

# Chuleta LAB ASO

---

created: 2024-12-07T15:09

updated: 2024-12-15T13:46

---

## Comandos utilizados en las prácticas de laboratorio

---

- `sudo su`: *Entra en modo superusuario*
- `lspci`: *Lis todos los dispositivos PCI conectados al sistema.*
  - `-v / -vv`: *Muestra información detallada de cada dispositivo (vv más detallada)*
  - `-k`: *Indica que controladores de kernel están manejando cada dispositivo.*
  - `-nn` *Muestra IDs + Nombres de cada dispositivo*
  - `-s <bus:slot.func>`: *Filtrá y muestra información solo para el dispositivo especificado por el bus, slot y función* ->  
`lspci -s 00:1f.2`.
- `lsusb`: *Lista todos los dispositivos conectados al bus USB*
  - `-v / -vv`: *Muestra información detallada de cada dispositivo (vv más detallada)*
  - `-t`: *Presenta los dispositivos en una estructura de árbol, mostrando la jerarquía de puertos.*
  - `-s <bus:slot.func>`: *Filtrá y muestra información solo para el dispositivo especificado por el bus, slot y función* ->  
`lsusb -s 00:1f.2`.
  - `-D <device>`: *Muestra información detallada para el dispositivo USB especificado.*
- `dmesg`: *Muestra los mensajes del búfer de anillos del kernel.*
- `mount`: *Se utiliza para montar sistemas de archivos.* -> `mount [opciones] dispositivo punto_de_montaje`
  - `-t <tipo>`: *Especifica el tipo de sistema de archivos (ext4)*
  - `-o <opciones>`:
    - `-ro`: *Montar como read only*
    - `-rw`: *Montar con permisos de escritura y lectura*
    - `uid`: *Establece el usuario propietario*
  - `-U`: Monta por UUID del dispositivo
  - `-B`: *Hace un "bind mount" que significa montar un directorio en otro lugar manteniendo el mismo contenido*
- `gdisk`: *Herramienta de particionamiento en Linux que se utiliza para trabajar con tablas de partición en formato GPT* -> `sudo gdisk <dispositivo>`\*Al ejecutarse se entra en un modo interactivo: (Se pueden crear hasta 128 particiones sin ninguna restricción)
  - `?`: Muestra una lista de todos los comandos posibles
  - `p`: Muestra la tabla de particiones actual
  - `n`: Crea una nueva partición
  - `d`: Crea una nueva partición
  - `t`: Cambia el tipo de partición
  - `w`: Escribe los cambios en el disco y sale del programa
  - `q`: Sale del programa **sin guardar**
  - `x`: Entra en el modo experto donde se puede
    - Cambiar tamaño encabezado GPT
    - Reparar tablas de partición corruptas
- `partprobe`: *Recarga la tabla de particiones.*
- `mkfs`: *Crea un sistema de archivos (define cómo se organizan y almacenan los datos en una partición).*
  - `mkfs -t <fstype> <dispositivo>`
    - `-t`:
    - `fstype`: ext4 (predeterminado para linux) , btrfs, vfat (para el USB `/dev/usb1` según el enunciado),etc.
- `mkswap`: *Prepara la partición para que el sistema operativo la reconozca como espacio de intercambio.*
- `chroot`: *cambia el directorio raíz aparente para un proceso y sus hijos creando un entorno aislado donde ese proceso solo tiene acceso a su nuevo directorio raíz.*
- `passwd`: *Cambia la contraseña del usuario*
- `tune2fs`: *Permite ajustar y modificar los parámetros de sistemas de archivos ext2, ext3 y ext4* -> `tune2fs [opciones] <dispositivo>`

- `-L`: \*Cambia o establece una etiqueta para identificar el sistema de archivos: `tune2fs -L "Miparticion" /dev/sda1`
- `-i`: Establece el tiempo máximo entre dos verificaciones automáticas por fsck, Formatos: d(días), w(semanas), m (meses).
- `grep`: Busca texto dentro de archivos o salidas de otros comandos.
  - `-r`: Busca recursivamente en subdirectorios.
  - `-i`: Ignora mayúsculas y minúsculas.
  - `-n`: Muestra el número de línea donde aparece el patrón.
  - `-v`: Muestra las líneas que no coinciden
  - `--color`: Resalta las coincidencias encontradas.
- `auto <ethernet IF>`: Indica que la interfaz se configure automáticamente al iniciar el sistema -> `auto eth0`\*
- `iface <ethernet IF> inet static`: Define la configuración de la interfaz como estática (IP fija) -> `iface eth0 inet static`
- `address <dirección IP>`: Especifica la dirección IP asignada a la interfaz -> `address 10.10.41.50`
- `network 10.10.41.0`: Define la dirección de red asociada a la interfaz -> `network 10.10.41.0`
- `netmask <mask>`: Establece la máscara de subred para la interfaz -> `netmask 255.255.255.0`
- `gateway <direccion IP>`: Define la puerta de enlace predeterminada para la interfaz -> `gateway 10.10.41.1`.
- `apt update`: Descarga la info más reciente de los paquetes disponibles desde los repositorios.
- `apt install <paquete>`: instala uno o más archivos del repositorio
- `sftp username@hostname`: Te conectas a un servidor sftp
  - `mget file_pattern`: Dentro de sftp permite descargarte más de un archivo a la vez.
- `ls`: Lista los directorios hijos de un directorio padre.
  - `-la` Lista también los archivos ocultos.
- `tar`: Herramienta para agrupar múltiples archivos en un único archivo (no comprimido por defecto). Combinado con opciones como `-z` o `-j`, soporta compresión gzip o bzip2 respectivamente.
- `ln`: Comando para crear enlaces entre archivos. Los enlaces simbólicos (`-s`) apuntan a la ubicación del archivo original, mientras que los hard links son duplicados referenciales en el sistema de archivos.
- `df -h I grep /carpeta`: Comprueba que la partición está montada en /carpeta.
- `echo`: Imprime por la salida un mensaje.
- `touch`: Modifica el acceso del archivo y modifica sus tiempos de última vez editado. Si no existe el archivo, lo crea.
- `mv`: Mueve un archivo de un directorio a otro/Lo renombra `mv /root /root.old`
- `passwd`: Cambia la contraseña de un usuario.
- `swapoff`: Desactiva una partición de intercambio.
- `shutdown 0`: Apaga el sistema inmediatamente.
- `tune2fs`: Ajusta parámetros del FS ext2/ext3/ext4.
  - `-i <intervalo>`: Cambia el intervalo de chequeo del filesystem (ej: `tune2fs -i 28 /dev/sda2`).
- `grep`: Busca texto dentro de archivos. (Ej: `grep -r "Debian GNU/Linux 11" /etc`)
- `ip link, ip addr, ip route`: Comandos para configurar y mostrar información de interfaz de red, direcciones IP y rutas. (ej: `ip addr add`, `ip route add default via ...`, `ip link set dev eth0 up`).
- `ifup / ifdown`: Activa o desactiva interfaces de red definidas en `/etc/network/interfaces`.
- `sftp usuario@servidor`: Conexión a servidor SFTP.
- `ls`: Lista archivos/directorios en el servidor remoto.
- `get <archivo>`: Descarga un archivo.
- `mget <patrón>`: Descarga múltiples archivos que coincidan con un patrón.
- `dpkg`: Gestor de paquetes base de Debian.
  - `--install paquete.deb`: Instala un paquete .deb.
  - `-r`: Desinstala dejando configuración.
  - `-P (purge)`: Desinstala borrando configuración.
- `apt update`: Actualiza la información de paquetes disponibles.
- `apt install <paquete>`: Instala paquetes desde el repositorio.
- `apt upgrade`: Actualiza los paquetes instalados a sus últimas versiones.
- `apt info <paquete>`: Muestra info de un paquete.
- `apt search <patrón>`: Busca paquetes.
- `apt clean`: Limpia la caché de paquetes descargados.
- `dpkg-reconfigure`: Reconfigura un paquete ya instalado.

- `make`: *Compila el código fuente según las directrices del Makefile.*
- `make install`: *Instala los binarios compilados en las ubicaciones finales del sistema.*
- `make clean`: *Elimina archivos temporales de compilación.*
- `make uninstall`: *Desinstala lo instalado previamente (si el Makefile lo soporta).*

## Guia Vi

---

### Modos VI

#### Command mode:

Modo utilizado para ejecutar comandos como guardar archivos, ejecutar comandos, mover el cursor, cortar (**yank**), pegar líneas o palabras, buscar/reemplazar. Todo lo escrito se interpreta como un comando.

#### Insert Mode:

Modo que permite insertar texto en el archivo, para entrar **pulsar i** desde el modo de comando.

 **PULSAR "ESC" DOS VECES PARA IR AL MODO COMANDO**

### Comandos VI

- **SALIR DE VI:** `:q` *Sale de VI siempre y cuando no haya cambios realizados, en caso de haber hecho algun cambio te dará una advertencia*
- **SALIR DE VI SIN GUARDAR:** `:q!` *Sale de VI sin guardar aunque hayas hecho algun cambio*
- **SALIR Y GUARDAR:** `ZZ` / `!wq` *Sale y guarda el archivo*
- **GUARDAR:** `:w` *Guarda el archivo*
- **COPIAR:** `yy` *Copia la linea actual*
- **PEGAR:**
  - `p` *Pega la linea copiada a partir de la posición del cursor.*
  - `P` *Pega la linea copiada antes del cursor.*
- **BORRAR:**
  - `x` *Borra el caracter donde apunta el cursor (supr).*
  - `X` *Borra el caracter anterior al cursor (backspace).*
  - `dw` *Borra desde el cursor hasta el final de la palabra.*
  - `d^` *Borra desde el principio de la linea hasta la posición del cursor.*
  - `d$` *Borra desde la posición del cursor hasta el final de la linea.*
  - `dd` *Borra toda la línea en la que se encuentra el cursor.*
- **BÚSQUEDA DE PALABRAS Y CARÁCTERES**
  - `/` *Busca hacia abajo del archivo*
  - `?` *Busca hacia arriba en el archivo*

## Guía Systemd and systemctl

---

### El proceso init y la evolución hacia systemd

El proceso **init** es el primer proceso que se inicia en sistemas Unix (PID=1) y permanece activo hasta el apagado del sistema. Su rol es arrancar los demonios y servicios de usuario, montar sistemas de archivos, gestionar hardware, etc. Históricamente, existían dos grandes enfoques para el init:

- **BSD-style**: basado en scripts en `/etc/rc/`
- **System V (SysVinit)**: basado en "runlevels"

Ambos tenían como inconveniente el arranque secuencial y, por ende, mayor lentitud.

Nuevas alternativas surgieron para reemplazar estos sistemas, siendo **systemd** la más adoptada en diversas distribuciones Linux (pese a ciertas críticas). Esta herramienta proporciona un arranque paralelo y una gestión centralizada de servicios y unidades, mejorando rendimiento y flexibilidad.

### systemctl y su interacción con systemd

`systemctl` es el comando principal para interactuar con `systemd`. Permite listar, iniciar, detener, reiniciar y configurar servicios, además de gestionar sockets, puntos de montaje y más. La autocompletación (TAB) ayuda a descubrir sus opciones.

## Conceptos clave de systemd

- **Units (Unidades):** Son recursos manejados por systemd: servicios (ej. `ssh.service`), sockets, puntos de montaje (`.mount`), temporizadores (`.timer`), etc. Existen 11 tipos de unidades.
- **Unit files:** Cada unidad se describe mediante un archivo de configuración de texto plano, cuyo nombre refleja el tipo de la unidad (ej. `nginx.service`, `boot.mount`). Se organizan en secciones (`[Unit]`, `[Install]`, `[Service]`, ...), con directivas clave-valor.
- **Estado de las unidades:** Al arrancar, las unidades se cargan en memoria y muchas quedan activas, esperando eventos (ej. conexión de un dispositivo, apertura de un socket) que disparen acciones del demonio correspondiente.
- **Dependencias:** Las unidades pueden requerir otras unidades, especificándose con directivas como `Wants`, `Requires`, `After`, `Before` en la sección `[Unit]` del archivo de la unidad.

## Operaciones básicas con systemctl

- **Listar unidades:**
  - `systemctl list-units`: lista unidades cargadas y activas.
  - `systemctl list-units --all`: incluye también las inactivas.
  - `systemctl list-units --type <tipo>`: filtra por tipo (ej. `service`).
  - `systemctl list-units --failed`: muestra unidades en estado fallido.
- **Obtener información de una unidad:**
  - `systemctl status <unidad>`: estado y últimos mensajes del log.
  - `systemctl cat <unidad>`: muestra el archivo de unidad.
  - `systemctl list-dependencies <unidad>`: árbol de dependencias.
  - `systemctl show <unidad>`: propiedades detalladas.
- **Controlar el estado de las unidades:**
  - `systemctl stop <unidad>`: desactiva inmediatamente la unidad.
  - `systemctl start <unidad>`: activa inmediatamente la unidad.
  - `systemctl restart <unidad>`: reinicia la unidad.
  - `systemctl enable <unidad>`: la unidad se activará en el próximo arranque.
  - `systemctl disable <unidad>`: la unidad no se activará en el siguiente arranque.
- **Editar configuración:**
  - `systemctl edit <unidad>`: abre el archivo de la unidad en un editor (determinado por `SYSTEMD_EDITOR`, `EDITOR` o `VISUAL`, o por defecto nano/vim/vi).

## Ecosistema systemd

- **Journald y journalctl:**
  - `systemd-journald.service` es el daemon del journal.
  - `journalctl` permite consultar el log. Ejemplos:
    - `journalctl --since "1 hr ago"`: últimos mensajes de la última hora.
    - `journalctl -n N`: últimas N entradas.
    - `journalctl --disk-usage`: uso en disco del journal.
    - `journalctl -u <unidad>`: filtra por una unidad específica.
- **Otros servicios:**
  - `systemd-logind.service`: gestiona sesiones de login.
  - `systemd-networkd.service`: gestiona la configuración de red.
  - `systemd-halt.service`, `systemd-poweroff.service`, `systemd-reboot.service`, etc.: gestionan apagado y reinicio del sistema.

## Operaciones avanzadas

- **systemd --user:** Permite a usuarios crear su propia instancia systemd para gestionar servicios personales.
- **Unidades enmascaradas (masked):** Una unidad enmascarada no puede ser iniciada ni habilitada hasta ser desenmascarada.
- **Conexión remota:** `systemctl --host=nombre_host` permite interactuar con systemd de otra máquina.

- **Estados detallados:** `systemctl --state=help` enumera todos los estados y sub-estados posibles.

