



Universidad
de Huelva



Escuela Técnica Superior de Ingeniería
Universidad de Huelva

Memoria de Prácticas

ADMINISTRACIÓN DE SERVIDORES

Grado en Ingeniería Informática

Autor: Alejandro Gordillo Pedraza

16 de octubre de 2023

Resumen

Esta práctica ha consistido en instalar dos servidores en distintos contenedores y aprender a monitorizar ambos en dos situaciones.

Estas vendrían siendo una en estado normal y otra en un estado de estrés. Durante esta práctica compararemos el estado de los servidores y comprobaremos nuestra capacidad de monitorización y compresión. Cabe destacar que, para la realización de estas práctica y posiblemente próximas prácticas, haremos uso de ProxMox.

Índice general

1. Práctica de monitorización

1.1 Creación el entorno de pruebas.....	4
1.2 Estudio de las capacidades de un servidor.....	4
1.3 Monitorización del servidor	7
1.4 Monitorización del servidor ante situaciones de estrés	12
1.5 Conclusión.....	13

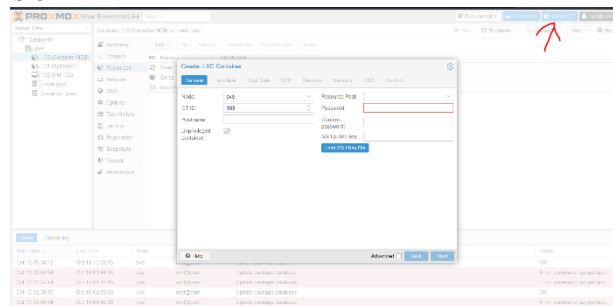
Capítulo 1

Práctica de monitorización

1.1 Creación el entorno de pruebas

Para crear un nuevo contenedor en Proxmox accederemos a nuestra maquina e iniciaremos sesión con el usuario con permisos que vayamos a usar para nuestro contenedor.

Después pulsaremos en el botón de Crear CT y nos abrirá un menú de configuración para la creación del contenedor. Aquí podremos ponerle un nombre y una contraseña a nuestro CT. Captura se puede ver una previsualización antes de darle a finalizar para comprobar si todo lo que hemos puesto está bien.



Seguimos haciendo los siguientes pasos, donde nos pedirá que elijamos una cantidad de cores, el tamaño del disco y el tamaño de la memoria de la que va a disponer nuestro contenedor.

Para finalizar llegaremos al paso final donde comprobaremos que este todo en orden y ya crearemos nuestro Contenedor.

1.2 Estudio de las capacidades de un servidor

Obtenga la información referente a la capacidad de el o los procesadores (puede consultar el fichero `/proc/cpuinfo`).

```
[root@Sysbench:~]# cat /proc/cpuinfo
processor       : 0
vendor_id      : GenuineIntel
cpu family     : 6
model          : 85
model name     : Intel(R) Xeon(R) Silver 4110 CPU @ 2.10GHz
stepping      : 4
microcode     : 0x1
cpu MHz        : 2095.076
cache size     : 16384 KB
physical id    : 0
siblings       : 4
core id        : 0
cpu cores      : 4
apicid         : 0
initial apicid : 0
fpu            : yes
fpu_exception  : yes
cpuid level    : 22
wp             : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fx
sr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon rep_good nopl xtopology cpuid t
sc_known_freq npi pcamuldq vmx ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer
aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch cpuid_fault invpcid_single pti ssbd bna
lghb stibp tpr_shadow vmml flexpriority ept vpid ept_ad fsgbase tsc_adjust bmi1 hle avx2 smep bmi2 erms
s invpcid rtm mpx avx512f avx512dq rdseed adx smap clflushopt clwb avx512cd avx512bw avx512vl xsaveopt x
savec xgetbv1 xsaues arat unip pku ospke arch_capabilities
bugs           : cpu_meltdown spectre_v1 spectre_v2 spec_store_bypass l1tf mds swapsys taa
bogomips       : 4190.15
clflush size   : 64
cache alignment : 64
address sizes   : 40 bits physical, 48 bits virtual
power management:

processor       : 1
vendor_id      : GenuineIntel
cpu family     : 6
model          : 85
model name     : Intel(R) Xeon(R) Silver 4110 CPU @ 2.10GHz
stepping      : 4
microcode     : 0x1
cpu MHz        : 2095.076
cache size     : 16384 KB
physical id    : 0
siblings       : 4
core id        : 2
cpu cores      : 4
apicid         : 2
initial apicid : 2
```

```
root@Sysbench:~# free
              total        used        free      shared  buff/cache   available
Mem:           524288        40380        315360        10560        168548        483908
Swap:          1048576           0         1048576

root@Sysbench:~# vmstat
procs-----memory-----swap-----io-----system-----cpu-----
 r b swpd free buff cache si so bi bo in cs us sy id wa st
0 0 1 0 315128 0 168548 0 0 39 8 7 6 0 0 100 0 0

root@Sysbench:~#
```

```
root@ContainerMDB:~# free
              total        used        free      shared  buff/cache   available
Mem:           524288        38636        38680        10164        446972        485652
Swap:          1048576           0         1048576

root@ContainerMDB:~# vmstat
procs-----memory-----swap-----io-----system-----cpu-----
 r b swpd free buff cache si so bi bo in cs us sy id wa st
0 0 0 0 38540 0 447104 0 0 39 8 7 6 0 0 100 0 0

root@ContainerMDB:~#
```

