

Instalando VirtualBox en Debian 9 Stretch

Sí, lo sabemos: **VirtualBox** es de Oracle y sí, algún día lo pueden restringir a privativo (de hecho los "complementos" tienen licencia privativa) **pero vamos a seguir aprovechando este hipervisor mientras su licencia sea libre**. Encontramos que es el más fácil de manejar y de ser necesario (como nos ha ocurrido antes) podremos manejarlo cómodamente por la línea de comandos. Hoy traemos su instalación en el novedoso **Debian 9 Stretch** en una portátil con procesador de 32 bits debido a un programita que verifica que el procesador sea de 32 bits ¡VirtualBox no lo puede engañar! (en realidad nadie es infalible).

Introducción.

Nuestra portátil con procesador de 32 bits, doble núcleo, se conecta de manera inalámbrica a nuestra red de área local, y allí tenemos un usuario debidamente creado, así que desempolvemos nuestros conocimientos sobre [SSH en este artículo](#) para tener conexión continua con ese ordenador (por supuesto, también sirve una conexión temporal, haremos una excepción a la regla). Después colocamos nuestras credenciales y listo, estamos conectados. Por nuestra parte usaremos **sudo** constantemente, ustedes allá pueden ejecutar "**su -**" para no tener que escribirlo a cada rato.

Descargando e instalando la llave de seguridad.

VirtualBox tiene su página web en protocolo **https** así que tranquilamente ejecutamos **wget** con la debida confianza de que está encriptada (mientras viaja) la información que pondremos en nuestros discos duros:

```
wget -q https://www.virtualbox.org/download/oracle_vbox_2016.asc -O oracle.asc
```

A continuación escribimos el comando **apt-key add** para agregar a la lista de claves de **Debian** de esta manera al descargar cualquier programa que vayamos a instalar nuestro sistema operativo aplicará los algoritmos necesarios y nos garantizará que lo que descargamos está correcto y que no proviene de un tercero.

```
sudo apt-key add oracle.asc
```

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

Nosotros colocamos el archivo con el nombre "oracle.asc" pero bien pueden ustedes utilizar una sola línea si borramos ese nombre de fichero y a la final del anterior código le agregamos un caracter tubería "|" seguido del siguiente comando **apt-key add**:

```
wget -q https://www.virtualbox.org/download/oracle_vbox_2016.asc -O- | apt-key add -
```

De cualquiera de las dos maneras quedará la llave instalada (a menos que tanga algún inconveniente como nos sucedió a nosotros, al final os explicamos).

Agregando el repositorio de VirtualBox.

En esta oportunidad nos decantamos por instalar en un fichero aparte la dirección de VirtualBox, como para variar y practicar, vamos a comentarlo paso por paso:

- Como bien es sabido por todos y todas **apt** es el programa predeterminado en **Debian** (y la mayoría de sus *distros*) como el encargado de mantener actualizado el software, vía terminal de comandos, en nuestros equipos.
- Cuando instalamos **Debian** y por cualquier razón no tengamos conexión con Internet, este sistema operativo colocará por defecto en el directorio **/etc/apt/** un fichero llamado **sources.list** donde especificará que instalamos desde un CD o DVD (o un dispositivo que haga las veces de unidad óptica) y nos pedirá amablemente que lo insertemos cada vez que vayamos a instalar un software que necesitemos.
- Para poder tener la versión más reciente y aprobada por **Debian** desde luego que tendremos que conectarnos por Internet: lo que tenemos que hacer es comentar las líneas que apuntan hacia los dispositivos que usamos para instalar con un "#" al principio de cada línea (ATENCIÓN: **Debian** viene en tres DVD's, recomendamos tenerlos siempre, haberlos descargado, porque nunca sabemos si falla el Internet *además que ahorra ancho de banda si instalamos primero desde disco local y luego actualizamos por Internet: los archivos que no hayan cambiado no tenemos necesidad de descargarlos ¿nos siguen el hilo?*). Dicho archivo quedaría **parecido o similar a esto**:

```
# # deb cdrom:[Debian GNU/Linux 9.2.0 _Stretch_ - Official i386 DVD Binary-1 20171007-13:33]/ stretch main # deb cdrom:[Debian GNU/Linux 9.2.0 _Stretch_ - Official i386 DVD Binary-1 20171007-13:33]/ stretch main # Line commented out by installer because it failed to verify: deb http://security.debian.org/debian-security stretch/updates main # Line commented out by installer because it failed to verify: deb-src http://security.debian.org/debian-security stretch/updates main # stretch-updates, previously known as 'volatile' # A network mirror was not selected duri
```

```
ng install. The following entries # are provided as examples, but you should amend them as appropriate # for your mirror of choice. # deb http://deb.debian.org/debian/ stretch-updates main deb-src http://deb.debian.org/debian/ stretch-updates main # Backports repository # deb http://deb.debian.org/debian stretch-backports main contrib non-free # deb http://deb.debian.org/debian stretch-backports-sloppy main contrib non-free
```

