

ISE - Práctica 1, Semana 1 (28/9/2020)

Timestamp

15:38

Descripción (Comando/Herramienta)

Daniel Castillo Seville (casel@ugr.es) se presenta y explica las prácticas. La P1 es instalar SO con RAID. Esta semana lo haremos con **Ubuntu Server** y las siguientes con **CentOS**. También explica las demás prácticas y la dinámica.
Los exámenes se harán vía SWAD. Si no podemos seguir el hilo en local, seguir al profesor y repetir después.

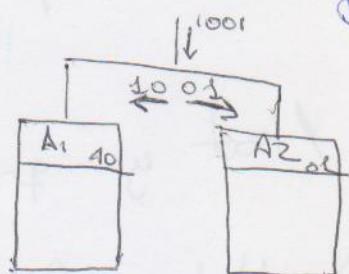
15:51

Instalación de Ubuntu Server con LVM

Un servidor es una máquina que sirve para dar servicios informáticos.

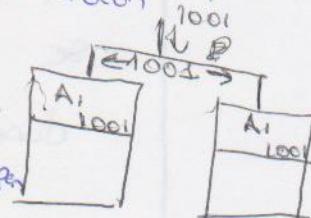
Una Máquina Virtual es una emulación de una máquina en otra física. Hay que señalar que sirve para virtualizar SOs, algo distinto de un container, que virtualiza un servicio.

RAID (Redundant Array of Independent Discs) es un sistema de backup para que, en caso de fallo, se pueda recuperar la información. Hay varios tipos de RAID:

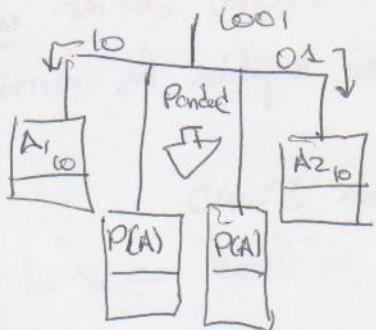
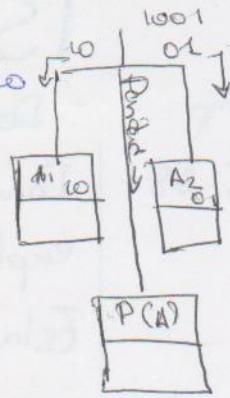


RAID 0: Da más velocidad ya que la mitad de los datos van a un disco y la otra a otro. El inconveniente es que, si uno de los discos falla, no se puede recuperar toda la información.

RAID 1: Sirve para redundancia, ya que duplica los datos en ambos discos. Puede dar tolerancia a fallos, a no ser que se rompan todos los discos.



RAID 5: Se compone de 3 discos, diferente de RAID 0. En el tercer disco se almacena la paridad, así que si uno de los dos primeros falla con esa paridad se puede reconstruir.

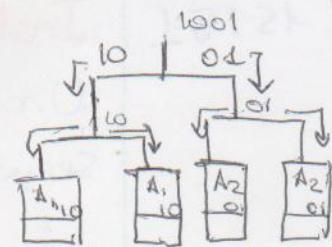


RAID 6: Como RAID 5, pero con 2 discos para paridad. Si se rompe un disco de paridad, no se pierde la info ya que está respaldada.

RAID 10: Combinación de RAID 1 y

RAID 0. Es un RAID 0 con

2xRAID 1. Combina la redundancia para tolerancia de fallos con la rapidez.



16:13

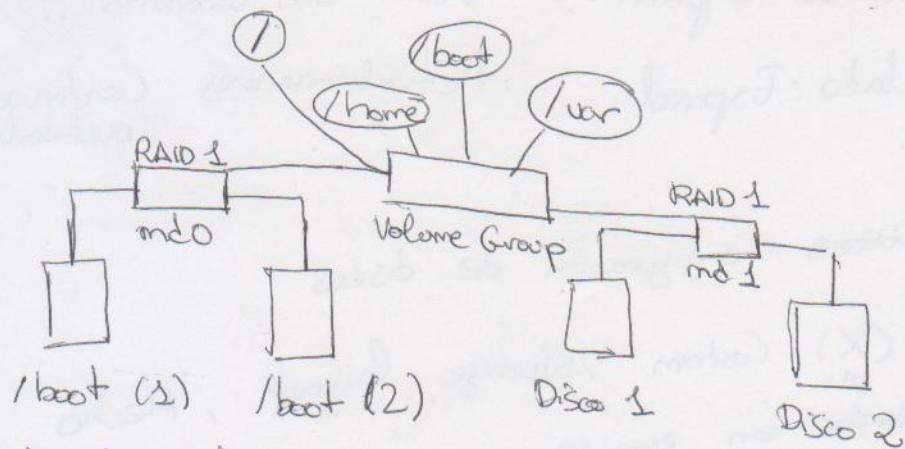
Para enter que información sensible se puede filtrar o eliminar del disco, los discos y volúmenes se encriptan para evitar problemas legales. Por ej.

No se puede encriptar /boot ya que es el disco de arranque.

LVM (Logical Volume) Abstacta los directorios del sistema de almacenamiento, haciendo que se le asigne en un disco duro parte del directorio sistema.

16:39

Viven desde un esquema:



A esta abstracción a bajo nivel se le denomina PV (Physical Volume). Estos se agrupan en un Volume Group. Y a partir de ahí se crean los volúmenes lógicos (Logical volumes). En caso de que fallese un volumen, se le asigna otro volume group y se apunta el LV al ese volume group.

16:26

Abrimos Virtual Box.

Es útil usar snapshots para volver a un estado anterior.