Los comandos **free** y **vmstat** pueden servir para monitorizar el uso de la memoria. Use ambos comandos para obtener la información referente a la capacidad de la memoria (total de la memoria, memoria virtual usada, espacio destinado a las cachés y buffers, etc).

```
root@Sysbench:~# df
Filesystem            1K-blocks    Used Available Use% Mounted on
/dev/mapper/pve-vm--101--disk--0 8191416 725596 7030008 10% /
none                  492          4      488    1% /dev
udev                 4049500          0 4049500    0% /dev/tty
tmpfs                 4077380          0 4077380    0% /dev/shm
tmpfs                 4077380      10608 4066772    1% /run
tmpfs                 4077380          0 4077380    0% /sys/fs/cgroup
tmpfs                 52432          0   52432    0% /run/user/0

root@Sysbench:~#
```

```
root@ContainerMDB:~# df
Filesystem            1K-blocks    Used Available Use% Mounted on
/dev/mapper/pve-vm--100--disk--0 8191416 1044004 6711600 14% /
none                  492          4      488    1% /dev
udev                 4049500          0 4049500    0% /dev/tty
tmpfs                 4077380          0 4077380    0% /dev/shm
tmpfs                 4077380      10416 4066964    1% /run
tmpfs                 4077380          0 4077380    0% /sys/fs/cgroup
tmpfs                 52432          0   52432    0% /run/user/0

root@ContainerMDB:~#
```

El comando **df** muestra el espacio libre y ocupado en los distintos sistemas de ficheros que tiene montados el sistema, obtenga la información referente a la capacidad de los discos.

```
root@Sysbench:~# df
Filesystem            1K-blocks    Used Available Use% Mounted on
/dev/mapper/pve-vm--101--disk--0 8191416 725596 7030008 10% /
none                  492          4      488    1% /dev
udev                 4049500          0 4049500    0% /dev/tty
tmpfs                 4077380          0 4077380    0% /dev/shm
tmpfs                 4077380      10608 4066772    1% /run
tmpfs                 4077380          0 4077380    0% /sys/fs/cgroup
tmpfs                 52432          0   52432    0% /run/user/0

root@Sysbench:~#
```

```
root@ContainerMDB:~# df
Filesystem            1K-blocks    Used Available Use% Mounted on
/dev/mapper/pve-vm--100--disk--0 8191416 1044004 6711600 14% /
none                  492          4      488    1% /dev
udev                 4049500          0 4049500    0% /dev/tty
tmpfs                 4077380          0 4077380    0% /dev/shm
tmpfs                 4077380      10416 4066964    1% /run
tmpfs                 4077380          0 4077380    0% /sys/fs/cgroup
tmpfs                 52432          0   52432    0% /run/user/0

root@ContainerMDB:~#
```

Los comandos **lsblk** e **iostat** muestran información sobre los distintos sistemas de ficheros montados. En lo que se refiere al nombre del dispositivo, ¿aprecia alguna diferencia?

```
root@Sysbench:~# lsblk
lsblk: dm-1: failed to get device path
lsblk: dm-8: failed to get device path
lsblk: dm-6: failed to get device path
lsblk: dm-4: failed to get device path
lsblk: dm-2: failed to get device path
lsblk: dm-0: failed to get device path
lsblk: dm-7: failed to get device path
lsblk: dm-1: failed to get device path
lsblk: dm-2: failed to get device path
lsblk: dm-0: failed to get device path
lsblk: dm-3: failed to get device path
lsblk: dm-5: failed to get device path
lsblk: dm-3: failed to get device path
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sr0    11:0  1  886M  0 rom
sda     8:0  0 128G  0 disk
|-sda2  8:2  0 512M  0 part
|-sda3  8:3  0 127.5G  0 part
^-sda1  8:1 1007K  0 part

root@Sysbench:~#
```

```
root@ContainerMDB:~# lsblk
lsblk: dm-1: failed to get device path
lsblk: dm-8: failed to get device path
lsblk: dm-6: failed to get device path
lsblk: dm-4: failed to get device path
lsblk: dm-2: failed to get device path
lsblk: dm-0: failed to get device path
lsblk: dm-7: failed to get device path
lsblk: dm-1: failed to get device path
lsblk: dm-2: failed to get device path
lsblk: dm-0: failed to get device path
lsblk: dm-3: failed to get device path
lsblk: dm-5: failed to get device path
lsblk: dm-3: failed to get device path
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sr0    11:0  1  886M  0 rom
sda     8:0  0 128G  0 disk
|-sda2  8:2  0 512M  0 part
|-sda3  8:3  0 127.5G  0 part
^-sda1  8:1 1007K  0 part

root@ContainerMDB:~#
```

Aparecen los mismos errores en los mismos contenedores, creo que pueda ser debido a que la BD de udevdb no este disponible. Puesto que el comando lsblk lee el sistema de archivos sysfs y el sistema udevdb. Dado que me salta un error no encuentro diferencias aparentes.

[root@Sysbench ~]# iostat										[root@ContainerDB ~]# iostat									
Linux 5.4.106-1-pve (Sysbench) 10/18/23 _x86_64_ (4 CPU)										Linux 5.4.106-1-pve (ContainerDB) 10/18/23 _x86_64_ (4 CPU)									
avg-cpu: %user %nice %system %iowait %steal %idle										avg-cpu: %user %nice %system %iowait %steal %idle									
0.01 0.00 0.00 0.00 0.00 99.99										0.01 0.00 0.00 0.00 0.00 99.99									
Device: tps kB_read/s kB_wrtn/s kB_read kB_wrtn										Device: tps kB_read/s kB_wrtn/s kB_read kB_wrtn									
sda0 0.00 0.00 0.00 2096 0										sda0 0.00 0.00 0.00 2096 0									
sda 2.84 105.07 21.64 178046091 36672200										sda 2.84 105.05 21.64 178046091 36672146									
dm-0 0.00 0.00 0.00 3264 4										dm-0 0.00 0.00 0.00 3264 4									
dm-1 2.21 0.17 18.99 283197 32178452										dm-1 2.21 0.17 18.99 283197 32178412									
dm-2 0.00 0.00 0.01 1220 15912										dm-2 0.00 0.00 0.01 1220 15912									
dm-3 0.60 0.61 3.73 1030559 6317264										dm-3 0.60 0.61 3.73 1030559 6317264									
dm-4 0.60 0.61 3.73 1030559 6317264										dm-4 0.60 0.61 3.73 1030559 6317264									
dm-6 0.35 0.55 1.94 927073 3283688										dm-6 0.35 0.55 1.94 927073 3283688									
dm-7 0.25 0.06 1.49 108866 2522396										dm-7 0.25 0.06 1.49 108866 2522396									
dm-8 0.00 0.00 0.00 5325 0										dm-8 0.00 0.00 0.00 5325 0									
[root@Sysbench ~]#										[root@ContainerDB ~]#									

Estos serían los datos en el caso de usar iostat, el cual no funcionaba hasta que no realice la instalación del “paquete” de comandos. “sudo yum install sysstat -y”

Obtenga la información referente a la capacidad de la red.