## Command Reference

# Unix/Linux Command Reference

**FOSSwire**.com

File Commands	System Info
<code>ls</code> - directory listing	<code>date</code> - show the current date and time
<code>ls -al</code> - formatted listing with hidden files	<code>cal</code> - show this month's calendar
<code>cd dir</code> - change directory to <i>dir</i>	<code>uptime</code> - show current uptime
<code>cd ..</code> - change to home	<code>w</code> - display who is online
<code>pwd</code> - show current directory	<code>whoami</code> - who you are logged in as
<code>mkdir dir</code> - create a directory <i>dir</i>	<code>finger user</code> - display information about <i>user</i>
<code>rm file</code> - delete <i>file</i>	<code>uname -a</code> - show kernel information
<code>rm -r dir</code> - delete directory <i>dir</i>	<code>cat /proc/cpuinfo</code> - cpu information
<code>rm -f file</code> - force remove <i>file</i>	<code>cat /proc/meminfo</code> - memory information
<code>rm -rf dir</code> - force remove directory <i>dir</i> *	<code>man command</code> - show the manual for <i>command</i>
<code>cp file1 file2</code> - copy <i>file1</i> to <i>file2</i>	<code>df</code> - show disk usage
<code>cp -r dir1 dir2</code> - copy <i>dir1</i> to <i>dir2</i> ; create <i>dir2</i> if it doesn't exist	<code>du</code> - show directory space usage
<code>mv file1 file2</code> - rename or move <i>file1</i> to <i>file2</i> if <i>file2</i> is an existing directory, moves <i>file1</i> into directory <i>file2</i>	<code>free</code> - show memory and swap usage
<code>ln -s file link</code> - create symbolic link <i>link</i> to <i>file</i>	<code>whereis app</code> - show possible locations of <i>app</i>
<code>touch file</code> - create or update <i>file</i>	<code>which app</code> - show which <i>app</i> will be run by default
<code>cat &gt; file</code> - places standard input into <i>file</i>	
<code>more file</code> - output the contents of <i>file</i>	
<code>head file</code> - output the first 10 lines of <i>file</i>	
<code>tail file</code> - output the last 10 lines of <i>file</i>	
<code>tail -f file</code> - output the contents of <i>file</i> as it grows, starting with the last 10 lines	
Process Management	Compression
<code>ps</code> - display your currently active processes	<code>tar cf file.tar files</code> - create a tar named <i>file.tar</i> containing <i>files</i>
<code>top</code> - display all running processes	<code>tar xf file.tar</code> - extract the files from <i>file.tar</i>
<code>kill pid</code> - kill process id <i>pid</i>	<code>tar czf file.tar.gz files</code> - create a tar with Gzip compression
<code>killall proc</code> - kill all processes named <i>proc</i> *	<code>tar xzf file.tar.gz</code> - extract a tar using Gzip
<code>bg</code> - lists stopped or background jobs; resume a stopped job in the background	<code>tar cjf file.tar.bz2</code> - create a tar with Bzip2 compression
<code>fg</code> - brings the most recent job to foreground	<code>tar xjf file.tar.bz2</code> - extract a tar using Bzip2
<code>fg n</code> - brings job <i>n</i> to the foreground	<code>gzip file</code> - compresses <i>file</i> and renames it to <i>file.gz</i>
	<code>gzip -d file.gz</code> - decompresses <i>file.gz</i> back to <i>file</i>
File Permissions	Network
<code>chmod octal file</code> - change the permissions of <i>file</i> to <i>octal</i> , which can be found separately for user, group, and world by adding:	<code>ping host</code> - ping <i>host</i> and output results
<ul style="list-style-type: none"> <li>• 4 - read (r)</li> <li>• 2 - write (w)</li> <li>• 1 - execute (x)</li> </ul>	<code>whois domain</code> - get whois information for <i>domain</i>
Examples: <code>chmod 777</code> - read, write, execute for all <code>chmod 755</code> - rwx for owner, rx for group and world For more options, see <code>man chmod</code> .	<code>dig domain</code> - get DNS information for <i>domain</i>
	<code>dig -x host</code> - reverse lookup <i>host</i>
	<code>wget file</code> - download <i>file</i>
	<code>wget -c file</code> - continue a stopped download
SSH	Installation
<code>ssh user@host</code> - connect to <i>host</i> as <i>user</i>	Install from source: <code>./configure</code>
<code>ssh -p port user@host</code> - connect to <i>host</i> on port <i>port</i> as <i>user</i>	<code>make</code>
<code>ssh-copy-id user@host</code> - add your key to <i>host</i> for <i>user</i> to enable a keyed or passwordless login	<code>make install</code>
	<code>dpkg -i pkg.deb</code> - install a package (Debian)
	<code>rpm -Uvh pkg.rpm</code> - install a package (RPM)
Searching	Shortcuts
<code>grep pattern files</code> - search for <i>pattern</i> in <i>files</i>	<code>Ctrl+C</code> - halts the current command
<code>grep -r pattern dir</code> - search recursively for <i>pattern</i> in <i>dir</i>	<code>Ctrl+Z</code> - stops the current command, resume with <code>fg</code> in the foreground or <code>bg</code> in the background
<code>command   grep pattern</code> - search for <i>pattern</i> in the output of <i>command</i>	<code>Ctrl+D</code> - log out of current session, similar to <code>exit</code>
<code>locate file</code> - find all instances of <i>file</i>	<code>Ctrl+W</code> - erases one word in the current line
	<code>Ctrl+U</code> - erases the whole line
	<code>Ctrl+R</code> - type to bring up a recent command
	<code>!!</code> - repeats the last command
	<code>exit</code> - log out of current session



\* use with extreme caution.

## Práctica 1 Installation of the OS

### Descripción de la práctica

- Identificamos los componentes hardware del sistema:

- `lspci`: Lista todos los dispositivos que están conectados al bus PCI.
- `lsusb`: Lista todos los dispositivos conectados al bus USB.

Part	Hardware model	Device Name
Network Card	Intel corporation ethernet connection (14) I219-V	eno1
Internal Hard Drive	Intel corporation device 43d2	NVME On1
USB Hard Drive	Toshiba America Inc External 3.0	sda

2. Terminamos de llenar la tabla de arriba usando el comando `dmesg`, recordando que \*\*los dispositivos normalmente están almacenados en el directorio /dev .

3. Si tienes un sistema con dos discos NVMe y un disco SATA:

#### 3.1 Primer NVMe:

- Controlador: /dev/nvme0
- Namespace: /dev/nvme0n1
- Primera partición: /dev/nvme0n1p1

#### 3.2. Segundo NVMe:

- Controlador: /dev/nvme1
- Namespace: /dev/nvme1n1
- Primera partición: /dev/nvme1n1p1

#### 3.3 SATA:

- Primer disco SATA: /dev/sda

4. El primer paso para instalar el sistema es particionar el disco en el dispositivo USB. (/dev/sda) y para ello desmontamo previamente todos los dispositivos usb

- `umount /dev/usb`

5. Nos movemos al /dev y ejecutamos:

- `sudo gdisk /dev/sda`.

6. Creamos las siguientes particiones usando los comandos de gdisk correspondientes.

Device	Code	Size	Mount-point	Comments
/dev/usb1	EFI System Partition (ESP) (EF00)	512MB	/boot/efi	This partition will hold the necessary EFI Boot information. It MUST be formatted with FAT32 (vfat) filesystem
/dev/usb2	Linux (8304)	30GB+	/	This partition will hold the main system, a typical Debian installation requires around 5GB, however, when we add more software it may grow considerably. Use at least 30Gb
/dev/usb3	Linux (8300)	5GB	/usr/local	We don't use too much this partition during the course, with 5GB it should be enough, in a real system it depends on the actual requirements regarding self-compiled applications
/dev/usb4	Linux (8302)	100GB+	/home	This one in general is where most space should be devoted. You can put 100GB or more
/dev/usb5	Swap (8200)	2xRAM		Put twice the size of the machine's RAM <sup>1</sup>
/dev/usb6	Linux (8300)	20GB		Reserved for future use, you don't need to create it now

7. Una vez hemos creado las particiones necesarias, empezamos inicializando el área swap (doble de la RAM) (tampoco es obligatorio)

- `mkswap device`

8. Seguidamente creamos el sistema de archivos (define cómo se organizan y almacenan los datos en una partición, sin éste no podría ni guardar ni leer datos en una partición).

```
# Formatear las particiones
mkfs.vfat /dev/usb1          # Formatear /boot/efi como vfat
```

```

mkfs.ext4 /dev/usb2          # Formatear / como ext4
mkfs.ext4 /dev/usb3          # Formatear /usr/local como ext4
mkfs.ext4 /dev/usb4          # Formatear /home como ext4
mkswap /dev/usb5             # Configurar el swap
swapon /dev/usb5             # Activar el swap
mkfs.ext4 /dev/usb6          # Formatear la partición reservada como ext4

# Crear directorio principal para montaje
mkdir /linux

# Montar las particiones
mount /dev/usb2 /linux        # Montar /
mkdir -p /linux/boot/efi      # Montar /boot/efi
mount /dev/usb1 /linux/boot/efi # Montar /boot/efi
mkdir -p /linux/usr/local     # Montar /usr/local
mount /dev/usb3 /linux/usr/local # Montar /usr/local
mkdir -p /linux/home           # Montar /home
mount /dev/usb4 /linux/home

```

Es importante crear los directorios después de montar el /linux pero antes de montar el resto.

9. Instalaremos el sistema base que se encuentra en un servidor SFTP de la siguiente forma:

```

bash cd /linux sftp aso@asoserver.pc.ac.upc.edu #Ponemos la password As0RoCkSHaRd! get aso-install.tar.gz
exit tar xzf aso-install.tar.gz rm aso-install.tar.gz

```

10. Terminamos de montar los sistemas auxiliares

```

for i in /dev /dev/pts /proc /sys /run; do
    mount -B $i /linux/$i
done

```

Este script toma algunos directorios clave del sistema (que son necesarios para que el sistema operativo funcione correctamente, como /dev y /proc) y los “clona” en otro directorio (en este caso /linux). Esto se hace para que, cuando estés dentro del entorno de /linux, sigas teniendo acceso a estos sistemas esenciales.

11. Ahora pasamos a configurar la tabla sistemas de archivos (/etc/fstab)

```

vim /linux/etc/fstab
device(/dev/usb5 en caso del ejemplo) none swap defaults 0 0 #Añadir la partición swap
device / ext4 defaults 0 1 # Añadir la partición de root
device /boot/efi cf

```

Primer número: **DUMP** (0/1) si debe realizar una copia de seguridad con dump. Casi siempre 0.

Segundo número: (fsck): 1 se verifica al inicio (Normalmente root /), 2 Verificación en otras particiones (después de la raíz).

/proc y /sys no tienen dispositivos adjuntos porque son sistemas de archivos virtuales diseñados para dar información sobre el sistema y el kernel en lugar de interactuar directamente con hardware físico tal y como lo harían los archivos de dispositivos en /dev.

12. Cambiamos el directorio root del sistema y usamos temporalmente el software instalado en el sistema en vez del disponible en el sistema:

```
chroot /linux
```

El comando chroot cambia el directorio raíz aparente para un proceso y sus hijos creando un entorno aislado donde ese proceso solo tiene acceso a su nuevo directorio raíz.

13. Configuramos el teclado:

```

dpkg-reconfigure locales
dpkg-reconfigure console-data
dpkg-reconfigure keyboard-configuration

```

14. Para evitar tener que modificar la tabla de particiones cada vez que encendemos un ordenador, usaremos un **bootloader**, un set de programas residentes en el disco duro que permiten al usuario cargar otros sistemas operativos, instalaremos GRUB:

```
grub-install --target=x86_64-efi /dev/usb
```

Asumiendo que la ya hemos realizado estos comandos previamente:

```
mount /dev/usb2 /linux  
mount /dev/usb1 /linux/boot/efi
```

Para no tener que editar el archivo `/boot/grub/grub.cfg` ejecutamos `update-grub` que crea automáticamente este archivo.

15. Ahora cambiaremos las contraseñas que como ya sabemos se encuentran en el archivo `/etc/passwd` con el comando:  
`passwd`

16. Desmontamos todo lo que hemos montado antes, en caso de no saber que particiones están montadas actualmente se puede utilizar el comando:

```
mount | grep linux
```

**IMPORTANTE:** Desmontar en orden inverso al que se ha montado:

```
umount /linux/boot/efi  
umount /linux/usr/local  
umount /linux/home  
  
umount /linux  
  
# Y si quieres desactivar swap:  
swapoff /dev/usb5 #Suponiendo que esta en usb5  
sudo shutdown 0
```

17. Ahora nos encargaremos de la **Post-Configuración**. Para ello empezamos dandole a la tecla F12 para ir al menú de BOOT de la BIOS, seleccionamos *Toshiba harddrive from the UEFI boot partition* y no la Legacy, ya que no hemos configurado con mecanismo MBR y si todo ha funcionado correctamente deberíamos tener dos cuentas de usuario, root y aso. Para cambiar a superusuario como aún no tenemos `sudo` instalado podemos hacerlos ejecutando `su`.

18. Cambiamos la frecuencia de checks del filesystem en el usb2 a 28 días.

```
tune2fs -i 28 /dev/sda2
```

Con `tune2fs` también podemos cambiar el porcentaje de bloques reservados (modificar el espacio reservado para el superusuario (-m)).

19. Hay varios mensajes que aparecen durante el proceso de login del sistema, queremos cambiar alguno de ellos. Todos están localizados en `/etc`. Queremos cambiar el mensaje que dice algo parecido a "Debian GNU/Linux 11". Para encontrarlo usamos el comando:

```
'grep -r "Debian GNU/Linux 11" /etc'
```

Y encontramos que el archivo es `/etc/issue`.

20. Modificamos el MOTD (Message of the day) modificando el `/etc/motd`

21. Pasamos a configurar la red, se puede hacer via **manual** o **automática**:

**Manual:**

Hacemos Flush de la conexión actual asegurarnos que está activo.

```
ip link show #  
ip link set dev <ethernet IF> down  
ip link set dev <ethernet IF> up
```

The commands for configuring the IP\_address and his default route is:

```
sudo ip addr add <IP_address> / <netmask> dev <interface>\nip link set dev eno1 up\nip route add default via 10.10.41.1
```

Y justo después creamos el archivo /etc/resolv.conf con la información DNS.

De esta forma tendríamos que configurarlo cada vez que encendemos el ordenador. Para evitar esto lo configuraremos para que se configure automáticamente al momento de iniciar.