1. Maquina > Nueva. Ponemos nombre, sistema Ubuntu (64 bits)
2. Memoria. Recomendar 4 GB de RAM
3. Creamos nuevo disco VDI, reservado dinámicamente, 10 GB
4. Propiedades > Almacenamiento en el controlador ~~SATA~~ Creamos otro disco SATA siguiendo paso 3.
5. Instalamos el disco de instalación arrastrando la unidad óptica con el disco la .iso

16:35

Iniciamos VBox

Al empezar la instalación se nos pone la selección de idioma.

Idioma: Español

Red: Sin cambios

Teclado: Español

Actualizaciones

Configurar sin
actualizar

16:42

~~Disks~~ Configuración de discos

(X) Custom storage layout, Hecho
Hacer con espacio

Particiones: (desde "Add GPT partition")
300 M, leave unformatted, Crear: Crea el boot.
Se genera una partición bios.grub en un disco
duo, pero se debe crear otro

~~Add as bootable device~~, leave unformatted, Crear: Crea la part. 3
Crear Software raid:
md 0, creamos las particiones de 300 M.
md 1, creamos las otras particiones.

Ahora sí, creamos las particiones lógicas.

md 0: -, ext 4, /boot

Crear grupo de volúmenes VM

[x] md 1, [x] cifrar, Passphrase:

↳ Se crea Vg0

Vg0 > Create logical volume

8G, ext 4, /
1G, ext 4, /home
-, swap

RAID
hardware es
costoso con
el mismo
propósito de
hacer RAID

RAID
software es
a costa 0,
para uso personal

17:02

Confirmar acción destrucción? Confirmar
Configuración de perfil

Usuario, nombre equipo... : varoi
Contraseña, prácticas, ISF

¿Instalar cosas? Tabulamos, Hecho

Instalando sistema...

En caso de que salga la actualización,
descargamos.

Ubuntu se reinicia

17:13

Please unlock disk dm-crypt0: prácticas, ISF
varoi login: varoi

Password: prácticas, ISF

lsblk asegura las partitions.

lspci | grep Eth lista el Ethernet

Saltimes de la MV, Configuración de Virtualbox

Herramientas > ~~Red~~ Red > General

Vbox net0 : IPv4 192.168.56.1

Propiedades > Red > Adaptador 2

Habilitar, Adaptador solo anfitrion, vbox net0

Reiniciamos Ubuntu

Sudo nano /etc/network/interfaces. Editar interfaces

17:28

17:24

Escribimos en el fichero

auto enp0s8

iface enp0s8 netmask static

address 192.168.56.105

Sudo ip link set enp0s8

Sudo nano /etc/netplan/00-installer-config.yaml

Escribimos

{enp0s8:

dhcp: file

addresses: [192.168.56.105/24]

//version: 2

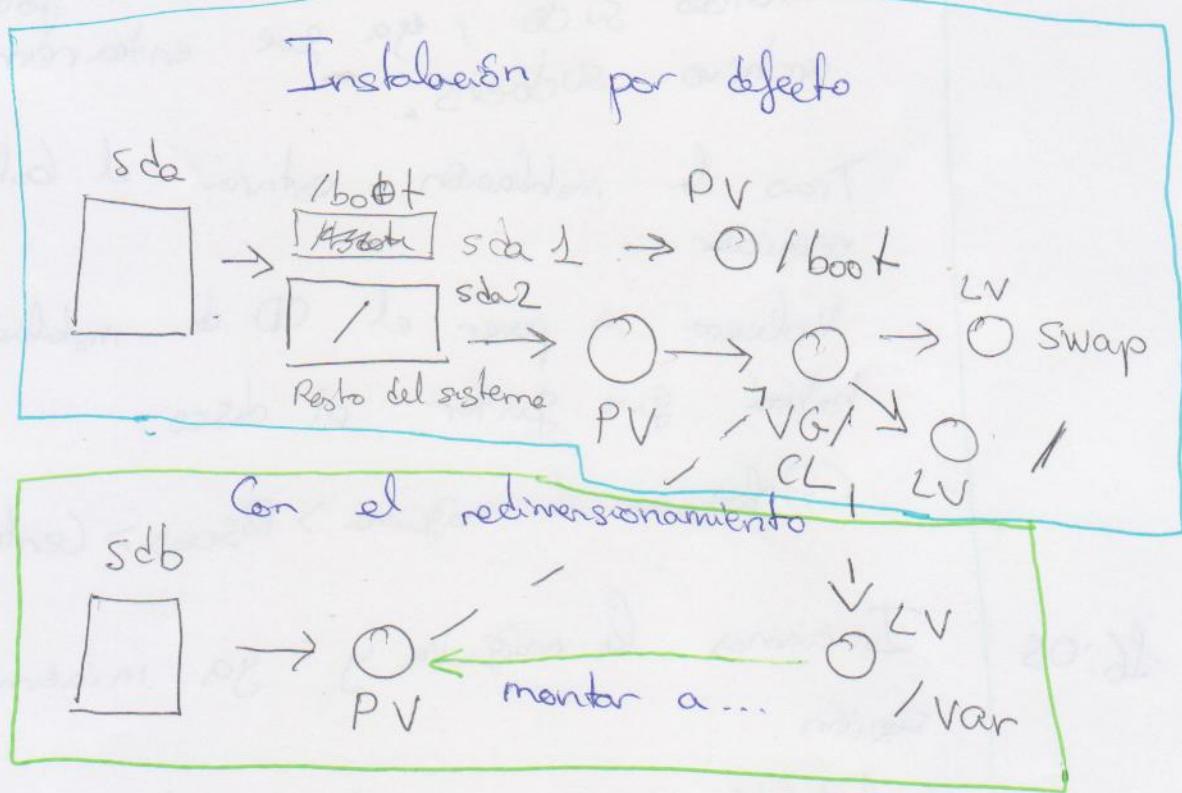
Sudo netplan apply para guardar cambios

Práctica 1 - Semana 2

Instalación de CentOS

Instalaremos un CentOS por defecto y una comanda, configuraremos los LV para aumentar la memoria.

