Born2BeRoot

Qué es una máquina virtual

Una máquina virtual es un software capaz de instalar un sistema operativo dentro de sí mismo, haciendo pensar al sistema operativo que está alojado en un ordenador real. Con las máquinas virtuales podemos crear dispositivos virtuales que se comportarán de la misma forma que los dispositivos físicos, utilizando su propia CPU, memoria, interfaz de red y almacenamiento.
Esto es posible porque la máquina virtual está alojada en un dispositivo físico, que es el que proporciona los recursos de hardware a la máquina virtual. El programa de software que crea máquinas virtuales es el hipervisor. El hipervisor es responsable de aislar los recursos de la máquina virtual del hardware del sistema y realizar las implementaciones necesarias para que la máquina virtual pueda utilizar estos recursos.
Los dispositivos que proporcionan los recursos de hardware se denominan máquinas host o hosts. Las diferentes máquinas virtuales que se pueden asignar a un host se denominan invitados o máquinas invitadas. El hipervisor utiliza una parte de la CPU, el almacenamiento, etc. de la máquina host y los distribuye entre las diferentes máquinas virtuales.

Cómo funciona una máquina virutal

La virtualización nos permite compartir un sistema con múltiples entornos virtuales. El hipervisor gestiona el sistema de hardware y separa los recursos físicos de los entornos virtuales. Los recursos se gestionan según las necesidades, desde el anfitrión hasta los invitados. Cuando un usuario de una VM realiza una tarea que requiere recursos adicionales del entorno físico, el hipervisor administra la solicitud para que el sistema operativo invitado pueda acceder a los recursos del entorno físico.

Propósito de las máquinas virtuales

Puede haber varias máquinas virtuales en el mismo host y cada una de ellas estará aislada del resto del sistema. Gracias a esto, podemos ejecutar diferentes sistemas operativos en nuestra máquina. Para cada máquina virtual, podemos ejecutar una distribución de sistema operativo diferente. Cada uno de estos sistemas operativos se comportará como si estuviera alojado en un dispositivo físico, por lo que tendremos la misma experiencia al usar un sistema operativo en una máquina física y en una máquina virtual.

Una vez que sabemos cómo funcionan, es buena idea ver todas las ventajas que obtenemos al usar máquinas virtuales:
-Diferentes máquinas invitadas alojadas en un ordenador pueden ejecutar diferentes sistemas operativos, por lo que tendremos diferentes sistemas operativos trabajando en la misma máquina.
-Proporcionan un entorno en el que probar de forma segura los programas inestables para ver si afectarán al sistema o no.
-Conseguimos un mejor uso de los recursos compartidos.
-Reducimos costes reduciendo la arquitectura física.
-Son fáciles de implementar porque proporcionan mecanismos para clonar una máquina virtual en otro dispositivo físico.

Elección del sistema operativo: CentOs vs Debian

Ambos son una distribución Linux.
CentOs utiliza el formato de paquete RPM, con YUM/DNF como administrador de paquetes. Es mejor para usuarios más experimentados, como las empresas que quieren una distribución más confiable por el apoyo RHEL.
Debian utiliza el formato de paquete DEB con dpkg/ATP y es mejor para principiantes de Linux. Es mucho más sencillo de configurar y tiene más usuarios, con lo que hay más respuestas online en el caso de tener problemas de configuración.

APT – Advanced Package Tool

APT es una colección de herramientas usadas para instalar, actualizar, eliminar y administrar paquetes de software en Debian y sus sistemas operativos (Ubuntu, Linux Mint, etc.).
Funciona mediante repositorios o directorios especiales con colecciones de paquetes de software.

Aptitude

Aptitude es una herramienta avanzada de empaquetado que agrega una interfaz de usuario a la funcionalidad.
Esto permite a un usuario buscar un paquete de forma interactiva e instalarlo o eliminarlo.

APT vs Aptitude

APT está restringido a una línea de comandos sin interfaz de usuario. En cambio, Aptitude es superior: tiene una interfaz interactiva predeterminada de solo texto y mejor gestión de paquetes.

AppArmor

AppArmor es un marco de control de acceso obligatorio. Limita los programas de acuerdo con un conjunto de reglas que especifican qué archivos pueden acceder a un programa determinado.
Ayuda a proteger el sistema contra vulnerabilidades.