### Automática:

```
vim /etc/network/interfaces\nauto <ethernet IF>\niface <ethernet IF> inet static\naddress 10.10.41.??? # (ip que te da el profesor)\nnetwork 10.10.41.0\nnetmask 255.255.255.0\ngateway 10.10.41.1\n\nifup <ethernet IF>\nifdown <ethernet IF> # Para comprobar si se han configurado correctamente
```

### Con DHCP:

```
vim /etc/network/interfaces\nauto <ethernet IF>\niface <ethernet IF> inet dhcp
```

## Práctica 2 Application Management

### Resumen rápido preguntas del principio:

- Para conectarse a un sever sftp usamos: `sftp username@hostname`
- Dentro de **sftp**:
  - ls: Lista directorios
  - get: Obtiene un archivo de un directorio.
  - mget file\_pattern: Obtiene muchos archivos de un directorio.
- Para comprimir: `tar -cvzf archivocomprimido.tar.gz archivo 1 archivo 2 archivo 3`
  - -c: Crea un archivo tar.
  - -v: Modo verbose (muestra los archivos añadidos al tar).
  - -z: Comprime con gzip.
  - -f: Especifica el nombre del archivo tar.
- Y para descomprimir: `tar -xvzf`, donde la x indica que extrae los archivos.
- Para crear hard y softlinks funciona de la siguiente manera:
  - **hardlink**: `ln existing_file hardlink_name` -> `ln file.txt hardlink.txt`
  - **softlink**: `ln -s target_file link_name`.
- El PATH envioromnt variable significa

### Introducción

### Resumen

La instalación de software en un sistema operativo consiste en copiar los archivos de la aplicación en las ubicaciones adecuadas y, si hace falta, ajustar parámetros de configuración. En sistemas modernos, gestionar las dependencias es fundamental, ya que antes de instalar un nuevo software hay que asegurarse de tener instaladas las librerías o paquetes de los que depende.

### Sistemas de gestión de software

En distribuciones como Debian (la base de ASO Linux), el software se organiza en paquetes `.deb`. Estos paquetes incluyen binarios, librerías, archivos de configuración, manuales y documentación, así como información sobre sus dependencias. De esta

forma, es fácil instalar, actualizar y desinstalar programas sin tener que resolver manualmente las dependencias.

## Entorno gráfico en UNIX: X-Window

El sistema X-Window (X11, X) es un protocolo que ofrece las bases para mostrar interfaces gráficas en sistemas tipo UNIX. Funciona mediante un modelo cliente/servidor: el **X server** se encarga de la salida gráfica y de recibir la entrada del usuario, mientras que las aplicaciones (clientes) solicitan la creación de ventanas y reaccionan a eventos. X por sí solo no define el aspecto de las ventanas, menús o botones; esto queda en manos de:

- **Window Managers (Gestores de ventanas)**: Controlan la posición, aspecto y gestión de las ventanas. Ejemplos: `kwin`, `gnome-shell`, uno muy ligero: `pekwm`
- **Display Managers (Gestores de sesión)**: Muestran una pantalla de login gráfica y controlan el inicio de la sesión. Ejemplos: `XDM`, `GDM`, `SDDM`, `lightDM`.
- **Desktop Environments (Entornos de escritorio)**: Ofrecen una interfaz unificada con íconos, menús, barras de tareas y aplicaciones integradas (p.ej. GNOME, KDE). Estos entornos se apoyan en un gestor de ventanas y un display manager, y emplean una biblioteca gráfica (GTK+, QT, etc.).

\*Si usas `pekwm + lightDM` no necesitas desktop environment, se hace todo desde la terminal

La tabla siguiente muestra algunos ejemplos:

Entorno de Escritorio	Window Manager	Display Manager	Biblioteca Gráfica
GNOME	gnome-shell	GDM	GTK+
KDE	Kwin	SDDM	QT
Xfce	Xfwm4	LightDM	GTK+
LXDE	Openbox	LXDM	QT

## Instalación de paquetes binarios (Debian)

Para instalar paquetes `.deb` manualmente, se utiliza el comando `dpkg`. Este comando permite:

- Instalar paquetes: `dpkg --install paquete.deb`
- Desinstalar (sin borrar archivos de configuración)
- Purgar (desinstalar borrando además los archivos de configuración)
- Listar paquetes instalados
- Ver archivos de un paquete

## Diferencia entre desinstalar y purgar un paquete

- **Desinstalar (remove)**: Elimina los archivos binarios del paquete, pero deja intactos los archivos de configuración.
- **Purgar (purge)**: Elimina tanto los archivos binarios como los archivos de configuración, dejando el sistema como si el paquete nunca se hubiese instalado.

## Instalación con dependencias

Si un paquete tiene dependencias, es necesario instalarlas antes o simultáneamente. Por ejemplo, si al intentar instalar `lynx` falla por no tener `lynx-common`, primero habrá que instalar `lynx-common`.

### ¿Qué comando usaste para instalar lynx sin el -common?

Se intentó con:

```
dpkg --install lynx.deb
```

Esto falló debido a la falta de `lynx-common`. Tras instalar:

```
dpkg --install lynx-common.deb
```

Se pudo completar la instalación de `lynx`.

La instalación y gestión de paquetes en Debian puede realizarse con distintas herramientas, pasando desde el nivel básico (`dpkg`) hasta el avanzado (APT), que resuelve automáticamente las dependencias y facilita la instalación de entornos gráficos completos, compiladores y software externo.

## APT y repositorios

Primero debes configurar el repositorio en /etc/apt/sources.list, por ejemplo:

```
deb http://ftp.es.debian.org/debian/ stable main non-free contrib_
```

Luego, actualiza el índice de paquetes:

```
apt update
```

Para actualizar todos los paquetes instalados a su última versión:

```
apt upgrade
```

Para obtener información de un paquete:

```
apt info nombre-paquete
```

## Instalación del entorno gráfico X-Window

Instalar el servidor X:

```
apt install x-window-system
```

## Entornos de escritorio

Debian proporciona metapaquetes (task-< entorno >-desktop) que instalan entornos completos:

Listar todos los paquetes:

```
apt list
```

Filtrar metapaquetes task----desktop:

```
apt list | grep '^task-.*-desktop'
```

O usar la búsqueda:

```
apt search desktop
```

Elegir e instalar un entorno, por ejemplo GNOME:

```
apt install task-gnome-desktop
```

Si algo falla, reconfigura el paquete afectado:

```
dpkg-reconfigure nombre-paquete
```

## Preparar el sistema para compilar

Instalar compilador, librerías y Firefox:

```
apt install gcc libc6-dev firefox
```

Limpiar la caché de paquetes:

```
apt clean
```

apt clean elimina todos los paquetes descargados de la caché, mientras que apt autoclean solo elimina los obsoletos.

## Instalar binarios fuera de repositorios

Descomprimir JDK en una carpeta de tu elección:

```
tar -xvzf jdk-11.0.14_linux-x64_bin.tar.gz
```

Mover la carpeta resultante:

```
mv jdk-11.0.14 /opt/java1.11
```

Verificar la versión:

```
/opt/java1.11/bin/java -version
```

Hacer lo mismo para otra versión (ej. Java 1.17) en /opt/java1.17.

El comando java -version del sistema no mostrará estas nuevas versiones porque no están en el PATH por defecto. Para solucionarlo, crea enlaces simbólicos:

```
ln -s /opt/java1.11/bin/java /usr/local/bin/java
```

Así java -version usará la versión 1.11. Para acceder a distintas versiones sin cambiar el PATH por defecto, crea enlaces simbólicos con nombres distintos:

```
ln -s /opt/java1.17/bin/java /usr/local/bin/java1.17  
ln -s /opt/java1.11/bin/java /usr/local/bin/java1.11
```

De esta forma puedes ejecutar java1.17 -version o java1.11 -version según necesites.##### instalación desde código fuente\*

En algunos casos es necesario instalar software directamente desde el código fuente, ya sea porque no hay un paquete disponible o para adaptar mejor la aplicación a nuestro sistema. El proceso típico incluye descargar el código fuente, configurarlo, compilarlo, instalarlo, y finalmente limpiar los archivos temporales.

### Pasos generales:

1. Descargar el archivo fuente (por ejemplo asosh-0.1.tar.gz) desde el servidor especificado.
2. Descomprimir el código fuente en el directorio designado, normalmente /usr/src.

```
tar -xvzf asosh-0.1.tar.gz -C /usr/src
```

¿Qué comando usaste para descomprimir?

```
sftp aso@asoserver.pc.ac.upc.edu + AsORoCkSHaRd!  
cd sources + get asosh-0.1.tar.gz +exit  
tar xvzf asosh-0.1.tar.gz
```

Por ejemplo, el anterior si estabas en el directorio donde se descargó el tar.

3. Examinar el contenido del directorio del código fuente. Allí suele haber un script configure que permite ajustar la instalación (por defecto, asosh se instala en /usr/local).

Para que se instale en /usr/local/asosh:

```
./configure --prefix=/usr/local/asosh
```

¿Qué parámetros se usaron?

```
La opción --prefix=/usr/local/asosh.
```

4. Si la configuración da error por falta de librerías de desarrollo (headers), por ejemplo un mensaje indicando que no encuentra cierto header (.h):

### *¿Cuál es el error reportado?*

Alguna librería requerida no está presente (por ejemplo falta una librería de desarrollo).

### *¿Razón del error?*

Las cabeceras de desarrollo no están instaladas.

### *¿Cómo lo solucionaste?*

Instalar el paquete de desarrollo correspondiente:

```
apt install libreadline-dev
```

5. Una vez solucionadas las dependencias, ejecutar de nuevo:

```
./configure --prefix=/usr/local/asosh
```

Si termina sin errores, procedemos a compilar:

```
make
```

Este paso no requiere permisos de root.

6. Instalar el software ya compilado (coloca los binarios y demás archivos en su ubicación final):

```
sudo make install
```

7. Una vez instalado, es buena idea eliminar los archivos temporales generados durante la compilación:

### *¿Qué comando se usa para borrar archivos temporales?*

```
make clean
```

Además, se pueden revertir los pasos de instalación:

### *¿Qué argumento se puede usar para deshacer la instalación?*

```
make uninstall
```

Este proceso es estándar para la mayoría de software distribuido en código fuente: configurar (./configure), compilar (make), instalar (make install), limpiar (make clean) y, si es necesario, desinstalar (make uninstall).

## Práctica 3: Scripts

Todos los intérpretes de comandos incorporan un lenguaje de programación con sentencias de control de flujo, asignación de variables, funciones, etc. Los usuarios pueden escribir programas (\*\*shells\*\*cripts\*\*) utilizando este lenguaje para automatizar la ejecución de secuencias de comandos. Los shellscripts serán interpretados por el shell.

### Creación de un Shellscrip

#### 1. Crear el archivo:

- Invoca un editor de textos y escribe el código correspondiente.

#### 2. Primera línea del script:

- `#!/bin/bash`

Indica el shell que interpretará el programa.

#### 3. Guardar el archivo:

- Guarda el script con un nombre adecuado, por ejemplo, `mi_script.sh`.

#### 4. \*\*Dar permisos de ejecución:

```
chmod 700 mi_script.sh  
````
```

#### 5. \*\*Ejecutar el script:\*\*

```
```bash
./mi_script.sh
```

## Comentarios en Shellscripts

- Utiliza el carácter `#` para insertar comentarios dentro del shellscript.

## Consulta del Manual

- Para buscar información en el manual sobre las sentencias propias de **\*\*bash\*\*** relacionadas con la programación de shellscripts, ejecuta:

```
```bash
man bash
```

```
### Comandos Comunes en Shellscripts
#### sleep
• **Descripción:**  
    Detiene la ejecución del shellscript durante el número de segundos indicado como parámetro.  
• **Ejemplo:**  
    sleep 5  
    Retorna el control pasados 5 segundos.  
#### echo
• **Descripción:**  
    Muestra un mensaje en la salida estándar. Permite imprimir el valor de las variables.  
• **Ejemplos:**  
    echo Hola  
    Muestra el mensaje "Hola".  
    echo Valor de home: $HOME  
    Muestra el contenido de la variable HOME.  
    echo -n Sin salto de línea  
    No muestra salto de línea (parámetro -n).  
    echo -e "\ta\t"  
    El parámetro -e activa la interpretación de caracteres especiales como \t (tabulador).  
**test**  
  
• **Descripción:**  
    Evalúa condiciones respecto a archivos (existencia, permisos, tipo), cadenas de caracteres (igualdad, etc.) o numéricas (igualdad, desigualdad, mayor que, etc.). No escribe nada en la salida estándar, pero devuelve un código de estado ($?).  
    0 si la condición es verdadera.  
    Diferente de 0 si la condición es falsa.  
• **Ejemplos:**  
`test -d /bin; echo $?`  
Escribe 0 porque /bin es un directorio.  
`test -w /bin; echo `  
Escribe 1 porque no se puede escribir en /bin.  
`test hola = adeu; echo $?`  
Escribe 1 porque las cadenas son diferentes.  
`test 4 -gt 5; echo $?`  
Escribe 1 porque 4 no es mayor que 5.  
`test 3 -ne 6; echo $?`  
Escribe 0 porque 3 no es igual a 6.  
`test 3 -gt 2 -a 5 -lt 7; echo $?`  
Escribe 0 porque se cumplen ambas condiciones (-a indica and).  
  
#### expr
• **Descripción:**  
    Evalúa una expresión aritmética y muestra el resultado en la salida estándar.  
• **Ejemplos:**
```

```
`expr 3 + 4`  
Muestra 7. Es necesario que existan espacios en blanco entre los operandos y el operador.  
'expr 3 \* 4'  
Muestra 12. Es necesario proteger el operador * con el carácter de escape \ para evitar que el shell lo interprete como un comodín.
```

#### read  
• **Descripción:**

Permite leer una línea de la entrada estándar y guardarla en una variable. También puede guardar las palabras que componen la línea en variables diferentes.

- **Ejemplos:**

```
'read lin; echo L: $lin'
```

Carga la línea leída en la variable lin.

