Instale Slackware en un servidor Dedibox BareMetal de online.net

Este tutorial explica cómo instalar y arrancar Slackware Linux en online.net Dedibox BareMetal Server Start Family. Se centra en los servidores a los que no tiene acceso físico y aquellos que no admiten la exposición remota del hardware (es decir, no hay KVM sobre IP). La configuración de este tipo de servidores es posible a través de Web interface. Afortunadamente o no, esta interfaz no admite de forma nativa la instalación de Slackware. No te preocupes, nos las arreglaremos sin embargo.

Aunque las primeras partes del tutorial son específicas del servidor Dedibox, el resto es más genérico. Esto significa que la información proporcionada aquí se aplica por igual a otras ofertas de alojamiento, que solo deben proporcionar un sistema operativo de rescate similar (más información sobre el sistema operativo de rescate más abajo). Por el contrario, las primeras partes se aplican bien a las distribuciones de Linux que no sean Slackware. Si necesita un control detallado sobre el proceso de instalación en el servidor Dedibox, está en un buen lugar.

1. Dedibox rescue OS

El sistema operativo de rescate es un sistema operativo en el que puede iniciar su servidor utilizando la interfaz web. Este sistema operativo le permite realizar tareas de mantenimiento, si su sistema operativo principal no se inicia o si necesita acceder al servidor, mientras pasa por alto su sistema operativo principal. Hay varias versiones del sistema operativo de rescate para elegir; estaremos usando uno basado en Linux.

Puede (y debe) conectarse al sistema operativo de rescate a través de SSH. Una de las características del sistema operativo de rescate es que es volátil, lo que significa que los cambios realizados en él se pierden después del reinicio. Pero lo que es más importante, las claves de host SSH se regeneran cada vez que se inicia el sistema operativo de rescate, lo que hace que la key fingerprint del host SSH también se modifique, cada vez que reinicie. Esto hace que la verificación de la autenticidad del servidor sea un poco incómoda.

Una de las formas de verificar la autenticidad del host es abrir el ticket de soporte técnico de online.net, solicitando que se le proporcione la key fingerprint host del sistema de rescate. Y debido a que la key fingerprint cambia con cada reinicio, no es deseable reiniciar la máquina durante la instalación de Slackware. No es un gran drama, es posible instalar con éxito y luego arrancar Slackware en todo su esplendor con solo un reinicio al final del viaje. También puede conectarse sin asegurarse de la autenticidad del host, tener un juego e incluso una instalación de prueba. Una vez que esté familiarizado con el entorno, puede reiniciar nuevamente en el sistema operativo de rescate, solicitar asistencia técnica para la key fingerprint, limpiar el disco duro y realizar la instalación y configuración final. Solo ten en cuenta que sin verificar la autenticidad del host, eres susceptible a los ataques MITM. Aunque el sistema de archivos de rescate es volátil y puede limpiar el contenido del disco para asegurarse de que esté limpio y seguro, aún así, el propio hardware podría ser el objetivo del ataque. Y hoy en día, el hardware en realidad está ejecutando software (firmware) más a menudo.

Afortunadamente, existe un método alternativo que no implica el compromiso del soporte técnico. Puede leer los detalles completos en LinuxQuestions.org: Verificación de la autenticidad del host

(SSH) después de iniciar sesión, por encima del terminal en serie seguro y accesible. En el tutorial en sí, solo me centraré en cómo hacer las cosas.

2. Consola serial

La interfaz web proporciona una opción de consola de serie, que le permite interactuar con el hardware del servidor. Es de uso limitado con el sistema operativo de rescate (es decir, no es posible iniciar sesión), pero se puede utilizar completamente con el sistema operativo principal. Si se configura correctamente, le permitirá ver el proceso de arranque del sistema operativo principal y también tener un inicio de sesión de terminal, en caso de que la conexión SSH no esté disponible por cualquier motivo. (Pero no esperes nada lujoso, es solo una consola de serie después de todo).