Asegurarse de que la máquina no tiene un entorno gráfico en el momento del lanzamiento

No es así para todas las máquinas virtuales, pero en este caso específico:
Si al encender te carga el escritorio, tiene un entorno gráfico. Está mal.
Si al encender te carga solo la consola, no tiene un entorno gráfico. Está bien.
También se puede ejecutar:

```
Echo $XDG_CURRENT_DESKTOP
```

Si no aparece nada, es que no tiene un entorno gráfico, está bien.

Comprobar que el servicio UFW se inicia

```
ufw status
```

Comprobar que el servicio SSH se inicia

```
systemctl status ssh
```

Verificar que el sistema operativo es Debian o CentOS

uname -a

Comprueba que se ha añadido un usuario con el login del alumno evaluado y que pertenece a los grupos “sudo” y “user42”

Podemos listar todos los grupos a los cuales pertenece un usuario en concreto:

groups <nombre_del_usuario>

Comprueba que se han implementado las reglas impuestas sobre la política de contraseñas

-Crea un nuevo usuario

sudo adduser <nuevo_usuario>

-Asígnale una contraseña de tu elección, respetando las reglas.

getent passwd <nuevo_usuario>

-Cómo se han configurado las reglas solicitadas

vi /etc/login.defs
Cambiamos PASS_MAX_DAYS a 30
Cambiamos PASS_MIN_DAYS a 2
Cambiamos PASS_WARN_AGE a 7

Cargamos la nueva configuración de contraseñas en root:

chage root -M 30 -m 2 -W 7
chage root -l

Si quisiéramos configurarla para un usuario, cambiaríamos <root> por el nombre del usuario.

También utilizamos PAM – Pluggable Authentication Modules.
Los módulos de autenticación conectables establecen los requisitos de calidad de contraseñas predeterminadas para la contraseña del sistema.
Para poder utilizarlo, se debe instalar un módulo adicional, libpam-pwquality.

-Crear un grupo llamado “evaluando” y asígnalo a este usuario

su -
sudo addgroup evaluando
usermod -aG evaluando <nuevo_usuario>

-Verifica que el usuario pertenece al grupo “evaluando”

Groups <nombre_del_usuario>

-Ventajas de esta política de contraseñas y ventajas y desventajas de su implementación

Ventajas: refuerza la seguridad del usuario.
Desventajas: se pierden los datos si el usuario no realiza los cambios en el término establecido.

Nombre de host y particiones

-Verifique que el nombre del host de la máquina es: login42

```
su –
hostnamectl
```

-Modifique este nombre del host reemplazando el inicio de sesión con el suyo y reinicie la máquina

```
su –
hostnamectl set-hostname <nuevo_host>
reboot
```

Comprobación:

```
su –
vi /etc/hosts
```

-Restaura la máquina al nombre del host original

```
su –
hostnamectl set-hostname <antiguo_host>
reboot
```

Comprobación:

```
su –
vi /etc/hosts
```

-Cómo visualizar las particiones de la máquina virtual

```
lsblk
```

-LVM - Administrador de volumen lógico

Es una capa de abstracción entre un dispositivo de almacenamiento y un sistema de archivos. Obtenemos muchas ventajas al usar LVM, pero la principal es que tenemos mucha más flexibilidad cuando se trata de administrar particiones.

Supongamos que creamos cuatro particiones en nuestro disco de almacenamiento. Si por algún motivo necesitamos ampliar el almacenamiento de las tres primeras particiones, no podremos porque no hay espacio disponible junto a ellas. En caso de que queramos ampliar la última partición, siempre tendremos el límite que impone el disco. Es decir, no podremos manipular particiones de forma amigable. Gracias a LVM, todos estos problemas están resueltos.

Al usar LVM, podemos expandir el almacenamiento de cualquier partición (ahora conocida como volumen lógico) cuando queramos sin preocuparnos por el espacio contiguo disponible en cada volumen lógico. Podemos hacer esto con el almacenamiento disponible ubicado en diferentes discos físicos (lo que no podemos hacer con las particiones tradicionales).

SUDO

-Comprobar que el programa sudo está correctamente instalado

```
systemctl status sudo
```

-Mostrar la asignación de su nuevo usuario al grupo sudo

```
groups sudo
```

-Valor y funcionamiento de sudo
Sudo podría equivaler a superusuario.
Pueden ejecutar cualquier cosa como si fuera root pero sin ser root.
De sudo depende la seguridad del sistema, por lo que hay que cumplir una serie de requisitos para tener una configuración fuerte para el grupo sudo:

username	ALL	=	(	ALL	:	ALL	)	ALL
	invitados			usuarios		grupos		comandos

ejemplo:
amurcia-42 ALL=(ALL:ALL)ALL

-Mostrar la aplicación de las reglas
sudo visudo /etc/sudoers

-Verifica que la carpeta /var/log/sudo/ existe y tenga al menos un arvhico. Verifica el contenido de la carpeta, debería haber un historial de los comandos utilizados con sudo
cd /var/log/sudo

