Tips and Tricks

Trucos para usuarios linux

- Como convertir ficheros .flac a .mp3 en el shell de linux con ffmpeg
- Teclas Inicio (Home) y Final (End) en ZSH y oh-my-zsh con Powershell
- Linux, paquetes instalados desde el shell
- Cómo instalar y activar el repositorio EPEL en Centos 7/8
- Conocer el tamaño de unas carpetas ignorando los enlaces duros (rsync)
- Ssh se sale (break) de un ciclo (loop) en un script bash
- Redis Failed to start Advanced key-value store.
- Comando find con -maxdepth excluyendo el propio directorio
- Du y los ficheros o directorios ocultos
- Como vaciar o eliminar emails antiguos en dovecot sin usar find
- rc.local en systemas Debian usando systemd. Ejemplo redis
- Bad Bots y la pesadilla del tráfico. Htaccess en Apache 2.4
- Sudo sin contraseña
- Reparar el fichero .zsh_history: zsh: corrupt history file /home/USER/.zsh_history
- El archivo hosts para trabajar en un servidor o ip distinto del de la resolución DNS
- Cambiar repositorios de Ubuntu al mirror de OVH
- ## Guía Rápida de find - Búsquedas Avanzadas
- APT Upgrade - Modos Desatendido vs Manual
- Crear una unidad lógica con el 100%
- Búsqueda y reemplazo recursivo con sed en directorios
- Instalación de Node.js LTS 22 y npm en Ubuntu con CloudLinux CageFS
- OpenVPN: cómo excluir destinos del túnel con `net_gateway` (el caso de apple.com / apple.es)

Como convertir ficheros .flac a .mp3 en el shell de linux con ffmpeg

Introducción

Algunas veces me descargo o me pasan algun disco de música clasica en forma .flac y la verdad, ni tengo el equipo para tal audición, ni tanto espacio en mis saturados discos. Asi que lo mejor es convertirlos a .mp3 (también puedes hacerlo a .ogg si eres muy OpenSource (aunque mp3 ya es formato abierto)

Convertir todos los ficheros .flac de un directorio a .mp3 con ffmpeg en linux con el shell

Bueno, ni que decir tiene que debes tener instalado **fmpeg** y los codecs, pero eso lo dejo para otro momento.

Recursivo

```
find -name "*.flac" -exec ffmpeg -i {} -acodec libmp3lame -ab 128k {}.mp3 \;
```

Un solo nivel

```
find -maxdepth 1 -name "*.flac" -exec ffmpeg -i {} -acodec libmp3lame -ab 128k {}.mp3 \;
```

Enlaces

- man ffmpeg
- find
- Los 28 comandos más útiles de FFmpeg

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Teclas Inicio (Home) y Final (End) en ZSH y oh-my-zsh con Powershell

Introducción

La verdad es que me encanta usar ZHS en combinación de Oh-My-Zsh y el tema PowerLevel10k pero tenía un problema con las teclas Inicio (Home) y Fin (End) que no funcionan. Al final lo solucioné y te cuento como lo hice.

Solución al problema

Para probar si te va a funcionar antes de editar el fichero de configuración de zsh, te aconsejo que ejecutes en el terminal y después pruebes las teclas:

```
❯ bindkey "\033[1~" beginning-of-line
❯ bindkey "\033[4~" end-of-line
```

Si te funciona (debería), es necesario editar el fichero ~/.zshrc y añadirlo.

Después ejecuta:

```
❯ source ~/.zshrc
```

Enlaces

- Candrew34 en github -> Cannot using home/end key after install oh-my-zsh
- List of zsh bindkey commands

- Binding Keys in Zsh – jdhaos's blog

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Linux, paquetes instalados desde el shell

Introducción

A veces necesitamos conocer que paquetes tenemos instalados en nuestra distribución linux. Y no usamos un entorno gráfico.

Distribuciones basadas en .deb

Para conocer qué paquetes están instalados en nuestra distribución linux, desde el shell ejecutaremos

```
# apt list --installed | grep nginx

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

nginx/stable,now 1.18.0-2~focal amd64 [installed,upgradable to: 1.20.1-1~focal]
```

Distribuciones basadas en .rpm

```
 rpm -qa | grep apache
ea-apache24-mod_bwlimited-1.4-47.52.2.cpanel.x86_64
...
ea-apache24-2.4.48-3.12.1.cpanel.x86_64
...
```

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Cómo instalar y activar el repositorio EPEL en Centos 7/8

Introducción

El repositorio EPEL (Extra Packages for Enterprise Linux) es un repositorio de un grupo de Fedora que crea, mantiene y administra una serie de paquetes **.rpm** ausentes o presentes en versiones anticuadas, para mejorar las capacidades de las distros basadas en Redhat (RHEL, CentOs, Scientific Linux, Fedora)

Sus instalación en un servidor con cPanel requiere ciertas normas para evitar problemas posteriores, que algunas veces pueden ser bastante graves para nuestro sistema.

Instalación EPEL (CentOs 7/8)

Instalación EPEL (CentOs 7/8)

```
[root@centos7 ~]# yum -y install epel-release
Complementos cargados:fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirror.tedra.es
 * extras: mirror.tedra.es
 * updates: mirror.tedra.es
Resolviendo dependencias
--> Ejecutando prueba de transacción
---> Paquete epel-release.noarch 0:7-11 debe ser instalado
--> Resolución de dependencias finalizada

Dependencias resueltas
```

```
================================================================================
================================================================================
=
 Package
Arquitectura                                Versión
Repositorio                                 Tamaño
================================================================================
================================================================================
=
Instalando:
 epel-release
noarch                                      7-11
extras                                       15 k

Resumen de la transacción
================================================================================
================================================================================
=
Instalar  1 Paquete

Tamaño total de la descarga: 15 k
Tamaño instalado: 24 k
Downloading packages:
epel-release-7-
11.noarch.rpm
                                                         |  15 kB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Instalando    : epel-release-7-11.noarch          1/1
  Comprobando   : epel-release-7-11.noarch     1/1

Instalado:
  epel-release.noarch 0:7-11

¡Listo!
```

Desactivación (cpanel consejo)

Por defecto un repositorio se instalan activados, lo cual es bastante peligroso en un servidor con cPanel o con otro panel intrusivo (el 99,9% lo son)

Deberemos editar el fichero de configuración del repositorio `/etc/yum.repos.d/epel.repo` editando la línea `enable=1` a `enable=0`

```
[epel]
name=Extra Packages for Enterprise Linux 7 - $basearch
#baseurl=http://download.fedoraproject.org/pub/epel/7/$basearch
metalink=https://mirrors.fedoraproject.org/metalink?repo=epel-7&arch=$basearch
failovermethod=priority
enabled=0
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
```

Cuando queramos instalar o actualizar algún paquete del repositorio epel deberemos usar la opción `--enablerepo=epel` en nuestro comando yum

Ejemplo

```
[root@centos7 ~]# yum --enablerepo=epel -y install snapd
```

Verificacion doble de desactivación

Como es importante, deberíamos hacer una doble verificación de que el repositorio no está activo, con el comando `yum repolist` que en caso de no estar activo, no lo mostrará.

```
[root@centos7 ~]# yum repolist
Complementos cargados:fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirror.tedra.es
 * extras: mirror.tedra.es
 * updates: mirror.tedra.es
id del repositorio
nombre del repositorio
estado
base/7/x86_64
CentOS-7 - Base    10.072
```

```
extras/7/x86_64
CentOS-7 - Extras     498
updates/7/x86_64
CentOS-7 - Updates   2.579
repolist: 13.149
```

Conocer los paquetes disponibles en EPEL

Es un comando sencillo que mostrará la lista de paquetes del repositorio.

```
[root@centos7 ~]# yum --disablerepo="*" --enablerepo="epel" list available
Complementos cargados:fastestmirror
Loading mirror speeds from cached hostfile
 * epel: mirror.eixamcoop.cat
Paquetes disponibles
0ad.x86_64                                  0.0.22-1.el7                  epel
0ad-data.noarch                             0.0.22-1.el7                  epel
0install.x86_64                             2.11-1.el7                    epel
2048-cli.x86_64                             0.9.1-1.el7                   epel
2048-cli-nocurses.x86_64                    0.9.1-1.el7                   epel
. . .
```

Lista de paquetes de EPEL disponibles

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Conocer el tamaño de unas carpetas ignorando los enlaces duros (rsync)

Rsync, enlaces duros y du

En mi trabajo uso rsync con un sistema de enlaces duros, como el **Time Machine** de Apple. Y a veces es bueno saber o conocer, el tamaño de las carpetas ignorando los enlaces duros, en los que esta basado este sistema de backup continuo.

du

```
$ du -hc --max-depth=1 path/
24G     rsync/2022-03-17-065908
1.6M    rsync/2022-03-17-072202
1.2G     rsync/2022-03-17-105858
1.1G     rsync/2022-03-17-074333
79.9G     rsync/
79.9G     total
```

- Man page command du
- How to get folder size ignoring hard links?

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como esta, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Ssh se sale (break) de un ciclo (loop) en un script bash

Introducción

Uso Rsync time backup para algunos proyectos, pero el caso de uno en particular con más de 8TB de ficheros a mantener en backup, y con múltiples usuarios, prefiero usar una estrategia de copia de seguridad por usuario.

Cuando programé el script bash rápido para hacer este trabajo, me encontré que la finalización del script se alcanzaba tras leer el completar la primera copia de seguridad.

Tras finalizar correctamente el proceso de sincronización.

El problema es que **ssh** lee desde la entrada estándar, por lo tanto, se come todas las líneas restantes.

Solución al problema en ssh

```
ssh $USER@$SERVER "COMMAND_IN_REMOTE" < /dev/null
```

También podemos usar `ssh -n` en lugar de la redirección a ninguna parte.

> -n' Redirects stdin from /dev/null (actually, prevents reading from stdin). This must be used when ssh is run in the background. A common trick is to use this to run X11 programs on a remote machine. For example, ssh -n shadows.cs.hut.fi emacs & will start an emacs on shadows.cs.hut.fi, and the X11 connection will be automatically forwarded over an encrypted channel. The ssh program will be put in the background. (This does not work if ssh needs to ask for a password or passphrase; see also the -f option.)