Para esto haremos uso del comando netstat -s

[root@Sysbench ~]# netstat -s									
Ip:									
1660289 total packets received									
2 with invalid addresses									
0 forwarded									
0 incoming packets discarded									
13070 incoming packets delivered									
9494 requests sent out									
Icmp:									
0 ICMP messages received									
0 input ICMP message failed.									
ICMP input histogram:									
0 ICMP messages sent									
0 ICMP messages failed									
ICMP output histogram:									
Tcp:									
353 active connections openings									
4 passive connection openings									
0 failed connection attempts									
7 connection resets received									
1 connections established									
11937 segments received									
8522 segments send out									
0 segments retransmitted									
1 bad segments received.									
55 resets sent									
InSumErrors: 1									
Udp:									
927 packets received									
0 packets to unknown port received.									
0 packet receive errors									
922 packets sent									
0 receive buffer errors									
0 send buffer errors									
IgnoredMulti: 206									
UdpLite:									
TcpExt:									
307 TCP sockets finished time wait in fast timer									
15 delayed acks sent									
8611 packet headers predicted									
525 acknowledgments not containing data payload received									
524 predicted acknowledgments									
1 congestion windows recovered without slow start by DSACK									
1 other TCP timeouts									
TCPLossProbes: 5									
TCPBacklogCoalesce: 1									
5 DSACKs received									
21 connections reset due to unexpected data									
5 connections reset due to early user close									
IPReversePathFilter: 1									
TCPRecvCoalesce: 6									
TCPOFOQueue: 895									
TCPAutoCorking: 57									
TCPOrigDataSent: 1708									
TCPDelivered: 2058									
TCPAckCompressed: 811									
IpExt:									
InBcastPkts: 211									
InOctets: 136363263									
OutOctets: 765845									
InBcastOctets: 69341									
InNoECTPkts: 1681203									
[root@Sysbench ~]#									

1.3 Monitorización del servidor

Determine los procesos activos y la relación que existe entre ellos.
Compruebe que el servidor Mariadb esté activo.

```
[root@ContainerMDB ~]# systemctl is-active mariadb
unknown
[root@ContainerMDB ~]# systemctl start mariadb.service
[root@ContainerMDB ~]# systemctl is-active mariadb
active
[root@ContainerMDB ~]#
```

Determine el consumo de los diferentes recursos identificando los procesos que más consuman.

```
[root@ContainerMDB ~]# systemctl start mariadb.service
[root@ContainerMDB ~]# systemctl is-active mariadb
active
[root@ContainerMDB ~]# ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  1.0 43568 5396 ?        Ss   Sep28   0:16 /usr/lib/systemd/systemd --system --deserialize 14
root        40  0.0  1.2 39092 6728 ?        Ss   Sep28   0:08 /usr/lib/systemd/systemd-journald
dbus        59  0.0  0.8 58120 4288 ?        Ss   Sep28   0:02 /usr/bin/dbus-daemon --system --address=systemd: --noforroot
logind      85  0.0  0.3 6524 1692 console Ss+  Sep28   0:00 /sbin/agetty --noclear --keep-baud console 115200,38400root
keep-baud 115200,38400,96root 87  0.0  0.8 90968 4692 ?        Ss   Sep28   0:00 login -- root
root        90  0.0  0.5 22736 2660 ?        Ss   Sep28   0:05 /usr/sbin/crond -n
root        244  0.0  0.9 102904 4784 ?       Ss   Sep28   0:01 /sbin/dhclient -1 -q -lf /var/lib/dhclient/dhclient--etroot
root        306  0.0  1.4 218556 7424 ?       Ssl  Sep28   2:52 /usr/sbin/rsyslogd -n
root        311  0.0  0.5 11832 3832 tty1     Ss   Sep28   0:00 bash
root        332  0.0  1.3 133448 7284 tty1     S    Sep28   0:00 sudo su
root        333  0.0  0.7 81992 4872 tty1     S    Sep28   0:00 su
root        334  0.0  0.5 11832 3844 tty1     S+   Sep28   0:00 bash
root       13588  0.0  1.8 155352 9676 ?       Ss   06:34   0:00 sshd: root@pts/2
root       13590  0.0  0.5 11836 3836 pts/2    Ss+  06:34   0:00 -bash
root       13843  0.0  1.8 155352 9788 ?       Rs   08:51   0:00 sshd: root@pts/3
root       13845  0.0  0.5 11836 3840 pts/3    Ss   08:51   0:00 bash
mysql      13846  0.0  0.4 9708 2568 ?        Ss   08:53   0:00 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
mysql      14110  0.0 16.3 969004 85844 ?       Sl   08:53   0:00 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql
[root@ContainerMDB ~]#
```