- Generalmente viene anotado las direcciones de los repositorios principales de **Debian** pero bien podemos buscar los *repositorios espejos* que estén más cercanos a nosotros (o al menos los que sean más rápidos a la hora de descargar).
- A su vez, dentro de la carpeta **/etc/apt/sources.list.d/** (sí, lo sabemos, es confuso pero *recuerden que la letra "d" después del punto es nemotécnico y significa "directory" o "directorio" en nuestro idioma*) podremos colocar tantos ficheros necesitemos o gustemos, y donde cada uno de ellos contendrán las direcciones de los repositorios necesarios para otras aplicaciones. Acá le colocamos el nombre de **"virtualbox.list"** que es el nombre de la aplicación, pero también podría ser el nombre de uno o varios repositorios, etcétera; las combinaciones son infinitas.
- En el fichero **"virtualbox.list"** escribiremos esto (usad vuestro editor de texto -sin interfaz gráfica- favorito: **nano, vi, vim, etc.**):

```
deb http://download.virtualbox.org/virtualbox/debian stretch contrib
```

- Guardamos el fichero y vamos a la siguiente sección.

Instalando VirtualBox.

- Escribimos **"sudo apt-get update"**.
- Al rato, cuando termine, colocamos **"apt install virtualbox-5.1"** o **"apt-get install virtualbox-5.1"**.
- Hasta acá debería llegar el trabajo, pero no es así si estamos rabajando con **Debian**, no: con este sistema operativo se debe tener siempre los ojos bien abiertos y la cafetera (o el termo) bien caliente, veamos.

Compilando los kernels en VirtualBox.

Acá recibimos dos tipos de mensaje, y aunque el primero dice ser una advertencia, es un mensaje [que ya nosotros conocemos muy bien](#): no hemos compilado los *kernels* para que funcione **VirtualBox**. El segundo mensaje se refiere a que no pudo iniciar la interfaz gráfica *¡pues claro, estamos conectados por SSH!*

Instalando module-assistant.

Lo primero que haremos es instalar **module-assistant** una herramienta GNU que no tiene mayor documentación ([acá hay una pequeña orientación](#)):

module-assistant is the command-line tool for handling module-source packages that have been prepared for the Debian distribution. It helps users to build and install module package(s) easily for one or more custom kernels.

Como podemos leer en inglés: *ayuda a los usuarios a construir e instalar paquete(s) modulares fácilmente para uno o más **kernels** personalizados.* ¡Pues que acá nosotros no tenemos ningún *kernel* personalizado! Pero el asunto es el siguiente (muchas veces los gringos son muy parcos en las descripciones) :

- Sí, es cierto, podemos tener *kernels* personalizados bien por nosotros mismos o por terceros ¡de hecho hasta hacen competencias de ordenadores especialmente cosntruidos para a ver quién compila más rápido!
- (Lo más común) salen nuevas versiones de *kernels*.
- (Se encadena con el punto anterior) ya sea *kernel* nuevo, normal o personalizado **nuestras aplicaciones bien se beneficiarán compilando justo a la medida de nuestras máquinas** ([consultad el punto 5 este artículo sobre 10 cosas que no sabías sobre GNU/Linux](#)).

Así que vamos e instalamos:

Instalando build-essential

También debemos instalar **build-essential** el cual garantiza que el entorno del compilador está correcto (pudiéramos decir que se encarga de los prerrequisitos). **¡ATENCIÓN! este software es más "gordito" y se lleva 26 megabytes a descargar y más de 100 megabytes en disco duro** aunque los ordenadores que tenemos actualmente son muy potentes igual no me deja de sorprender la cantidad de paquetes que debe instalar. Garantizar un buen entorno para compilar, nos atrevemos a decir, tiene tanto trabajo como compilar en sí mismo.

Compilando los *kernel* a VirtualBox.

Pues eso, que casi, casi que estamos listos para comenzar a usar VirtualBox, lo que tenemos que hacer ahora es ejecutar:

```
m-a prepare
```

¿Y dice que falta uno por actualizar? **Pues le damos *apt-get upgrade*, veamos qué falta:**

¡Faltaba *libxcursor*! Pues como decíamos, para compilar pues...

```
Obteniendo los fuentes de la versión del núcleo: 4.13.0-0.bpo.1-686-pae
apt-get install linux-headers-4.13.0-0.bpo.1-686-pae Leyendo lista de p
aquetes... Hecho Creando árbol de dependencias Leyendo la información
de estado... Hecho E: No se ha podido localizar el paquete linux-headers
-4.13.0-0.bpo.1-686-pae E: No se pudo encontrar ningún paquete usando «*
» con «linux-headers-4.13.0-0.bpo.1-686-pae» E: No se pudo encontrar nin
gún paquete con la expresión regular «linux-headers-4.13.0-0.bpo.1-686-pa
e» apt-get install build-essential Leyendo lista de paquetes... Hecho
Creando árbol de dependencias Leyendo la información de estado... Hech
o build-essential ya está en su versión más reciente (12.3). 0 actualiz
ados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados. ¡He
cho!
```

Volvemos a ejecutar **m-a prepare** y ahora sí vamos con **VirtualBox**.