Resulta que, en realidad, es posible iniciar sesión en el sistema operativo de rescate utilizando la consola de serie. Al menos funciona cuando se selecciona Ubuntu 16.04 amd64 como sistema operativo de rescate. Desafortunadamente, este tutorial fue escrito con el supuesto de que no era posible (¡lo que en realidad fue el caso!). Por esa razón, algunos pasos son más complejos de lo que serían si la consola en serie estuviera disponible desde el principio. Esto se aplica especialmente al paso de verificación de autenticidad SSH del servidor. Desafortunadamente, no actualizaré el tutorial para dar cabida a este descubrimiento.

3. Hacer que el sistema operativo de rescate esté disponible

Si acaba de comprar un servidor nuevo, debe venir sin un sistema operativo preinstalado. Desafortunadamente, significa también que el sistema operativo de rescate no está disponible todavía.

Para habilitar el sistema operativo de rescate: mediante la interfaz web, vaya a la página de administración de su servidor: *Server → Server list → (server name) Manage*. Si la única opción que ve es `[INSTALL]` , entonces el sistema operativo de rescate no está disponible y primero debe instalar uno de los sistemas operativos ofrecidos. (Nota de Purist: elija sabiamente el sistema operativo, ya que el ícono asociado con este sistema operativo representará su sistema Slackware a partir de entonces). Una vez que se complete la instalación del sistema operativo, se le presentarán más opciones: `[REBOOT]` , `[RESCUE]` , `[SERIAL_CONSOLE]` y `[INSTALL]` . Estás listo para irte ahora. Simplemente no arranques el sistema operativo de rescate todavía.

4. Preparando el acceso del sistema operativo de rescate a través de SSH

Tenga cuidado de no sobrescribir las claves SSH de su estación de trabajo en ~/.ssh

Ahora estamos a punto de crear dos pares de claves SSH. Se usará un par para iniciar sesión en el sistema operativo de rescate y el otro (solo parte de la clave pública) para la autenticación del sistema operativo de rescate. Escriba lo siguiente en su estación de trabajo:

```
$ mkdir dedibox_rescue_os_keys
$ cd dedibox_rescue_os_keys
$ ssh-keygen -t rsa -f login_key -N ''
$ ssh-keygen -t rsa -f auth_key -N ''
```

Mediante la interfaz web, vaya a la página de administración de claves SSH: *inicie sesión como (username) → SSH keys* y agregue *auth_key.pub* y *login_key.pub* public llaves. Tenga en cuenta que después de la adición exitosa, la página no se actualiza automáticamente y para ver las claves, debe hacer clic en el enlace *SSH keys* en el lado izquierdo.

Cargar las claves a través de la interfaz web tiene el efecto de ponerlas a disposición del sistema operativo de rescate. Cuando se inicia el sistema operativo de rescate, las claves simplemente se agregan al archivo *~/.ssh/authorized_keys* para el usuario en particular.

Estas claves pueden eliminarse de manera segura de la interfaz web una vez que haya terminado con la instalación de Slackware.

5. Arrancar y conectarse al sistema operativo de rescate

Usando la interfaz web, vaya a la página de administración de su servidor: *Server → Server list → (server name) Manage*. Habilite la consola serie haciendo clic en el botón `[SERIAL_CONSOLE]` y siga la guía. La consola se abrirá en una nueva pestaña del navegador. Tenga en cuenta que la conexión de la consola tiene un tiempo de caducidad, por lo que no se quedará allí para siempre. Además, no aparecen muchos mensajes relacionados con el arranque del sistema operativo de rescate, pero sigue siendo mejor que nada. El OS Rescue tarda un tiempo en iniciarse y poder verlo puede permitirle calmarse un poco (solo tenga en cuenta que hay un tiempo entre el momento en que los mensajes dejan de aparecer y el momento en que realmente puede conectarse a través de SSH). La razón por la que necesita iniciar `[SERIAL_CONSOLE]` antes de `[RESCUE]` es que el botón `[SERIAL_CONSOLE]` desaparece una vez que inicie el sistema operativo de rescate.

Aún es posible acceder a la consola serie, incluso si el botón `[SERIAL_CONSOLE]` no está visible. Solo tiene que ir a https://console.online.net/en/server/state/XXXXX/bmc address, reemplazando *XXXXX* parte con el número real de su servidor.