Verifique si existen clientes de red conectados.

```
[root@ContainerMDB ~]# netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 ContainerMDB:ssh        10.8.2.7:53371          ESTABLISHED
tcp        0      0 ContainerMDB:ssh        10.8.2.7:57532          ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags               Type               I-Node  Path
unix   10      [ ]                 DGRAM              11174775 /dev/log
unix    3      [ ]                 DGRAM              11174767 /run/systemd/notify
unix    2      [ ]                 DGRAM              11174768 /run/systemd/cgroups-agent
unix    2      [ ]                 DGRAM              11174772 /run/systemd/shutdown
unix    4      [ ]                 DGRAM              11174774 /run/systemd/journal/socket
unix    3      [ ]                 STREAM             CONNECTED 11177891
unix    3      [ ]                 STREAM             CONNECTED 11176462 /run/systemd/journal/stdout
unix    2      [ ]                 DGRAM              49877135
unix    2      [ ]                 DGRAM              11177179
unix    3      [ ]                 STREAM             CONNECTED 11176480 /run/systemd/journal/stdout
unix    2      [ ]                 DGRAM              11174861
unix    3      [ ]                 STREAM             CONNECTED 11176488 /run/dbus/system_bus_socket
unix    3      [ ]                 STREAM             CONNECTED 51609965
unix    3      [ ]                 STREAM             CONNECTED 11177185
unix    3      [ ]                 STREAM             CONNECTED 51609966 /run/dbus/system_bus_socket
unix    3      [ ]                 STREAM             CONNECTED 11177172
unix    3      [ ]                 STREAM             CONNECTED 11176747 /run/systemd/journal/stdout
unix    2      [ ]                 DGRAM              11182065
unix    2      [ ]                 DGRAM              51781265
unix    3      [ ]                 STREAM             CONNECTED 11177384
unix    2      [ ]                 DGRAM              51611079
unix    2      [ ]                 DGRAM              11176788
unix    3      [ ]                 STREAM             CONNECTED 11176455
unix    2      [ ]                 DGRAM              51581733
unix    3      [ ]                 STREAM             CONNECTED 11176415
unix    2      [ ]                 DGRAM              11183126
unix    3      [ ]                 STREAM             CONNECTED 11176456
unix    2      [ ]                 DGRAM              11176501
unix    2      [ ]                 DGRAM              11176622
unix    3      [ ]                 STREAM             CONNECTED 11176416 /run/systemd/journal/stdout
[root@ContainerMDB ~]#
```

En las conexiones a internet, todas salen en estado ESTABLISHED , el cual corresponde como conexión establecidas. Por lo que todas las redes están conectadas y disponibles.

Ejecute el comando **iostat** con la opción **-c** para comprobar el estado de uso de la CPU. Haga la monitorización a espacios regulares de 2 segundos durante 20 segundos. Eluso de la CPU, ¿sufré cambios significativos?

```
[root@ContainerMDB ~]# iostat -c 2
Linux 5.4.106-1-pve (ContainerMDB)      10/18/23      _x86_64_      (4 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.01    0.00    0.00    0.00    0.00   99.99

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.00    0.00    0.00    0.00    0.00  100.00

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.00    0.00    0.00    0.00    0.00  100.00

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.25    0.00    0.00    0.00    0.00   99.75

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.00    0.00    0.00    0.00    0.00  100.00

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.00    0.00    0.00    0.00    0.00  100.00

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.25    0.00    0.00    0.00    0.00   99.75

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.25    0.00    0.00    0.00    0.00   99.75

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.00    0.00    0.00    0.00    0.00  100.00

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.00    0.00    0.00    0.00    0.00  100.00

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.00    0.00    0.00    0.00    0.00  100.00

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.00    0.00    0.00    0.00    0.00  100.00

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.25    0.00    0.00    0.00    0.00   99.75

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.00    0.00    0.00    0.00    0.00  100.00

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.00    0.00    0.00    0.00    0.00  100.00
```

Como podremos comprobar en la captura siguiente, el servidor está casi en reposo porque no sufre grandes cambios muy significativos. Algunas veces está durante muy poco tiempo en modo usuario, pero casi siempre la CPU no está gestionando ninguna solicitud.

Mediante la opción **-d** del comando **iostat**, verifique a periodos regulares de 2 segundos el estado de las operaciones de entrada y salida.

```
[root@ContainerMDB ~]# iostat -d 2
Linux 5.4.106-1-pve (ContainerMDB)      10/18/23      _x86_64_      (4 CPU)
```

Device:	tps	kB_read/s	kB_wrtn/s	kB_read	kB_wrtn
sdd0	0.00	0.00	0.00	2096	0
sda	2.84	104.92	21.62	178671535	36820350
dm-0	0.00	0.00	0.00	3264	4
dm-1	2.21	0.17	18.95	283197	32273844
dm-2	0.00	0.00	0.01	1304	16424
dm-3	0.60	0.61	3.74	1036143	6375968
dm-4	0.60	0.61	3.74	1036143	6375968
dm-6	0.35	0.55	1.96	932593	3333628
dm-7	0.25	0.06	1.49	108930	2530564
dm-8	0.00	0.00	0.00	5325	0

Device:	tps	kB_read/s	kB_wrtn/s	kB_read	kB_wrtn
sdd0	0.00	0.00	0.00	0	0
sda	2.00	0.00	16.00	0	32
dm-0	0.00	0.00	0.00	0	0
dm-1	4.00	0.00	16.00	0	32
dm-2	0.00	0.00	0.00	0	0
dm-3	0.50	0.00	2.00	0	4
dm-4	0.50	0.00	2.00	0	4
dm-6	0.00	0.00	0.00	0	0
dm-7	0.50	0.00	2.00	0	4
dm-8	0.00	0.00	0.00	0	0

Device:	tps	kB_read/s	kB_wrtn/s	kB_read	kB_wrtn
sdd0	0.00	0.00	0.00	0	0
sda	5.50	384.00	12.00	768	24
dm-0	0.00	0.00	0.00	0	0
dm-1	2.00	0.00	10.00	0	20
dm-2	0.00	0.00	0.00	0	0
dm-3	0.50	0.00	2.00	0	4
dm-4	0.50	0.00	2.00	0	4
dm-6	0.50	0.00	2.00	0	4
dm-7	0.00	0.00	0.00	0	0
dm-8	0.00	0.00	0.00	0	0

Device:	tps	kB_read/s	kB_wrtn/s	kB_read	kB_wrtn
sdd0	0.00	0.00	0.00	0	0
sda	1.50	0.00	6.00	0	12
dm-0	0.00	0.00	0.00	0	0
dm-1	2.00	0.00	8.00	0	16
dm-2	0.00	0.00	0.00	0	0
dm-3	0.00	0.00	0.00	0	0
dm-4	0.00	0.00	0.00	0	0
dm-6	0.00	0.00	0.00	0	0
dm-7	0.00	0.00	0.00	0	0
dm-8	0.00	0.00	0.00	0	0