Solución en Rsync time backup

En este software no hay posibilidad de modificar o usar el parámetro -n asi que solo se puede hacer via redirección a ninguna parte `< /dev/null`

```
while IFS= read -r line
do
  case $line in
    appdata_ociz9efdik2y|transmission-daemon|updater-ociz9efdik2y)
       continue
    ;;
    *)
       echo "$line"
       /srv/mypath/diwan/rsync-time-backup/rsync_tmbackup.sh --rsync-set-flags "-D -zz --
numeric-ids --links --hard-links -rlt --no-perms --no-group --no-owner --itemize-changes" --
strate
gy "1:1 7:7 30:30"  -p 9999 root@mypath.domain.net:/data/"$line"
/srv/storage/mypath/rsync/"$line" < /dev/null
    ;;
   esac
done < "$input"
```

Agradecimientos

- ssh breaks out of while-loop in bash - duplicate
- While loop stops reading after the first line in Bash
- ssh(1) - Linux man page

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Redis Failed to start Advanced key-value store.

Introducción

En algún momento nuestro servidor **Redis** falla, y deja de estar operativo. Reiniciamos pero no arranca y en su lugar muestra un error cuando hacemos un `sudo systemctl status redis-server`

```
jun 02 17:43:33 abkrim-nox systemd[1]: Failed to start Advanced key-value store.
```

Por más que lo intentamos no lo consigue. Veamos que podemos hacer.

Analizando el problema

Systemctl logs

```
journalctl -xeu redis-server.service
░░ The job identifier is 3699 and the job result is done.
jun 02 17:45:14 abkrim-nox systemd[1]: redis-server.service: Start request repeated too
quickly.
jun 02 17:45:14 abkrim-nox systemd[1]: redis-server.service: Failed with result 'exit-code'.
░░ Subject: Unit failed
░░ Defined-By: systemd
░░ Support: http://www.ubuntu.com/support
░░
░░ The unit redis-server.service has entered the 'failed' state with result 'exit-code'.
jun 02 17:45:14 abkrim-nox systemd[1]: Failed to start Advanced key-value store.
```

Bueno ya tenemos una pista pero viendo los logs (bendita bitácora) podemos obtener más información.

```
sudo tail -n100  /var/log/redis/redis-server.log
…
```

```
7471:M 02 Jun 2022 17:48:00.413 * DB loaded from base file appendonly.aof.66.base.rdb: 0.000
seconds
7471:M 02 Jun 2022 17:48:00.893 # Bad file format reading the append only file
appendonly.aof.66.incr.aof: make a backup of your AOF file, then use ./redis-check-aof --fix
<filename.manifest>
```

Con esto vemos que nuestra configuración de redis esta configurado para usar una estrategia AOF (append-only file) para evitar perdidas de datos en caso de una terminación brusca (energía, kill -9,..) que no permita la escritura de los datos activos a disco.

`appendonly yes`

Y además de esto, nuestro fichero AOF esta corrupto. Así pues hay que recuperarlo.

Reparando el fichero AOF

El comando general es `redis-check-aof –fix ahora falta encontrar tanto el binario de la utilidad como el fichero.

En mi caso un Ubuntu 22.04 con redis instalado vía repositorio, siendo la versión 6.7

```
❯ sudo ls -lisah /var/lib/redis

total 16K
17170880 4,0K drwxr-x---  3 redis redis 4,0K jun  2 20:19 .
16777254 4,0K drwxr-xr-x 94 root  root  4,0K may 31 10:43 ..
17170910 4,0K drwxr-x---  2 redis redis 4,0K jun  2 18:34 appendonlydir
17170564 4,0K -rw-rw----  1 redis redis 1,6K jun  2 20:19 dump.rdb
❯ sudo ls -lisah /var/lib/redis/appendonlydir

total 836K
17170910 4,0K drwxr-x--- 2 redis redis 4,0K jun  2 18:34 .
17170880 4,0K drwxr-x--- 3 redis redis 4,0K jun  2 20:19 ..
17171350 4,0K -rw-rw---- 1 redis redis 1,6K jun  2 18:34 appendonly.aof.67.base.rdb
17171130 820K -rw-r----- 1 redis redis 815K jun  2 20:21 appendonly.aof.67.incr.aof
17171294 4,0K -rw-r----- 1 redis redis   92 jun  2 18:34 appendonly.aof.manifest
```

Así pues el comando sería

```
sudo /usr/bin/redis-check-aof --fix /var/lib/redis/appendonlydir/appendonly.aof.manifest

Start checking Multi Part AOF
Start to check BASE AOF (RDB format).
[offset 0] Checking RDB file appendonly.aof.66.base.rdb
[offset 26] AUX FIELD redis-ver = '7.0.0'
[offset 40] AUX FIELD redis-bits = '64'
[offset 52] AUX FIELD ctime = '1653893946'
[offset 67] AUX FIELD used-mem = '5178616'
[offset 79] AUX FIELD aof-base = '1'
[offset 81] Selecting DB ID 0
[offset 3048] Checksum OK
[offset 3048] \o/ RDB looks OK! \o/
[info] 19 keys read
[info] 9 expires
[info] 9 already expired
RDB preamble is OK, proceeding with AOF tail...
AOF analyzed: filename=appendonly.aof.66.base.rdb, size=3048, ok_up_to=3048, ok_up_to_line=1,
diff=0
BASE AOF appendonly.aof.66.base.rdb is valid
Start to check INCR files.
AOF appendonly.aof.66.incr.aof format error
AOF analyzed: filename=appendonly.aof.66.incr.aof, size=64241117, ok_up_to=64238121,
ok_up_to_line=4261202, diff=2996
This will shrink the AOF appendonly.aof.66.incr.aof from 64241117 bytes, with 2996 bytes, to
64238121 bytes
Continue? [y/N]: y
Successfully truncated AOF appendonly.aof.66.incr.aof
All AOF files and manifest are valid
```

Después de esto ya podremos iniciar redis

```
❯ sudo systemctl restart redis-server
❯ sudo systemctl status redis-server
● redis-server.service - Advanced key-value store
    Loaded: loaded (/lib/systemd/system/redis-server.service; enabled; vendor preset: enabled)

    Active: active (running) since Thu 2022-06-02 17:56:19 CEST; 7min ago
      Docs: http://redis.io/documentation,
            man:redis-server(1)
```

```
  Main PID: 12969 (redis-server)
    Status: "Ready to accept connections"
     Tasks: 6 (limit: 38330)
    Memory: 3.9M
       CPU: 1.538s
    CGroup: /system.slice/redis-server.service
            └─12969 "/usr/bin/redis-server 127.0.0.1:6379" "" "" "" "" "" "" ""

jun 02 17:56:18 abkrim-nox systemd[1]: Starting Advanced key-value store...
jun 02 17:56:19 abkrim-nox systemd[1]: Started Advanced key-value store.
```

Enlaces

- Redis persistence

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Comando find con -maxdepth excluyendo el propio directorio

Comando find con -maxdepth excluyendo el propio directorio

A veces es necesario ejecutar este comando de manera recursiva pero queremos obviar el propio directorio desde el que se ejecuta como por ejemplo para eliminar todos directorios de una carpeta de rsync con enlaces duros de forma ordenada.

```
find . -maxdepth 1 -type d | sed -r '/^\.$/d'
```

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Du y los ficheros o directorios ocultos

El comando du y los ficheros ocultos

Muchas veces al usar el comando `du` para localiza directorios con elevado o anormal consumo de espacio en disco nos topamos, con un directorio en el que la información que `du` nos ofrece no se ajusta a nivel directorio con el nivel subdirectorios.

```
❯ du -sh *
1,3G	api
7,1G	investigo
5,2G	sitelight
1,4G	sitelight2

❯ cd investigo
❯ du -sh *
179M	investigo/backup
20K	investigo/conf
408K	investigo/logs
275M	investigo/sitelight
3,2G	investigo/web
```

Nos faltan casi 4 Gb.

Comando du incluyendo los directorios ocultos

```
du -hs investigo/.[^.]*
4,0K	investigo/.bash_history
4,0K	investigo/.bash_logout
4,0K	investigo/.bashrc
3,4G	investigo/.cache
604K	investigo/.config
8,0K	investigo/.emacs.d
4,0K	investigo/.gitconfig
32K	investigo/.java
96K	investigo/.local
4,0K	investigo/.mysql_history
32M	investigo/.npm
18M	investigo/.oh-my-zsh
92K	investigo/.p10k.zsh
4,0K	investigo/.profile
4,0K	investigo/.shell.pre-oh-my-zsh
20K	investigo/.ssh
4,0K	investigo/.Xauthority
8,0K	investigo/.yarn
4,0K	investigo/.yarnrc
48K	investigo/.zcompdump
52K	investigo/.zcompdump-coresitelight-5.8
116K	investigo/.zcompdump-coresitelight-5.8.zwc
28K	investigo/.zsh_history
12K	investigo/.zshrc
```

Alternativas

```
du -sh .[!.]* * 2>/dev/null | sort -rh


# con shopt:
shopt -s dotglob && du -sh */ | sort -rh
```

- Man page command du
- du command does not parse hidden directories

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como esta, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Como vaciar o eliminar emails antiguos en dovecot sin usar find

Dovecot y los comandos olvidados (Re-editado 2025/07/31)

Muchas veces, y me incluyó yo, por vaguería y algo de desconocimiento, pues una se acostumbra a los fácil, usamos una combinación de `find` para hacer un vaciado de alguna cuenta o carpeta de correo que se llenó.

Pues bien, eso es mejor hacerlo con las herramientas del propio Dovecot (si es este el sistema de servidor IMAP que usamos)

Eliminación de correos IMAP por antigüedad

Ejemplo

```
doveadm expunge -u jane.doe@example.org mailbox Spam before 2w
```

Redicción 2025/07/31

Diferencia entre `savedbefore` y `before` en Dovecot:

`savedbefore`:

- Fecha cuando Dovecot **guardó físicamente** el email en el servidor
- Se actualiza con migraciones, restores, reorganizaciones del maildir
- **Problemático** después de mantenimientos del servidor

`before`:

- Fecha **real de recepción/entrega** del email
- Corresponde a cuando el email **llegó originalmente** al buzón
- **Más confiable** para limpiezas por antigüedad

Resumen práctico:

- `before` → Usa la fecha real/original del email
- `savedbefore` → Usa fecha técnica de almacenamiento (evitar tras restores)

En tu caso: después del restore complicado que mencionaste, todos los emails tienen `saved_date` reciente, por eso `savedbefore 7d` no encontraba nada, pero `before 7d` sí encuentra los 86K emails realmente antiguos.

