prueba con carga simulada de CPU, primero miramos el rendimiento de
cpu

```
oscar@srv-base-oscarhernandez:~$ vmstat 1
procs -----memory----- --swap-- -----io----- -system-- -----cpu-----
 r b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st gu
 1 0     0 1137700 25680 621044    0    0   203   234 1140    0  0  1 99  0  0  0
 0 0     0 1137700 25680 621044    0    0    0    0 1259  203  1  1 98  0  0  0
 0 0     0 1137700 25680 621044    0    0    0    0 1138  130  0  1 99  0  0  0
 0 0     0 1137700 25680 621044    0    0    0    0 1075   91  0  1 99  0  0  0
 0 0     0 1137700 25680 621044    0    0    0    0 1092  100  0  1 99  0  0  0
 0 0     0 1137700 25680 621044    0    0    0    0 1129  119  0  1 99  0  0  0
 0 0     0 1137700 25680 621044    0    0    0    0 1083   88  0  1 99  0  0  0
 0 0     0 1137700 25680 621044    0    0    0    0 1074   94  0  1 99  0  0  0
 0 0     0 1137700 25680 621044    0    0    0    0 1106  108  0  1 99  0  0  0
 0 0     0 1137700 25680 621044    0    0    0    0 1095  103  0  1 99  0  0  0
 0 0     0 1137700 25680 621044    0    0    0    0 1073   95  0  1 99  0  0  0
 0 0     0 1137700 25680 621044    0    0    0    0 1090  107  0  1 99  0  0  0
```

luego ejecutamos la prueba durante 60 segundos

```
oscar@srv-base-oscarhernandez:~$ stress-ng --cpu 2 --timeout 60s
stress-ng: info:  [2330] setting to a 1 min, 0 secs run per stressor
stress-ng: info:  [2330] dispatching hogs: 2 cpu
```

y esperamos los resultados

```
oscar@srv-base-oscarhernandez:~$ stress-ng --cpu 2 --timeout 60s
stress-ng: info:  [2330] setting to a 1 min, 0 secs run per stressor
stress-ng: info:  [2330] dispatching hogs: 2 cpu
stress-ng: info:  [2330] skipped: 0
stress-ng: info:  [2330] passed: 2: cpu (2)
stress-ng: info:  [2330] failed: 0
stress-ng: info:  [2330] metrics untrustworthy: 0
stress-ng: info:  [2330] successful run completed in 1 min, 0.01 secs
```