```
'read word1 word2; echo L1: $word1; echo L2: $word2'
```

Carga la primera palabra leída en word1 y la segunda (y las restantes) en word2.

#### exit

• **Descripción:**

Finaliza la ejecución del shellscript.

#### true / false

• **Descripción:**

Comandos que se usan para generar condiciones de control en bucles infinitos.

#### Comillas (Metacaracteres) en Shellscripts

El shell dispone de tres tipos de comillas:

- **Comillas simples (\*\*' \*\*)**  
Interpretan literalmente la cadena que están entre las comillas (sin expandir metacaracteres ni interpretar espacios en blanco como separadores).

• Ejemplo:

```
'echo '$PATH *'
```

Muestra la cadena literal \$PATH \*.

• **Comillas dobles (\*\*" \*\*)**

• Permiten interpretar el metacaracter \$ para reemplazar el valor de las variables.

• Ejemplo:

```
'echo "$PATH *'"
```

Muestra el valor de la variable PATH seguido de la lista de archivos del directorio actual.

• **Comillas inversas (\ ` `` `)**

- Ejecutan la comanda que está entre las comillas y usan el resultado como parámetro de otra comanda.
- Permiten interpretar el metacaracter \$.

• Ejemplo:

```
```bash
```

```
ls -l `which sort`
```

Utiliza el resultado de ejecutar which sort como parámetro para ls -l.

• **Escapar metacaracteres (\*\*)**

• Inhibe la expansión de metacaracteres.

• Ejemplo:

```
echo \*
```

Muestra el carácter \* en lugar de expandirlo a la lista de archivos.

## Variables

Repasa lo explicado en sesiones anteriores sobre variables.

### Control de Flujo

**if**

- **Sintaxis:**

```
if condición1
then
    sentencias1
[ elif condición2
then
    sentencias2 ]
...
[ else
    sentencias3 ]
fi
```

- **Descripción:**

Evaluá una condición. Si es verdadera, ejecuta sentencias1; si es falsa y condición2 es verdadera, ejecuta sentencias2; y así sucesivamente. Si todas las condiciones son falsas, ejecuta sentencias3.

- **Formas típicas de generar condiciones:**

- Comando test.
- Comando grep (retorna verdadero si encuentra la palabra).
- Otras comandadas que retornan verdadero si han ejecutado correctamente y falso en caso de error.

### while / until

- **Sintaxis:**

```
while condición
do
    sentencias
done
```

```
until condición
do
    sentencias
done
```

- **Descripción:**

- **while:** Evalúa la condición. Si es verdadera, ejecuta las sentencias dentro del bucle y vuelve a evaluar.
- **until:** Itera mientras la condición se evalúe como falsa.

### for

- **Sintaxis:**

```
for variable in lista_valores
do
    sentencias
done
```

- **Descripción:**

Itera sobre cada elemento de lista\_valores y ejecuta las sentencias internas. En cada iteración, variable toma el valor del elemento correspondiente.

- **Formas de generar lista\_valores:**

- *para iterar sobre los nombres de archivo del directorio actual.*
- *Comando entre comillas invertidas para iterar sobre cada palabra resultante de la ejecución de una comanda.*
- *\$ para iterar sobre los parámetros del shellscript.*

## case

- **Sintaxis:**

```
case palabra in
    patrón1) sentencias1 ;;
    patrón2) sentencias2 ;;
    ...
    patrónN) sentenciasN ;;
esac
```

- **Descripción:**

Busca el primer patrón que corresponda a palabra y ejecuta las sentencias asociadas.

- **Uso de patrones:**

- **a:** Palabras que comienzan con a.
- **a|b:** Palabras que comienzan con a o b.
- **:** Todas las palabras (usualmente como último patrón para manejar casos no coincidentes).

## Comentarios sobre las Sentencias de Control de Flujo

- Dentro de bucles (**for**, **while**, **until**):

- **break:** Sale del bucle.
- **continue:** Salta a la siguiente iteración del bucle.

- **Redirección:**

• Es posible redirigir la entrada estándar y/o la salida estándar de una sentencia de tipo for, while o until. Esta redirección afecta a todas las comandos ejecutados dentro del bucle.

- **Errores Comunes:**

- Olvidar palabras clave como done o no cerrar una cometa puede provocar el error de ejecución Unexpected EOF.

## Paso de Argumentos a los ShellsScripts

Cuando invocamos un shellscrip, podemos pasarle argumentos. Para acceder a ellos:

- **\$#:**

Permite obtener el número de parámetros del shellscrip.

- **\$\*\*:**

Contiene la lista de todos los parámetros del shellscrip\*

### Acceso Individual:

- El primer parámetro: \$1
- El segundo parámetro: \$2
- ...
- El noveno parámetro: \$9

- **Más de 9 Parámetros:**

- Para acceder a parámetros superiores a \$9, utiliza la comanda shift.

```
shift
```

- **Descripción:**

Descarta el valor de \$1, mueve el valor de \$2 a \$1, el de \$3 a \$2, ..., y coloca el valor del primer parámetro no accesible en \$9. Actualiza el valor de \$# y \$\*.

**Nota:** El efecto de shift no puede ser deshecho.

- **\$0:**

Siempre contiene el nombre del shellscrip.

## Comprobación de Argumentos

Cuando un shellscrip espera argumentos, es recomendable comprobar que el valor de \$# es el esperado.

### Ejemplo:

```

if [ $# -lt 2 ]; then
    echo "Uso: $0 arg1 arg2"
    exit 1
fi

```

Este script verifica que al menos se hayan pasado dos argumentos; de lo contrario, muestra un mensaje de *Usage* y finaliza.

## Inicio práctica 3

### Script para detectar a usuarios inválidos

Tenemos que hacer un script que determine qué usuarios en /etc/passwd son inválidos, es decir si el usuario está en el archivo passwd pero no tiene ningún archivo. Aunque también existen algunos usuarios que no tienen archivos pero sirven para ejecutar los *daemons* (programas que se ejecutan en segundo plano) y debemos añadir una opción que declare que éstos usuarios sí son válidos.

Tenemos que rellenar los huecos del siguiente script:

```

#!/bin/bash
p=0
function print_help
{
    echo "Usage: $1 [options]"
    echo "Possible options:"
    echo "-p validate users with running process"
}
if [ $# -gt 1 ]; then
    print_help $0
    exit
fi

while [ $# -gt 0 ]; do
    case $1 in
        "-p")
            p=1
            shift;;
        *)
            echo "Error: not valid option: $1"
            exit 1;;
    esac
done

for user in _____; do
    home=$(cat /etc/passwd | grep "^$user:" | cut -d: -f6)
    if [ -d $home ]; then
        num_fich=$(find "$home" -type f -user $user | wc -l)
    else
        num_fich=0
    fi

    if [ $num_fich -eq 0 ]; then
        if [ $p -eq 1 ]; then
            user_proc=_____
            if [ $user_proc -eq 0 ]; then
                echo "The user $user has no processes"
            fi
        else
            echo "The user $user has no files in $home"
        fi
    fi
done

```

- Por lo que parece tenemos que rellenar dos huecos, el primero lo rellenamos con:

```
$ cut -d: f1 /etc/passwd
```

- **cut**: como éste comando imprime las partes seleccionadas de cada linea de un ARCHIVO por la salida estándar, nos interesa para imprimir el primer campo (field, f1) del archivo /etc/passwd de cada usuario. La opción -d hace que se vean solo los nombres de usuario, sin esa opción se vería todo el archivo /etc/passwd. Al ser un comando, debe ir con \$ antes y entre paréntesis.
- Para llenar el segundo campo, debemos saber cómo sabemos si un usuario tiene procesos en ejecución y eso lo sabemos gracias al comando `ps -U username`, y para saber si NO tiene ningún proceso en ejecución lo podemos saber gracias a que no devuelva ninguna línea, con lo que podríamos hacer un `wc -l` del comando anterior que cuente las líneas.

```
user_proc=$(ps -U | wc -l)
user_proc=$((user_proc-1))
```

El -1 es para eliminar la línea automática que siempre se cuenta, la de los encabezados. También podría hacerse poniendo `--headers` antes de user\_proc.

El comando `shift` elimina el primer argumento posicional (\$1) de la lista de argumentos haciendo que el siguiente argumento pase a ser \$1. Esto se usa para iterar correctamente sobre las opciones una vez que se ha procesado una.

En el comando `grep "^\$user:"` sobre /etc/passwd, el símbolo ^ indica el inicio de la linea, con lo que `^\$user:` busca una línea que comience exactamente con el nombre del usuario seguido de dos puntos. Esto evita que se obtengan coincidencias parciales, asegurando que solo se localice la línea correspondiente a ese usuario específico. El carácter : es importante porque en el archivo /etc/passwd el formato es `usuario:contraseña:UID:GID:info:home:shell`, así el : tras el nombre de usuario garantiza que sea el campo correcto, no solo una subcadena del nombre.

## Detección de usuarios inactivos

Ahora extenderemos el script anterior para detectar usuarios inactivos, es decir un usuario que no tiene ningún proceso en ejecución, que hace mucho no inicia sesión y que no ha cambiado ninguno de sus archivos desde hace mucho tiempo. El periodo de tiempo se pasará por parámetro:

```
./BadUsers.sh -t 2d (indicates 2 days)
alvarez
aduran
xavim
marcg

./BadUsers.sh -t 4m (indicates 4 months)
xavim
marcg
```

## Otro ejemplo de script

Escribe un script que muestre un listado con los 10 procesos que mas tiempo consumen en CPU. El listado tiene que mostrar 10 líneas, una por proceso, donde cada linea debe mostrar el pid, el usuario propietario, el tiempo acumulado y el nombre del ejecutable. Las líneas deben estar ordenadas de más a menos tiempo consumido.

```
#!/bin/bash
ps -eo pid,user,cputime,comm --sort=-cputime | head -n 11
```

Resumen de como funciona:

- `ps` muestra los procesos actuales
- `-e` muestra todos los procesos
- `-o` hace que salga con un formato específico (pid,user,cputime,comm (comanda)).
- `--sort` es una opción de output, puedes añadir `+` o `-` para ordenar creciente o decrecientemente.
- `head` corta el output a `n` líneas (10 + el header), el propio `ps` no tiene una opción directa para controlar el número de líneas de output

Modifica el script de forma que podamos parametrizar el número de procesos a mostrar. Por defecto el script se comportará como el del apartado anterior, pero si recibe un parámetro este será el número de procesos que hay que listar.

```
DEFAULT_NUM=10
NUM_PROCS=${1:-$DEFAULT_NUM}

ps -eo pid,user,cputime,comm --sort=-cputime | head -n $((NUM_PROCS + 1))
```

## Práctica 4 gestión de usuarios

### Objetivos

- Crear nuevas cuentas de usuario y cambiar sus propiedades.
- Deshabilitar y eliminar cuentas de usuario de forma segura.

### Antes de empezar

Preguntas previas:

1. **¿En qué archivos se definen las bases de datos de usuarios, contraseñas y grupos?**
  - Usuarios y contraseñas: `/etc/passwd` (usuarios y algunos datos básicos), `/etc/shadow` (contraseñas cifradas y políticas de expiración).
  - Grupos: `/etc/group`.
2. **¿Cómo se pueden asignar UID para nuevos usuarios?**
  - El UID se asigna típicamente de forma automática al crear usuarios con `useradd` u `adduser`. Si se hace manualmente, se puede editar `/etc/passwd` y asignar un UID único no utilizado. Normalmente las distribuciones mantienen un rango de UID para usuarios del sistema (0-999) y otro para usuarios normales (1000 en adelante).
3. **¿Qué comandos pueden usarse para cambiar propietarios y permisos de un archivo y de todos los archivos en un directorio?**
  - Cambiar propietario: `chown`.
  - Cambiar grupo: `chgrp`.
  - Cambiar permisos: `chmod`.Para aplicar en un directorio y todos sus subdirectorios y archivos, se usa la opción `-R` (recursivo), por ejemplo: `chown -R usuario:grupo /ruta/del/directorio`.

En un sistema Linux, cada usuario tiene una cuenta que incluye toda su información, archivos y procesos. Los usuarios se identifican internamente con un UID (un número entero). Además, existe una base de datos que asocia el UID a un nombre de usuario (username), y que también contiene información adicional (como el directorio home, el shell de inicio, etc.).

La creación de un nuevo usuario implica:

- Asignar un UID.
- Modificar la base de datos de usuarios.
- Crear un grupo o asociar el usuario a uno existente.
- Copiar archivos de configuración inicial desde `/etc/skel` al directorio home del usuario.
- Opcionalmente asignar el usuario a varios grupos.

El uso de grupos permite dividir a los usuarios en categorías con diferentes permisos y privilegios.

### Perfil y entorno de usuario

Tras un login interactivo, el shell ejecuta ciertos archivos de configuración. En el caso de BASH:

- Se ejecuta `/etc/bash_profile` (o `/etc/profile`) para configurar el entorno global.
- Luego se ejecuta `~/.bash_profile` para la configuración del usuario.

En `/etc/skel` se almacenan archivos base que se copian al crear nuevos usuarios. Esto permite asegurar un entorno mínimo (variables PATH, prompt, etc.).

#### Ejemplo:

- Asegurar que el PATH de todos los usuarios contenga `/usr/local/bin`:

```
export PATH=$PATH:/usr/local/bin
```

Este comando toma el valor actual de PATH, y le agrega al final :/usr/local/bin. De esta forma, cuando el usuario ejecute comandos, también se buscarán en ese directorio.

#### Análisis del comando `export PATH=$PATH:/usr/local/bin`:

- `export PATH=`: indica que vamos a definir la variable de entorno PATH.
- `$PATH:/usr/local/bin`: Toma el valor actual de PATH (\$PATH) y agrega :/usr/local/bin.
- Con `export` hacemos la variable visible para procesos hijos.

#### Cambiar el prompt para el usuario aso:

Queremos que se vea así:

```
aso:CWD - date $
```

Donde CWD es el directorio actual y date es la fecha en formato día/mes hora:minuto.

Podemos modificar la variable de entorno PS1 en `~/.bash_profile` por:

```
PS1='aso:\w - \d \t \$'
```

- `\w` muestra el directorio actual.
- `\d` muestra la fecha en formato DOW Mon DD.
- `\t` muestra la hora actual HH:MM:SS.

Si se requiere un formato específico día/mes hora:minuto, se podría utilizar el comando date embbebido, por ejemplo:

```
PS1='aso:\w - $(date +"%d/%m %H:%M") \$'
```

#### ¿Qué cambios se aplicaron a la variable PATH?

- Se le agregó el directorio /usr/local/bin.

#### ¿En qué variable se define el prompt?

- En la variable de entorno PS1.

#### ¿Qué cambios se aplicaron a la variable del prompt?

- Se estableció un valor personalizado que muestra el nombre del usuario, el directorio actual y la fecha/hora en el formato deseado.

## Creación de usuarios manualmente

Se pide crear cuentas de usuario para cada miembro del grupo de laboratorio, asignándolos al grupo admin.