Usa siempre `before` para limpiezas normales.

Obtener la lista de buzones

Dado que los buzones se escriben en el shell de distinta manera, para usarse en el comando es bueno obtener la liista

```
doveadm mailbox list -u jane.doe@example.org
Archive
Mantenimientos
Mantenimientos/mysql
ASSP
Seguridad
Seguridad/inmunifyAV
Seguridad/Wordfence
Services
Services/Failed
LFD
[Gmail]
[Gmail]/Importantes
Junk
```

```
Trash
Sent
Drafts
INBOX
```

Conteo de mensajes

```
doveadm mailbox status -u  jane.doe@example.orgm messages Sistemas.correo
Sistemas.correo messages=5267
```

Purgado por asunto

```
doveadm expunge -u jane.doe@example.org mailbox 'Mantenimientos/mysql' HEADER Subject "Palabra Clave"
```

Purgado mas complejo HEADER y BODY

```
doveadm expunge -u jane.doe@example.org mailbox 'Mantenimientos/mysql' HEADER Subject "Palabra Clave" BODY "Otro texto"
```

- Consulta la documentación de Dovecot - Expunge
- doveadm: Delete messages older than date

Eliminación por linea de asunto

Ejemplo

```
doveadm expunge -u jane.doe@example.org mailbox INBOX subject Cron
```

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

rc.local en systemas Debian usando systemd. Ejemplo redis

Introducción

Algunos servicios como **redis** pueden requerir de ciertas configuraciones del sistema para su mejor fucnionamiento, como puede ser **Transparent Huge Pages (THP)** desactivado. Si puedes desactivarlo porque no afecta a otros servicios de tu servidor (atento a esto, que no es salir por la calle del medio sin pensar en las implicaciones) es bueno hacerlo.

```
3336867:M 04 Dec 2023 18:05:32.941 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This will create latency and memory usage issues with Redis. To fix this issue run the command 'echo never > /sys/kernel/mm/transparent_hugepage/enabled' as root, and add it to your /etc/rc.local in order to retain the setting after a reboot. Redis must be restarted after THP is disabled.
```

Solución

La solución pasa por desactivarlo en el sistema

```
echo never > /sys/kernel/mm/transparent_hugepage/enabled
```

Pero necesitamos ejecutar esto al reinicio, o perdermos la configuración deseada. La opcion normal sería añadirl al scripot de arranque, como muchos dicen, en `/etc/rc.local` pero distors basadas en Debia, o otras basadas en RedHat, no tiene ese fiochero porque usan ya de hace tiempo **systemd** para la gestión de estas cuestiones.

Crear un Servicio systemd para Desactivar THP

Crear un Archivo de Servicio systemd: Abre un nuevo archivo en el directorio de servicios de systemd con un editor de texto como nano o vim. Por ejemplo:

```
sudo nano /etc/systemd/system/disable-thp.service
```

Donde añadimos

```
[Unit]
Description=Disable Transparent Huge Pages (THP)

[Service]
Type=oneshot
ExecStart=/bin/sh -c 'echo never > /sys/kernel/mm/transparent_hugepage/enabled'

[Install]
WantedBy=multi-user.target
```

Lo habilitamos

```
sudo systemctl daemon-reload
sudo systemctl enable disable-thp.service
sudo systemctl start disable-thp.service
```

Verificamos

```
cat /sys/kernel/mm/transparent_hugepage/enabled
always madvise [never]
```

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Bad Bots y la pesadilla del tráfico. Htaccess en Apache 2.4

Cada vez esta pero el tema. Un ejercito de web scrappers, de personas dedicadas a vivir de crear contenido falso, indexable, o de robar imáganes, pulula por la red.

Una de las mejores formas de acabar con ellos es denegarles el acceso, en nuestro fichero `.htaccess`

.htaccess anti bad bots

La cuesgtión es añadir la lista de robots no deseados en nuestro fichero `.htaccces` usando para ello las directivas `setenvif`

```
# Start Bad Bot Prevention
<IfModule mod_setenvif.c>
# SetEnvIfNoCase User-Agent ^$ bad_bot
SetEnvIfNoCase User-Agent "^12soso.*" bad_bot
SetEnvIfNoCase User-Agent "^192.comAgent.*" bad_bot
SetEnvIfNoCase User-Agent "^1Noonbot.*" bad_bot
...

<Limit GET POST PUT>
  Order Allow,Deny
  Allow from all
  Deny from env=bad_bot
</Limit>
</IfModule>
# End Bad Bot Prevention
```

- `<IfModule mod_setenvif.c>` Esta directiva comprueba si el módulo mod_setenvif está habilitado. Si lo está, se ejecuta el código dentro de este bloque.
- `SetEnvIfNoCase User-Agent "^12soso.*" bad_bot`: Esta directiva establece una variable de entorno llamada bad_bot si el User-Agent comienza con "12soso".
- `<Limit GET POST PUT>`: Esta directiva limita las reglas dentro de este bloque a los métodos HTTP especificados (GET, POST y PUT).
- `Order Allow,Deny`: Define el orden en el que se aplican las reglas de acceso. Primero se aplican las reglas `Allow` y luego las reglas `Deny`
- `Allow from all`: Permite el acceso a todos por defecto.
- `Deny from env=bad_bot`: Deniega el acceso a cualquier solicitud que tenga la variable de entorno bad_bot establecida.
- los `</` cierra la directiva corrrespondiente

La lista

Hay muchas, pero esta lista es un buen punto de partida y se actualiza regularmente. Incluso puedes mantenerla con algun pequeño script.

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Sudo sin contraseña

Configuración y uso de `sudo` en sistemas linux

Introducción sobre el comando `sudo`

¿Qué Hace `sudo`?

El comando `sudo` (Super User DO) permite a un usuario autorizado ejecutar comandos con privilegios de otro usuario, normalmente el usuario root. Es una herramienta fundamental para la administración de sistemas, ya que facilita la realización de tareas que requieren permisos elevados sin necesidad de cambiar de usuario.

Alcances en seguridad y confiabilidad

- **Seguridad**: `sudo` mejora la seguridad al permitir un control granular sobre quién puede ejecutar qué comandos. Registra todas las actividades realizadas, lo que facilita la auditoría y el seguimiento de acciones administrativas.
- **Confiabilidad**: Reduce la necesidad de compartir la contraseña de root, limitando el acceso a privilegios elevados únicamente a usuarios específicos y comandos determinados.

Ubicación de su configuración

La configuración de `sudo` se encuentra principalmente en el archivo `/etc/sudoers`. Además, se pueden añadir configuraciones específicas en el directorio `/etc/sudoers.d/`.

> En Macos lo ficheros `/etc` realmente estan en `/private/etc/`

Usar `visudo` para la edición (aconsejado)

Editar el archivo `/etc/sudoers` directamente puede ser arriegado, ya que un error de sintaxis puede bloquear el acceso administrativo. Por ello, se recomienda utilizar el comando `visudo`, que verifica la sintaxis antes de aplicar los cambios.

Hacer un backup antes de editar

Antes de modificar el archivo `sudoers`, es prudente realizar una copia de seguridad:

```
sudo cp /etc/sudoers /etc/sudoers.backup_$(date +%Y%m%d)
```

Este comando crea una copia del archivo `sudoers` con la fecha actual, facilitando la restauración en caso de errores.

Editar con `visudo`

Para editar el archivo `sudoers` de manera segura:

```
sudo visudo
```

Este comando abre el archivo en el editor predeterminado configurado para `visudo` (generalmente `nano` o `vi`) y verifica la sintaxis al guardar.

Diferentes posibilidades de configuración

Permitir `sudo` sin pedir contraseña

Para que un usuario pueda ejecutar comandos con `sudo` sin necesidad de ingresar una contraseña, añade la siguiente línea en el archivo `sudoers`:

```
usuario ALL=(ALL) NOPASSWD:ALL
```

Ejemplo:

```
javier ALL=(ALL) NOPASSWD:ALL
```

Permitir `sudo` sin pedir contraseña pero limitado a algunos comandos

Para otorgar permisos de `sudo` sin contraseña pero restringidos a comandos específicos:

```
usuario ALL=(ALL) NOPASSWD:/ruta/al/comando1, /ruta/al/comando2
```

Si tenemos mas d eun comando puede ser mas pratico usar una variable, un fichero de .conf especifico.

```
touch /etc/sudoers.d/mi_usuario_sudo
visudo -f /etc/sudoers.d/mi_usuario_sudo
```

Añade la configuración deseada (es un ejemplo)

```
Cmnd_Alias PRTG = /usr/sbin/csf, /usr/local/directadmin/scripts/letsencrypt.sh, /usr/bin/ls,
/usr/bin/cat, /usr/bin/tail
admin ALL=(ALL) NOPASSWD: PRTG
```

Guadar y cerrar.

Ejemplo:

```
javier ALL=(ALL) NOPASSWD:/usr/bin/systemctl restart nginx, /usr/bin/systemctl status nginx
```

Atención: En sistemas como macOS, una configuración incorrecta que elimina la solicitud de contraseña puede bloquear el acceso administrativo si no existe otro usuario con privilegios de superadministrador.

Usar `/etc/sudoers.d/` para configuraciones específicas de usuarios

En lugar de modificar directamente el archivo `sudoers`, es posible crear archivos individuales para cada usuario en el directorio `/etc/sudoers.d/`. Esto facilita la gestión y evita conflictos.

Creación de un Archivo de Configuración para un Usuario

1. **Crear el Archivo:**

   ```
   sudo nano /etc/sudoers.d/usuario
   ```

2. **Añadir las Reglas de `sudo`:**
   Permitir `sudo` sin contraseña:

   ```
   usuario ALL=(ALL) NOPASSWD:ALL
   ```

   Permitir `sudo` sin contraseña pero limitado a ciertos comandos:

   ```
   usuario ALL=(ALL) NOPASSWD:/usr/bin/systemctl restart nginx, /usr/bin/systemctl
   status nginx
   ```

3. **Guardar y Cerrar el Archivo:**
   Presiona `Ctrl + X`, luego `Y` y `Enter` para guardar los cambios.
4. **Verificar la Sintaxis:**
   `visudo` automáticamente verifica la sintaxis al editar el archivo. Sin embargo, puedes comprobar manualmente ejecutando:

   ```
   sudo visudo -cf /etc/sudoers.d/usuario
   ```

   Este comando validará la configuración e inmformará de cualquier error.