Device:	tps	kB_read/s	kB_wrtn/s	kB_read	kB_wrtn
sdd0	0.00	0.00	0.00	0	0
sda	1.50	0.00	12.00	0	24
dm-0	0.00	0.00	0.00	0	0
dm-1	1.50	0.00	10.00	0	20
dm-2	0.00	0.00	0.00	0	0
dm-3	0.50	0.00	2.00	0	4
dm-4	0.50	0.00	2.00	0	4
dm-6	0.00	0.00	0.00	0	0
dm-7	0.50	0.00	2.00	0	4
dm-8	0.00	0.00	0.00	0	0

Device:	tps	kB_read/s	kB_wrtn/s	kB_read	kB_wrtn
sdd0	0.00	0.00	0.00	0	0
sda	0.50	0.00	2.00	0	4
dm-0	0.00	0.00	0.00	0	0
dm-1	0.00	0.00	0.00	0	0
dm-2	0.00	0.00	0.00	0	0
dm-3	0.50	0.00	2.00	0	4
dm-4	0.50	0.00	2.00	0	4
dm-6	0.50	0.00	2.00	0	4
dm-7	0.00	0.00	0.00	0	0
dm-8	0.00	0.00	0.00	0	0

Use el comando **w**, para obtener la carga de la CPU en los últimos 1, 5 y 15 minutos.
¿Se encuentra el sistema sobrecargado?

```
[root@ContainerMDB ~]# w
 09:43:17 up 19 days, 17:35,  3 users,  load average: 0.00, 0.00, 0.00
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
root      tty1                28Sep23 15days  0.03s  0.01s  bash
root      pts/2    10.8.2.7        09:11   31:50  0.00s  0.00s  -bash
root      pts/3    10.8.2.7        09:40    0.00s  0.00s  0.00s  w
[root@ContainerMDB ~]#
```

Como podemos apreciar en la imagen la CPU no se encuentra sobrecargada en ningún momento.

El comando **iostat** no devuelve información sobre la carga específica de una CPU, para estos casos usamos el comando **mpstat**. Compare la salida de ambos comandos.

Para la comparación he buscado alguna información acerca de los comandos, la cual adjunto para ayudarnos.

%irq: muestra el porcentaje de tiempo empleado por la CPU o las CPU para dar servicio a las interrupciones de hardware.

%soft: muestra el porcentaje de tiempo empleado por la CPU o las CPU para reparar las interrupciones del software

%guest: muestra el porcentaje de tiempo empleado por la CPU o las CPU para ejecutar un procesador virtual

```
[root@ContainerMDB ~]# mpstat 2
Linux 5.4.106-1-pve (ContainerMDB)      10/18/23      _x86_64_      (4 CPU)

10:35:01   CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
10:35:03   all     0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   100.00
10:35:05   all     0.25    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   99.75
10:35:07   all     0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   100.00
10:35:09   all     0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   100.00
10:35:11   all     0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   100.00
10:35:13   all     0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   100.00
10:35:15   all     0.25    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   99.75
10:35:17   all     0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   100.00
10:35:19   all     0.25    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   99.75
10:35:21   all     0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   100.00
10:35:23   all     0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   100.00
10:35:25   all     0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   100.00
10:35:27   all     0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   100.00
10:35:29   all     0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   100.00
10:35:31   all     0.25    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   99.75
10:35:33   all     0.25    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   99.75
10:35:35   all     0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   100.00
```

La utilidad **sar** (system activity report"), recopila información de distintos parámetros del sistema. Esta aplicación incluye dos shell scripts. El primer script, sa1,recopila datos de forma regular, mientras que el script sa2 se utiliza para crear los informes resumidos (uno por día en **/var/log/sa/sarDD**. Ambos scripts se ejecutan usando cron. Si queremos obtener los datos en tiempo real podemos invocar directamente al comando sar.

```
[root@ContainerMDB ~]# sar -u 2
Linux 5.4.106-1-pve (ContainerMDB)      10/18/23      _x86_64_      (4 CPU)

10:48:30      CPU      %user      %nice      %system      %iowait      %steal      %idle
10:48:32      all       0.00       0.00       0.00       0.00       0.00      100.00
10:48:34      all       0.25       0.00       0.00       0.00       0.00      99.75
10:48:36      all       0.00       0.00       0.00       0.00       0.00     100.00
10:48:38      all       0.25       0.00       0.00       0.00       0.00      99.75
10:48:40      all       0.00       0.00       0.00       0.00       0.00     100.00
10:48:42      all       0.00       0.00       0.00       0.00       0.00     100.00
10:48:44      all       0.25       0.00       0.00       0.00       0.00      99.75
10:48:46      all       0.00       0.00       0.00       0.00       0.00     100.00
10:48:48      all       0.00       0.00       0.00       0.00       0.00     100.00
10:48:50      all       0.50       0.00       0.00       0.00       0.00      99.50
10:48:52      all       0.00       0.00       0.00       0.00       0.00     100.00
10:48:54      all       0.25       0.00       0.00       0.00       0.00      99.75
10:48:56      all       0.00       0.00       0.00       0.00       0.00     100.00
10:48:58      all       0.00       0.00       0.00       0.00       0.00     100.00
10:49:00      all       0.25       0.00       0.00       0.00       0.00      99.75
```