#### Parámetros a definir:

- UID
- Nombre de usuario
- Directorio HOME
- Shell
- Grupos (principal y secundarios)

Por ejemplo:

#### Parámetros Usuario 1 Usuario 2

```
UID 1001 1002
```

```
Username quique marc
```

```
HOME /home/quique /home/marc
```

```
Shell /bin/bash /bin/bash
```

```
Groups admin (primario) admin (primario)
```

Para editar la base de datos de usuarios, se usa:

```
vipw
```

Esto abre /etc/passwd en un entorno seguro.

#### ¿Diferencia entre usar vipw y editar directamente con vi?

- `vipw` bloquea el fichero /etc/passwd y /etc/shadow para evitar corrupciones si otro administrador está editándolo simultáneamente. Garantiza integridad y seguridad. Si se usara `vi` directamente, podríamos dañar la consistencia del archivo si alguien más lo edita a la vez.

Usar `vigr` para editar /etc/group y crear o modificar grupos. Con `vipw -s` podemos editar /etc/shadow de forma segura.

#### ¿Qué grupos se han creado?

- Por ejemplo students.

#### ¿Qué usuarios forman parte de esos grupos?

- quique y marc en students.

- Otros usuarios según necesidad.

#### Desactivar una cuenta de usuario para que no pueda hacer login:

- Se puede colocar un \* o ! al inicio del campo de contraseña en /etc/shadow.
- Por ejemplo, cambiar el campo de contraseña del usuario en /etc/shadow a  
`! :usermod -L username`  
o manualmente con `vipw -s` colocando !.

#### Desactivar cuentas nuevas hasta terminar el proceso:

```
usermod -L juan
usermod -L maria
```

Crear el directorio HOME:

```
mkdir /home/quique
cp -r /etc/skel/* /home/quique
chown -R quique:quique /home/quique
chmod 700 /home/quique
```

#### ¿Qué comandos se han usado para cambiar el propietario?

```
chown -R juan:juan /home/juan
```

#### ¿Qué comandos para cambiar los permisos?

```
chmod 700 /home/juan
```

Asignar contraseña:

```
passwd quique
passwd marc
```

Esto actualiza /etc/shadow.

#### ¿Qué riesgos de seguridad existen al poner la contraseña en el archivo de usuarios?

- Si la contraseña estuviera en /etc/passwd (en texto plano o cifrado simple) podría ser leída por cualquier usuario ya que /etc/passwd es legible por todos. Por eso se usa /etc/shadow, que sólo es legible por root.

#### ¿Qué comando se usa para editar /etc/shadow de forma segura?

```
vipw -s
```

#### ¿Qué significan los parámetros en /etc/shadow?

/etc/shadow contiene:

- Nombre de usuario
- Contraseña cifrada
- Días desde el epoch del último cambio de contraseña
- Mínimo y máximo de días de validez, aviso de expiración, etc.

#### ¿Qué comando puede modificar parámetros de la contraseña?

- `chage` permite modificar las fechas de expiración de la contraseña, etc.

Usar `chfn` y `chsh` para establecer información del usuario:

- `chfn` juan permite cambiar el nombre completo, teléfono, etc.
- `chsh` juan permite cambiar el shell del usuario.

## Creación automática de usuarios

Utilizar useradd (o adduser) para crear cuentas:

- Profesores: emoranco, rserral
- Estudiante: student
- Cuentas asoXX para grupos de práctica, por ejemplo: aso01, aso02, etc.

Elegir nombres adecuados. Por ejemplo, emoranco y rserral para profesores, student para el estudiante genérico, y aso01 para un grupo.

```
#!/bin/bash

# Función para crear un usuario
create_user() {
    local username=$1
    local group=$2
    local shell=$3
    local home=$4

    # Crear grupo si no existe
    if ! grep -q "^${group}:" /etc/group; then
        echo "Creando grupo $group..."
        groupadd $group
    fi

    # Crear usuario con los parámetros especificados
    echo "Creando usuario $username..."
    useradd -m -d "$home" -s "$shell" -g "$group" "$username"

    # Asignar contraseña predeterminada (puede ser cambiada después)
    echo "$username:password" | chpasswd

    # Mostrar detalles
    echo "Usuario $username creado con éxito."
    echo "Home: $home | Grupo: $group | Shell: $shell"
    echo ""
}

# Crear profesores
echo "Creando cuentas de profesores..."
create_user "emoranco" "professors" "/bin/bash" "/home/emoranco"
create_user "rserral" "professors" "/bin/bash" "/home/rserral"

# Crear estudiante genérico
echo "Creando cuenta de estudiante..."
create_user "student" "students" "/bin/bash" "/home/student"

# Crear usuarios para grupos de laboratorio
echo "Creando cuentas de otros usuarios (grupos de laboratorio)..."
for i in $(seq -w 1 4); do
    username="aso$i"
    create_user "$username" "labgroup" "/bin/bash" "/home/$username"
done

echo "Todas las cuentas han sido creadas."
```

#### Permisos para diferentes grupos:

- Profesores: acceso a nivel de grupo a todos los archivos de todos los usuarios.
- Estudiantes: acceso a nivel de grupo a todos los archivos de todos los usuarios, excepto el de los profesores.
- Administradores: acceso a nivel de grupo sólo a los archivos de su propio grupo.
- Otros usuarios (asoXX): sin acceso a nivel de grupo a ningún archivo.

Estas condiciones dictan el uso de grupos y máscaras de creación (umask) apropiadas.

#### ¿Cómo configurar los permisos del directorio para que los archivos heredados tengan permisos adecuados?

- Ajustar la umask en /etc/profile o en el ~/.bash\_profile para los grupos deseados.
- Por ejemplo, para que los archivos creados por teachers tengan permisos group-read y group-write, se puede usar umask 002.
- Además, el bit **setgid** en directorios asegura que los nuevos archivos creados hereden el grupo del directorio, por ejemplo:

```
$> chmod g+s /home/teachersgroup
```

#### Eliminar y desactivar de usuarios

Para eliminar un usuario se deben:

- Borrar su correo, trabajos de impresión, trabajos cron y at.
- Eliminar su entrada en /etc/passwd, /etc/shadow y /etc/group.
- Borrar sus archivos, previo back up.

### ¿Cómo hacer backup de todos los archivos de un usuario?

- Usar find y tar:

```
find /home/username -user username -print | tar cvf backup_username.tar -T -
```

### Problema con nombres con espacios:

- xargs por defecto separa por espacios. Usar xargs -0 y find -print0 para manejar nombres con espacios (sin comprimir)

```
find / -user username -print0 | xargs -0 -I _ cp _ /backups
```

### ¿Qué comando para buscar y eliminar todos los archivos de un usuario?

- con -exec:

```
find / -user username -exec rm {} \;
```

- con xargs y excluyendo carpeta de backups:

```
find / -not \(-name"backups"\) -user username -print | xargs -I _ rm -rf _
```

Deshabilitar la cuenta antes de eliminar archivos:

- Cambiar el shell del usuario a un script que informe la desactivación:

```
chsh -s /usr/local/bin/failed-login.sh username
```

```
/usr/local/bin/failed-login.sh:
```

```
#!/usr/bin/bash

echo "Esta cuenta ha sido cerrada por razones de seguridad."

echo "Contacte al administrador."

read
```

- Asegurar permisos de ejecución: chmod +x /usr/local/bin/failed-login.sh

- Añadir /usr/local/bin/failed-login.sh a /etc/shells.

### ¿Cómo comprobar que la cuenta está desactivada?

- Intentar iniciar sesión con ese usuario y verificar que aparece el mensaje del script en vez de iniciar shell.

### Script para hacer backup, borrar archivos y desactivar cuenta:

```
#!/bin/bash
# Script: remove_user.sh
# Uso: remove_user.sh username

if [ $# -ne 1 ]; then
    echo "Uso: $0 username"
    exit 1
fi
user=$1

# 1. Backup de archivos
backup="backup_{user}.tar"

find / -user $user -print0 | xargs -0 tar cvf $backup
```

```
_# 2. Borrar archivos del usuario_
find / -user $user -exec rm {} \;

_# 3. Desactivar cuenta cambiando el shell_
chsh -s /usr/local/bin/failed-login.sh $user

echo "Cuenta $user desactivada y archivos respaldados en $backup."
```

## Sudo y Control de la Ejecución de Aplicaciones

Al igual que shutdown, hay otros comandos de administración que solo pueden ser ejecutados por el usuario root. Es una mala práctica de seguridad usar la cuenta root para ejecutar estos comandos. Para resolver este problema, se puede utilizar el comando sudo, que permite a un usuario ejecutar un comando con permisos de root. La configuración de qué aplicaciones puede ejecutar un usuario en particular se define en el archivo /etc/sudoers. Este archivo debe ser editado de manera segura usando el comando visudo.

### Pasos para Configurar sudo

#### 1. Editar el Archivo /etc/sudoers Usando visudo

Siempre utiliza `visudo` para editar el archivo /etc/sudoers ya que realiza comprobaciones de sintaxis antes de guardar los cambios, evitando errores que podrían bloquear el acceso administrativo.

```
sudo visudo
```

#### 2. Permitir al Grupo admin Ejecutar Todos los Comandos como Superusuarios

Añade la siguiente línea al final del archivo /etc/sudoers:

```
%admin ALL=(ALL:ALL) ALL
```

- %admin: Se refiere al grupo admin.
- ALL=(ALL:ALL): Permite ejecutar comandos como cualquier usuario y grupo.
- ALL: Permite ejecutar cualquier comando.

#### 3. Permitir al Grupo teachers Ejecutar Scripts Específicos y Binaries

Añade la siguiente línea para permitir que los usuarios del grupo teachers ejecuten el script de eliminación de usuarios y todos los binarios en /usr/local/teachers/bin:

```
%teachers ALL=(ALL:ALL) /usr/local/bin/remove_user.sh, /usr/local/teachers/bin/*
```

- %teachers: Se refiere al grupo teachers.
- /usr/local/bin/remove\_user.sh: Ruta al script específico que permite eliminar usuarios.
- /usr/local/teachers/bin/\*: Permite ejecutar cualquier binario dentro de este directorio.

#### • Como miembro del grupo admin:

```
sudo whoami
```

Debería devolver root.

#### • Como miembro del grupo teachers:

Ejecuta el script permitido:

```
sudo /usr/local/bin/remove_user.sh username
```

Debería ejecutarse sin solicitar permisos adicionales.

#### 6. Deshabilitar la Cuenta Root

#### Pasos para Deshabilitar la Cuenta Root

##### 1. Asegurarse de Tener Acceso Administrativo con sudo

Antes de deshabilitar root, verifica que puedes ejecutar comandos administrativos con un usuario del grupo admin.

```
sudo whoami
```

Debería devolver root.

## 2. Bloquear la Cuenta Root

Bloquea la contraseña de root para evitar inicios de sesión directos.

```
sudo passwd -l root
```

• -l: Bloquea la contraseña del usuario, evitando inicios de sesión directos.

## 3. Verificar que la Cuenta Root Está Bloqueada

Intenta cambiar a root:

```
su - root
```

Deberías recibir un mensaje de error indicando que la cuenta está deshabilitada.

## 4. Configurar sudo para Usar Solo Usuarios del Grupo admin

Asegúrate de que solo los miembros del grupo admin puedan usar sudo para tareas administrativas.

### Preguntas y Respuestas

#### 1. ¿Qué cambios se requieren en el archivo /etc/sudoers para habilitar la configuración descrita anteriormente?

- Para el Grupo admin:

```
%admin  ALL=(ALL:ALL)  ALL
```

Permite a todos los miembros del grupo admin ejecutar cualquier comando como cualquier usuario y grupo.

- Para el Grupo teachers:

```
%teachers  ALL=(ALL:ALL)  /usr/local/bin/remove_user.sh,  /usr/local/teachers/bin/*
```

Permite a los miembros del grupo teachers ejecutar el script específico remove\_user.sh y cualquier binario dentro de /usr/local/teachers/bin.

#### 2. ¿Cuáles son los pasos requeridos para deshabilitar la cuenta root?

- Verificar Acceso Administrativo:

• Asegúrate de que puedes ejecutar comandos administrativos con sudo desde una cuenta perteneciente al grupo admin.

- Bloquear la Cuenta Root:

• Ejecuta `sudo passwd -l root` para bloquear la contraseña de root.

- Verificar el Bloqueo:

• Intenta cambiar a root con `su - root` y verifica que no es posible iniciar sesión.

- Configurar sudo Adecuadamente:

• Asegúrate de que solo los usuarios autorizados (grupo admin) puedan usar sudo para tareas administrativas antes de bloquear a root.

## Practica 5 Backups

Resumen rápido preguntas del principio:

### Empaquetar y desempaquetar archivos con tar:

- Empaquetar:

```
tar -cvf archivo.tar archivo1 archivo2 directorio/
```

• -c: Crea un nuevo archivo tar.

• -v: Muestra el progreso en la terminal.

- -f: Especifica el nombre del archivo tar resultante.

- **Desempaquetar:**

```
tar -xvf archivo.tar
```

- -x: Extrae los archivos del tar.
- -v: Muestra el progreso en la terminal.
- -f: Especifica el archivo tar a extraer.

- **Hard link**

- Un hard link es una referencia adicional al mismo inodo en el sistema de archivos.

- **Crear un hard link**

```
ln archivo_a archivo_b
```

### Diferencia entre cp y ln:

- **Copiar archivos (cp):**

- Crea una copia independiente con un inodo diferente.
  - Modificar uno no afecta al otro.
  - Si se elimina uno, el otro permanece.
- **Crear un hard link (ln):**
- Ambos archivos comparten el mismo inodo.
  - Modificar uno afecta al otro.
  - El archivo no se elimina hasta que todos los enlaces son eliminados.

## Introducción

### Consideraciones para la Política de Back up

Antes de realizar back ups , el administrador debe decidir una política considerando aspectos como:

- **Tipo de Medio Físico:** Seleccionar el medio adecuado para los back ups considerando tamaño, costo, velocidad, disponibilidad, usabilidad y fiabilidad.
- **Archivos a los que hacer un back up:** Decidir qué archivos necesitan back up y dónde se encuentran. Los archivos más importantes son los de configuración y los de usuario (usualmente en /root y /home, respectivamente). Algunos archivos no requieren back up, como los temporales (/tmp) y los binarios del sistema (/bin, /sbin/).
- **Frecuencia y Programación:** Depende de la variabilidad de los datos. Por ejemplo, una base de datos puede requerir back ups múltiples diarios, mientras que un servidor web puede necesitar solo una copia diaria y otros sistemas de archivos una copia semanal.
- **Otros Aspectos:** Dónde almacenar las copias, cuánto tiempo mantenerlas y la rapidez necesaria para recuperar cada tipo de archivo.