6. arquitectura; en este caso, es:



15:43

Arrancamos VirtualBox

Naguara > Nueva, la llamaremos CentOs y seguimos como en la Semana anterior. En este caso el tamaño del uso por dato es de 8GB.

15:51

Tres le carga una GUI cargada:

Nos vamos a disco instalación por defecto. Al y aceptarlos la y empezar a instalar, cargar la configuración la contraseña root (prácticas, ISE) y el usuario (varios: prácticas, ISE; ser admin). Cuando lo haremos administrador, podrá usar comandos sudo , ya que estaremos en el archivo sudoers.

Tras la instalación, estará el botón de reiniciar.

Volverá a poner el CD de instalación, así que habrá que quitar el disco.

(Preferencias de máquina > Discos > Centros de conexión)

16:05

Iniciamos la máquina y ya iniciamos sesión.

lsblk: Vemos que los type de sdb1 y sdb2 son part, y que en vez de vg0-X se nombra cb-X

Nos falta ahora un 2º disco sdb.

16:15

Preferencias de máquinas > Discos > Crear y seguimos los pasos.

Volvemos a ver si está con lsblk llamado sdb.

Llamaremos con el comando lvm console pero pedirá permisos root. Para ello pondremos su root y escribimos la contraseña. Verás que existe pero no vale. Vamos a formatear sdb.

1 pvcreate Permite crear un volumen físico. Si malo (man pvcreate) explica argumentos y demás. Con ↑ + G veremos ejemplos. Salimos con Q

pvcreate /dev/sdb

2 pvdisplay muestra la info de los volúmenes físicos. sdb no muestra mucha info aparte del tamaño

3 vg extend añade PV a VG. En este caso añadiremos sdb a cl
vgextend cl /dev/sdb

4 Podemos ver a que sdb se asignó a cl con pvdisplay, sección "VG Name".

5 lv create crea un volumen lógico. Preguntas permisos.
-L indica el tamaño
-n indica el nombre

lv create -c 1G -n newver cl

6 Para ver si se ha hecho lv display.

7 Para acceder a la partición podemos usar

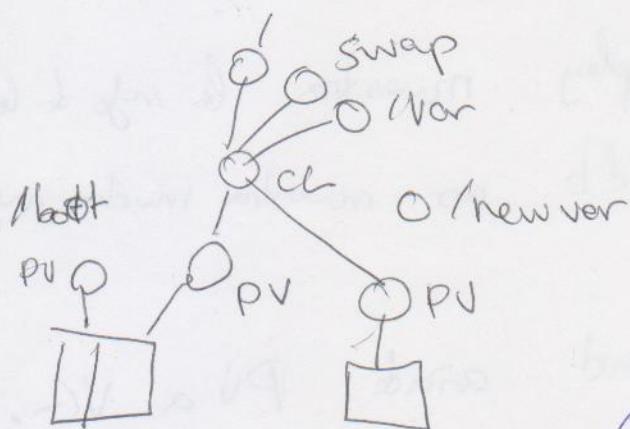
{ /dev/cl /newver
/dev/mapper/
cl-newver

8 Vamos a formatear newver
con ext4,

mkfs -t ext4

Comprobemos que se hizo y lo vemos
con lsblk

En este momento el logrante sera



El objetivo es
desmontar /newver
a /var.

Que da de asi:

- i) Traspasemos /var a /newver
- ii) Montamos y desmontamos
- iii) Modificaremos fstab, para que se monte siempre que se inicie el sistema.
- iv) Liberamos el espacio y movemos permanentemente a /var /old.

9 Creamos una carpeta para newver y así montar el disco.

mkdir /mnt/newver
y lo comprobamos → ls /mnt

10 Vamos a montar newver en /mnt/newver
mount /dev/cd /mnt/newver

Para comprobarlo lo veremos si está con
mount (leyendo sobre última linea)

11 Copiamos los archivos de /var a /newver

! No podemos copiarlo tal cual ya que
SELinux (Security Enhanced Linux) le asigna
contextos para darle una seguridad ante errores.
Si hicieramos una copia normal el sistema &
SELinux no lo reconocerá como parte del sistema

cp -a /var /var-OLD

Comprobaremos con ls -laZ /

12 Debería a cambiar hacer backups, se recomienda
cambiar a modo mantenimiento.
Systemctl { rescue } runlevel 2, reboot

4 haceremos la copia

cp -a /var/ /mnt/newver

4 Salimos del modo mantenimiento
Systemctl reboot

CentOS se reinicia.

! Se reporta un bug donde hay que meterse 2 veces en mantenimiento. Al volver al login se entra como root.

13 Vamos a poner a montar cada vez que se inicie /newvar

maso /etc/fstab
y añadimos la linea

/dev/mapper/cl -newvar /var ext4 defaults
0 0

14 Vamos a desmontar /newvar

umount -l /mnt/newvar

lsblk, con quanto se pide desmontar, que lo haya

15 Montemos los ficheros: mount -a y veremos que /newvar se ha cargado.

16 Montamos nuevo para un backup & ver y creamos /ver & nuevo
umount /dev/mapper/cl-newvar
mn /var /var-OLD
~~mkfs~~ /var
restore con /ver