Desesperadamente buscando a "vboxdrv".

En [nuestra entrada anterior](#) ya teníamos instalado, en uno de esos inacabables tanteos de instalación; a **vboxdrv.sh** pues bien, acá en nuestro flamante **Debian 9** pues que *aparentemente* no está. Con ayuda de **DuckDuckGo**, el potente buscador que respeta nuestra privacidad, encontramos este artículo donde dan cuenta de la ubicación de dicho fichero que contiene el guion a ejecutar.

```
root@AcerAspireOne:/# /usr/lib/virtualbox/vboxdrv.sh setup vboxdrv.sh: S
topping VirtualBox services. vboxdrv.sh: Building VirtualBox kernel modu
les. This system is not currently set up to build kernel modules (system
extensions). Running the following commands should set the system up co
```

```
rrectly: apt-
get install linux-
headers-4.13.0-0.bpo.1-686-pae
  (Th
e last com
mand may fail if you
r system is not fully updated.)
  apt-get install linux-
headers-686-pae vboxdrv.sh: failed: Look at /var/log/vbox-
install.log to find out what went wrong.
```

¡y surge otro error más interesante!

Usando un *kernel* demasiado "avanzado".

En la sección anterior resaltamos en rojo la orden recomendada: esencialmente **VirtualBox** nos está pidiendo el código fuente de nuestro *kernel* para poder compilar... el problema es que **linux-headers-4.13.0-0.bpo.1-686-pae** *no se encuentra en los repositorios normales sino que pertenece a los "backports", software que aún no ha sido probado correctamente*. Iniciamos nuestra búsqueda en los paquetes de *debian.org*, encontramos y ejecutamos `wget` a las ubicaciones que están más cerca de nosotros acá en Venezuela (desde Brasil):

```
wget http://ftp.br.debian.org/debian/pool/main/l/linux/linux-headers-4.13
.0-0.bpo.1-686_4.13.13-1~bpo9+1_i386.deb  dpkg -i linux-
headers-4.13.0-0.bpo.1-686_4.13.13-1~bpo9+1_i386.deb
```

Pero entonces declara que necesita lo siguiente:

```
dpkg: problemas de dependencias impiden la configuración de linux-headers
-4.13.0-0.bpo.1-686:  linux-headers-4.13.0-0.bpo.1-686 depende de linux-
headers-4.13.0-0.bpo.1-common (= 4.13.13-1~bpo9+1); sin embargo:  El paq
uete `linux-headers-4.13.0-0.bpo.1-common' no está instalado.  linux-hea
ders-4.13.0-0.bpo.1-686 depende de linux-kbuild-4.13; sin embargo:  El p
aquete `linux-kbuild-4.13' no está instalado.
```

Así que debemos instalar **linux-headers-4.13.0-0.bpo.1-common** y **linux-kbuild-4.13**, vamos de

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

nuevo a Brasil y regresamos:

```
wget http://download.unesp.br/linux/debian/pool/main/l/linux/linux-headers-4.13.0-0.bpo.1-common_4.13.13-1~bpo9+1_all.deb
wget http://ftp.br.debian.org/debian/pool/main/l/linux/linux-kbuild-4.13_4.13.13-1~bpo9+1_i386.deb
dpkg -i linux-kbuild-4.13_4.13.13-1~bpo9+1_i386.deb
dpkg -i linux-headers-4.13.0-0.bpo.1-686_4.13.13-1~bpo9+1_i386.deb
```