### Estrategia de back up

Con la información anterior, es posible decidir una estrategia de back up que incluye la frecuencia y el tipo de copias. Una estrategia común es realizar copias completas e incrementales:

- **Copias Completas (Nivel 0):** Semanales.
- **Copias Incrementales (Nivel 1 o superior):** Diarias.

Si la tasa de cambio de archivos es muy alta, se puede modificar el modelo para incluir:

- **Nivel 0:** Copia mensual.
- **Nivel 1:** Copias incrementales semanales.
- **Nivel 2:** Copias incrementales diarias.

### Herramientas para Implementar la Estrategia

En este laboratorio utilizaremos `tar` y `rsync`. Se usará el mismo disco para realizar los back ups , aunque en un entorno real esto no es recomendable debido al alto riesgo de que una pérdida de datos afecte también a los back ups .

## Pasos para Configurar la Partición de Back Up

### 1. Crear el directorio /backup:

Este directorio servirá como punto de montaje para la nueva partición donde se almacenarán los back ups .

```
mkdir /backup
```

### 2. Crear una nueva partición en el espacio libre disponible:

Utiliza herramientas como gdisk para crear una nueva partición en el espacio libre del disco.

```
gdisk /dev/sdX
```

- Dentro de `gdisk`, sigue los pasos interactivos:
- Presiona n para crear una nueva partición.
- Selecciona el tipo de partición, número y tamaño según tus necesidades.
- Presiona w para escribir los cambios en el disco.

### 3. Formatear la partición con el sistema de archivos btrfs:

Formatea la nueva partición para prepararla para su uso.

```
mkfs.btrfs /dev/sdXn
```

- Reemplaza `/dev/sdXn` con el identificador de la partición creada (por ejemplo, `/dev/sdb1`).

### 4. Montar la partición en el directorio /backup:

Monta la partición formateada en el directorio `/backup`.

```
mount /dev/sdXn /backup
```

### 5. Cambiar los permisos del directorio /backup para restringir el acceso a root:

Ajusta los permisos para que solo el usuario root pueda acceder al directorio, protegiendo así los back ups .

```
chmod 700 /backup
```

```
chown root:root /backup
```

### Protección Adicional para el Directorio /backup

Para aumentar la seguridad del directorio `/backup`, es recomendable montar la partición en modo de solo lectura la mayor parte del tiempo, permitiendo el modo de lectura-escritura únicamente cuando se realicen back ups . Esto previene modificaciones no autorizadas y protege la integridad de los datos respaldados.

- Montar en modo de solo lectura:

```
mount -o remount,ro /dev/sdXn /backup
```

- Montar en modo de lectura-escritura:

```
mount -o remount,rw /dev/sdXn /backup
```

### Respuesta a la Pregunta Planteada

#### ¿Qué debes hacer para montar automáticamente esta nueva partición en modo de solo lectura al inicio del sistema?

Para montar automáticamente la nueva partición en modo de solo lectura durante el arranque del sistema, es necesario editar el archivo `/etc/fstab`. Este archivo contiene las especificaciones de montaje de los sistemas de archivos que se deben montar al iniciar el sistema.

### Pasos para Configurar el Montaje Automático en Modo de Solo Lectura:

#### 1. Identificar el UUID de la partición:

Es recomendable usar el UUID en lugar del nombre del dispositivo para evitar problemas si cambian las designaciones de los dispositivos.

```
blkid /dev/sdXn
```

- Esto devolverá una línea similar a:

```
/dev/sdXn: UUID="xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" TYPE="btrfs"
```

## 2. Editar el archivo `/etc/fstab`:

Abre el archivo `/etc/fstab` con un editor de texto, por ejemplo, vim.

```
vim /etc/fstab
```

## 3. Añadir una línea para la nueva partición con opciones de montaje en ro:

```
UUID="xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" /backup btrfs defaults,ro 0 2
```

- `UUID=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx`: Identificador único de la partición.
- `/backup`: Punto de montaje.
- `btrfs`: Sistema de archivos.
- `defaults,ro`: Opciones de montaje (defaults incluye opciones estándar y `ro` especifica modo de solo lectura).
- `0 2`: Opciones de dump y fsck

Monta todas las particiones especificadas en `/etc/fstab` sin reiniciar.

```
mount -a
```

- Confirma que la partición `/backup` esté montada en modo de solo lectura:

```
mount | grep /backup
```

- Deberías ver una línea que indica que `/backup` está montado con las opciones `ro`.

## Backups completos

Para realizar un back up completo del directorio `/root`, utilizamos el comando tar, que permite empaquetar y opcionalmente comprimir archivos y directorios. Es fundamental que los nombres de los archivos de back up sean descriptivos e incluyan información sobre el contenido, la fecha y hora del back up, así como si el back up es completo o incremental. Esto facilita la gestión y restauración de los back ups .

### Incluir la Fecha Automáticamente en el Nombre del Archivo de Back up

Para incluir automáticamente la fecha y hora en el nombre del archivo de back up, podemos utilizar el comando date dentro de una sustitución de comandos `$()`. Por ejemplo:

```
backup-etc-level0-$(date +%Y%m%d-%H%M).tar
```

Este comando generará un nombre de archivo con el formato `backup-etc-level0-202401131230.tar` si se ejecuta el 13 de enero de 2024 a las 12:30.

### Comando para Hacer una Copia Completa del Directorio `/root`

Para crear una copia completa del directorio `/root`, utilizamos el siguiente comando tar:

```
tar -cvf /backup/backup-root-level0-$(date +%Y%m%d-%H%M).tar -C /root .
```

- `tar`: Comando para empaquetar archivos.
- `-c`: Crea un nuevo archivo tar.
- `-v`: Muestra el progreso en la terminal (modo verbose).

- -f: Especifica el nombre del archivo tar resultante.
- /backup/backup-root-level0-\$(date +%Y%m%d%H%M).tar: Ruta y nombre del archivo de back up, incluyendo la fecha y hora.
- -C
- /root: Directorio que se va a hacer un back up.

### Compresión del Archivo de Back up

Para comprimir el archivo de back up, se añade la opción -z al comando tar, que utiliza gzip para la compresión. El comando modificado sería:

```
tar -cvfz /backup/backup-root-level0-$(date +%Y%m%d%H%M).tar.gz /root
```

- -z: Comprime el archivo usando gzip.

### Consideraciones de Seguridad al Comprimir Archivos de Back up

Aunque comprimir los archivos de back up reduce el espacio que ocupan y puede facilitar su transferencia, **no es recomendable en términos de seguridad** por las siguientes razones:

- **Exposición de Datos Sensibles:** La compresión no encripta los datos, lo que significa que cualquier persona con acceso al archivo comprimido puede extraer y leer su contenido.
- **Integridad de los Datos:** La compresión puede introducir vulnerabilidades si no se maneja correctamente, aunque esto es menos común.
- Tener que descomprimir i comprimir cada vez que se necesitan los datos **introduce mucho overhead** y puede hacer que el proceso tarde aún más.

### Excluir Archivos Específicos del Back up

A veces, es necesario excluir ciertos archivos del back up completo. Para ello, se puede crear un archivo de exclusiones que liste los archivos o directorios a omitir. Por ejemplo, creamos un archivo llamado excludes.txt con el siguiente contenido:

```
/tmp
/bin
/sbin
```

### Comando para Crear el Archivo de Exclusiones:

```
echo -e "/tmp\n/bin\n/sbin" > /backup/excludes.txt
```

### Comando para Excluir Archivos en el Back up

Para excluir los archivos listados en excludes.txt, se añade la opción --exclude-from al comando tar:

```
tar -cvf /backup/backup-root-level0-$(date +%Y%m%d%H%M).tar --exclude-from=/backup/excludes.txt /root
```

- --exclude-from=/backup/excludes.txt: Especifica el archivo que contiene la lista de archivos/directorios a excluir.

### Verificación de la Integridad del back up con sha512sum

Es importante verificar que los archivos de back up no hayan sido modificados después de su creación. Para ello, utilizamos sha512sum para generar una firma digital que asegura la integridad del archivo.

### Comando para Generar la Firma SHA512:

```
sha512sum /backup/backup-root-level0-(date +%Y%m%d%H%M).tar > /backup/backup-root-level0-$(date +%Y%m%d%H%M).tar.asc
```

### Desglose del Comando:

- sha512sum /backup/backup-root-level0-\$(date +%Y%m%d%H%M).tar: Calcula el hash SHA512 del archivo de back up.
- >: Redirecciona la salida al archivo especificado.
- /backup/backup-root-level0-\$(date +%Y%m%d%H%M).tar.asc: Archivo donde se almacena la firma SHA512.

*Si quieres un archivo .asc con todos los sha512sum de los backups, haz append en vez de redireccionar la salida*

```
sha512sum /path/to/backup/file >> /path/to/signature.asc
```

### Respuestas a las Preguntas de 5.5.1

1. **¿Cómo puedes incluir automáticamente la fecha en el nombre del archivo de back up?**

Utilizando el comando date dentro de una sustitución de comandos \$(). Por ejemplo:

```
backup-etc-level0-$(date +%Y%m%d-%H%M).tar
```

2. **¿Qué comando has usado para hacer la copia completa del directorio /root?**

```
tar -cvf /backup/backup-root-level0-$(date +%Y%m%d%H%M).tar /root
```

3. **Si queremos comprimir el archivo de back up, ¿qué opción debes añadir al comando tar?**

Añadir la opción -z para comprimir con gzip:

```
tar -cvfz /backup/backup-root-level0-$(date +%Y%m%d%H%M).tar.gz /root
```

4. **¿Por qué no es generalmente bueno en términos de seguridad comprimir el archivo de back up?**

• **Exposición de Datos Sensibles:** La compresión no encripta los datos, permitiendo que cualquier persona con acceso al archivo comprimido pueda extraer y leer su contenido.

• **Integridad de los Datos:** Aunque menos común, la compresión puede introducir vulnerabilidades si no se maneja correctamente.

• **Dependencia de Herramientas:** Requiere herramientas de descompresión compatibles durante la restauración.

5. **¿Qué comando has añadido al comando tar para excluir archivos?**

Añadir la opción --exclude-from especificando el archivo de exclusiones:

```
tar -cvf /backup/backup-root-level0-$(date +%Y%m%d%H%M).tar --exclude-from=/backup/excludes.txt /root
```

6. **¿Cómo has usado el comando sha512sum para producir la firma SHA512?**

Generando el hash y redireccionándolo a un archivo con extensión .asc:

```
sha512sum /backup/backup-root-level0-$(date +%Y%m%d%H%M).tar > /backup/backup-root-level0-$(date +%Y%m%d%H%M).tar.asc
```

## Back ups Incrementales

Un back up incremental captura únicamente los archivos que han cambiado desde el último back up (ya sea completo o incremental). Esto optimiza el espacio y el tiempo necesarios para realizar los back ups , especialmente en sistemas con grandes volúmenes de datos que cambian frecuentemente.

### Modificaciones Previas al back up Incremental

Antes de realizar un back up incremental, es necesario modificar algunos archivos en el directorio /root:

- **Crear nuevos archivos y subdirectorios:**

```
mkdir /root/nuevo_directorio  
touch /root/nuevo_directorio/nuevo_archivo.txt
```

- **Modificar el contenido de algunos archivos:**

```
echo "Contenido actualizado" >> /root/existente.txt
```

- **Usar el comando touch para cambiar la fecha de modificación de algunos archivos:**

```
touch /root/existente.txt
```

### Realizar un back up Incremental con tar

Para realizar un back up incremental, se utiliza la opción --newer de tar, que incluye únicamente los archivos modificados desde una fecha específica o desde la última modificación de un archivo de referencia.

#### Ejemplo de Comando para back up Incremental:

```
tar -cvf /backup/backup-root-level1-$(date +%Y%m%d%H%M).tar --newer="2024-01-13 12:30" /root
```

O poniendo directamente que los archivos se hayan modificado en el último x tiempo:

```
name=$(date '+%Y%m%d-%H%M')
tar -zcvf /backup/"$name.tar.gz" --newer-mtime="24 hours ago" -C /etc .
```

"X [time\_mesurement] ago", como "time\_mesurement" puedes poner: second(s), minute(s), hour(s), day(s), month(s), year(s)...

O utilizando un archivo de referencia:

```
tar -cvf /backup/backup-root-level1-$(date +%Y%m%d%H%M).tar --newer=/backup/backup-root-level0-202401131230.tar /root
```

### Generar Firma SHA512 para el back up Incremental

```
sha512sum /backup/backup-root-level1-$(date +%Y%m%d%H%M).tar > /backup/backup-root-level1-$(date +%Y%m%d%H%M).tar.asc
```

### Respuestas a las Preguntas de 5.5.2

#### 1. ¿Qué comando has usado para hacer la copia incremental?

```
tar -cvf /backup/backup-root-level1-$(date +%Y%m%d%H%M).tar --newer=/backup/backup-root-level0-202401131230.tar /root
```

#### 2. ¿Qué problema potencial tiene el uso del archivo de back up completo para obtener la fecha del back up al hacer la copia incremental?

El principal problema es que el proceso de back up completo puede llevar mucho tiempo, lo que puede generar discrepancias en la fecha si el back up incremental se realiza mientras el back up completo aún está en progreso. Además, si el back up completo falla o se interrumpe, la referencia para el back up incremental puede no ser precisa.

#### 3. ¿Cómo puedes resolver este problema?

Para solucionar este problema, se puede utilizar un archivo de referencia que se actualice solo cuando el back up completo se haya completado exitosamente. De esta manera, los back ups incrementales siempre se basarán en una referencia estable y consistente.

### Realizar un Segundo back up Incremental

Después de realizar más modificaciones en el directorio /root:

- **Crear nuevos archivos y subdirectorios:**

```
mkdir /root/otro_directorio
touch /root/otro_directorio/otro_archivo.txt
```

- **Modificar el contenido de algunos archivos:**

```
echo "Otro contenido actualizado" >> /root/existente.txt
```

- **Usar el comando touch para cambiar la fecha de modificación de algunos archivos:**

```
touch /root/existente.txt
```

- Eliminar algunos de los archivos generados para el primer back up incremental:

```
rm /root/nuevo_directorio/nuevo_archivo.txt
```

#### Comando para el Segundo back up Incremental:

```
tar -cvf /backup/backup-root-level2-$(date +%Y%m%d%H%M).tar --newer=/backup/backup-root-level1-202401131245.tar /root
```

#### Generar Firma SHA512 para el Segundo back up Incremental:

```
sha512sum /backup/backup-root-level2-$(date +%Y%m%d%H%M).tar > /backup/backup-root-level2-$(date +%Y%m%d%H%M).tar.asc
```

#### 4. ¿Cómo puedes verificar que el contenido del back up es el mismo que el directorio original?

Comparando el contenido del back up con el directorio original utilizando herramientas como diff, o verificando la integridad con las firmas SHA512 previamente generadas.