-Intenta ejecutar un comando a través de sudo.
sudo echo “Hello”

UFW – Uncomplicated FireWall

-Comprobar que UFW está instalado
ufw status

-Comprobar que funciona correctamente
Para que dé información sobre el registro, las políticas predeterminadas y los nuevos perfiles:
sudo uwf status verbose

-Qué es UFW
UFW es una interfaz para iptables. Está orientada a simplificar el proceso de configuración de un firewall.
Iptables es difícil para usuarios inexpertos y UFW sirve para proteger fácilmente la red.

-Listar las reglas activas en UFW
Para obtener el oden y número de identificación de todas las reglas activas:
sudo uwf status numbered

-Agregar una nueva regla para abrir el puerto 8080
ufw allow 8080
Además, dentro de la aplicación de VirtualBox, deberemos añadir manualmente el puerto 8080 dentro de Network.

-Elimina la nueva regla
Enumeramos las reglas:
sudo ufw status numbered
Eliminamos la regla según el número que hayamos obtenido previamente:

sudo ufw delete <número>

SSH – Secure SHell

-Comprobar que SSH está instalado y comprobar que funciona

systemctl status ssh

-Qué es SSH

SSH es un protocolo de comunicación red que permite que dos ordenadores se comuniquen y compartan datos.
La comunicación entre los ordenadores está encriptada, por lo que es adecuado para redes no seguras. Cuenta con:
-Capa de seguridad
-Acceso remoto: garantiza el acceso cifrado para usuarios y procesos.
-Transferencia de archivos: SFTP – protocolo seguro de transferencia de archivos administrado por SSH. Es una forma segura de manipular archivos a través de una red.
-Tunneling: técnica de encapsulación que proporciona transferencia de datos segura. Es útil para acceder a materiales en línea confidenciales para empresas desde redes inseguras.
-Gestión de red: gestiona la infraestructura de red y otras partes del sistema.

-Verifica que solo utiliza el puerto 4242

Tenemos que ver que solo aparece

Port 4242
PermitRootLogin no:

Find / -name “sshd_config”
vi /etc/ssh/_sshd_config

-Usar SSH con el usuario recién creado

En la máquina virtual:

ip address

En la terminal:

ssh <nombre_de_usuario>@dirección_ip -p 4242
wall “Escribe aquí tu texto”

Supervisión de guiones

-Cómo funciona el script

su – #para conectarse a la raíz

apt install net-tools

#Copiar el siguiente script:

```
#!/bin/sh
total_mem_kilo=$(free --kilo | grep "Mem:" | awk '{print $2}')
total_mem_mega=$(free --mega | grep "Mem:" | awk '{print $2}')
used_mem_kilo=$(free --kilo | grep "Mem:" | awk '{print $3}')
used_mem_mega=$(free --mega | grep "Mem:" | awk '{print $3}')
memory_per=`free --kilo | grep "Mem:" | awk '{printf("%.2f", $3*100/$2)}'`
disk_usage_gb=$(df  --total -h | grep "total" | awk '{print $2 "b"}')
disk_usage_m=$(df  --total -m| grep "total" | awk '{print $3}')
cpu_physical=`cat /proc/cpuinfo | grep physical\ id | sort | uniq | wc -l`
cpu_load=`top -n1 -b | grep "%Cpu(s):" | awk '{print $2}'`

if [ `lsblk | grep lvm | wc -l` -gt 0 ]
then
        lvm="yes"
else
        lvm= "no"
fi

wall "  #Architecture: `uname -a`
        #CPU physical : `cat /proc/cpuinfo | grep physical\ id | sort | uniq
| wc -l`
        #vCPU : `cat /proc/cpuinfo | grep processor | wc -l`
        #Memory Usage: $used_mem_mega/$total_mem_mega"MB "($memory_per%)
        #Disk Usage: $disk_usage_m/$disk_usage_gb
        #CPU load: $cpu_load%
        #Last boot: `who -b | awk '{print $3 " " $4}'`
        #LVM use: $lvm
        #Connexions TCP : ` netstat | grep "ESTABLISHED" | wc -l` ESTABLISHED
        #User log: `who | wc -l`
        #Network: IP `hostname -I`(`ip a | grep link/ether | awk '{print
$2}'`)
        #Sudo : `journalctl -q | grep sudo | grep TTY | wc -l` cmd"
```