Ahora vuelva a la página de administración del servidor y haga clic en "[RESCUE]'". Cuando se le solicite la selección del sistema operativo, seleccione `[Ubuntu_16.04_amd64]` y luego haga clic en `[CLICK_HERE_TO_LAUNCH_THE_RESCUE_SYSTEM]` . Después de un tiempo, se le presentarán los detalles necesarios para conectarse al sistema operativo de rescate a través de SSH. También puede cambiar a la ventana de salida `[SERIAL_CONSOLE]` para monitorear lo que está sucediendo.

Una vez que el sistema operativo de rescate esté completamente iniciado, conéctese desde su estación de trabajo:

```
$ ssh username@x.y.z.w -i ./login_key
```

Cuando se le pregunte *Are you sure you want to continue connecting (yes/no)?*, responda que sí.

Podría ser una buena idea agregar temporalmente la key fingerprint del host del SSH de rescate al archivo / /.ssh/known_hosts en su estación de trabajo. Esto le permitirá volver a iniciar sesión (por ejemplo, en el caso de una conexión rota) en el sistema operativo de rescate sin la necesidad de repetir el procedimiento de autenticación que se describe a continuación. Solo recuerde eliminar la key fingerprint una vez que haya terminado o si el procedimiento de autenticación descrito a continuación falla.

Una vez que haya iniciado sesión, para autenticar el sistema operativo de rescate, en el lado del servidor escriba lo siguiente:

```
$ cat ~/.ssh/authorized_keys
```

y luego compare las claves públicas impresas en el terminal con las claves públicas *auth_key.pub* y *login_key.pub* que generó anteriormente. Si las dos claves entre el servidor y la estación de trabajo coinciden, está seguro de ir. (Una vez más, si desea comprender los detalles, consulte el LinuxQuestions.org mencionado anteriormente thread).

Ahora, en el lado del servidor, le sugiero que primero inicie el programa *screen* y luego inicie sesión como root (la contraseña se encuentra en la página de detalles de la conexión del sistema operativo de rescate):

```
$ screen
$ sudo su -
```

(Siendo paranoico, cambio las contraseñas de usuario y root proporcionadas por*online.net*).

No entraré en detalles del programa *screen* , pero la razón por la que queremos usarlo es su capacidad para mantener abierto el terminal remoto, incluso si la conexión SSH (o más bien la red) se interrumpe o si cierra accidentalmente la Ventana. Normalmente, tal evento rompería el proceso de instalación. Si eso le sucede a usted, y utiliza *screen* , puede recuperar el terminal remoto (¡con todos

los comandos iniciados aún en ejecución!), Simplemente conectándose a través de SSH y luego volviendo a conectar el llamado *screen* sesión:

```
$ screen -r
$ # Sometimes, detaching the session first is needed:
$ screen -rd SESSION_PID
```

Todo lo anterior significa también que no puede detener la ejecución remota de comandos simplemente cerrando la ventana del terminal local; *screen* sesión se mantendrá en el extremo remoto hasta que la cierre explícitamente. Por cierto, también puede usar esta funcionalidad para reducir el tráfico de red durante la fase de instalación, es decir, una vez que los paquetes se empezaron a instalar y no necesitan atención, puede desconectarse de la sesión y volver a adjuntarla un tiempo más tarde para verificar el estado. Utiliza la combinación de teclado Ctrl+A D para separar de *screen* sesión. Verifique en Internet cómo usar el programa *screen* (la página de manual es en realidad enormemente larga).

6. Configuración del entorno del instalador de Slackware (chroot)

La principal motivación detrás de este HOWTO es el hecho de que *online.net* no proporciona un método directo de instalación de Slackware en sus servidores Dedibox. Y eso es genial! El punto es, si tal método existiera, sería algo desconocido para un usuario de Slackware: un instalador basado en GUI / Web. Mientras que ser colocado en el shell del instalador de Slackware le permite instalar y configurar el sistema de la manera que desee (TM). Todas las opciones avanzadas están disponibles sin mucha molestia. Y vamos a usarlos todos. : - ^

El procedimiento detallado a continuación es conocido y común. Vamos a *chroot* a desempacar Slackware *initrd*image y ejecutamos *setup* desde allí.