1.4 Monitorización del servidor ante situaciones de estrés

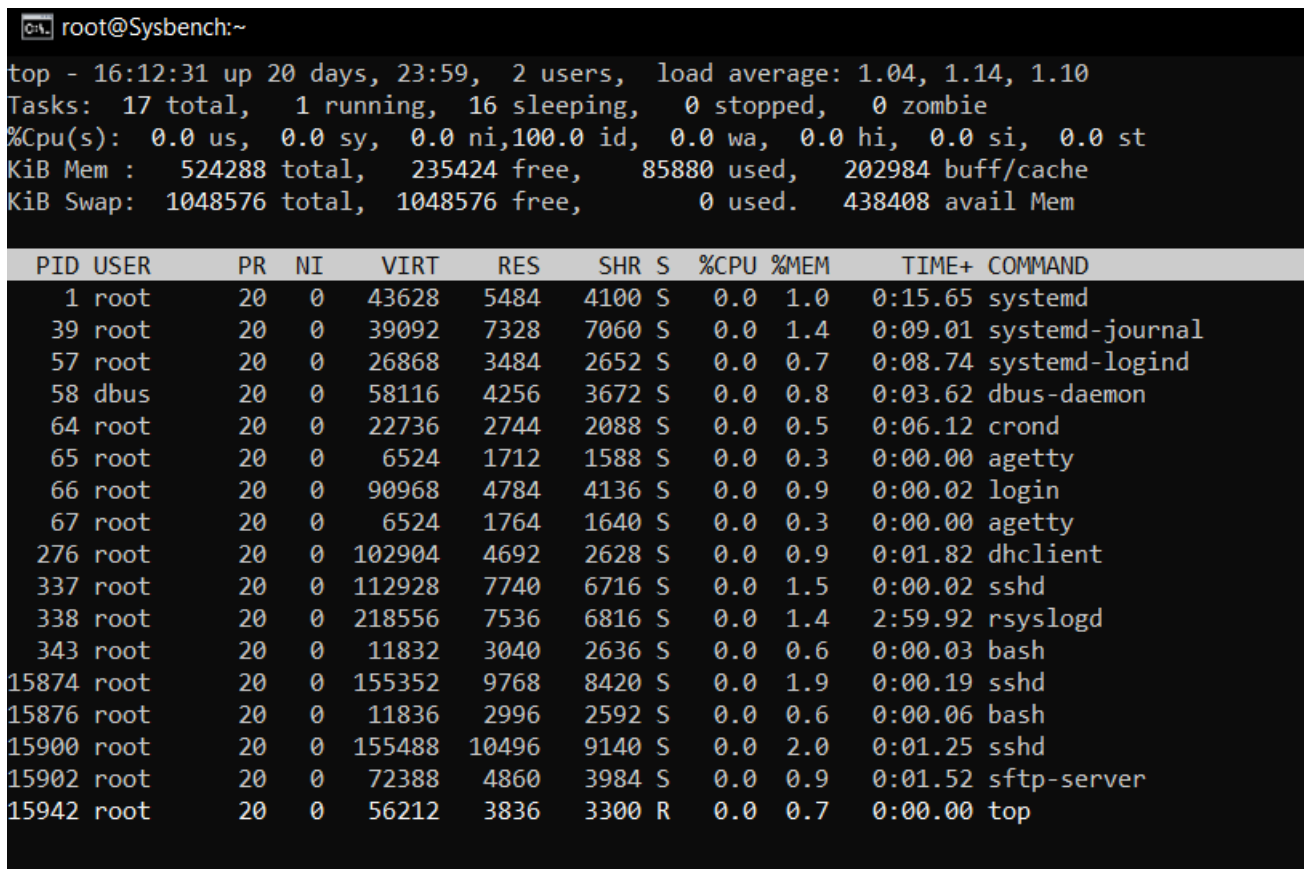
Cree el esquema de base de datos sbtest y dótele de 5 tablas con 50000 filas cada una

Todos los comandos que pondré en los siguientes apartados y en este se ejecutaron en sysbench aunque los cambios se hicieron en Mariadb.

```
/usr/share/sysbench/oltp_read_write.lua --threads=4 --mysql-host=192.168.50.176  
--mysql-user=root --mysql-password=**** --mysql-port=3306 --tables=5 --table-  
size=50000 prepare
```

Aquí se puede ver que la ip de mariadb es 192.168.50.27 y que el usuario donde se crearon las tablas era el root.

En la siguiente captura podemos ver el nivel de estrés de la base de datos de mariadb mediante el comando top.



PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	43628	5484	4100	S	0.0	1.0	0:15.65	systemd
39	root	20	0	39092	7328	7060	S	0.0	1.4	0:09.01	systemd-journal
57	root	20	0	26868	3484	2652	S	0.0	0.7	0:08.74	systemd-logind
58	dbus	20	0	58116	4256	3672	S	0.0	0.8	0:03.62	dbus-daemon
64	root	20	0	22736	2744	2088	S	0.0	0.5	0:06.12	crond
65	root	20	0	6524	1712	1588	S	0.0	0.3	0:00.00	agetty
66	root	20	0	90968	4784	4136	S	0.0	0.9	0:00.00	login
67	root	20	0	6524	1764	1640	S	0.0	0.3	0:00.00	agetty
276	root	20	0	102904	4692	2628	S	0.0	0.9	0:01.82	dhclient
337	root	20	0	112928	7740	6716	S	0.0	1.5	0:00.02	sshd
338	root	20	0	218556	7536	6816	S	0.0	1.4	2:59.92	rsyslogd
343	root	20	0	11832	3040	2636	S	0.0	0.6	0:00.03	bash
15874	root	20	0	155352	9768	8420	S	0.0	1.9	0:00.19	sshd
15876	root	20	0	11836	2996	2592	S	0.0	0.6	0:00.06	bash
15900	root	20	0	155488	10496	9140	S	0.0	2.0	0:01.25	sshd
15902	root	20	0	72388	4860	3984	S	0.0	0.9	0:01.52	sftp-server
15942	root	20	0	56212	3836	3300	R	0.0	0.7	0:00.00	top

En la imagen anterior se puede apreciar que la memoria apenas se usa, y ya ni hablemos de la Cpu pero me equivoque de contenedor al usar el comando top la primera vez y no pude comprobar correctamente el consumo. Debería dar unos valores alrededor de 35% de memoria y cerca del 1% de cpu si no me equivoco.

Haga un test de tipo oltp_read_only durante 150 segundos empleando 3 hebras.