Resumen de comandos clave

```
# Crear una copia de seguridad del archivo sudoers
sudo cp /etc/sudoers /etc/sudoers.backup_$(date +%Y%m%d)

# Editar el archivo sudoers de manera segura
sudo visudo

# Permitir a un usuario ejecutar todos los comandos sin contraseña
usuario ALL=(ALL) NOPASSWD:ALL

# Permitir a un usuario ejecutar comandos específicos sin contraseña
usuario ALL=(ALL) NOPASSWD:/ruta/al/comando1, /ruta/al/comando2

# Crear un archivo de configuración específico para un usuario
sudo nano /etc/sudoers.d/usuario

# Verificar la sintaxis de un archivo en sudoers.d
```

```
sudo visudo -cf /etc/sudoers.d/usuario
```

Consideraciones Adicionales

- **Permisos de Archivos:** Asegúrate de que los archivos en `/etc/sudoers.d/` tengan permisos correctos (generalmente 0440) para evitar problemas de seguridad.

  ```
  sudo chmod 0440 /etc/sudoers.d/usuario
  ```

- **Evitar Errores de Sintaxis:** Siempre utiliza `visudo` o editores diseñados para manejar la configuración de `sudo` para prevenir errores que puedan bloquear el acceso administrativo.
- **Uso Responsable de `NOPASSWD`:** Otorgar permisos sin contraseña debe hacerse con cautela, limitando el acceso solo a los comandos estrictamente necesarios para minimizar riesgos de seguridad.
- **Documentación y Auditoría:** Mantén una documentación clara de las configuraciones realizadas y revisa periódicamente los permisos otorgados para asegurar que siguen siendo necesarios y seguros.
- En **MacOS** una configuración erronea del usuario administrador puede ser fatal y muy complicada la recuperación del desastre. Se recomienda por esto y po rmuchas mas cosas, tener siempre un segundo usuario SuperAdmin en un sistema MacOs

Conclusión

El uso adecuado de `sudo` es esencial para la administración segura y eficiente de sistemas Linux. Configurarlo correctamente, utilizando herramientas como `visudo` y aplicando buenas prácticas de seguridad, garantiza que los usuarios puedan realizar tareas administrativas sin comprometer la integridad y seguridad del sistema.

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Reparar el fichero .zsh_history: zsh: corrupt history file /home/USER/.zsh_history

Introducción

A ves por lo que sea se corrompe el fichero de historia de ZSH `.zsh_history` y nos aparece el mensaje en el shell, cuando entramos al shell, actualizamos Oh My Zsh

```
zsh: corrupt history file /home/USER/.zsh_history
```

Se puede corregir el archivo de historial corrupto de Zsh (`.zsh_history`) siguiendo estos pasos. La corrupción del archivo de historial puede ocurrir por diversas razones, como cierres inesperados de la terminal o conflictos de escritura. Aquí te muestro cómo solucionarlo:

Pasos para Corregir un Archivo de Historial Corrupto en Zsh

1. **Hacer una Copia de Seguridad**: Antes de realizar cualquier cambio, es recomendable hacer una copia de seguridad del archivo de historial corrupto:

   ```
   cp ~/.zsh_history ~/.zsh_history_backup
   ```

2. **Eliminar Líneas Corruptas**: Abre el archivo de historial con un editor de texto como `nano` o `vim` y elimina las líneas corruptas o caracteres extraños. Puedes usar el siguiente comando para abrirlo con `nano`:

   ```
   nano ~/.zsh_history
   ```

Busca líneas que no tengan sentido o que estén incompletas y elimínalas.

3. **Reparar el Archivo de Historial**: Puedes intentar reparar el archivo de historial usando el siguiente comando, que eliminará líneas corruptas automáticamente:

   ```
   strings ~/.zsh_history > ~/.zsh_history_clean
   mv ~/.zsh_history_clean ~/.zsh_history
   ```

   El comando `strings` extrae solo las cadenas de texto legibles, lo que puede ayudar a limpiar el archivo.
4. **Recargar el Historial**: Después de limpiar el archivo, recarga el historial en la sesión actual de Zsh:

   ```
   source ~/.zshrc
   ```

5. **Verificar el Historial**: Asegúrate de que el historial se carga correctamente y que no hay mensajes de error al iniciar una nueva sesión de terminal.

Notas Adicionales

- **Permisos**: Asegúrate de tener los permisos adecuados para editar el archivo `.zsh_history`.
- **Consistencia**: Si experimentas problemas frecuentes con el historial, verifica que no haya múltiples instancias de Zsh escribiendo en el archivo al mismo tiempo.

Siguiendo estos pasos, deberías poder corregir el archivo de historial corrupto de Zsh y evitar problemas futuros.

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

El archivo hosts para trabajar en un servidor o ip distinto del de la resolución DNS

Introudccion fichero de sistema: hosts

El archivo `hosts` es un archivo de texto simple que se utiliza para mapear nombres de host a direcciones IP. Esto es útil para redirigir nombres de dominio a direcciones IP específicas, lo que puede ser especialmente útil durante las migraciones de sistemas para probar configuraciones antes de pasar a producción.

Ubicaciones del Archivo `hosts`

- **Windows**: `C:\Windows\System32\drivers\etc\hosts`
- **Linux**: `/etc/hosts`
- **MacOS**: `/private//etc/hosts`

Uso del Archivo `hosts`

El archivo `hosts` permite redirigir un nombre de dominio a una dirección IP específica. Esto es útil para pruebas locales o para verificar un sitio web en un servidor diferente antes de cambiar los registros DNS.

Formato del Archivo `hosts`

El archivo `hosts` tiene un formato simple, donde cada línea contiene una dirección IP seguida de uno o más nombres de host. Los comentarios se pueden agregar usando el símbolo `#`.

```
# Example of a hosts file entry
127.0.0.1   localhost
192.168.1.100   example.com www.example.com
```

Ejemplo de Uso para Migraciones

Supongamos que estás migrando un sitio web a un nuevo servidor con la dirección IP `192.168.1.100` y quieres probar el sitio antes de actualizar los registros DNS. Puedes editar el archivo `hosts` para redirigir el dominio a la nueva IP:

1. **Abrir el Archivo `hosts`**:
   - Usa un editor de texto con privilegios de administrador para abrir el archivo `hosts` en tu sistema operativo.
2. **Agregar una Entrada para el Dominio**:
   - Añade una línea en el archivo `hosts` con la nueva dirección IP y el dominio que deseas redirigir.

   ```
   192.168.1.100   example.com www.example.com
   ```

3. **Guardar los Cambios**:
   - Guarda el archivo y cierra el editor.
4. **Verificar la Redirección**:
   - Abre un navegador web e ingresa el dominio `example.com`. Deberías ser redirigido a la dirección IP especificada en el archivo `hosts`.

Consideraciones

- **Privilegios de Administrador**: Necesitarás privilegios de administrador para editar el archivo `hosts`.(uso de sudo, su...)
- **Cache de DNS**: Puede ser necesario limpiar la caché de DNS después de editar el archivo `hosts` para que los cambios surtan efecto. En Windows, puedes usar el comando `ipconfig /flushdns`. En MacOS y Linux, puedes usar `dscacheutil -flushcache` o `sudo systemd-resolve --flush-caches`.
- **Temporalidad**: Recuerda que los cambios en el archivo `hosts` son locales a tu máquina y no afectan a otros usuarios o dispositivos.

Este método es muy útil para pruebas y verificaciones antes de realizar cambios en los registros DNS públicos.

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Cambiar repositorios de Ubuntu al mirror de OVH

Objetivo

Actualizar los repositorios APT de Ubuntu para que utilicen el mirror de OVH:

- Mejora la velocidad y disponibilidad si el servidor está en OVH o cerca geográficamente
- Evita usar mirrors como `es.archive.ubuntu.com` o `archive.ubuntu.com`, que pueden tener latencias o caídas

Mirror recomendado para OVH el suyo propio

```
http://ubuntu.mirrors.ovh.net/ftp.ubuntu.com/ubuntu/
```

Sistemas con /etc/apt/sources.list (formato clásico)

Copia de seguridad

```
cp /etc/apt/sources.list /etc/apt/sources.list.bak
```

Sustitución automática del mirror

```
sed -i
's|http://.*\.archive\.ubuntu\.com/ubuntu|http://ubuntu.mirrors.ovh.net/ftp.ubuntu.com/ubuntu|
g' /etc/apt/sources.list
sed -i 's|http://\([a-
z]*\.\)\?archive\.ubuntu\.com/ubuntu|http://ubuntu.mirrors.ovh.net/ftp.ubuntu.com/ubuntu|g'
/etc/apt/sources.list
```

- Cambia cualquier mirror genérico o regional por el de OVH
- Aplica a todas las entradas (main, security, updates, etc.)