restar con devolver el contexto de SELinux
a /var

17 Montamos el nuevo /var
Mount -a
... y reiniciamos
reboot

18 Comprobemos que funcione
lsblk
ls -lat /
mount

Práctica 1, Semana 3

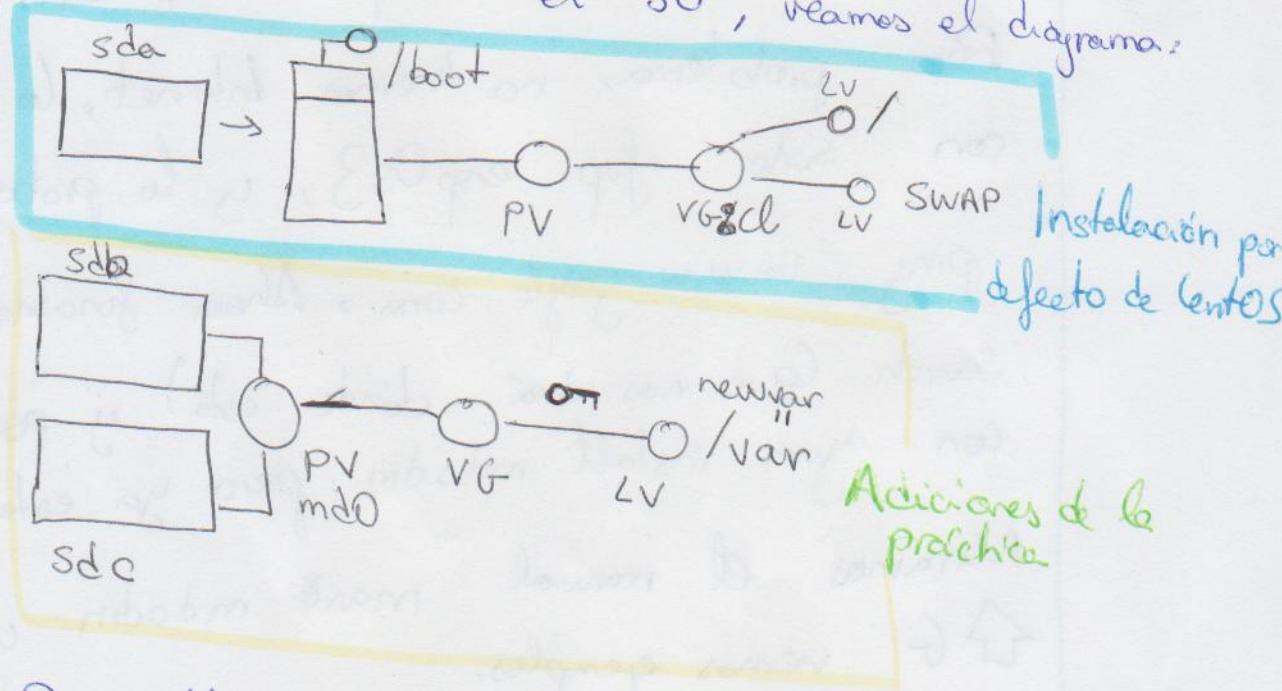
RAID y encriptación en CentOS

15:43

Nos vamos a VirtualBox y hacemos esto
ver la instalación por defecto de CentOS
(para ver cómo se hace, consultar log de la Semana
2)

15:49

Mientras instalamos el SO, veamos el diagrama:



Para ello:

- 1 Añadir los discos
- 2 Crear la RAID
- 3 Crear md0
- 4 Crear el VG
- 5 Crear el LV
- 6 Cifrar el sistema
- 7 Montar /var en /var

15:57

Con la instalación hecha y el disco ^{sacado} ~~sacado~~, comprobamos que el sistema inicie normalmente y vemos con `lsblk` la instalación por defecto. Y ~~reiniciamos~~ salimos del sistema para proceder a colocar y configurar los discos.

16:05

Vamos a agregar 2 discos duros en el controlador SATA (ver guías anteriores para saber cómo) e iniciamos.

Comprobemos con `lsblk` que existen `sdb` y `sdc`.

Vamos a buscar con `yum search` el paquete `mdadm`, herramienta para hacer los RAID.

Pero problema: no tenemos Internet. Lo solucionamos con `sudo ifop enp0s3`, y lo probamos con `ping www.google.com`. Ahora funciona `yum search` (que nos dirá dónde está) y podemos instalarlo con `yum install mdadm`, pero ya está instalado.

Miramos el manual `man mdadm` y con  `G` vemos ejemplos.

Para hacer la raid 1 "md0" con `sdb` y `sdc` usaremos `mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sdb /dev/sdc`, y guardaremos. Comprobamos la raid con `lsblk`. Si de los dos discos cuelga `md0`, está bien.

16:20

Creamos una PV con la RAID
PV create /dev/md0

y la vemos con pv display o pvs (es más compacto)

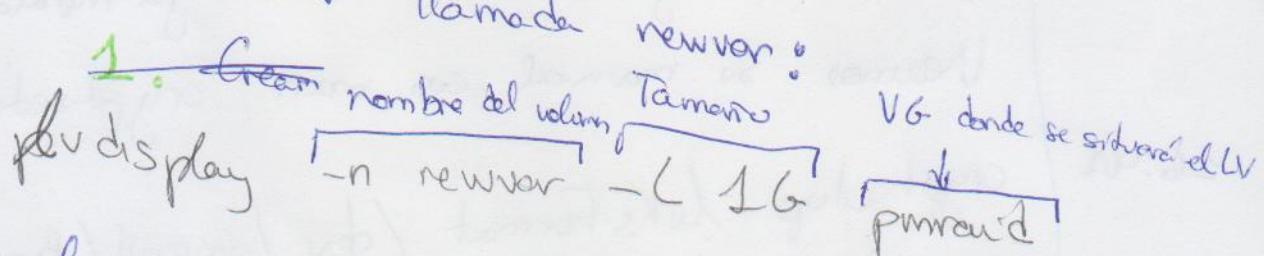
16:22

Creamos una VG llamada pmraid con
vgcreate pmraid /dev/md0

y lo comprobamos con vg display o vgs (+compacto)

16:24

Creamos una LV llamada newvar



y lo comprobamos con lv display o lvs (+compacto)

16:27

Para la encriptación se usa LUKS (Linux
Unified Key Setup)

1 LVM on LUKS → Usado por Ubuntu 20.04 con 2 métodos alternativos:

Supongamos todo el contenido
del disco. cifran todo el contenido → Primero
eso creará los volúmenes logicos. → Carga decifrada (unape
2 LUKS on LVM → Se crean primero los volúmenes logicos. → Segundad → Carga decifrada (unape
y luego se cifren (cada LV por separado)

+ carga una por cada (y)
+ control (cada partición tiene su contraseña)

Usaremos LVM on LUKS. Su bien se ha hecho
el LV, no está formateado ni asignado.