#### 5. ¿Cómo puedes verificar, usando el comando sha512sum, la integridad del back up?

Calculando el hash SHA512 del archivo de back up y comparándolo con el hash almacenado en el archivo .asc. Por ejemplo:

```
sha512sum -c /backup/backup-root-level1-202401131245.tar.asc
```

Si los hashes coinciden, la integridad del back up está asegurada.

#### 5.5.3 Restaurando un back up

La restauración de un back up implica descomprimir y extraer los archivos respaldados en su ubicación original. Es crucial restaurar los back ups en el orden correcto para asegurar que las copias incrementales se apliquen adecuadamente sobre el back up completo.

#### Simular Pérdida de Datos y Restauración

##### 1. Renombrar el Directorio /root para Simular Pérdida de Datos:

```
mv /root /root.old
```

##### 2. Restaurar el Directorio desde los back ups :

Para restaurar completamente el directorio /root, se deben restaurar primero el back up completo y luego los back ups incrementales en el orden en que fueron creados.

##### Orden de Restauración:

- Restaurar el back up completo (Nivel 0).
- Restaurar el primer back up incremental (Nivel 1).
- Restaurar el segundo back up incremental (Nivel 2).

##### 3. Comandos Utilizados para Restaurar los back ups :

- Restaurar el back up Completo:

```
tar -xvf /backup/backup-root-level0-202401131230.tar -C /
```

##### • Restaurar el Primer back up Incremental:

```
tar -xvf /backup/backup-root-level1-202401131245.tar -C /
```

##### • Restaurar el Segundo back up Incremental:

```
tar -xvf /backup/backup-root-level2-202401131300.tar -C /
```

#### 4. ¿Qué ocurrió con los archivos que se habían eliminado antes del segundo back up incremental?

Al restaurar los back ups completos e incrementales, los archivos eliminados antes del segundo back up incremental no se restauran automáticamente, ya que los back ups incrementales solo agregan o modifican archivos, pero no eliminan aquellos que ya fueron eliminados en el sistema original.

#### 5. ¿Cómo puedes detectar que algunos archivos han sido eliminados? ¿Cuándo esos archivos serán restaurados desde los back ups ?

- **Detección de Eliminaciones:**

Utilizando herramientas de comparación como diff para comparar el estado actual del directorio /root con el estado de los back ups restaurados.

- **Restauración de Archivos Eliminados:**

Para restaurar archivos que fueron eliminados, se deben extraer específicamente desde el back up completo o desde el back up incremental donde fueron incluidos antes de ser eliminados.

#### Restauración de un Fragmento del back up

Para restaurar únicamente una subcarpeta específica del directorio /root, se puede utilizar la opción --extract con una ruta específica.

#### Ejemplo de Comando para Restaurar una Subcarpeta:

```
#para listar el contenido sin descomprimir/desempaquetar:  
tar -tf nombre.tar.gz  
# una vez encontrado el nombre del subdirectorio, extrae solo ese haciendo:  
tar -xvf nombre.tar.gz -C /root ./nombre_subdirectorio/
```

- -tf: Alistar archivos
- -xvf: Opciones para extraer, verbose, y especificar archivo.
- /backup/backup-root-level0-202401131230.tar: Archivo de back up desde el cual extraer.
- /root/subdirectorio: Subdirectorio específico que se desea restaurar.
- -C /: Directorio raíz donde se restaurará la subcarpeta.

## 5.6 Backups con rsync

Hasta ahora, hemos almacenado los back ups en la misma máquina que contiene los datos. Sin embargo, lo más común es tener varias máquinas para respaldar y almacenar las copias en una máquina central utilizando la red.

Para ello, podemos usar el comando **rsync** -> permite copiar un directorio (o conjunto de archivos) a otro directorio a través de una conexión de red.

**rsync** utiliza un algoritmo de suma de verificación eficiente para transmitir solo las diferencias entre los dos directorios, al mismo tiempo que comprime los archivos para una transmisión más rápida.

Esta herramienta permite copiar archivos desde o hacia un directorio ubicado en una máquina remota, o directorios de la misma máquina. Lo que no permite es copiar directorios entre dos máquinas remotas. Además, rsync permite copiar enlaces, dispositivos y preservar permisos, propietarios y grupos. También soporta listas de exclusión y conexiones remotas usando Secure Shell (SSH) entre otras posibilidades (para más información, consulte man rsync).

### 5.6.1 Haciendo back ups a Tráves de una Red

Como se mencionó anteriormente, rsync puede respaldar una máquina remota. Esto se puede hacer con rsh, o poniendo rsync en modo servidor, pero puede ser peligroso porque una máquina local en la red podría estar capturando los datos de la conexión.

Para resolver estos problemas, rsync permite conexiones seguras usando ssh.

#### Pasos Requeridos para Instalar y Activar el Servidor SSH

1. **Instalar el Servidor SSH:**

En distribuciones basadas en Debian:

```
sudo apt-get update  
sudo apt-get install openssh-server
```

2. **Iniciar y Habilitar el Servicio SSH:**

```
sudo systemctl start ssh  
sudo systemctl enable ssh
```

### 3. Verificar que el Servidor SSH Está Corriendo:

```
sudo systemctl status ssh
```

## Habilitar el Acceso del Usuario Root a Través de SSH

Por razones de seguridad, el acceso del usuario root a través de SSH está deshabilitado por defecto. Para habilitarlo, sigue estos pasos:

### 1. Editar el Archivo de Configuración de SSH:

```
sudo nano /etc/ssh/sshd_config
```

### 2. Buscar la Línea que Contiene PermitRootLogin:

### 3. Descomentar la Línea y Cambiar su Valor a yes:

```
#PermitRootLogin prohibit-password  
PermitRootLogin yes
```

### 4. Reiniciar el Servicio SSH para Aplicar los Cambios:

```
sudo systemctl restart ssh
```

**Nota:** Habilitar el acceso del usuario root a través de SSH puede representar un riesgo de seguridad. Se recomienda utilizar claves SSH en lugar de contraseñas y restringir el acceso por IP si es posible.

Para realizar back ups completos utilizando rsync, seguiremos estos pasos:

## 5.6.2 Haciendo backups completos

### 1. Crear un Directorio para los back ups en la Partición /backup:

```
mkdir -p /backup/rsync-backup/
```

### 2. Ejecutar el Comando rsync:

Comando Proporcionado:

```
rsync -avz /root -e ssh root@localhost:/backup/rsync-backup/
```

- -a (**Archive**): Modo de archivo. Equivale a -rlptgoD. Preserva enlaces simbólicos, permisos, propietarios, grupos, tiempos y dispositivos.
- -v (**Verbose**): Modo verbose. Muestra información detallada sobre el proceso de sincronización.
- -z (**Compress**): Comprime los datos durante la transferencia para acelerar la transmisión a través de la red.

## Crear un Archivo en /root y Realizar el Mismo Comando rsync

### 1. Crear un Archivo Nuevo:

```
touch /root/nuevo_archivo.txt
```

### 2. Ejecutar el Comando rsync de Nuevo:

```
rsync -avz /root -e ssh root@localhost:/backup/rsync-backup/
```

### 3. Eliminar el Archivo y Ejecutar rsync Nuevamente:

```
rm /root/nuevo_archivo.txt
rsync -avz /root -e ssh root@localhost:/backup/rsync-backup/
```

### ¿Qué Ocurrió con el Archivo Eliminado?

El archivo eliminado (nuevo\_archivo.txt) ya no existe en el directorio /root, pero **no se elimina automáticamente del directorio de back up** /backup/rsync-backup/. Esto se debe a que, por defecto, rsync solo actualiza o agrega archivos, pero no elimina aquellos que han sido eliminados en el directorio fuente.

### ¿Qué Opción de rsync Permite una Sincronización Exacta de los Dos Directorios?

La opción --delete permite que rsync elimine en el directorio de destino aquellos archivos que ya no existen en el directorio fuente, logrando una sincronización exacta.

**Comando con la Opción** --delete:

```
rsync -avz --delete /root -e ssh root@localhost:/backup/rsync-backup/
```

### ¿Cómo Puedes Hacer una Copia de Todos los Archivos en el Directorio /root Excepto Aquellos que Tienen una Extensión .txt?

Utilizando la opción --exclude para excluir archivos con la extensión .txt.

**Comando:**

```
rsync -avz --exclude='*.txt' /root -e ssh root@localhost:/backup/rsync-backup/
```

### ¿Cuál es la Diferencia entre rsync /source /destination y rsync /source /destination/?

Con /destination/, si /destination ya existe, rsync creará un subdirectorio llamado source dentro de /destination. Por ejemplo, los archivos de /source se copiarán a /destination/source/. Si no, rsync copia /source directamente dentro de /destination sin crear un subdirectorio adicional. Es decir, los archivos de /source se copiarán directamente a /destination/.

### 5.6.3 Haciendo back ups Incrementales Inversos

Como se vio en la sección anterior, cada vez que haces una copia y sincronizas, el directorio donde tienes el espejo es exactamente igual al directorio fuente. Esto puede ser un problema en algunas situaciones, ya que no permite controlar los cambios realizados en los archivos. Para resolver esto, puedes usar las opciones --backup y --backup-dir de rsync. Los back ups generados con estas opciones se llaman **inversos** porque la copia completa es la más reciente, a diferencia de tar.

#### Script Simple para Hacer back ups Incrementales Inversos con rsync

```
#!/bin/bash
# Directorio fuente_
SOURCE_DIR=/root
# Directorio de destino_
DEST_DIR=/backup/rsync-backup/
# Archivo de exclusiones: lista de archivos a excluir_
EXCLUDES=/backup/excludes.txt
# Nombre de la máquina de back up_
BSERVER=localhost
# Fecha para el back up: año, mes, día, hora, minuto, segundo_
BACKUP_DATE=$(date +%Y%m%d%H%M%S)
# Opciones para rsync_
OPTS="--ignore-errors --delete-excluded --exclude-from=$EXCLUDES \ --delete --backup --backup-dir=$DEST_DIR/backups/$BACKUP_DATE -av"
```

```
_# Transferencia real_
rsync $OPTS $SOURCE_DIR root@$BSERVER:$DEST_DIR/complet
```

- SOURCE\_DIR: Directorio que se va a respaldar.
- DEST\_DIR: Directorio donde se almacenarán los back ups .
- EXCLUDES: Archivo que contiene la lista de archivos/directorios a excluir del back up.
- BSERVER: Nombre o dirección IP de la máquina de back up.
- BACKUP\_DATE: Fecha y hora actual en formato YYYYMMDDHHMMSS.
- OPTS: Opciones adicionales para rsync.

#### • **Opciones de rsync :**

- --ignore-errors: Ignora errores durante la transferencia.
- --delete-excluded: Elimina archivos excluidos del destino.
- --exclude-from: Especifica el archivo de exclusiones.
- --delete: Elimina archivos en el destino que ya no existen en la fuente.
- --backup: Realiza back ups de los archivos que se van a sobrescribir o eliminar.
- --backup-dir: Directorio donde se almacenarán los archivos respaldados.
- -a: Modo archivo.
- -v: Modo verbose.

### Proceso de back up

#### 1. Crear un Archivo en el Directorio Fuente y Sincronizar con el Script:

```
touch /root/nuevo_archivo_inverso.txt
./script_back_up.sh
```

#### 2. Modificar el Archivo Nuevo y Sincronizar de Nuevo:

```
echo "Contenido actualizado" >> /root/nuevo_archivo_inverso.txt
./script_back_up.sh
```

#### 3. Eliminar el Archivo y Sincronizar de Nuevo:

```
rm /root/nuevo_archivo_inverso.txt
./script_back_up.sh
```

### ¿Qué Ocurre con el back up Cuando el Archivo Nuevo es Modificado?

Cuando el archivo nuevo (nuevo\_archivo\_inverso.txt) es modificado y se ejecuta nuevamente el script de rsync, el archivo actualizado se copia al directorio de destino (/backup/rsync-backup/complet/). Además, la versión anterior del archivo se mueve al directorio de back ups (/backup/rsync-backup/backups/20240113130000/), preservando así las versiones anteriores.

### ¿Qué Ocurre Cuando el Archivo es Eliminado?

Al eliminar el archivo (nuevo\_archivo\_inverso.txt) y ejecutar nuevamente el script de rsync, el archivo es eliminado del directorio de destino (/backup/rsync-backup/complet/). Sin embargo, una copia del archivo eliminado se almacena en el directorio de back ups (/backup/rsync-backup/backups/20240113130000/), permitiendo su recuperación si es necesario.

### 5.6.4 Backups con Snapshots

Una posibilidad que ofrece rsync es realizar back ups incrementales donde, utilizando las propiedades de los hard link, las copias incrementales parecen copias completas. Esto se logra aprovechando que los hard links permiten que múltiples nombres de archivo apunten al mismo inodo, lo que facilita la creación de snapshots que consumen menos espacio en disco.

#### Hard links \*

El nombre de archivo no representa el archivo en sí mismo, es solo un hard link al inodo. Esto permite que un archivo (inode) tenga más de un hard link. Por ejemplo, si tienes un archivo llamado file\_a, puedes crear un enlace a él llamado file\_b:

```
ln file_a file_b
```

Usando el comando stat es posible saber cuántos hard links tiene un archivo:

```
stat file_a
```

### 1. ¿Cómo puedes detectar si file\_a y file\_b pertenecen al mismo inodo?

Utilizando el comando stat en ambos archivos y comparando el número de inodo.

```
stat file_a  
stat file_b
```

Si ambos muestran el mismo número de inodo (Inode:), entonces pertenecen al mismo inodo.

### 2. ¿Qué ocurre con file\_b si hay cambios en el contenido de file\_a?

Dado que ambos archivos apuntan al mismo inodo, cualquier cambio en el contenido de file\_a también se refleja en file\_b, ya que ambos son el mismo archivo a nivel de sistema de archivos. Cambian las fechas de ACCESS, CHANGE y MODIF.

### 3. ¿Qué ocurre con file\_b si hay cambios en los permisos de file\_a?

Ambos tendrán los mismos permisos.