Con explicaciones:

```
#!/bin/sh
total_mem_kilo=$(free --kilo | grep "Mem:" | awk '{print $2}')
```

Grep para selección, print para imprimir

free – enseña la cantidad de memoria

```
total_mem_mega=$(free --mega | grep "Mem:" | awk '{print $2}')
used_mem_kilo=$(free --kilo | grep "Mem:" | awk '{print $3}')
used_mem_mega=$(free --mega | grep "Mem:" | awk '{print $3}')
memory_per=`free --kilo | grep "Mem:" | awk '{printf("%.2f", $3*100/$2)}'`
```

Operación matemática para saber la memoria que tenemos

```
disk_usage_gb=$(df  --total -h | grep "total" | awk '{print $2 "b"}')
```

df – espacio en el disco utilizado por el sistema de fichero

```
disk_usage_m=$(df  --total -m| grep "total" | awk '{print $3}')
cpu_physical=`cat /proc/cpuinfo | grep physical\ id | sort | uniq | wc -l`
cpu_load=`top -n1 -b | grep "%Cpu(s):" | awk '{print $2}'`
```

Top -n1 -b – me da la información de lo que está pasando ahora mismo. Dice qué usuario está ejecutando qué, cuánto tiempo y qué porcentaje de espacio ocupa. Solo lista el número 1, el que más se esté usando. -b es para que no vaya actualizando continuamente

```
if [ `lsblk | grep lvm | wc -l` -gt 0 ]
then
        lvm="yes"
else
        lvm= "no"
fi
```

Si existe lvm (logical volumen manager) sí. Si no, lvm no.

Cada físico puede tener varios procesadores.

Me da cuantos tengo físicos y los vCPU me da procesador.

```
wall "  #Architecture: `uname -a`
```

uname -a – da información del sistema, incluyendo el usuario que estamos utilizando

```
        #CPU physical : `cat /proc/cpuinfo | grep physical\ id | sort | uniq
| wc -l`
        #vCPU : `cat /proc/cpuinfo | grep processor | wc -l`
        #Memory Usage: $used_mem_mega/$total_mem_mega"MB "($memory_per%)
        #Disk Usage: $disk_usage_m/$disk_usage_gb
        #CPU load: $cpu_load%
        #Last boot: `who -b | awk '{print $3 " " $4}'`
```

who – quien hay conectado en el ordenador, quién está activo. Normalmente aunque no haya nada hay 2 abiertos.

who -b - enseña la utltima vez que se ha encendido el ordenador

```
        #LVM use: $lvm
        #Connexions TCP : ` netstat | grep "ESTABLISHED" | wc -l` ESTABLISHED
```

netstat grep “established” porque queremos saber el número de conexiones activas: no que estés escuchando, sino las establecidas. Con el wc -l me dice la cantidad.

netstat – genera visualizaciones que muestran el estado de la red y estadísticas del protocolo

```
        #User log: `who | wc -l`
        #Network: IP `hostname -I`(`ip a | grep link/ether | awk '{print
$2}'`)
        #Sudo : `journalctl -q | grep sudo | grep TTY | wc -l` cmd"
```

journalctl – permite ver los mensajes del Kernel o núcleo del sistema. Puede ser más específico. Es un diario del sistema: logins, inicios de sistema, etc.
Cuando liste la cantidad de comandos sudo, buscamos en el journalctl -q (-q para que no te muestre los warnings)
Buscamos solo los TTY y SUDO a la vez, porque es una forma de asegurar que solo lista una vez el comando que hemos ejecutado.
El TTY debe estar activado por razones de seguridad.

whoami – dice quién es
lsblk – listblocks: da todas las particiones del ordandor.
sda: solid disk A. Me mostrará, por ejemplo, que dentro de SDA, tengo 3 particiones.

-Qué es cron
Crontab es una línea de comandos cron. Cron es un programador de trabajos en sistemas operativos similares a Linux. Para que se ejecuten trabajos periódicamente en horas, fechas o intervalos fijos.

-Cómo se ha configurado el script para que se ejecute cada 10 minutos
crontab -e -u sbin
*/10 * * * * /sbin/home/monitoring.sh

Si queremos que sea cada minuto:
crontab -e -u sbin
*/1 * * * * /sbin/home/monitoring.sh

El 1 son los minutos.
El * seguido del 1 son las horas (1-23)
El siguiente * son los días del mes (1-31)
El siguiente * son los meses (1-12)
El siguiente * son los días de la semana (0-6)