Si por casualidad se ~~borrara~~^{soltara} el contenido, se puede borrar con shred, que hace poco legible el contenido con funciones matemáticas. Si se da con la función de inversa, se podrá leer. También se hace algo similar con el código, cambiando la nomenclatura con ~~variables~~^{nombres de} variables y funciones sin sentido.

16:43 LVM on LUKS

Usaremos el paquete cryptsetup con yum search / yum install. Estaba ya instalado.

Vamos a crear su manual con man cryptsetup

16:47 cryptsetup luksFormat /dev/pmaud/renewvar
encriptará la partición. **Reverde escribir YES**
en mayúscula para seguir. Por la contraseña predeterminada

16:50 Abriremos de nuevo la RAID cifrada con
cryptsetup luksOpen /dev/mapper/pmaud-renewvar.
pmaud-renewvar-crypt.

Al entrar comprobaremos con lsblk se abre un apartado
en pmaud-renewvar llamado pmaud-renewvar-
crypt

16:54 Vamos a hacer el backup. Lo ponemos en modo
mantenimiento con systemctl isolate rescue
Reverde el bug de doble login!

16:57

Formatemos la partición encriptada.

mkfs -t ext4 /dev/mapper/pmraid-newvar-crypt

Creamos una carpeta temporal en /mnt
mkdir /mnt/newvar

y comprobamos con ls /mnt/

17:00

Montamos el disco

mount /dev/mapper/pmraid-newvar-crypt /mnt/newvar

y copiamos los archivos

cp -a /var/ /mnt/newvar

Comprobamos los contextos de SELinux

ls -lZ /var

, ls -lZ /mnt/newvar

17:03

Añadimos newvar a fstab

nano /etc/fstab

>> /dev/mapper/pmraid-newvar-crypt (TAB) /var
(TAB x3) ext (TAB) default 0 0

Movemos /var a /var-OLD

mv /var /var-OLD

Desmontamos /mnt/newvar

y creamos /var con su contexto | mkdum /var
restore con /var

Montamos desde fichero
mant -a

Y probamos con lsblk

17:13

Vamos a automatizar el desencriptado con el archivo
/etc/crypttab.

El formato en el archivo sería

*
pmraid -newvar-encrypt UUID=<...> none
Partición desencriptada ID de la partición

Para pillar la UUID cogeremos los blkid
de la partición tipo crypto-LUKS. La encavaremos
según el tipo y lo llevaremos a /etc/crypttab
blkid | grep crypto >> /etc/crypttab

Luego formateamos el cauce en el archivo.
nano /etc/crypttab

Tendrá que quedar como en ④

17:20

Guardemos y reiniciemos. Si todo salva bien, se
pedirá la contraseña de
encriptación. En lsblk se verá el sistema montado.
Si hubiera un fallo se rehacienda en modo
mantenimiento.

Práctica 2 ISE

- Semana 1

Timestamp

Comando, explicación, diagramas...

Creación de una red interna

16:20

En Virtual box, en la máquina un nuevo adaptador solo ~~anfitrión~~ que sea la misma que sea la misma. CentOS habilitaremos que sea la misma.

Damos de alta en CentOS los adaptadores enp0s3 y enp0s8

Ruta: /etc/sysconfig/network-scripts/

ifcfg-enp0s3

Cambiamos ONBOOT a yes

ifcfg-enp0s8

Ponemos lo siguiente:

Tres ellos, reiniciamos CentOS

16:22

Cuando se reinicie, veremos con ip addr que los adaptadores estén con state UP, google.es para ver que enp0s3 y pongamos a funcionar

16:30

Instalamos Ubuntu Server y comprobaremos en la `ifconfig` que tenemos ambos adaptadores y que `enp0s8` tiene una IP acabada en `105` (el de Centos acaba en `110`)

16:35

En Ubuntu Server,

`Ping google.es`

`Ping 192.168.56.110`

En Centos,

`Ping google.es`

`Ping 192.168.56.105`

Con ello comprobaremos que ambas máquinas se conectan a Internet y a la otra máquina.

Instalación y configuración de servicios

Un servicio es un programa que escucha por un socket y procesa todo lo que se recibe para enviar una respuesta.

Un cortafuegos es una barrera que protege nuestros puertos de los ataques. Se edita con `iptables`, aunque Centos y Ubuntu lo abren con `firewall-cmd` o `ufw` (uncomplicated firewall). SSH: Secure Shell, es un protocolo de administración remota a otra máquina en otro lugar de forma segura. Al instalarlo hay un cliente y un servidor, los cuales se configuran con ssh (client) o sshd (server, daemons).

Lo malo es que no admite cosas en 2º plano. Así, por ej., se puede cortar la descarga de un archivo si se cierra ssh.

Para ello se hace uso de herramientas como screen que ponen el ssh como segundo plano.

Para temas de seguridad, hay 2 grandes herramientas:

fail2ban: Es un ~~fuerte~~ programa que evita ataques DDoS haciendo que si ~~atacan~~ más de 3 veces se ^{intenta conectarse} banea durante un tiempo.

rkHunter: Comprueba el sistema en busca de vulnerabilidades.