Nuestro Slackware *chroot* se quedará sin el sistema operativo de rescate de Ubuntu, que se está quedando sin sistema de archivos basado en RAM. El tamaño del sistema de archivos RAM es de ~ 8.0 GiB (que depende de la cantidad total de RAM del sistema). Sugerencia: 8.0 GiB es más que suficiente para mantener el árbol completo de paquetes de Slackware, si es necesario (es decir, si desea descargar los paquetes antes de ejecutar *setup*).

Ahora, vamos a configurar el instalador de Slackware*chroot*:

```
$ mkdir -p ~/slackware-chroot
$ cd ~/slackware-chroot
$ wget https://slackware.osuosl.org/slackware64-14.2/isolinux/initrd.img
$ gunzip -cd initrd.img | cpio -dvim
$ mount --bind /proc proc
$ mount --bind /sys sys
```

```
$ mount --bind /dev dev
$ mount --bind /dev/pts dev/pts
$ mount --bind /run run
$ touch etc/resolv.conf
$ mount --bind /etc/resolv.conf etc/resolv.conf
```

And run the chroot:

```
$ chroot ~/slackware-chroot /bin/bash --login
$ cd
```

El montaje de /etc/resolv.conf proporciona DNS a *chroot*.

Una vez que haya *chrooted*, es posible que desee jugar con la variable de entorno *TERM* , que influirá en la forma en que se muestran *dialogs*. De forma predeterminada, *TERM = linux* y no funciona bien si usa *screen* o si se conecta desde un emulador de terminal que se ejecuta bajo X.

Para obtener los mejores resultados con *screen*, use lo siguiente:

```
$ export TERM=screen
```

y si no usa *screen* , pero se está conectando desde X:

```
$ export TERM=xterm
```

Finalmente, para el terminal virtual (VT), deje el valor predeterminado:

```
$ export TERM=linux
```

```
¡Bienvenido al disco de instalación de Slackware Linux!
```

7. Particionamiento

Ahora puede particionar el disco a su gusto, con dos excepciones. En este tutorial, usaré una partición separada para ser montada en el directorio */boot* . Este directorio contendrá los archivos de configuración del kernel, initrd y bootloader. Lo que nos lleva a la segunda excepción: como gestor de arranque, voy a usar *syslinux* instalado en MBR, lo que significa que el disco tiene que usar el tipo de etiqueta MBR. Si necesitas GPT, estás por tu cuenta. 

El sistema de archivos de la partición */ boot* debe ser compatible con *syslinux* . *ext4* hará el trabajo bien. Y cuando se trata del tamaño, 128 MiB es suficiente.

Las siguientes instrucciones destruirán los datos en el disco..

Puede usar su herramienta de partición favorita, por ejemplo, *fdisk* , *cfdisk* , etc. Voy a utilizar *parted* . Tenga en cuenta que el disco debe estar en estado “unmanaged”, es decir, los servicios como LVM2 o el software RAID (mdadm) deben estar desactivados. Tuve muchos dolores de cabeza al liberar el disco desde el control de LVM2, cuando estaba jugando con él (pero lo logré ).

Por cierto, el disco ya contendrá la tabla de particiones, que se creó cuando instalamos uno de los sistemas operativos de reserva para habilitar el sistema operativo de rescate. La tabla de particiones se puede limpiar con los siguientes comandos:

```
$ dd count=1 bs=512 conv=notrunc if=/dev/zero of=/dev/sda
$ partprobe
```

Los siguientes comandos parted crearán la etiqueta MBR y la partición / boot:

```
$ parted /dev/sda mklabel msdos
$ # Start at 1 MiB in the hope of a correct alignment:
$ parted -a optimal /dev/sda mkpart primary 1MiB 129MiB
$ # Set bootable flag:
$ parted /dev/sda set 1 boot on
```

Con la partición */ boot [/dev/sda1]* en su lugar, puede particionar el espacio restante de la forma que mejor se adapte a sus necesidades. Voy a usar LVM2 para administrar el disco, así que creo una partición grande [*/dev/sda2*] que se pasará a LVM2. El procedimiento de configuración para habilitar LVM2 se describe en Apéndice A. El siguiente comando parted creará la partición requerida:

```
$ # Passing "-a optimal" automatically aligns at the last sectors of the
disk.
$ # The start and end offsets have to be given explicitly:
$ parted -a optimal /dev/sda mkpart primary 129MiB 100%
$ # Set the 'lvm' flag only if you plan to use LVM2 for managing the disk.
$ parted /dev/sda set 2 lvm on
```