Sistemas con /etc/apt/sources.list.d/ubuntu.sources (formato deb822)

Copia de seguridad

```
cp /etc/apt/sources.list.d/ubuntu.sources /etc/apt/sources.list.d/ubuntu.sources.bak
```

Sustitución automática del mirror

```
sed -i 's|URIs: http://.*\.archive\.ubuntu\.com/ubuntu|URIs:
http://ubuntu.mirrors.ovh.net/ftp.ubuntu.com/ubuntu|g' /etc/apt/sources.list.d/ubuntu.sources
sed -i 's|URIs: http://security\.ubuntu\.com/ubuntu/|URIs:
http://ubuntu.mirrors.ovh.net/ftp.ubuntu.com/ubuntu/|g' /etc/apt/sources.list.d/ubuntu.sources
```

- Cambia cualquier URI de mirror genérico o regional por el de OVH
- Aplica a todas las entradas del archivo formato deb822

Verificación

```
apt update
```

- Asegúrate de que todos los paquetes se descargan desde el mirror de OVH
- No deberían aparecer errores ni advertencias de conexión

Recomendaciones

- Este cambio puede repetirse en otros ficheros dentro de `/etc/apt/sources.list.d/`
- En entornos con múltiples servidores, se recomienda automatizar la sustitución con:
  - Scripts en bash
  - Ansible
  - Herramientas de orquestación o ejecución remota (como pssh)

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Guía Rápida de find - Búsquedas Avanzadas

□□ Buscar por Antigüedad

Archivos más antiguos que:

```
# Más de 1 año
find /directorio -type f -mtime +365

# Más de 1 mes (30 días)
find /directorio -type f -mtime +30

# Más de 1 semana
find /directorio -type f -mtime +7

# Más de 1 día
find /directorio -type f -mtime +1
```

Archivos modificados en los últimos:

```
# Últimas 24 horas
find /directorio -type f -mtime -1

# Últimos 7 días
find /directorio -type f -mtime -7

# Últimos 30 días
find /directorio -type f -mtime -30
```

Rango de fechas:

```
# Entre 7 y 30 días
find /directorio -type f -mtime +7 -mtime -30
```

```
# Entre 1 y 6 meses
find /directorio -type f -mtime +30 -mtime -180
```

□□ Buscar por Tamaño

Mayor que:

```
# Mayor que 100MB
find /directorio -type f -size +100M

# Mayor que 1GB
find /directorio -type f -size +1G

# Mayor que 10KB
find /directorio -type f -size +10k
```

Menor que:

```
# Menor que 1MB
find /directorio -type f -size -1M

# Menor que 100KB
find /directorio -type f -size -100k
```

Tamaño exacto:

```
# Exactamente 1MB
find /directorio -type f -size 1M
```

Rangos de tamaño:

```
# Entre 1MB y 10MB
find /directorio -type f -size +1M -size -10M

# Entre 100KB y 1MB
find /directorio -type f -size +100k -size -1M
```

□□ Combinaciones Útiles

Archivos grandes y antiguos:

```
# Archivos > 100MB y > 1 año
find /directorio -type f -size +100M -mtime +365

# Archivos > 1GB y > 30 días
find /directorio -type f -size +1G -mtime +30
```

Con acciones:

```
# Listar con detalles
find /directorio -type f -size +100M -mtime +30 -ls

# Eliminar archivos (¡CUIDADO!)
find /directorio -type f -size +1G -mtime +365 -delete

# Listar tamaños legibles
find /directorio -type f -size +100M -exec ls -lh {} \;
```

□□ Unidades de Medida

Unidad	Significado
`c`	bytes
`k`	KB (1024 bytes)
`M`	MB (1024 KB)
`G`	GB (1024 MB)

□ Tipos de Tiempo

Opción	Significado
`-mtime`	Modificación
`-atime`	Acceso
`-ctime`	Cambio de atributos

□□ Ejemplos Prácticos

```
# Logs antiguos y grandes
find /var/log -name "*.log" -size +10M -mtime +7

# Archivos temporales antiguos
find /tmp -type f -mtime +1 -delete

# Archivos grandes en home
find /home -type f -size +500M -mtime +30 -ls
```

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

APT Upgrade - Modos Desatendido vs Manual

Problema común

Por defecto, `apt upgrade -y` se queda **colgado esperando input** cuando hay conflictos en archivos de configuración modificados, rompiendo la automatización en servidores. En situaciones de mantenimiento desatendido, es necesario obviarlo.

Tipos de preguntas en apt upgrade

1. Confirmación inicial (lista de paquetes)

```
The following packages will be upgraded:
  zabbix-agent2 nginx mysql-server (y otros 25 paquetes)
Do you want to continue? [Y/n]
```

2. Conflictos de configuración (el problema real)

```
Configuration file '/etc/zabbix/zabbix_agent2.conf'
 ==> Modified (by you or by a script) since installation.
 ==> Package distributor has shipped an updated version.
   What would you like to do about it ?  Your options are:
    Y or I  : install the package maintainer's version
    N or O  : keep your currently-installed version
      D     : show the differences between the versions
      Z     : start a shell to examine the situation
```

Soluciones para upgrades desatendidos

Método 1: Opciones en línea de comandos (puntual)

```
# Mantener siempre la configuración actual (recomendado)
sudo apt -o Dpkg::Options::="--force-confold" upgrade -y

# Usar siempre la nueva configuración del paquete
sudo apt -o Dpkg::Options::="--force-confnew" upgrade -y

# Comando más completo (recomendado para scripts)
sudo DEBIAN_FRONTEND=noninteractive apt \
  -o Dpkg::Options::="--force-confold" \
  -o Dpkg::Options::="--force-confdef" \
  upgrade -y
```

> Cada uno de los métodos tiene su necesidad de entenderlo. La opcion **usar la nueva configuración**, puede ser muy problemática, pues por norma general, muchos o simplemenet algunos ficheros de configuración pueden estar modificados y su reseteo a la versión nueva, puede dejar el servicio inconsistente e incluso en algunos casos, perder la accesibilidad al servidor (sshd). El desatendido, en el que queremos conservar el fichero atcual, es lo más común, pero puede hacernos olvidar que a veces hay cambios que hay que aplicarlos. Vivimos tiempos de inmediatez, y de pocos recursos para administración de sistemas, y esto puede ser un problema.

Método 2: Configuración global permanente (recomendado)

```
# Crear archivo de configuración global
sudo tee /etc/apt/apt.conf.d/50unattended-upgrades << 'EOF'
Dpkg::Options {
    "--force-confold";
    "--force-confdef";
}
EOF
```

Resultado después de crear el archivo:

- `sudo apt upgrade -y` → Totalmente desatendido
- `sudo apt upgrade` → Solo pregunta confirmación inicial, conflictos automáticos

Método 3: Variable de entorno (temporal)

```
# Para la sesión actual
export DEBIAN_FRONTEND=noninteractive
sudo -E apt upgrade -y

# Para un comando específico
sudo DEBIAN_FRONTEND=noninteractive apt upgrade -y
```

Opciones de dpkg explicadas

Opción	Descripción	Equivale a presionar
`--force-confold`	Mantener configuración actual	**N** (No)
`--force-confnew`	Usar nueva configuración del paquete	**Y** (Yes)
`--force-confdef`	Usar respuesta por defecto si existe	Automático
`--force-confask`	Preguntar siempre (modo interactivo)	Pregunta manual
`--force-confmiss`	Instalar archivos de configuración faltantes	Automático

Sobreescribir configuración global (uso manual)

Cuando tienes configuración global pero quieres control manual

```
# Desactivar todas las opciones automáticas (preguntará todo)
sudo apt -o Dpkg::Options::="" upgrade

# Forzar modo interactivo completo
sudo apt -o Dpkg::Options::="--force-confask" upgrade

# Usar nuevas configuraciones temporalmente
sudo apt -o Dpkg::Options::="--force-confnew" upgrade

# Modo interactivo con interfaz gráfica
sudo DEBIAN_FRONTEND=dialog apt upgrade
```

Para casos específicos

```
# Solo preguntar en conflictos (recomendado para revisión)
sudo apt -o Dpkg::Options::="--force-confask" -o Dpkg::Options::="--force-confdef" upgrade

# Mostrar diferencias antes de decidir
sudo apt -o Dpkg::Options::="--force-confask" upgrade
# (Luego presionar 'D' para ver diferencias)
```

Verificar configuración actual

```
# Ver toda la configuración de apt
apt-config dump
```

```
# Ver solo opciones de dpkg
apt-config dump | grep -i dpkg

# Ver archivos de configuración de apt
ls -la /etc/apt/apt.conf.d/

# Verificar contenido del archivo personalizado
cat /etc/apt/apt.conf.d/50unattended-upgrades
```

Casos de uso recomendados

Para servidores de producción

Conservará el fichero original

```
# Configuración global permanente (Método 2)
sudo tee /etc/apt/apt.conf.d/50unattended-upgrades << 'EOF'
Dpkg::Options {
    "--force-confold";
    "--force-confdef";
}
EOF

# Uso diario
sudo apt update && sudo apt upgrade -y
```

Para actualizaciones críticas o manuales

```
# Revisar manualmente cada conflicto
sudo apt -o Dpkg::Options::="--force-confask" upgrade

# Ver diferencias antes de decidir
# Presionar 'D' cuando aparezca el conflicto
```

Para scripts automatizados

```
#!/bin/bash
# Script de actualización desatendida
sudo DEBIAN_FRONTEND=noninteractive apt update
sudo DEBIAN_FRONTEND=noninteractive apt \
  -o Dpkg::Options::="--force-confold" \
  -o Dpkg::Options::="--force-confdef" \
  upgrade -y

# Log de cambios
echo "Upgrade completed at $(date)" >> /var/log/auto-upgrade.log
```

Para desarrollo/testing

```
# Aplicar siempre nuevas configuraciones
sudo apt -o Dpkg::Options::="--force-confnew" upgrade -y

# O mantener configuraciones personalizadas
sudo apt -o Dpkg::Options::="--force-confold" upgrade -y
```

Comandos de emergencia

Si apt se cuelga durante upgrade

```
# En otra terminal/SSH
sudo killall apt apt-get dpkg

# Reparar estado de paquetes
sudo dpkg --configure -a
sudo apt --fix-broken install
```

Limpiar locks si es necesario

```
sudo rm /var/lib/dpkg/lock-frontend
sudo rm /var/cache/apt/archives/lock
sudo dpkg --configure -a
```

Configuración recomendada paso a paso

Instalación inicial (una sola vez)

```
# 1. Crear configuración global
sudo tee /etc/apt/apt.conf.d/50unattended-upgrades << 'EOF'
Dpkg::Options {
    "--force-confold";
    "--force-confdef";
}
EOF

# 2. Verificar que funciona
sudo apt update
apt-config dump | grep -i dpkg
```

Uso diario

```
# Actualización desatendida completa
sudo apt update && sudo apt upgrade -y

# Actualización con revisión manual ocasional
sudo apt update
sudo apt -o Dpkg::Options::="--force-confask" upgrade
```

Notas importantes

- `--force-confold` es la opción más segura para producción (mantiene tus configuraciones)

- `--force-confnew` puede sobreescribir configuraciones importantes
- **Siempre hacer backup** de configuraciones críticas antes de upgrades mayores
- **La configuración global** se aplica a todos los usuarios del sistema
- **Puedes combinar opciones** para casos específicos
- **Las opciones de línea de comandos** siempre sobreescriben la configuración global

> Hacer backups regulares de /etc son **altamente reomendables** (como todos los backups o copias de respaldo)

Troubleshooting común

Problema: "dpkg was interrupted"

```
sudo dpkg --configure -a
sudo apt --fix-broken install
```

Problema: "Could not get lock"

```
sudo lsof /var/lib/dpkg/lock-frontend
sudo killall apt apt-get
sudo rm /var/lib/dpkg/lock-frontend
```

Problema: Paquetes retenidos (held back)

```
# Ver paquetes retenidos
apt list --upgradable

# Forzar actualización
sudo apt full-upgrade
```

Esta configuración te permitirá tener upgrades completamente automatizados por defecto, pero con la flexibilidad de tomar control manual cuando sea necesario.