SSH en Ubuntu Server

16:55

Ubuntu trae el cliente pero no el server de ssh:
Tenemos que instalarlo via task sel.
Para ello lo instalamos sudo apt install tasksel

16:58 Al iniciar task sel con sudo task sel sale una pantalla de la que seleccionaremos OpenSSH server, pulsaremos y damos a OK

17:00

Cuando se instale podemos
sudo systemctl status sshd.service
y veremos que está activa en el puerto 22.

17:01

Cambiamos el puerto de sshd:

sudo nano /etc/ssh/sshd-config.

y descomentamos Port 22022 → Cambiamos los hackers normalmente atacan el puerto 22, despista.
Si se permitiera los hackers solo podrían lograr la root
Permit Root login no → tendrían que forcejar la pass

Reiniciamos con ^{sudo} systemctl ~~status~~ restart sshd.service y
comprobamos su estado. Ahora ssh escucha en :22022

12:05 Vamos a habilitar el puerto en el cortafuegos ^{con}
^{sudo} ufw allow 22022

En CentOS, nos conectamos con

ssh user@192.168.56.103 -p 22022
si lo hacemos con 22 se deneg.

y veremos que se inicia una sesión de Ubuntu.

12:18 Vamos a automatizar el logon:

En CentOS

Creamos una clave pública-privada
ssh-keygen
la enviamos a Ubuntu
ssh-copy-id user@192.168.56.103
-p 22022

En Ubuntu

Sudo nano /etc/ssh/sshd-config
Editamos Password Authentication no.

En Gatos

Antes de nada -> cambiar el puerto en el archivo /etc/ssh/sshd.conf
Instalamos semanage:

Buscamos

E instalamos gdm provides semanage

Buscamos el puerto con

sudo semanage port -l
y lo filtramos con

sudo semanage port -l | grep ssh

En el firewall habilitemos el puerto

sudo firewall-cmd (--permanent) --add-port=2222/tcp
y reiniciamos el servicio sshd

sudo systemctl restart sshd.service.

Para la creación de clave pública/privada seguimos lo mismo que en Ubuntu Server pero con 192.168.56.160

Práctica 2 - Semana 2

Timestamp

Git, control de versiones y backups

Copia de seguridad Es un empaquetamiento de los datos tras la cual todos los cambios se mueve en otro sitio. Se guarda en cada backup Control de versiones Es un empaquetamiento de los datos pero donde no se copian todos los archivos cada vez, sino que se van guardando las diferencias.

Para copias de seguridad:

Copia binaria dd, copia bit a bit los datos. Se usa para particiones fijas
Copia y empaquetamiento cpio, cp, tar
Sincronización rsync, rsnapshot
Instantáneas

Aquí, en Ubuntu Server algunas maneras de hacer backup

dd if = prueba/ of = pruebaBackup.img. Copia binaria
cp prueba/ pruebaBackup/. Copia de carpetas básicas
ls | cpio > pruebaBackup.cpio ✓ Creación
cpio -idv < pruebaBackup.cpio ✓ Extracción
tar cvf pruebaBackup.tar.gz ✓ Extracción
tar xvzf pruebaBackup.tar.gz ↑ Creación ↑ Extracción
Pipe cpio, copia de seguridad de carpetas
Tar, archivos comprimidos.

R Sync Copias de seguridad por snapshots, versión sería.

r sync prueba/ ^o proveer Backup Rsync /
Copia segundaria local

r sync -e "ssh -p 2222" prueba/ & varoi@<IP maq>
varoi@<IP maq> ruta
Copia segundaria por (por ej.:)

r sync prueba Backup Rsync

↑

Restauración de copia de seguridad. /home/varoi/ubuntuBackup

Copia segundaria por ssht

varoi@192.168.56.110:

Para control de cambios:

git

Las ventajas para I&G son seguir los cambios, planificar entornos de prueba y poner un pie en los DevOps.

Hay muchos comandos y mucha documentación para profundizar.

▷ Cómo funciona?

Se basa en 3 zonas

Workspace / Directorio de trabajo Zona donde se trabaja con los archivos

Stage Zona de registro de cambios, donde decimos qué archivos se están cambiando

Repositorio En forma de commits - se registran los cambios.

También hay repositorios remotos.

16:30

Comandos de git.

git init (inicializar repo)

git remote add origin <url> (inicializar repo remoto)

git add / rm Añadir / Eliminar cosas

git commit -m " " Registrar cambios

git diff | ➔ Diferencia entre worktree y zona de cambios
--staged Diferencia entre zona de cambios y repos local
HEAD Diferencia entre repositorio local y worktree

git log Muestra los cambios hashes de los commits realizados

git reset

git checkout HEAD <fichero>

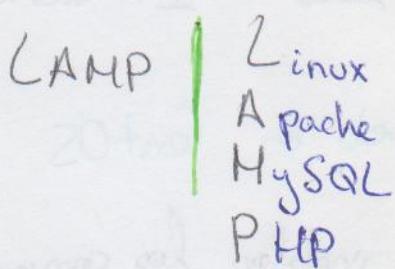
Práctica 2 - Semana 3

Timestamp

15:42

Instalación de un servidor web en Ubuntu Server

Arrancamos Ubuntu Server para instalar LAMP



15:46

Procedemos a instalarlo con tasksel
sudo tasksel

Cogemos el marcador LAMP Server pero el paquete 34 genera un error apt-get failed porque en el repositorio no está. Tendríamos que actualizarlo.
sudo apt-get update

Volvemos a arrancar tasksel, marcamos LAMP server y le damos a instalar (Tab > OK)

Instalará y configurará los servidores. Por ej.; MySQL Server se escucha en el puerto 3306 por defecto

Un problema al instalar LAMP desde tasksel es que los paquetes estén desactualizados. Para ello solventarlo, se podría solucionar la pila a mano.