Informe al kernel sobre los cambios de partición:

```
$ partprobe
```

Vamos a verificar el resultado:

```
$ parted /dev/sda print
Model: ATA SAMSUNG MZ7LN256 (scsi)
Disk /dev/sda: 256GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End    Size   Type     File system  Flags
 1      1049kB  135MB  134MB  primary  ext4         boot
 2      135MB   256GB  256GB  primary               lvm
```

8. Instalación de Slackware (programa de instalación)

Si prefiere descargar los paquetes requeridos por usted mismo, en lugar de dejar que el programa *setup* lo haga por usted (en realidad lo prefiero así), ahora es el momento.

Quiero usar *rsync* para descargar los paquetes de antemano. Como Slackware *chroot* no proporciona el comando *rsync* , debemos dejar Slackware *chroot* por un momento, usar *rsync* del sistema operativo de rescate de Ubuntu y luego regresar:

```
$ # Exit from Slackware chroot:
$ exit
$ mkdir -p ~/slackware-chroot/packages
$ cd ~/slackware-chroot/packages
$ # Download only: a, ap, d, l, n packages series (I don't need GUI apps),
$ # pay !attention! to the "." at the end of command line:
$ rsync -vaz
rsync://rsync.osuosl.org/slackware/slackware64-14.2/slackware64/{a,ap,d,l,n}
.
$ wget
https://slackware.osuosl.org/slackware64-14.2/slackware64/CHECKSUMS.md5
$ # Verify the checksums of the downloaded packages:
$ grep -P "\./(a|ap|d|l|n)/" CHECKSUMS.md5 | md5sum -c --quiet
$ mkdir -p ~/slackware-chroot/patches
$ cd ~/slackware-chroot/patches
$ Download patches, pay !attention! to the "." at the end of command line:
$ rsync -vaz
rsync://rsync.osuosl.org/slackware/slackware64-14.2/patches/packages .
$ wget https://slackware.osuosl.org/slackware64-14.2/patches/CHECKSUMS.md5
$ # Verify the checksums of the downloaded patches:
$ grep -P "\./packages" CHECKSUMS.md5 | md5sum -c --quiet
$ # Return to Slackware chroot:
$ chroot ~/slackware-chroot /bin/bash --login
$ cd
$ # Remember to update TERM as described earlier:
$ export TERM=screen
```

Teniendo todas las particiones en su lugar, ahora es el momento de ejecutar el programa *setup* y realizar la instalación como lo sabe. Solo recuerde formatear y montar la partición */boot* cuando se le solicite mediante *setup* . Cuando se le solicite, omita *LILO* instalación, ya que vamos a utilizar *syslinux* en su lugar. Si descargó los paquetes de antemano, dirija *configuración* al directorio */paquetes* premontado. De lo contrario, use *setup* para descargar los paquetes por usted.

No reinicie la máquina cuando el programa de instalación se ofrezca al final de la instalación.

9. Chroot recién instalado de Slackware

Sí, uno más *chroot* para tratar.  La configuración del sistema recién instalado se realiza mejor desde el propio sistema. Y vamos a hacer exactamente eso. Cuando el programa *setup* terminó su trabajo, dejó el sistema de archivos raíz de Slackware (y algunos más) montado en */mnt* . Y no hay nada que nos impida desde *chrooting* a este directorio, lo que significa que realmente podemos "iniciar sesión" en el sistema recién instalado sin reiniciar.

Este sistema es de alguna manera limitado (es decir, no hay servicios en ejecución), pero tiene todas las herramientas necesarias para realizar los pasos de configuración finales antes de reiniciar el servidor.