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Crear una unidad lógica con el 100%

A veces, voy con prisa. Se me olvida siempre.

```
lvcreate -n [NOMBRE] -l 100%FREE [NOMBRE_VOLUMEN_A_USAR]
mkfs.xfs /dev/[NOMBRE_VOLUMEN_A_USAR]/[NOMBRE]
blkid -s UUID -o value /dev/[NOMBRE_VOLUMEN_A_USAR]/[NOMBRE]
# editar fstab
# UUID=[UUID] /[PUNTO_DE_MONTAJE] xfs defaults,nofail 0 2
```

Búsqueda y reemplazo recursivo con sed en directorios

Contexto

Estrategias para localizar y modificar cadenas de texto en múltiples archivos de forma recursiva, con capacidad de filtrar por tipo de archivo y excluir líneas comentadas.

Escenario típico

- Necesidad de cambiar nombres de variables, funciones o constantes en toda una base de código
- Refactorización de nombres de configuración
- Migración de valores legacy a nuevas nomenclaturas
- Requiere precisión para no modificar comentarios ni código inactivo

Estrategias de búsqueda

1. Localización básica (análisis previo)

```
# Buscar en todos los archivos
find . -type f -exec grep -l "PATRON_BUSQUEDA" {} \;

# Con contexto (3 líneas antes y después)
find . -type f -exec grep -n -B3 -A3 "PATRON_BUSQUEDA" {} + | less

# Múltiples patrones (OR lógico)
```

```
find . -type f -exec grep -l "PATRON_1\|PATRON_2\|PATRON_3" {} \;
```

2. Filtrado por tipo de archivo

```
# Solo archivos con extensión específica
find . -type f -name "*.php" -exec grep -l "PATRON" {} \;
find . -type f -name "*.js" -exec grep -l "PATRON" {} \;
find . -type f -name "*.py" -exec grep -l "PATRON" {} \;

# Múltiples extensiones
find . -type f \( -name "*.php" -o -name "*.inc" \) -exec grep -l "PATRON" {} \;

# Excluir directorios específicos
find . -type f -name "*.php" -not -path "*/vendor/*" -not -path "*/node_modules/*" \
  -exec grep -l "PATRON" {} \;
```

3. Análisis con numeración de líneas

```
# Ver archivo:línea:contenido
find . -type f -name "*.php" -exec grep -Hn "PATRON" {} \;

# Con AWK para formato personalizado
find . -type f -name "*.php" -exec awk '/PATRON/ {print FILENAME ":" NR ":" $0}' {} +
```

Estrategias de reemplazo

1. Reemplazo básico (SIN backup automático)

```
# Reemplazar en todos los archivos de un tipo
find . -type f -name "*.php" -exec sed -i 's/PATRON_VIEJO/PATRON_NUEVO/g' {} \;

# Múltiples reemplazos en una pasada
find . -type f -name "*.php" -exec sed -i '
```

```
  s/PATRON_1/NUEVO_1/g
  s/PATRON_2/NUEVO_2/g
  s/PATRON_3/NUEVO_3/g
' {} \;
```

Nota: `-i` modifica archivos en sitio. En BSD/macOS usar `sed -i ''`

2. Reemplazo CON backup automático

```
# Crea archivo.php.bak antes de modificar
find . -type f -name "*.php" -exec sed -i.bak 's/PATRON_VIEJO/PATRON_NUEVO/g' {} \;

# Limpiar backups después de verificar
find . -type f -name "*.php.bak" -delete
```

3. Excluir líneas comentadas

PHP/JavaScript/C-style

```
# Excluir líneas que empiezan por #, //, /* o *
find . -type f -name "*.php" -exec sed -i.bak '/^\s*[#*\/]/! s/PATRON/NUEVO/g' {} \;
```

Desglose del patrón:

- `/^\s*[#*\/]/!` → Negación: aplica solo si NO cumple
- `^\s*` → Inicio de línea con posibles espacios/tabs
- `[#*\/]` → Caracteres de comentario: `#`, `*`, `/`
- `!` → Negación lógica

Versión robusta (múltiples tipos de comentario)

```
find . -type f -name "*.php" -exec sed -i.bak '
/^\s*#/! {
  /^\s*\/\//! {
    /^\s*\/\*/! {
      /^\s*\*/! {
        s/PATRON_1/NUEVO_1/g
        s/PATRON_2/NUEVO_2/g
      }
```

```
    }
  }
}
' {} \;
```

Esto excluye:

- `# comentario estilo shell/Python`
- `// comentario estilo C++/JavaScript`
- `/* inicio de bloque comentario */`
- `* líneas dentro de bloque comentario`

Python

```
# Excluir líneas que empiezan por #
find . -type f -name "*.py" -exec sed -i.bak '/^\s*#/! s/PATRON/NUEVO/g' {} \;
```

Shell scripts

```
# Excluir líneas que empiezan por #
find . -type f -name "*.sh" -exec sed -i.bak '/^\s*#/! s/PATRON/NUEVO/g' {} \;
```

4. Alternativa con Perl (más flexible)

```
# Perl permite regex más complejas
find . -type f -name "*.php" -exec perl -i.bak -pe '
  s/PATRON_VIEJO/PATRON_NUEVO/g unless /^\s*[#*\/]/
' {} \;

# Múltiples patrones con hash de sustituciones
find . -type f -name "*.php" -exec perl -i.bak -pe '
  BEGIN {
    %replace = (
      "PATRON_1" => "NUEVO_1",
      "PATRON_2" => "NUEVO_2",
      "PATRON_3" => "NUEVO_3"
    );
  }
  unless (/^\s*[#*\/]/) {
```

```
    s/$_/$replace{$_}/g for keys %replace;
  }
' {} \;
```

Flujo de trabajo recomendado

```
# PASO 1: Análisis inicial (¿qué se va a cambiar?)
find . -type f -name "*.php" -exec grep -Hn "PATRON_VIEJO" {} \; | less

# PASO 2: Análisis excluyendo comentarios
find . -type f -name "*.php" -exec awk '
  !/^\s*[#*\/]/ && /PATRON_VIEJO/ {
    print FILENAME ":" NR ":" $0
  }
' {} + | less

# PASO 3: Prueba en un solo archivo
sed -i.test '/^\s*[#*\/]/! s/PATRON_VIEJO/PATRON_NUEVO/g' ruta/archivo_prueba.php
diff ruta/archivo_prueba.php ruta/archivo_prueba.php.test

# PASO 4: Aplicar con backup (EJECUTAR)
find . -type f -name "*.php" -exec sed -i.bak '/^\s*[#*\/]/! s/PATRON_VIEJO/PATRON_NUEVO/g' {}
\;

# PASO 5: Revisar cambios (si hay control de versiones)
git diff
git status

# PASO 6: Validar funcionamiento
# [Ejecutar tests, verificar aplicación]

# PASO 7: Limpiar backups si todo OK
find . -name "*.php.bak" -delete
```

Casos de uso comunes

Cambio de prefijo de base de datos

```
# MySQL/MariaDB: cambiar prefijo de tablas en dumps
find . -type f -name "*.sql" -exec sed -i.bak 's/prefijo_viejo_/prefijo_nuevo_/g' {} \;
```

Refactorización de nombres de clase

```
# PHP: cambiar nombre de clase
find . -type f -name "*.php" -exec sed -i.bak '/^\s*[#*\/]/! {
  s/class OldClassName/class NewClassName/g
  s/OldClassName::/NewClassName::/g
  s/new OldClassName/new NewClassName/g
}' {} \;
```

Migración de configuración

```
# Cambiar nombres de constantes de configuración
find . -type f \( -name "*.php" -o -name "*.inc" \) -exec sed -i.bak '/^\s*[#*\/]/! {
  s/OLD_CONFIG_NAME/NEW_CONFIG_NAME/g
  s/OLD_DB_HOST/NEW_DB_HOST/g
  s/OLD_API_KEY/NEW_API_KEY/g
}' {} \;
```

Actualización de rutas

```
# Cambiar rutas legacy
find . -type f -name "*.php" -exec sed -i.bak '/^\s*[#*\/]/! {
  s|/var/www/old_path/|/var/www/new_path/|g
  s|old-domain\.com|new-domain.com|g
}' {} \;
```

Limitaciones conocidas

1. Comentarios multilínea

El método basado en `^\s*` solo detecta comentarios al **inicio de línea**. No detecta:

```
$var = "value"; /* comentario inline */ $otra = "PATRON"; // no se excluirá
```

Para estos casos, considerar un parser específico del lenguaje.

2. Cadenas literales

Si `PATRON` aparece dentro de strings, será reemplazado:

```
$sql = "SELECT * FROM old_table"; // se cambiará "old_table"
```

Para evitarlo se requiere análisis sintáctico completo.

3. Bloques multilínea

Comentarios que abarcan múltiples líneas:

```
/*
 * Este es un comentario
 * con PATRON_VIEJO que
 * no será detectado en líneas 2-3
 */
```

Solo la primera línea (`/*`) se excluirá.

