Cuando administramos máquinas virtuales Debian en Proxmox resulta muy útil disponer de una **consola serie independiente de la red, SSH y del entorno gráfico**. De esta forma podremos acceder a la máquina incluso aunque tengamos problemas de red, falle el servidor SSH o estemos utilizando una tarjeta gráfica física mediante PCI passthrough. Proxmox permite añadir un puerto serie virtual a una máquina virtual y utilizarlo como consola interactiva desde su interfaz web o directamente desde el terminal del nodo Proxmox.

Para disponer de una consola serie completamente funcional tendremos que configurar correctamente **tres elementos diferentes**:

1. GRUB, para poder visualizar y controlar el gestor de arranque mediante el puerto serie.
2. El kernel de Linux, para poder visualizar los mensajes de arranque y la salida que posteriormente podremos consultar mediante dmesg.
3. systemd, para disponer de un prompt de inicio de sesión mediante ttyS0.

Es importante diferenciar estos tres elementos, ya que podemos encontrarnos con situaciones en las que GRUB se visualice correctamente por la consola serie pero, una vez iniciado el kernel, desaparezca toda la salida y nunca aparezca el prompt de inicio de sesión.

Agregar un puerto serie a la máquina virtual

En primer lugar tendremos que añadir un puerto serie virtual a nuestra máquina virtual.

Desde la interfaz web de Proxmox accederemos a «Máquina virtual» > «Hardware» > «Botón Agregar» > «Puerto serie»

Seleccionaremos 0 (A menos que antes ya lo hayamos seleccionado por alguna otra razón). También podemos añadirlo directamente desde el terminal del nodo Proxmox:

```
qm set ID_VM --serial0 socket
```

Por ejemplo, para una máquina virtual con ID 100:

```
qm set 100 --serial0 socket
```

Utilizar serial0 como consola de Proxmox

Si queremos que al abrir la consola de la máquina virtual desde la interfaz web de Proxmox aparezca directamente el puerto serie, configuraremos serial0 como dispositivo de consola. Desde el nodo Proxmox podemos hacerlo mediante:

```
qm set ID_VM --vga serial0
```

La configuración de la máquina virtual contendrá entonces:

```
serial0: socket
vga: serial0
```

Configurar vga como serial0 no crea una tarjeta gráfica virtual adicional. Simplemente indica a Proxmox que debe utilizar el puerto serie como consola integrada para esa máquina virtual.

Utilizar serial0 junto con una GPU mediante PCI passthrough

Esta configuración resulta especialmente útil cuando utilizamos una tarjeta gráfica física mediante PCI passthrough. En este tipo

de máquinas virtuales es habitual encontrarnos inicialmente con:

```
vga: none
```

...ya que no queremos añadir una tarjeta gráfica virtual de Proxmox. Podemos sustituir esa configuración por:

```
vga: serial0
```

...sin interferir con la GPU física que tengamos asignada mediante PCI passthrough. Por ejemplo, una configuración podría contener:

```
hostpci0: 0000:03:00.0,pcie=1,x-vga=1
serial0: socket
vga: serial0
```

De esta forma tendremos dos salidas independientes:

```
GPU mediante PCI passthrough
  |
  +--- Monitor físico
    |
    +--- Entorno gráfico de Debian


serial0
  |
  +--- Consola web de Proxmox
    |
    +--- ttyS0
      |
      +--- Arranque de Linux
      +--- Mensajes del kernel
      +--- Login en modo texto
```

Habilitar el inicio de sesión mediante ttyS0

Dentro de Debian tendremos que habilitar un getty sobre el primer puerto serie para que systemd muestre un prompt de inicio de sesión mediante ttyS0. Ejecutaremos:

```
sudo systemctl enable --now serial-getty@ttyS0.service
```

Podemos comprobar su estado mediante:

```
systemctl status serial-getty@ttyS0.service
```

Si el servicio está correctamente iniciado deberíamos disponer posteriormente de un prompt similar al siguiente:

```
Debian GNU/Linux 13 servidor ttyS0

servidor login:
```

Sin embargo, habilitar únicamente este servicio **no garantiza** que podamos ver todo el proceso de arranque. También tendremos que indicar explícitamente al kernel de Linux que utilice ttyS0 como consola.

Configurar el kernel para utilizar ttyS0

Este es uno de los pasos más importantes de toda la configuración. Si el kernel no recibe el parámetro correspondiente, podremos ver GRUB o incluso los primeros mensajes del firmware UEFI, pero la salida desaparecerá cuando Linux empiece a arrancar. Un síntoma típico es que la consola serie se quede detenida mostrando algo parecido a:

```
EFI stub: Loaded initrd from LINUX_EFI_INITRD_MEDIA_GUID device path
```

En realidad el sistema continúa arrancando con normalidad, pero el kernel deja de enviar su salida al puerto serie. Para solucionarlo editaremos:

```
sudo nano /etc/default/grub
```

Si, por ejemplo, tenemos actualmente:

```
GRUB_CMDLINE_LINUX="net.ifnames=0 biosdevnames=0"
```

...añadiremos las dos consolas:

```
GRUB_CMDLINE_LINUX="net.ifnames=0 biosdevnames=0 console=tty0 console=ttyS0,115200n8"
```

Si no tenemos ningún otro parámetro que necesitemos conservar, también podemos utilizar:

```
GRUB_CMDLINE_LINUX="console=tty0 console=ttyS0,115200n8"
```

La opción tty0 mantiene disponible la consola local, mientras que ttyS0 activa la consola serie utilizando una velocidad de 115200 baudios, 8 bits de datos, sin paridad y un bit de parada.