Para evitar sorpresas desagradables, montamos algunos sistemas de archivos posiblemente necesarios, antes de *chrooting* :

```
$ cd /mnt
$ mount --bind /run run
$ mount --bind /dev/pts dev/pts
$ # Only needed if you downloaded the patches and want to apply them:
$ mkdir run/patches
$ mount --bind /patches/packages run/patches
$ # Enter Slackware Chroot (tm):
$ chroot /mnt /bin/bash --login
$ cd
$ # Remember to update TERM as described earlier:
$ export TERM=screen
```

10. Aplicar parches

El siguiente conjunto de comandos aplicará todos los parches disponibles y le informará sobre cualquier archivo *.new* para tratar:

```
$ find /run/patches -name \*.txz -exec upgradepkg {} \;
$ find /etc -name \*.new
```

11. Bootloader (syslinux)

El siguiente conjunto de comandos instalará el *syslinux* bootloader:

```
$ extlinux --install /boot
$ dd count=1 bs=440 conv=notrunc if=/usr/share/syslinux/mbr.bin of=/dev/sda
```

Luego, cree el *archivo de configuración* syslinux:

```
$ cat << EOF > /boot/syslinux.cfg
PROMPT  0
TIMEOUT 0
DEFAULT vmlinuz-generic
SERIAL  1 9600

LABEL vmlinuz-generic
  KERNEL vmlinuz-generic
  APPEND console=ttyS1,9600 printk.time=0 quiet ipv6.disable=1 ro
  INITRD initrd-generic.gz
EOF
```

Esta configuración permitirá que los mensajes aparezcan en la consola de serie. También especifico algunos parámetros del kernel (*printk.time=0 quiet*) para silenciar considerablemente su salida (aún aparecerían mensajes de error). Como no quiero molestarme con IPv6, lo deshabilito en el nivel del kernel (*ipv6.disable=1*). Como puede ver, usaremos el kernel genérico con initrd. Esta es la única forma (es decir, por medio de initrd) de que el LVM2 pueda ser funcional.

Tenga en cuenta que las rutas del kernel y initrd especificadas en *syslinux.cfg* tienen que ser relativas al directorio */boot* . Esto se debe a que *syslinux* no puede leer desde la partición raíz basada en LVM2, por lo que algo como */boot/vmlinuz-generic* no funcionaría (/ en la partición LVM2 en mi caso).

12. Disco RAM inicial (initrd)

Cree el *initrd* archivo de configuración:

```
$ cat << EOF > /etc/mkinitrd.conf
# mkinitrd.conf
# See "man mkinitrd.conf" for details on the syntax of this file
#
#SOURCE_TREE="/boot/initrd-tree"
#CLEAR_TREE="0"
OUTPUT_IMAGE="/boot/initrd-generic.gz"
KERNEL_VERSION="$( readlink /boot/vmlinuz-generic | rev | cut -f1 -d- | rev
)"
#KEYMAP="us"
MODULE_LIST="ext4"
#LUKSDEV="/dev/sda2"
#LUKSKEY="LABEL=TRAVELSTICK:/keys/alienbob.luks"
ROOTDEV="/dev/vg0/rootfs"
ROOTFS="ext4"
#RESUMEDEV="/dev/sda2"
#RAID="0"
LVM="1"
#UDEV="1"
#MODCONF="0"
#MICROCODE_ARCH="/boot/intel-ucode.cpio"
```

```
WAIT="0"
EOF
```

Como dice el comentario, consulte la página del manual *mkinitrd.conf* para obtener más información.

En particular, asegúrese de que su *MODULE_LIST* , *ROOTDEV* y *ROOTFS* estén definidos correctamente. Si no necesita el soporte de LVM2, puede establecer *LVM = "0"* (o comentarlo).

Lo notable es cómo *KERNEL_VERSION* se deriva automáticamente, no para el núcleo en ejecución, sino para el núcleo instalado (que puede ser más nuevo o más antiguo que el que está en ejecución). */etc/mkinitrd.conf* proviene de la secuencia de comandos */sbin/mkinitrd* , por lo que es posible usar comandos de shell dentro del archivo de configuración. En la instalación estándar de Slackware, encontrará que */boot/vmlinuz-generic* es en realidad un enlace simbólico:

```
$ ls -l /boot/vmlinuz-generic
lrwxrwxrwx 1 root root 22 Dec 13 00:44 /boot/vmlinuz-generic -> vmlinuz-
generic-4.4.38
```

Así que el siguiente código:

```
KERNEL_VERSION="$( readlink /boot/vmlinuz-generic | rev | cut -f1 -d- | rev
)"
```

simplemente extraerá el número de versión de *installed kernel image*. Y finalmente, para crear el *initrd* , ejecute el siguiente comando:

```
$ mkinitrd -c -F
```

Tenga en cuenta que no es necesario ejecutar el comando *syslinux* relacionados después de crear o actualizar la *initrd* image. Esto es diferente de *LILO* , donde tiene que ejecutar el comando *lilo* después de cambiar *initrd* image.