Ha sido laboriosa la instalación de **VirtualBox** pero no nos rendimos, así que ahora que si tenemos el código fuente tanto genéricos, su específico para arquitectura i386 y "kbuild" para trabajarlos vemos que es la hora de por fin volver a instalar a **VirtualBox** (esta vez desde [/sbin/vboxconfig](#) que es otra manera de continuar la instalación, intentaremos por todas las formas y maneras):

```
root@AcerAspireOne:/home/javier# /sbin/vboxconfig
Created symlink /etc/systemd/system/multi-user.target.wants/vboxdrv.service ? /lib/systemd/system/vboxdrv.service.
Created symlink /etc/systemd/system/multi-user.target.wants/vboxballoonctrl-service.service ? /lib/systemd/system/vboxballoonctrl-service.service.
Created symlink /etc/systemd/system/multi-user.target.wants/vboxautostart-service.service ? /lib/systemd/system/vboxautostart-service.service.
Created symlink /etc/systemd/system/multi-user.target.wants/vboxweb-service.service ? /lib/systemd/system/vboxweb-service.service.
vboxdrv.sh: Stopping VirtualBox services.
vboxdrv.sh: Building VirtualBox kernel modules.
This system is currently not set up to build kernel modules.
Please install the Linux kernel "header" files matching the current kernel for adding new hardware support to the system.
The distribution packages containing the headers are probably: linux-headers-686-pae linux-headers-4.13.0-0.bpo.1-686-pae
This system is currently not set up to build kernel modules.
Please install the Linux kernel "header" files matching the current kernel for adding new hardware support to the system.
The distribution packages containing the headers are probably:
: linux-headers-686-pae linux-headers-4.13.0-0.bpo.1-686-pae
There were problems setting up VirtualBox. To re-start the set-up process, run /sbin/vboxconfig as root.
root@AcerAspireOne:/home/javier#
```

¡Pues nada, sigue igual! Ya estando un poco cansados, hacemos seguimiento de las dependencias y ejecutamos lo que creemos nos hace falta:

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

```
wget http://ftp.br.debian.org/debian/pool/main/l/linux/linux-headers-4.13.0-0.bpo.1-686-pae_4.13.13-1~bpo9+1_i386.deb dpkg -i linux-headers-4.13.0-0.bpo.1-686-pae_4.13.13-1~bpo9+1_i386.deb
wget http://ftp.cl.debian.org/debian/pool/main/l/linux-latest/linux-headers-686-pae_4.13+86~bpo9+1_i386.deb dpkg -i linux-headers-686-pae_4.13+86~bpo9+1_i386.deb
```

El que persevera ALCANZA (también se cansa, ya vamos a un merecido descanso):

```
root@AcerAspireOne:/home
/javier# /sbin/vboxconfig Created symlink /etc/systemd/system/multi-user.target.wants/vboxdrv.service
Created symlink /lib/systemd/system/vboxdrv.service.
Created symlink /etc/systemd/system/multi-user.target.wants/vboxballoonctrl-service.service
Created symlink /etc/systemd/system/multi-user.target.wants/vboxautostart-service.service
Created symlink /etc/systemd/system/multi-user.target.wants/vboxweb-service.service.
vboxdrv.sh: Stopping VirtualBox services.
vboxdrv.sh: Building VirtualBox kernel modules.
vboxdrv.sh: Starting VirtualBox services. root@AcerAspireOne:/home/javier#
```

En el proceso anterior, compilar los *kernels* tardó como tres minutos, claro es un ordenador del 2008, ya tiene casi 10 años de vida que para los ordenadores es una eternidad pero a la final nos queda el error de que no puede lanzar o encontrar la interfaz gráfica:

Que se soluciona, por supuesto, sentándose físicamente en la consola de la máquina de marras o bien por escritorio remoto.

VirtualBox tiene comandos de texto para levantar sin entorno gráfico las máquinas virtuales, *incluso tiene opción para crearlas y configurarlas incluso "por escritorio remoto" en la instalación del sistema operativo (para este ejemplo debemos hacer notar que las "tarjetas de red" de las máquinas virtuales deberán estar en "modo puente" y tener su propia dirección IP en nuestra red de área local "LAN")*. El proceso esencialmente consiste en configurar y crear la máquina virtual apuntando la ruta a la imagen ISO del sistema operativo a instalar, arrancar la máquina desde esa unidad de almacenamiento **y conectarse a ella mediante VirtualBox que nos pasará todo lo que suceden en la instalación del sistema operativo hasta que todo esté instalado**. Luego deberemos configurar ese sistema operativo para que acepte conexiones por escritorio remoto. *Para conectarnos "por remoto" al instalar el sistema operativo apuntaremos a la dirección IP de la máquina real o anfitrión. Cuando ya tengamos el sistema operativo instalado y configurado apuntaremos a la dirección IP del ordenador virtual.*

Conclusión.

Nunca debemos rendirnos, si acaso perdermos batallas pero no la guerra ya que en el software libre SIEMPRE disponemos del código fuente, siempre tendremos una solución, por muy compleja que ésta sea. Nuestra siguiente entrada en este vuestro humilde blog versará sobre la creación de máquinas virtuales por línea de comandos.

Fuentes consultadas.

En idioma castellano.

En idioma inglés.

- «How to Install VirtualBox [on Debian 9 Stretch](#)» by Xiao Guoan.
- «How to Install VirtualBox 5.1 on [CentOS 7 / RHEL 7 / Fedora 26](#)» by Pradeep Kumar.
- «[How to build a kernel module with DKMS on Linux](#)» by Dan Nanni.

Código fuente de vboxdrv.sh