En este ejercicio el comando que utilicé fue:

```
/usr/share/sysbench/oltp_read_only.lua --threads=3 --events=0 --time=150 --mysql-host=192.168.50.176 --mysql-user=root --mysql-password=**** --mysql-port=3306 --tables=5 --table-size=50000 --range-selects=off --db-ps-mode=disable --report-interval=1 run
```

The image contains two terminal screenshots. The left screenshot shows the output of the 'top' command in a container named 'ContainerMDB'. It displays system statistics and a list of processes. The 'mysql' process (PID 14110) is shown with 169.0% CPU usage and 47.0% memory usage. The right screenshot shows the output of the 'sysbench' command in a container named 'Sysbench'. It displays a series of performance metrics over 69 seconds, showing a steady increase in transactions per second (tps) from approximately 2619 to 2697.

```
root@ContainerMDB:~
top - 16:18:28 up 21 days, 11 min,  2 users,  load average: 3.39, 1.79, 1.33
Tasks: 20 total,  1 running, 19 sleeping,  0 stopped,  0 zombie
%Cpu(s): 69.3 us,  0.0 sy,  0.0 ni, 30.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 524288 total,  364 free, 267016 used, 256908 buff/cache
KiB Swap: 1048576 total, 1048572 free,  4 used. 257272 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
14110 mysql    20   0 1253848 246368 14836 S 169.0  47.0   28:39.07 mysqld
   1 root      20   0  43568   5388   4016 S   0.0   1.0    0:18.19 systemd
  40 root      20   0  39092   7228   6944 S   0.0   1.4    0:09.13 systemd-journal
  59 dbus      20   0  58120   4232   3656 S   0.0   0.8    0:03.73 dbus-daemon
  82 root      20   0  26868   3256   2428 S   0.0   0.6    0:08.77 systemd-logind
  85 root      20   0   6524   1652   1528 S   0.0   0.3    0:00.00 agetty
  86 root      20   0   6524   1676   1548 S   0.0   0.3    0:00.00 agetty
  87 root      20   0  90968   4636   3988 S   0.0   0.9    0:00.03 login
  90 root      20   0  22736   2660   2012 S   0.0   0.5    0:06.45 crond
 244 root      20   0 102904   4196   2136 S   0.0   0.8    0:02.00 dhclient
 305 root      20   0 112928   7836   6808 S   0.0   1.5    0:00.04 sshd
 306 root      20   0 218556   7508   6576 S   0.0   1.4   3:02.75 rsyslogd
 311 root      20   0  11832   3032   2632 S   0.0   0.6    0:00.01 bash
 332 root      20   0 133448   6848   5716 S   0.0   1.3    0:00.01 sudo
 333 root      20   0  81992   4024   3484 S   0.0   0.8    0:00.00 su
 334 root      20   0  11832   3044   2624 S   0.0   0.6    0:00.02 bash
13946 mysql    20   0   9708   2568   2280 S   0.0   0.5    0:00.01 mysqld_safe
16123 root      20   0 155352   9792   8440 S   0.0   1.9    0:00.05 sshd
16125 root      20   0  11836   2936   2672 S   0.0   0.6    0:00.00 bash
16139 root      20   0  56208   3848   3304 R   0.0   0.7    0:00.01 top

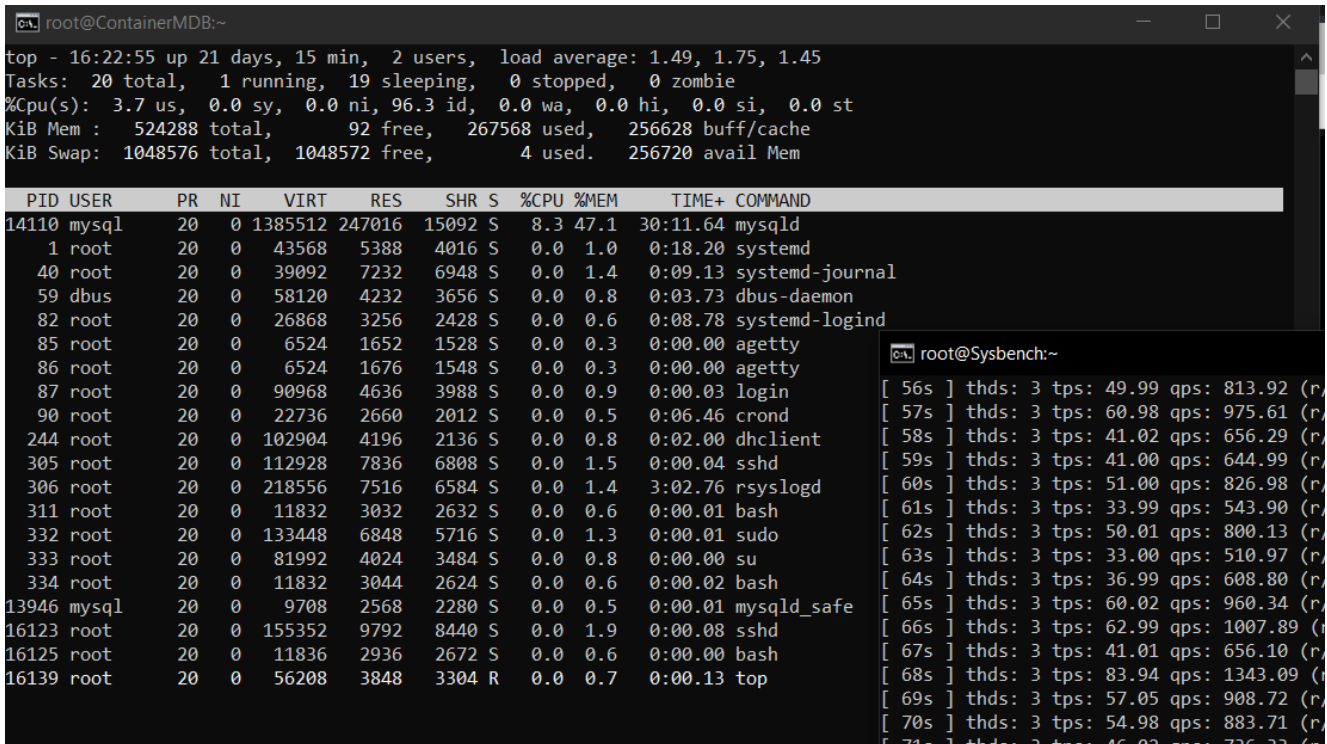
root@Sysbench:~
[ 42s ] thds: 3 tps: 2619.84 qps: 31
[ 43s ] thds: 3 tps: 2432.01 qps: 29
[ 44s ] thds: 3 tps: 2594.05 qps: 31
[ 45s ] thds: 3 tps: 2581.52 qps: 30
[ 46s ] thds: 3 tps: 2406.80 qps: 28
[ 47s ] thds: 3 tps: 2277.60 qps: 27
[ 48s ] thds: 3 tps: 2255.11 qps: 27
[ 49s ] thds: 3 tps: 2318.96 qps: 27
[ 50s ] thds: 3 tps: 2365.26 qps: 28
[ 51s ] thds: 3 tps: 2427.78 qps: 29
[ 52s ] thds: 3 tps: 2417.99 qps: 29
[ 53s ] thds: 3 tps: 2355.95 qps: 28
[ 54s ] thds: 3 tps: 2269.39 qps: 27
[ 55s ] thds: 3 tps: 2304.84 qps: 27
[ 56s ] thds: 3 tps: 2392.16 qps: 28
[ 57s ] thds: 3 tps: 2596.01 qps: 31
[ 58s ] thds: 3 tps: 2281.99 qps: 27
[ 59s ] thds: 3 tps: 2317.88 qps: 27
[ 60s ] thds: 3 tps: 2379.96 qps: 28
[ 61s ] thds: 3 tps: 2301.50 qps: 27
[ 62s ] thds: 3 tps: 2351.24 qps: 28
[ 63s ] thds: 3 tps: 2269.43 qps: 27
[ 64s ] thds: 3 tps: 2343.93 qps: 28
[ 65s ] thds: 3 tps: 2398.39 qps: 28
[ 66s ] thds: 3 tps: 2528.05 qps: 30
[ 67s ] thds: 3 tps: 2427.99 qps: 29
[ 68s ] thds: 3 tps: 2603.97 qps: 31
[ 69s ] thds: 3 tps: 2697.99 qps: 32
```