13. Habilitar el acceso a la consola serie

A partir de ahora, la configuración de la consola serie en */boot/syslinux.cfg* permite interactuar con el gestor de arranque y también ver los mensajes del kernel, pero no permite el inicio de sesión de root a través del puerto serie. Si desea habilitarlo, descomente la siguiente línea en */etc/inittab* :

```
s2:12345:respawn:/sbin/agetty -L ttyS1 9600 vt100
```

y la siguiente línea en */etc/securetty* :

```
ttyS1
```

Es posible que también desee comentar las siguientes líneas en */etc/inittab* :

```
#c1:12345:respawn:/sbin/agetty --noclear 38400 tty1 linux
```

```
#c2:12345:respawn:/sbin/agetty 38400 tty2 linux
#c3:12345:respawn:/sbin/agetty 38400 tty3 linux
#c4:12345:respawn:/sbin/agetty 38400 tty4 linux
#c5:12345:respawn:/sbin/agetty 38400 tty5 linux
#c6:12345:respawn:/sbin/agetty 38400 tty6 linux
```

y las siguientes líneas en */etc/securetty* :

```
#tty1
#tty2
#tty3
#tty4
#tty5
#tty6
```

tty [1-6] son para los mensajes de inicio de sesión estándar de VT, pero como no tenemos teclado ni pantalla, no podemos hacer uso de ellos.

14. Finalizando la instalación

Hemos terminado con la instalación y configuración inicial de Slackware Linux.  Ahora puedes preparar el sistema para reiniciar y, bueno, reiniciar. Antes de hacer eso, también podría considerar mirar el Appendix B, donde explico cómo preparar las cosas de SSH, para que después de reiniciar, pueda conectarse al servidor con SSH de inmediato. (De lo contrario, tendrá que iniciar sesión en la consola serie para realizar las otras tareas de configuración).

Primero, prepare el disco duro para salir con seguridad a la fase de reinicio. Tenga en cuenta que, solo desmonte las particiones */mnt/boot* y */mnt* ya que estas son las únicas particiones del disco duro que tengo. Si tiene más particiones de disco montadas, también debe desmontarlas:

```
$ # Exit freshly installed Slackware chroot:
$ exit
$ umount -v /mnt/boot
$ mount -v -o remount,ro /mnt
$ # This never hurts:
$ sync
$ # Only needed if LVM2 is used:
$ vgchange -an --ignorelockingfailure
$ # This never hurts again:
$ sync
$ # Shouldn't be needed, but just in case:
$ sleep 3
```

Now, go to the server management page and press `[BOOT_IN_NORMAL_MODE]` button. You can observe the reboot process on the serial console.

A. Configuración de la gestión del disco LVM2

Las siguientes instrucciones destruirán los datos en el disco.

Antes de continuar con la partición de LVM2, si el disco ya está bajo el control de LVM2, primero debe desactivarse. Yo uso el siguiente conjunto de comandos para hacerlo:

```
$ lvscan
$ ( cd /dev/mapper && lvchange -an $(pvs --noheadings -o vg_name) )
$ vgscan
$ vgchange -an $(pvs --noheadings -o vg_name)
$ pvscan
$ pvremove -ffy $(pvs --noheadings -o pv_name)
$ partprobe
```

Para obtener más información sobre LVM2, vaya a https://wiki.archlinux.org/index.php/LVM

Recuerde que el disco ya se particionó con MBR en el capítulo “Particionamiento” y */dev/sda2* ya existe. El siguiente conjunto de comandos activará LVM2 en */dev/sda2* y creará las particiones (las particiones LVM2 se ubicarán encima de */dev/sda2* partición):

```
$ pvcreate /dev/sda2
$ pvdisplay
$ vgcreate vg0 /dev/sda2
$ vgdisplay
$ # Create the partitions (logical volumes):
$ lvcreate -L 12G vg0 -n rootfs
$ lvdisplay
$ vgchange -ay
```

NOTA:

- Lo bueno de LVM2 es que puede agregar fácilmente más particiones más adelante.
- He elegido tamaños de partición que se adaptan a mis necesidades actuales, dejando un espacio libre significativo. LVM2 puede aumentar fácilmente los tamaños más adelante si es necesario.
- No he creado la partición de intercambio. El servidor tiene más que suficiente de RAM. Pero si es necesario, se puede agregar fácilmente más adelante.