Alternativas avanzadas

Ripgrep (rg) + sd

```
# Búsqueda ultrarrápida
rg "PATRON" --type php

# Reemplazo con sd (más intuitivo que sed)
```

```
find . -name "*.php" -exec sd "PATRON_VIEJO" "PATRON_NUEVO" {} \;
```

Usando AG (The Silver Searcher)

```
# Listar archivos
ag -l "PATRON" --php

# Con contexto
ag "PATRON" --php -C 3
```

Scripts dedicados por lenguaje

Para refactorizaciones complejas, usar herramientas especializadas:

- **PHP**: PHP-CS-Fixer, Rector
- **JavaScript**: jscodeshift, lebab
- **Python**: rope, bowler

Verificación post-cambio

```
# Contar ocurrencias antes y después
echo "Antes:"
find . -name "*.php" -exec grep -o "PATRON_VIEJO" {} \; | wc -l

echo "Después:"
find . -name "*.php" -exec grep -o "PATRON_NUEVO" {} \; | wc -l

# Verificar que no quedan ocurrencias del patrón viejo
find . -name "*.php" -exec grep -H "PATRON_VIEJO" {} \;

# Si hay control de versiones
git diff --stat
git diff | grep -E "^\+.*PATRON_NUEVO" | wc -l
```

Consejos de seguridad

1. **Siempre hacer backup** antes de modificaciones masivas
2. **Usar control de versiones** (git commit antes de ejecutar)
3. **Probar en un archivo** antes de aplicar a todo el proyecto
4. **Revisar diff completo** antes de commit final
5. **Ejecutar suite de tests** después de cambios
6. **Considerar crear rama temporal** para refactorizaciones grandes

Troubleshooting

"Device or resource busy"

```
# Puede ocurrir con archivos abiertos. Solución: cerrar editores
lsof | grep nombre_archivo
```

"Permission denied"

```
# Verificar permisos
find . -type f -name "*.php" ! -writable

# Corregir si es necesario
find . -type f -name "*.php" -exec chmod u+w {} \;
```

Sed no modifica archivos

```
# Verificar sintaxis específica del sistema
sed --version  # GNU sed
sed -i.bak ...  # funciona en GNU

# En macOS/BSD
sed -i '' ...   # requiere argumento vacío
```

Última actualización: Octubre 2025
Compatibilidad: Linux (GNU sed), *BSD/macOS (requiere ajustes en `-i`)
Herramientas: find, sed, grep, awk, perl

Instalación de Node.js LTS 22 y npm en Ubuntu con CloudLinux CageFS

Contexto

Guía para instalar Node.js 22 LTS en Ubuntu cuando el servidor utiliza CloudLinux con CageFS. La documentación oficial de CloudLinux está orientada a RPM, pero este proceso está validado para sistemas Debian/Ubuntu.

Prerequisitos

- Ubuntu (validado en versiones 20.04+)
- CloudLinux con CageFS activo
- Acceso root
- Usuario con configuración mysql/mariadb ya tuneada (socket o .my.cnf)

Parte 1: Instalación base de Node.js

1. Verificar estado actual (opcional)

```
# Ver si ya tienes Node instalado
node --version
npm --version
```

```
# Ver repositorios de Node configurados
grep -r "nodesource" /etc/apt/sources.list.d/
```

2. Limpiar instalaciones previas (si existen)

```
# Eliminar repositorios antiguos de NodeSource
sudo rm -f /etc/apt/sources.list.d/nodesource.list
sudo rm -f /etc/apt/keyrings/nodesource.gpg

# Eliminar versiones antiguas
sudo apt remove nodejs npm -y
sudo apt autoremove -y
```

3. Instalar Node.js 22 LTS desde NodeSource

```
# Descargar e instalar el script de setup oficial
curl -fsSL https://deb.nodesource.com/setup_22.x | sudo -E bash -

# Instalar Node.js (incluye npm automáticamente)
sudo apt install -y nodejs

# Verificar instalación en sistema base
node --version    # Debería mostrar v22.x.x
npm --version     # Debería mostrar 10.x.x
```

4. Instalar herramientas de compilación

```
# Necesarias para compilar módulos nativos de npm
sudo apt install -y build-essential
```

Parte 2: Integración con CloudLinux CageFS

Contexto de CageFS

CageFS crea un sistema de archivos virtualizado para cada usuario, aislándolos del sistema real. Los binarios instalados en el sistema base **no son visibles automáticamente** dentro de las jaulas de los usuarios.

5. Registrar Node.js en CageFS

```
# Añadir nodejs como paquete conocido por CageFS
# Esto le indica a CageFS que debe copiar estos binarios al skeleton
cagefsctl --addrpm nodejs
```

Nota: El parámetro `--addrpm` funciona correctamente en Ubuntu aunque el nombre sugiera RPM. CageFS detecta automáticamente el gestor de paquetes del sistema.

6. Actualizar el skeleton de CageFS

```
# Forzar actualización del skeleton (estructura base de las jaulas)
cagefsctl --force-update
```

Salida esperada:

```
Copying /usr/bin/node to /usr/share/cagefs-skeleton/usr/bin/node
Creating symlink /usr/share/cagefs-skeleton/usr/bin/corepack to
../lib/node_modules/corepack/dist/corepack.js
Creating symlink /usr/share/cagefs-skeleton/usr/bin/npm to ../lib/node_modules/npm/bin/npm-
cli.js
Creating symlink /usr/share/cagefs-skeleton/usr/bin/npx to ../lib/node_modules/npm/bin/npx-
cli.js
Updating users ...
Updating user user1 ...
Updating user user2 ...
```

```
Updating user user3 ...
```

Qué hace este comando

- **Copia binarios**: `/usr/bin/node` → `/usr/share/cagefs-skeleton/usr/bin/node`
- **Crea symlinks**: npm, npx, corepack dentro del skeleton
- **Actualiza jaulas**: Propaga los cambios a todos los usuarios enjaulados
- **Preserva aislamiento**: Cada usuario ve su propia copia independiente

Verificación

7. Probar en usuario enjaulado

```
# Cambiar a un usuario con CageFS activo
su - user1

# Verificar disponibilidad de Node.js
node -v
# Output esperado: v22.11.0 (o versión actual de la rama 22.x)

npm -v
# Output esperado: 10.9.0 (o superior)

# Verificar ubicación (dentro de la jaula)
which node
# Output esperado: /usr/bin/node
```

8. Verificación adicional desde root

```
# Ver qué binarios están en el skeleton
ls -la /usr/share/cagefs-skeleton/usr/bin/ | grep node

# Verificar symlinks
ls -la /usr/share/cagefs-skeleton/usr/bin/{npm,npx,corepack}
```

```
# Ver usuarios con CageFS activo
cagefsctl --list-enabled
```

Troubleshooting

Node no visible dentro de CageFS

```
# Verificar que el usuario tiene CageFS activo
cagefsctl --user user1

# Forzar actualización de un usuario específico
cagefsctl --force-update-user user1

# Reinicializar completamente CageFS (último recurso)
cagefsctl --reinit
```

Permisos incorrectos

```
# Verificar permisos del binario en skeleton
ls -la /usr/share/cagefs-skeleton/usr/bin/node

# Debería ser ejecutable (755 o similar)
# Si no lo es:
chmod 755 /usr/share/cagefs-skeleton/usr/bin/node
cagefsctl --force-update
```

Actualización de Node.js

```
# Cuando NodeSource publique una nueva versión 22.x
sudo apt update
sudo apt upgrade nodejs

# IMPORTANTE: Después de actualizar, regenerar skeleton
cagefsctl --force-update
```

Diferencias Ubuntu vs CentOS/RPM

Aspecto	Ubuntu (DEB)	CentOS (RPM)
Instalación	apt + NodeSource	yum/dnf + NodeSource
Comando CageFS	Mismo (`--addrpm` funciona)	Mismo
Ubicación binarios	`/usr/bin/node`	`/usr/bin/node`
Documentación oficial	Escasa para DEB	Abundante para RPM
Funcionamiento	Idéntico una vez instalado	Idéntico

Notas importantes

- **Node 22 LTS** tiene soporte hasta abril 2027
- **npm viene incluido** con Node.js
- **CageFS es transparente** para el usuario final
- **Actualizaciones automáticas**: NodeSource mantiene la rama 22.x
- **Sin sudo para usuarios**: Los usuarios no necesitan permisos especiales
- **Aislamiento preservado**: Cada usuario ve su propia instancia

Resultado esperado

- □ Node.js disponible en sistema base
- □ Node.js disponible dentro de todas las jaulas CageFS
- □ npm y npx funcionales para todos los usuarios
- □ Actualizaciones automáticas del sistema afectan a todas las jaulas

Comandos de verificación rápida

```
# Desde root
node --version && npm --version

# Desde usuario enjaulado
su - user1 -c "node --version && npm --version"

# Listar paquetes registrados en CageFS
```

```
cagefsctl --list-packages | grep node
```

Validado en: Ubuntu 22.04 LTS con CloudLinux + CageFS
Fecha: Octubre 2025
Node.js: v22.11.0 LTS (Iron)

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

OpenVPN: cómo excluir destinos del túnel con `net_gateway` (el caso de apple.com / apple.es)

> Documento de referencia operativa. Origen: el Mac del operador navega con túnel completo de las VPN propias de OpenVPN (dos servidores Ubuntu 24 con `openvpn-server@server`, servicio `openvpn-server@server.service`), y apple.es / apple.com devolvían `503` mientras iCloud y App Store entraban en bucle silencioso. Solución aplicada y verificada en ambos servidores el 2026-08-27.
>
> **Palabras clave para búsqueda:** OpenVPN, split tunneling, `net_gateway`, `redirect-gateway`, exclusión de destinos, `push route`, Apple, `17.0.0.0/8`, macOS, Safari, 503, iCloud, App Store, OpenVPN Connect, system extension, SplitTunnel, CDN Akamai, Fastly, Private Relay, Acceso privado a iCloud, `mask.apple-dns.net`, ODOH, tcpdump, curl HEAD 403.

El diagnóstico correcto: no es per-app, es por destino

macOS no soporta enrutado por aplicación (per-app routing) con clientes OpenVPN estándar — ni Tunnelblick, ni Viscosity, ni OpenVPN Connect. Las apps de terceros que lo prometen (p. ej. SplitTunnel) usan extensiones de red y DNS que interceptan todo el tráfico; en esta estación resultaron inestables y se retiraron.