En este podemos observar cómo la CPU ha pasado de estar en casi reposo a un 172,4% aunque la memoria se ha mantenido casi igual que antes.

Haga un test de tipo oltp_read_write durante 150 segundos empleando 3 hebras.

Por último, el comando aquí era el mismo que en el otro pero esta vez era de lectura y escritura:

```
/usr/share/sysbench/oltp_read_write.lua --threads=3 --events=0 --time=150 --mysql-host=192.168.50.176 --mysql-user=root --mysql-password=**** --mysql-port=3306 --tables=5 --table-size=50000 --range_selects=off --db-ps-mode=disable --report-interval=1 run
```



The screenshot shows two terminal windows. The top window, titled 'root@ContainerMDB:~', displays the output of the 'top' command, showing system statistics and a list of running processes. The bottom window, titled 'root@Sysbench:~', shows the output of the sysbench test, displaying performance metrics over time.

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
14110	mysql	20	0	1385512	247016	15092	S	8.3	47.1	30:11.64	mysqld
1	root	20	0	43568	5388	4016	S	0.0	1.0	0:18.20	systemd
40	root	20	0	39092	7232	6948	S	0.0	1.4	0:09.13	systemd-journal
59	dbus	20	0	58120	4232	3656	S	0.0	0.8	0:03.73	dbus-daemon
82	root	20	0	26868	3256	2428	S	0.0	0.6	0:08.78	systemd-logind
85	root	20	0	6524	1652	1528	S	0.0	0.3	0:00.00	agetty
86	root	20	0	6524	1676	1548	S	0.0	0.3	0:00.00	agetty
87	root	20	0	90968	4636	3988	S	0.0	0.9	0:00.03	login
90	root	20	0	22736	2660	2012	S	0.0	0.5	0:06.46	crond
244	root	20	0	102904	4196	2136	S	0.0	0.8	0:02.00	dhclient
305	root	20	0	112928	7836	6808	S	0.0	1.5	0:00.04	sshd
306	root	20	0	218556	7516	6584	S	0.0	1.4	3:02.76	rsyslogd
311	root	20	0	11832	3032	2632	S	0.0	0.6	0:00.01	bash
332	root	20	0	133448	6848	5716	S	0.0	1.3	0:00.01	sudo
333	root	20	0	81992	4024	3484	S	0.0	0.8	0:00.00	su
334	root	20	0	11832	3044	2624	S	0.0	0.6	0:00.02	bash
13946	mysql	20	0	9708	2568	2280	S	0.0	0.5	0:00.01	mysqld_safe
16123	root	20	0	155352	9792	8440	S	0.0	1.9	0:00.08	sshd
16125	root	20	0	11836	2936	2672	S	0.0	0.6	0:00.00	bash
16139	root	20	0	56208	3848	3304	R	0.0	0.7	0:00.13	top

Time	thds	tps	qps
56s	3	49.99	813.92
57s	3	60.98	975.61
58s	3	41.02	656.29
59s	3	41.00	644.99
60s	3	51.00	826.98
61s	3	33.99	543.90
62s	3	50.01	800.13
63s	3	33.00	510.97
64s	3	36.99	608.80
65s	3	60.02	960.34
66s	3	62.99	1007.89
67s	3	41.01	656.10
68s	3	83.94	1343.09
69s	3	57.05	908.72
70s	3	54.98	883.71
71s	3	46.02	736.33

En este caso baja considerablemente el consumo de cpu pero aumenta levemente el consumo de memoria.

1.5 Conclusión.

Para Finalizar en esta primera practica hemos podido aprender la creación de contenedores en ProxMox e instalar nuestro entorno de trabajo en el mismo. También hemos aprendido a como se encuentra/monitoriza una base de datos tanto en estado de reposo como en estado de estrés. Esto último permitiéndonos ver como respondo a estas circunstancias.

En conclusión, lo que puedo apreciar es que no es bueno ninguno de los extremos, porque si tenemos el servidor siempre en casi reposo quiere decir que estamos malgastando recursos y dinero en nuestro entorno de trabajo, lo cual no es bueno ni a nivel personal si es un entorno propio, ni a nivel de empresa. Y tampoco es nada bueno que nuestro servidor se encuentre siempre en estado de estrés exprimiéndole hasta el último recurso, si este fuera el caso durante bastante tiempo y no en momentos puntuales deberíamos hacer un estudio y ver que necesitamos ampliar del mismo para

un funcionamiento correcto de este.