¿Cómo lo comprobamos?

Apache sudo systemctl status apache2

MySQL sudo systemctl mysql status mysql

PHP php -a

15:56

¿Y cómo sabemos si sirve?

Vamos al navegador de nuestro sistema y ponemos la IP (192.168.56.105). Debería mostrar una imagen de Ubuntu y por algún lado "It works!"

Instalación de un servidor web en CentOS

16:06

Vamos a usar yum para instalar los componentes de la pila:

1. Apache: El paquete no se llama apache ni apache2, sino httpd

sudo yum install httpd

Ahora, si se le hace el status, se queda inactivo.

Vamos a activarlo:

sudo systemctl enable httpd

sudo systemctl start httpd

Ahora, en el navegador, al poner la IP (192.168.56.10)

No va. Tenemos que crear las reglas en firewall-cmd

firewall-cmd --permanent

--add-service=http

firewall-cmd --reload

Al volver al navegador, nos dará otra página distinta de inicio (patrocinada por Apache y CentOS)

2. MySQL. sudo yum install mysql mysql-server

Hacemos lo mismo para activar "mysql"

A partir de aquí, para instalarlo de forma segura, haremos los siguientes pasos:

mysql - secure - installation

Nos va indicando unos pasos:

1. Setear una contraseña a root

2. Eliminar al usuario anonymous

3. Borrar conexiones a root desde remoto

Podemos igualmente meternos @ root desde ssh.

4. Eliminar la base de datos test

5. Recargar los privilegios de las tablas

Con ello evitaremos las vulnerabilidades en un entorno de producción.

3. PHP

¡Sencilla! yum install php

Para probarlo, php -a

4. Funciones extra de conexión PHP-MYSQL

yum install php-mysqlnd.x86_64

Para probar todo hacemos un script de PHP.

cd /var/www/html

w (o nuevo) index.php

PHP | 2.php

```
$link = mysqli_connect('127.0.0.1:3306', 'root', 'practicas1SE');
if (!$link) die ("No me pude conectar " . mysqli_error());
else print ('Conectado!');
mysqli_close ($link);
?> PHP info();
```

Pero no se conecta en el navegador, hay que configurar.

vi /etc/httpd/conf/httpd.conf

y cambiamos index.html por index.php.

Funciona, pero no se conecta. Es debido a SQL y SELinux.

Miremos los booleans de SELinux

getsebool -a

Mucho texto. Veamos a ver si los relacionados con httpd

getsebool -a | grep httpd

Mucho (mores) texto, pero ya se ven bools como

httpd-network-connect-db → off

pongamos el flag a on

setsebool -P httpd-network-connect-db on

y lo comprobemos viendo que el flag está on

Al entrar en el navegador ya funciona.

Extras : Fail2Ban y Screen

fail2ban ~~esta~~ Evita el acceso a otros tres equivocarse
ante el tiempo.

En Centos se instala

sudo yum install epel-release

sudo yum install fail2ban

y se activa el servicio

Systemctl status fail2ban
Systemctl enable fail2ban
Systemctl start fail2ban
Systemctl status fail2ban

16:58

Al lanzar jail2ben-client status se dice que no hay jails o servicios de bareo. Se necesita \$ para servicio.

Para ello vamos a
por el archivo pide **no modificar paul.conf** y en su
lugar usar un paul.local.

Copiamos y editamos

cp paul.conf paul-local

vi
nano paul.local

Vamos a la zona de JAILS y de ahí a SSHD.
Cambiamos los siguientes parámetros

enabled=true

Port = 22022

17:06

Vamos a intentar barearnos.
Antes de nada ponemos en /etc/ssh/sshd-config

Password Authentication a yes.

Cogemos la consola de Windows y tratamos de conectarnos
sin éxito. Tras unos intentos se banearía al usuario con
un timeout.

¿Y cómo desbarea? jail

Con jail2ben-client set \$ unbenip <IP>

Screen: SSH en 2º plano

17:14 Instalamos screen

sudo install screen

Para usarlo ~~esa~~, entramos en el ssh e iniciamos un nuevo Screen / entiendo así a un ssh en 2º plano. Es en bash normal pero que podremos salir con Ctrl+A+D

Haciendo Screen ls veremos la lista de screens.
Podremos salir.

Al volver a conectarse, pon screen -r.

Para eliminarlo, se hace screen -X -S <ID> quit

ISE - Práctica 3

Semana 1

Timestamp

Hay 3 tipos de monitores

A nivel hardware Te monitoriza todo aquél hardware que deseas del sistema. Como ejemplos: lspci, lsusb, lshw o los monitores desde BIOS, y los mensajes del Kernel mediante dmesg (Linux), que es útil para problemas con el hardware o los periféricos.

A nivel software En Linux hay un directorio especial /proc para ver archivos que configuran el sistema de forma transparente. También está /var/log.

Para ver, por ejemplo, los RAID, consultaremos /proc/mdstat. Monitores generales Monitoran todo, ~~desde~~ tanto hardware como software, que son programas externos al sistema. Algunos de ellos son:

Nagios

Nagios Muy usado, que se ha transformado a un proyecto empresarial llamado Naemon

Ganglia Usado en la UGR, son usados en servidores de altas prestaciones, monitorizan sistemas de cómputo distribuidos.

Zabbix Es de instalación sencilla, y la que veremos en las prácticas. Tiene mucha documentación.

Cacti y AWstats

16:39

Cargar la RAID de Ubuntu y restaurarla

Vamos a simular que un disco se rompa. Para ello, antes de iniciar Ubuntu Server, vamos a **Almacenamiento > Disco 2** y le damos check a **Correctable en caliente**. Y lo iniciamos.