El problema real con Apple no era la app que navega, sino **el destino**: gran parte de la infraestructura de Apple vive en su propio bloque `17.0.0.0/8` (iCloud, App Store, activación, servidores corporativos), y ese bloque rechaza o degrada tráfico que llega desde IPs de

datacenter/VPN. La solución quirúrgica es **excluir destinos del túnel**, no aplicaciones: el tráfico hacia `17.0.0.0/8` sale por el gateway físico pre-VPN y todo lo demás sigue cifrado por el túnel.

El mecanismo: `net_gateway`

En la config del servidor OpenVPN, el keyword `net_gateway` significa «la gateway que tenía el cliente ANTES de levantar el túnel» (su router doméstico u office). Se combina con `push` para que el servidor la inyecte como ruta en cada cliente al conectar:

```
push "route 17.0.0.0 255.0.0.0 net_gateway"
```

- Con `redirect-gateway def1` activo, OpenVPN secuestra el tráfico con dos rutas `/1` (`0.0.0.0/1` y `128.0.0.0/1`) que cubren todo el espacio IPv4; la ruta `17.0.0.0/8` es **más específica** y gana sin tocar nada más.
- Es un cambio **solo de servidor**: los clientes (Mac, iPhone, iPad, Linux) no necesitan configurar nada.
- El `push` se aplica al **conectar**: los clientes ya conectados no lo ven hasta reconectar.
- Sobrevive a updates de macOS y reinicios del cliente: la ruta se re-aplica en cada reconexión por ser server-side (verificado tras un update de macOS con reinicio posterior: ruta activa y extensiones fantasma de SplitTunnel intactas).

Aplicación en el servidor (patrón con backup e idempotencia)

Config en `/etc/openvpn/server/server.conf` (Ubuntu 24, unidad `openvpn-server@server`). Procedimiento por servidor:

```
cd /etc/openvpn/server

# Idempotente: solo añade si no está
grep -q "route 17.0.0.0 255.0.0.0 net_gateway" server.conf || {
  cp -a server.conf server.conf.$(date +%F).bak
  printf '\n# Exclude Apple (17.0.0.0/8) from tunnel - route via pre-VPN gateway\npush "route
17.0.0.0 255.0.0.0 net_gateway"\n' >> server.conf
}
```

```
# Reinicio controlado (corta las sesiones activas; los clientes reconectan solos)
systemctl restart openvpn-server@server
sleep 2
systemctl is-active openvpn-server@server
ss -ulnp | grep 1194
journalctl -u openvpn-server@server --since "1 minute ago" --no-pager | grep -E
"Completed|error"
```

Verificación server-side esperada: servicio `active`, socket UDP en escucha y `Initialization Sequence Completed` en el journal.

Verificación en dos puntos (REGLA 0)

1. Desde el servidor: journal limpio tras el reinicio (arriba).

2. Desde el cliente conectado (Mac aquí), verificar el efecto y **un control negativo** para demostrar que el resto del tráfico SÍ sigue por la VPN:

```
# La ruta push llegó (debe existir la línea "17" apuntando a tu router LAN)
netstat -rn -f inet | grep -E "^0/1|^128\.0/1|^17"
#    0/1  y 128.0/1  → 10.8.0.1  utunX   (túnel completo activo)
#    17              → 192.168.x.1  en0    (exclusión Apple)

# El detalle de ruta hacia una IP de Apple sale por la interfaz física
route -n get 17.253.144.10 | grep -E "gateway|interface"
#    gateway: 192.168.x.1 / interface: en0

# CONTROL: el resto del tráfico sale por la IP de la VPN (no por casa)
curl -s https://api.ipify.org; echo

# Apple accesible con el túnel activo
curl -sI https://www.apple.com/library/test/success.html | head -1   # HTTP/2 200
```

Cuidado con las páginas que redirigen: `apple.es` responde `301` a `https://www.apple.com/es/` (servida por Akamai, FUERA del bloque 17/8). Un `curl -I` de una sola pata puede dar «correcto»

en el primer salto y que el fallo esté en el siguiente. Verificar SIEMPRE la cadena completa:

```
curl -sIL -o /dev/null -w "%{http_code} -> %{url_effective}\n" https://www.apple.es/
#    200 -> https://www.apple.com/es/
```

Y una trampa de cliente HTTP que imita a un bloqueo: `curl -sIL` envía peticiones **HEAD**, y algunos endpoints de auth de Apple (`idmsa.apple.com/IDMSWebAuth/signin.html`) responden `403` a HEAD pero `200` a GET. Un `403` en una cadena HEAD no significa que la página falle en el navegador; antes de culpar a la red, repetir con GET:

```
curl -sL -o /dev/null -w "%{http_code}\n"
"https://idmsa.apple.com/IDMSWebAuth/signin.html?..."
#    200 (el 403 de la cadena HEAD era un artefacto del método)
```

Si la IP de `api.ipify.org` es la del servidor VPN y el `route get` de Apple responde con tu router LAN, las dos cosas a la vez están probadas.

CDN de Apple fuera del bloque: añadir `/32` puntuales

`www.apple.com` y buena parte de los assets estáticos se sirven por Akamai (`www.apple.com.edgekey.net`), IPs que **no** están en `17.0.0.0/8`. Eso normalmente funciona igual por el túnel; si alguna página o descarga concreta sigue fallando, localiza su IP y exclúyela puntualmente:

```
# Localizar la IP real del dominio que falla
dscacheutil -q host -a name <dominio>.apple.com | grep ip_address

# En AMBOS servidores: ruta /32 específica
push "route A.B.C.D 255.255.255.255 net_gateway"
```

Reiniciar y reconectar para aplicarla. Es el mismo mecanismo; la receta genérica vale para cualquier destino problemático (un banco que bloquee IPs de datacenter, un servicio que exija IP doméstica, etc.).

Diagnóstico: captura de DNS y SYNs con tcpdump

Cuando las pruebas por consola están verdes pero un navegador falla, la captura de cabeceras (sin contenido, privacidad intacta) dice exactamente con quién habla el navegador y por dónde sale. Lanzarla en background y dejarla corriendo mientras se reproduce el fallo:

```
nohup sudo tcpdump -i any -n -l -c 4000 \
  '(udp port 53) or (tcp[tcpflags] & tcp-syn != 0 and dst port 443)' \


  > /tmp/captura.txt 2>&1 & disown
```

Particularidades medidas en macOS:

- `-i any` funciona (interfaz PKTAP); no existe `timeout` en BSD, acotar con `-c` (líneas).
- `tcp[tcpflags] & tcp-syn != 0` casa también los SYN-ACK; al analizar, filtrar solo los salientes (`grep "Flags \[S"` y contar el destino en el campo 5).
- Lanzado desde un shell no interactivo (agente, script), el proceso hijo **muere cuando muere el shell padre**: imprescindible `nohup … & disown`; comprobar el PID y el fichero de salida antes de dar por iniciada la captura.
- Análisis: nombres consultados con `grep "A?"` y destinos SYN con conteo por IP; correlacionar las IPs desconocidas con el mapa host→IP del flujo depurado (`dscacheutil -q host -a name <dominio>`).

El caso Safari residual: iCloud Private Relay

Con la exclusión `17/8` aplicada y el flujo de login completo verificado por `curl`, Safari podía seguir mostrando `503` en la redirección post-login de Apple ID — y una ventana privada no lo evitaba. Causa: **iCloud Private Relay (Acceso privado a iCloud)** activo.

- El Relay solo afecta al tráfico de **Safari** (y alguna app de sistema): `curl`, Chrome, Firefox o cualquier CLI van directos y no lo usan. Por eso todo lo verificado por consola estaba verde y solo el navegador fallaba — el síntoma «solo Safari» es su firma.
- Con el Relay activo, el tráfico de Safari no sale ni por casa ni por el túnel: va por los proxies de egreso de Apple, cuyas IPs **rotan entre peticiones**, y su DNS se resuelve vía ODOH (`mask.apple-dns.net`, `mask.icloud.com`). Una sesión de Apple ID creada desde una IP de egreso y continuada desde otra dispara el filtrado anti-abuso de Apple → `503` justo en

la redirección post-login.
- La ventana privada NO desactiva el Relay: es un ajuste de la cuenta iCloud, no del modo de navegación.

Detección (prueba directa, no fiarse de procesos): la captura tcpdump durante el fallo muestra consultas DNS recurrentes por `mask.apple-dns.net` / `mask.icloud.com` y conexiones a los proxies de egreso de Apple (`17.248.x.x`) mezcladas con el tráfico del sitio. La ausencia de procesos tipo `netsesamd`/`relayd` en `ps` NO prueba que el Relay esté apagado; el ajuste en Ajustes del Sistema es la referencia.

Solución: Ajustes del Sistema → [Apple ID] → iCloud → **Acceso privado a iCloud** → desactivar, cerrar Safari (`Cmd+Q`) y reabrir. Si se quiere conservar el Relay, la alternativa es usar un navegador no-WebKit (Chrome/Firefox) para los sitios hostiles a VPN: el Relay no les afecta.

Nota sobre IPv6

Estos servidores hacen `push "block-ipv6"`, así que con el túnel activo todo va por IPv4 y la exclusión `17.0.0.0/8` cubre a Apple completa. Si algún día se retira el `block-ipv6`, la exclusión solo cubriría el IPv4 de Apple; habría que valorar la contrapartida IPv6 (`route` IPv6 con `net_gateway` o bloqueo selectivo).

Nota sobre SplitTunnel y su herencia en macOS

Si se probó SplitTunnel (boundarylabs) y se desinstaló arrastrándolo a la papelera, quedan restos que CleanMyMac no ve:

- **Extensiones de sistema registradas** (`systemextensionsctl list`): `ca.boundarylabs.SplitTunnel.Extension` y `ca.boundarylabs.SplitTunnel.DNSExtension` quedan como `[activated disabled]`. En ese estado no cargan código ni interceptan nada, pero siguen listadas.
- **No se pueden des-registrar por CLI con SIP activado**: `sudo systemextensionsctl uninstall <teamID> <bundleID>` responde «cannot be used if System Integrity Protection is enabled». La vía soportada es reinstalar la app y usar SU desinstalación (deactivation cooperativa), o vivir con el registro fantasma, que es inocuo.
- Restos de usuario sí se limpian a mano: `~/Library/Group Containers/group.ca.boundarylabs.*`, `~/Library/HTTPStorages/ca.boundarylabs.*` y `~/Library/Application Scripts/group.ca.boundarylabs.*`.