B. Configuración del servidor SSH (antes de reiniciar)

Debe estar en *chroot* del sistema Slackware recién instalado para realizar los pasos de configuración que se detallan a continuación..

Si activó el servicio *sshd* durante *setup* , se iniciará automáticamente la próxima vez que se inicie el sistema Slackware. Desafortunadamente, no podrás conectarte a él por dos razones:

1. las claves de host aún no se generan, por lo que no podrá verificar la autenticidad del host y, por supuesto, no querrá conectarse sin poder verificarla,
2. la autenticación de clave pública del usuario no está configurada y, por supuesto, no desea iniciar sesión con la autenticación de contraseña.

Para resolver el primer problema, debemos realizar manualmente la tarea que normalmente realizarían los scripts de inicio de Slackware cuando el sistema se inicie por primera vez. La generación de las claves de host básicamente se reduce al siguiente comando:

```
$ ssh-keygen -A
```

Y luego, para obtener key fingerprint del host (me atengo a RSA):

```
$ ssh-keygen -l -f /etc/ssh/ssh_host_rsa_key
```

Configurar la autenticación de clave pública es un poco más engorroso, pero aún está lejos de ser una ciencia espacial. : - ^ Primero, debe cargar su clave pública de su estación de trabajo al servidor. Ejecute el siguiente comando en la estación de trabajo:

```
$ scp -i ./login_key ~/.ssh/id_rsa.pub user@x.y.z.w:~
```

NOTA: El comando anterior carga la clave pública *~/.ssh/id_rsa.pub*, pero para la autenticación de transferencia, usa la misma clave que cargó anteriormente mediante la interfaz web.

Ahora, de vuelta al servidor, cree el directorio *~/.ssh* requerido:

```
$ mkdir ~/.ssh
```

Te dije que usaras el programa *screen* al principio, ¿verdad? Ahora le haremos uso. La clave pública que cargó anteriormente se ha colocado en el directorio de inicio *user* del sistema operativo de rescate de Ubuntu. Necesitamos cambiarle el nombre al archivo *authorized_keys* en el directorio *~/.ssh* de la nueva instalación de Slackware:

```
$ # Detach from screen session, you'll be dropped to Ubuntu rescue OS:
(keyboard) Ctrl+a d
```

```
$ # Login as root:
$ sudo su -
$ mv /home/user/id_rsa.pub /root/slackware-
chroot/mnt/root/.ssh/authorized_keys
$ # Exit root login:
$ exit
$ Re-attach to screen session:
$ screen -r
```

Asegure la propiedad y los permisos correctos, de lo contrario *ssh* no nos dejará entrar:

```
$ chown root:root ~/.ssh
$ chown root:root ~/.ssh/authorized_keys
$ chmod 0700 ~/.ssh
$ chmod 0600 ~/.ssh/authorized_keys
```

NOTA:

1. Si no ha usado *screen* , simplemente abriría la segunda conexión SSH para realizar la tarea anterior. Alternativamente, podría salir de todos los *chroots* y luego ejecutarlos nuevamente, pero ¿quién querría hacer eso? 😉
2. Recuerde que la configuración de red del servidor correcta debe estar en su lugar para que pueda conectarse a través de SSH después de reiniciar.

En este punto, todas las piezas deben estar en su lugar y debe poder iniciar sesión con éxito en su nueva instalación de Slackware después de que se reinicie el servidor.

NOTA: Sé que permito el inicio de sesión de root a través de SSH. Tengo que vivir con eso. : - ^

Sources

- Escrito originalmente por Andrzej Telszewski
- Traducido por: Victor 2019/02/05 21:40 (UTC)

howtos, author atelszewski