Tras hacer un **lsblk**, vamos a "cargárnoslo".

Para ello, en Virtualbox, vamos a la misma ruta y clicamos en .

Vamos a ver qué pasa con **Sudo cat /proc/mounts**. Debería salir **[U-]**.

16:41

Ahora nos lo vamos a cargar a priori. Antes de nada, vamos a restaurar a una instantánea anterior

Vamos a la configuración de Virtualbox en la ruta y nos cargamos el primer disco e iniciamos.

Ahora todo iba a molir self, evidentemente. Esperamos a que initramfs cargue para salir del apuro.

cryptsetup: "Cannot process volume group vg 0"

cryptsetup: Waiting for encrypted source device

UUID=...

16:45

Con 2 tabulaciones podemos ver todos los dispositivos que nos permiten usar con el filesystem en RAID.

Con dmseg se podría leer la información para tratar de hacer debug. Con cat /proc/mdstat veremos que el RAID está inactive.

Vamos a intentar levantar los RAID

mdadm -R /dev/ md0

mdadm -R /dev/ md1

Y salimos del initramfs con exit, del cual saldrá y bootará Ubuntu.

Nuestra prioridad ahora es restaurar la RAID con otro disco duro.

Salimos de Virtualbox, creamos otro disco pero ahora de forma que el antiguo ocupe el puerto SATA 0 y el nuevo otro puerto SATA.

Al arrancar no habrá problemas de boot.

Ahora hay que hacer a mano las particiones, y replicar con ellos las 2 RAID.

Para crear las particiones usaremos fdisk

fdisk /dev/sdb.

Vamos creando las particiones:

sdb1: n, p, 1, -, 4096 (1 MiB)

sdb2: n, p, 2, 4097, 618496 (300 MiB)

sdb3: n, p, 3, -, - (9'7 MiB)

Todos ellos con n w guardados saldrán.

Ahora sí, vamos a recuperar el RAID

Sudo mdadm --add /dev/md0 /dev/sdb2
Sudo mdadm --add /dev/md0 /dev/sdb3

Para comprobar el estado de la restauración, podemos a tiempo real
User Sudo watch -n 1 /proc/mdstat

Para este recuperar el grub, tenemos que user

Sudo grub-install /dev/sdb1
(cuidademente ha de estar formateado)

Con dmseg se podría leer la información para tratar de hacer debug. Con cat /proc/mdstat veremos que el RAID está muerto.

Vamos a intentar levantar los RAID

mdadm -R /dev/md0

mdadm -R /dev/md1

Y salimos del initramfs con exit, del cual saldrá y booteará Ubunto.

Nuestra prioridad ahora es restaurar la RAID con otro disco duro.

Salimos de Virtualbox, creamos otro disco duro ahora de forma que el antiguo ocupe el puerto SATA 0 y el nuevo otro puerto SATA.

Al arrancar no habrá problemas de boot.

Ahora hay que hacer a mano las partitions, y replicar con ellas los 2 RAID.

Para crear las partitions usaremos fdisk

fdisk /dev/sdb.

Vamos creando las partitions:

sdb1: n, p, 1, -, 4096 (1 MiB)

sdb2: n, p, 2, 4097, 618496 (300 MiB)

sdb3: n, p, 3, -, - (9'7 MiB)

Todos ellos con w guardados en lazos.

Práctica 3 - Semana 2

Automatización

Para automatizar el estado del RAID podríamos usar este script de Python.

```
import re
```

Se podría mejorar si se pudiera automatizar para lantarse cada cierto tiempo. Se puede automatizar de dos formas:

/etc/systemd/systemd
crontab

También, para tratar ficheros, usaremos en el Shell

grep busca patrones y ficheros
find
awk
sed para expresiones regulares

A nivel de scripts, podemos hacerlos en Python o PHP

A nivel de plataforma podríamos usar Ansible, que monitoriza, instala y comprueba su ejecución de forma usual.

Gestionando la automatización como servicio

Añadiremos en /etc/systemd/system los ficheros

mon-raid.timer

mon-raid.service

Añadiremos el servicio

systemctl enable mon-raid.timer
systemctl start mon-raid.timer

y

para ver los logs del servicio:

sudo journalctl -u mon-raid --since="yesterday"
(Cambiar "yesterday" por la franja de tiempo deseada)

ISE - Práctica 4 - Parte explicativa

Phoronix

Es una plataforma de benchmarks bajo la página openbenchmarking.org. Contiene un conjunto de benchmarks para probar y estresar los componentes.

No está en los repositorios oficiales de Ubuntu/CentOS.

Hay que descargarlo u/a wget

Apache Benchmark (ab) y JMeter

Apache Benchmark es un benchmark diseñado para hacer simulaciones de peticiones HTTP a un servidor web para ver cuántos pueden servir. Permiten concurrencia y un nº fijo de peticiones, pero se hace en el mismo orden y tiempo.

JMeter es un software de Benchmarking más complejo, que permite concurrencia y nº de peticiones más compleja, pero esta incluye aleatoriedad.

Docker y los contenedores

El contenedor se basa en el host para ejecutarse, en lugar de emular una máquina virtual. Uno de los más populares es Docker, que lo hacen más rápido y más ligero que virtualizando.

Para instalar Docker, hay que arrancar el repos a Ubuntu

Dockerfiles

Al iniciar un contenedor de Docker se ejecuta este archivo. Normalmente se importan una imagen de contenedor base y se añaden otras instrucciones para personalizar el arranque y distribución.

docker-compose.yml

A modo de playbook de Ansible, este YAML define los Dockerfiles a ejecutar para que al hacer docker-compose up se ejecuten todos ellos.

Ajustes del servidor.

El ajuste fino se puede hacer usando sysctl. Esto permite que el Kernel active ~~y pare~~ e impida que se introduzcan.