Después regeneraremos la configuración de GRUB:

```
sudo update-grub
```

y reiniciaremos:

```
sudo reboot
```

Comprobar que el kernel está recibiendo la configuración

Este paso nos permite confirmar que los parámetros configurados en GRUB están llegando realmente al kernel.

Una vez iniciado Debian ejecutaremos:

```
cat /proc/cmdline
```

Deberíamos encontrar entre los parámetros:

```
console=tty0 console=ttyS0,115200n8
```

Por ejemplo:

```
BOOT_IMAGE=/boot/vmlinuz-6.12.107+deb13-amd64 root=UUID=XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX ro net.ifnames=0
biosdevnames=0 console=tty0 console=ttyS0,115200n8
```

Si los parámetros console no aparecen en /proc/cmdline, significa que **el kernel no está utilizando la consola serie**, aunque GRUB sí pueda mostrarse correctamente por ella. Esta comprobación resulta especialmente útil para diagnosticar situaciones en las que la consola parece funcionar durante el arranque inicial pero deja de mostrar información justo cuando empieza a ejecutarse Linux.

Mostrar también GRUB mediante la consola serie

Si además queremos poder visualizar e interactuar con el menú de GRUB desde serial0, tendremos que configurar el propio gestor de arranque para utilizar el puerto serie.

Editaremos nuevamente:

```
sudo nano /etc/default/grub
```

...y podremos utilizar una configuración como la siguiente:

```
GRUB_CMDLINE_LINUX_DEFAULT=""

GRUB_CMDLINE_LINUX="net.ifnames=0 biosdevnames=0 console=tty0 console=ttyS0,115200n8"

GRUB_TERMINAL="console serial"

GRUB_SERIAL_COMMAND="serial --unit=0 --speed=115200 --word=8 --parity=no --stop=1"
```

Si no necesitamos los parámetros net.ifnames y biosdevnames, podemos dejarlos fuera:

```
GRUB_CMDLINE_LINUX_DEFAULT=""

GRUB_CMDLINE_LINUX="console=tty0 console=ttyS0,115200n8"

GRUB_TERMINAL="console serial"

GRUB_SERIAL_COMMAND="serial --unit=0 --speed=115200 --word=8 --parity=no --stop=1"
```

Después actualizaremos GRUB:

```
sudo update-grub
```

y reiniciaremos la máquina virtual:

```
sudo reboot
```

Las tres capas de la consola serie

Para evitar confusiones podemos resumir la configuración en tres partes claramente diferenciadas.

```
GRUB mediante serial0
    |
    +--- GRUB_TERMINAL="console serial"
    |
```

```
    +--- GRUB_SERIAL_COMMAND="serial ..."


Kernel y dmesg mediante serial0
    |
    +--- console=tty0
    |
    +--- console=ttyS0,115200n8


Inicio de sesión mediante serial0
    |
    +--- serial-getty@ttyS0.service
```

Necesitamos estas tres capas si queremos disponer de una consola serie completamente funcional desde el inicio del arranque hasta el prompt de autenticación.

Comprobar la configuración de la máquina virtual en Proxmox

Desde el nodo Proxmox podemos comprobar la configuración de nuestra máquina virtual mediante:

```
qm config ID_VM
```

Para una máquina virtual que utilice la consola serie deberíamos encontrar, como mínimo:

```
serial0: socket
vga: serial0
```

Si además utilizamos una GPU mediante PCI passthrough tendremos algo parecido a:

```
hostpci0: 0000:03:00.0,pcie=1,x-vga=1
serial0: socket
vga: serial0
```

Acceder a la consola desde el terminal del nodo Proxmox

No estamos obligados a utilizar la interfaz web de Proxmox para acceder a serial0.

También podemos conectarnos directamente desde el shell del nodo Proxmox mediante:

```
qm terminal ID_VM
```

Por ejemplo:

```
qm terminal 100
```

Nos conectaremos directamente al mismo ttyS0 que utiliza la consola web.

Comprobar que ttyS0 existe dentro de Debian

Si seguimos teniendo problemas podemos comprobar que el kernel haya detectado correctamente el puerto serie virtual.

```
dmesg | grep ttyS
```

También podemos comprobar directamente la existencia del dispositivo:

```
ls -l /dev/ttyS0
```

Y verificar nuevamente el getty asociado:

```
systemctl status serial-getty@ttyS0.service
```

Con esta configuración tendremos una **consola administrativa de emergencia completamente independiente de la red y del entorno gráfico**. Si utilizamos una GPU mediante PCI passthrough podremos seguir empleándola normalmente desde un monitor físico mientras mantenemos ttyS0 disponible desde Proxmox para diagnosticar el arranque, consultar los mensajes del kernel o iniciar sesión en modo texto.