### 4. ¿Qué ocurre con file\_b si copias otro archivo, sobrescribiendo file\_a con cp file\_c file\_a?

Si sobrescribes file\_a con cp file\_c file\_a, file\_a ahora apuntará a un nuevo inodo que contiene el contenido de file\_c. Sin embargo, file\_b seguirá apuntando al inodo original de file\_a, manteniendo su contenido previo.

### 5. ¿Y si lo sobrescribes con la opción --remove-destination de cp?

Usar cp --remove-destination file\_c file\_a primero elimina file\_a y luego crea una nueva copia con el contenido de file\_c. Esto rompe el hard link entre file\_a y file\_b. Ahora, file\_a apunta al nuevo contenido, mientras que file\_b sigue apuntando al inodo original con el contenido previo.

### 6. ¿Y qué ocurre con file\_b si file\_a es eliminado?

Si eliminas file\_a (rm file\_a), file\_b seguirá existiendo y apuntando al mismo inodo. El archivo no se elimina del sistema de archivos hasta que se eliminan **todos** los hard links que apuntan a él. Por lo tanto, file\_b mantendrá el contenido original incluso después de eliminar file\_a.

## Back ups tipo Snapshot

Los back ups tipo snapshot permiten crear múltiples copias de seguridad que parecen copias completas de un sistema de archivos sin requerir todo el espacio en disco necesario para almacenar todas las copias. Esto se logra combinando los comandos rsync y cp -al, aprovechando las propiedades de los enlaces duros para minimizar el uso de espacio.

El proceso general para crear back ups snapshot utilizando cp y rsync es el siguiente:

```
_# Eliminar la copia más antigua_  
rm -rf backup.3  
_# Mover las copias existentes una posición hacia atrás_  
mv backup.2 backup.3  
mv backup.1 backup.2  
_# Crear una nueva copia utilizando enlaces duros_  
cp -al backup.0 backup.1  
  
_# Sincronizar el directorio fuente con la copia más reciente_  
rsync -a --delete source_directory/ /backup.0/
```

### 1. Eliminar la Copia Más Antigua:

```
rm -rf backup.3
```

Elimina la copia de back up más antigua para hacer espacio para la nueva copia.

### 2. Mover las Copias Existentes:

```
mv backup.2 backup.3  
mv backup.1 backup.2
```

Desplaza las copias de back up actuales hacia posiciones más antiguas.

### 3. Crear una Nueva Copia con Enlaces Duros:

```
cp -al backup.0 backup.1
```

Crea una nueva copia (backup.1) que comparte los mismos inodos que backup.0, utilizando enlaces duros (-l) y preservando los atributos (-a).

### 4. Sincronizar con rsync:

```
rsync -a --delete source_directory/ /backup.0/
```

Sincroniza el directorio fuente con backup.0, copiando solo las diferencias y eliminando archivos que ya no existen en la fuente.

## Ventajas y Desventajas

### • Ventajas:

- **Eficiencia en el Uso de Espacio:** Las copias incrementales solo almacenan las diferencias, reduciendo el espacio necesario.
- **Simulación de Copias Completas:** Las copias snapshot parecen copias completas, facilitando la restauración.

### • Desventajas:

- **Permisos y Propiedades:** Las copias de back up de días anteriores heredan los permisos y propiedades del back up actual, lo que puede no ser deseable.

## Uso de la Opción --link-dest de rsync

Para optimizar el proceso y preservar permisos y propietarios de copias anteriores, se puede utilizar la opción --link-dest de rsync, eliminando la necesidad del comando cp -al.

## Comandos Actualizados:

```
_# Eliminar la copia más antigua_  
rm -rf backup.3  
_# Mover las copias existentes una posición hacia atrás_  
mv backup.2 backup.3  
mv backup.1 backup.2  
mv backup.0 backup.1  
  
_# Sincronizar con `rsync` utilizando `--link-dest`_  
rsync -a --delete --link-dest=../backup.1 source_directory/ /backup.0/
```

• **--link-dest=../backup.1:** Indica a rsync que use backup.1 como referencia para enlaces duros. Los archivos que no han cambiado se enlazan al back up anterior, optimizando el uso de espacio y preservando permisos y propietarios.

## Script para back ups Snapshot

A continuación, se presenta un script básico para realizar back ups snapshot utilizando rsync con la opción --link-dest. Este script debe ser adaptado según las necesidades del sistema.

```
#!/bin/bash  
_# Sección: Ubicaciones de Archivos_  
SOURCE_DIR=/root  
DEST_DIR=/backup/rsync-backup  
EXCLUDES=/backup/excludes.txt  
BSERVER=localhost  
BACKUP_DATE=$(date +%Y%m%d%H%M%S)  
  
_# Opciones para rsync_  
OPTS="--ignore-errors --delete-excluded --exclude-from=$EXCLUDES \  
--delete --backup --backup-dir=$DEST_DIR/backups/$BACKUP_DATE -av"
```

```
_# Transferencia real usando `--link-dest`_
rsync -a --delete --link-dest=../backup.1 $SOURCE_DIR/ $DEST_DIR/backup.0/
```

- SOURCE\_DIR: Directorio que se respalda (/root).
- DEST\_DIR: Directorio donde se almacenan los back ups (/backup/rsync-backup).
- EXCLUDES: Archivo que contiene la lista de archivos/directorios a excluir.
- BSERVER: Máquina de back up (en este caso, localhost).
- BACKUP\_DATE: Fecha y hora actual para identificar el back up.

• **Opciones de rsync:**

- --ignore-errors: Ignora errores durante la transferencia.
- --delete-excluded: Elimina archivos excluidos del destino.
- --exclude-from: Especifica el archivo de exclusiones.
- --delete: Elimina archivos en el destino que ya no existen en la fuente.
- --backup: Realiza back ups de los archivos que se van a sobrescribir o eliminar.
- --backup-dir: Directorio donde se almacenarán los archivos respaldados.
- -a: Modo archivo.
- -v: Modo verbose.
- --link-dest=../backup.1: Utiliza backup.1 como referencia para enlaces duros.

## Respuestas a las Preguntas de 5.6.5

1. **¿Cómo es el comando rsync en el script para hacer las copias snapshot?**

El comando rsync en el script utiliza la opción --link-dest para crear enlaces duros con el back up anterior, optimizando el uso de espacio y preservando permisos y propietarios.

```
rsync -a --delete --link-dest=../backup.1 /root/ /backup/rsync-backup/backup.0/
```

2. **¿Qué ocurre con el back up después de la ejecución del script?**

Cada vez que se ejecuta el script:

- La copia más antigua (backup.3) se elimina.
- Las copias existentes se mueven una posición hacia atrás (backup.2 a backup.3, backup.1 a backup.2, backup.0 a backup.1).
- Se crea una nueva copia (backup.0) que parece una copia completa pero utiliza enlaces duros para compartir archivos inalterados con backup.1.

Esto permite tener cuatro copias de back up que parecen completas pero que ocupan espacio equivalente a la suma del directorio fuente más los cambios en los últimos tres días.

3. **¿Cuál es el tamaño del directorio /backup.0 y de otros directorios de back up?**

• /backup.0: Tamaño equivalente al directorio fuente actual más los cambios recientes.

• **backup.1, backup.2, backup.3:** Tamaño mínimo ya que comparten archivos inalterados mediante enlaces duros. Solo aumentan en tamaño si hay nuevos cambios en los archivos.

4. **¿Dónde está la información más reciente después de una restauración?**

La información más reciente se encuentra en el directorio backup.0, ya que es la copia sincronizada más reciente del directorio fuente.

5. **¿Qué comando se puede usar para recuperar los datos?**

Para restaurar los datos desde el back up más reciente (backup.0), se puede utilizar el comando rsync de la siguiente manera:

```
rsync -a --delete /backup/rsync-backup/backup.0/ /root/
```

- -a: Modo archivo, preserva permisos, propietarios, grupos, etc.
- --delete: Elimina archivos en el destino que no existen en el origen.
- /backup/rsync-backup/backup.0/: Directorio de back up más reciente.
- /root/: Directorio de destino donde se restaurarán los archivos.

Ésta práctica consiste en inciarnos al cloud público (AWS). Ésta práctica se divide en tres sub apartados:

## Poner en marcha un sitio de Wordpress en la nube

### 1. Crear una base de datos MySql con Amazon RDS

a. Open the [AWS Management Console](#). When the screen loads, enter *RDS* in the search bar, then select **RDS** to open the service console.

The screenshot shows the AWS Management Console search results for 'RDS'. The search bar at the top contains 'RDS'. On the left, there's a sidebar with various service links. The main area shows the 'Services' section with the 'RDS' card highlighted by a red box. The RDS card includes a star icon, the text 'Managed Relational Database Service', and a 'Top features' section with links to 'Dashboard', 'Databases', 'Query Editor', 'Performance Insights', and 'Schemas'. Below this is another service card for 'AWS Glue DataBrew' and then cards for 'Kinesis' and 'Amazon GameSparks'. At the bottom, there are sections for 'Features' and 'Dashboards'.

b. Choose the **Create database** button to get started.

The screenshot shows the Amazon RDS Dashboard. On the left, there's a sidebar with links like 'Dashboard', 'Databases', 'Query Editor', etc. The main area has a heading 'Resources' and a 'Create database' button highlighted by a red arrow. Below it, there's a list of resources: DB Instances (0/40), Allocated storage (0 TB/100 TB), Increase DB instances limit, DB Clusters (0/40), Reserved instances (0/40), Snapshots (8), and a manual section for DB Cluster (0/100) and DB Instance (0/100). To the right, there's a 'Recommended for you' sidebar with links to 'Amazon RDS Backup and Restore using AWS Backup', 'Time-Series Tables in PostgreSQL', 'Migrate SSRS to RDS for SQL Server', and 'Build RDS Operational Tasks'.

c. The first step is to choose the database engine you want to use. Amazon RDS supports six different engines, from popular open-source options like MySQL and PostgreSQL, to commercial options like Oracle and Microsoft SQL Server, to a cloud-native option called Amazon Aurora that was custom-built to take advantage of the cloud.

WordPress uses MySQL, so select **Standard create** for the database creation method and choose the **MySQL** engine.

WordPress uses MySQL, so select **Standard create** for the database creation method and choose the **MySQL** engine.

RDS > Create database

## Create database

**Choose a database creation method** [Info](#)

**Standard create**  
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

**Easy create**  
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

**Engine options**

**Engine type** [Info](#)

Amazon Aurora 

MySQL 

MariaDB 

PostgreSQL 

Oracle 

Microsoft SQL Server 

d. In the **Templates** section of the creation wizard, there is an option to only show options that are available in the [AWS Free Tier](#). Select this option to complete the learning in this guide without incurring costs.

In a production setup, you may want to use features of Amazon RDS that are outside the Free Tier. These include:

- A larger database instance class, for improved performance
- [Multi-AZ deployments](#), for automatic failover and recovery in the event of an infrastructure issue
- [Provisioned IOPS for disk storage](#), for faster I/O performance

aws | Services |  [Option+S] | N. Virginia ▾

MySQL 8.0.28

**Templates**  
Choose a sample template to meet your use case.

**Production**  
Use defaults for high availability and fast, consistent performance.

**Dev/Test**  
This instance is intended for development use outside of a production environment.

**Free tier**  
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. [Info](#)

**Availability and durability**

**Deployment options** [Info](#)  
The deployment options below are limited to those supported by the engine you selected above.

**Multi-AZ DB Cluster - new**  
Creates a DB cluster with a primary DB instance and two readable standby DB instances, with each DB instance in a different Availability Zone (AZ). Provides high availability, data redundancy and increases capacity to serve read workloads.

**Multi-AZ DB instance (not supported for Multi-AZ DB cluster snapshot)**  
Creates a primary DB instance and a standby DB instance in a different AZ. Provides high availability and data redundancy, but the standby DB instance doesn't support connections for read workloads.

**Single DB instance (not supported for Multi-AZ DB cluster snapshot)**  
Creates a single DB instance with no standby DB instances.



e. Next, you will specify the authentication settings for your MySQL deployment. These include the database name and the master username and password.

In the **Settings** section, enter **wordpress** as your **DB instance identifier**. Then specify the master username and password for your database. Choose a strong, secure password to help protect your database. Store the username and password for safekeeping as you will need it in a later module.

The screenshot shows the AWS RDS 'Settings' configuration page. The 'DB instance identifier' field contains 'wordpress'. The 'Master username' field contains 'admin'. The 'Master password' and 'Confirm password' fields both contain masked text (\*\*\*\*\*). Red boxes highlight the 'wordpress', 'admin', and both password fields.

DB instance identifier [Info](#)  
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.  
**wordpress**

Master username [Info](#)  
Type a login ID for the master user of your DB instance.  
**admin**

Master password [Info](#)  
\*\*\*\*\*  
Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

Confirm password [Info](#)  
\*\*\*\*\*

f. After setting your username and password, you can select key details about your MySQL deployment. This includes the instance configuration and storage details.

The default settings will work for this guide. You will use a small instance class that is suitable for testing or small-scale applications, and it fits within the AWS Free Tier. If you don't want to use the AWS Free Tier, you could set a larger instance class or alter the storage configuration options.

## Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

### DB instance class [Info](#)

- Standard classes (includes m classes)
- Memory optimized classes (includes r and x classes)
- Burstable classes (includes t classes)**

**db.t3.micro**  
2 vCPUs 1 GiB RAM Network: 2,085 Mbps

**Include previous generation classes**

## Storage

Storage type [Info](#)

General Purpose SSD (gp2)

Baseline performance determined by volume size



Allocated storage

20

GiB

(Minimum: 20 GiB. Maximum: 6,144 GiB) Higher allocated storage can improve IOPS performance.

Storage autoscaling [Info](#)

Provides dynamic scaling support for your database's storage based on your application's needs.

Enable storage autoscaling

Enabling this feature will allow the storage to increase after the specified threshold is exceeded.

Maximum storage threshold [Info](#)

Charges will apply when your database autoscales to the specified threshold

1000

GiB

Minimum: 22 GiB. Maximum: 6,144 GiB

g. Next, you can configure connectivity and network settings. Amazon RDS instances must be created in an [Amazon VPC](#), which is a logically separate network where your provisioned resources will live.

VPCs are an advanced topic outside the scope of this guide. Fortunately, AWS has created a default VPC in each Region in your account. The default VPC is already selected for you, and you can launch your Amazon RDS instance in this VPC.

**Connectivity** [Info](#)

**Compute resource**  
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

**Don't connect to an EC2 compute resource**  
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

**Connect to an EC2 compute resource**  
Set up a connection to an EC2 compute resource for this database.

**Virtual private cloud (VPC)** [Info](#)  
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

Default VPC (vpc-16af336e) ▾  
Only VPCs with a corresponding DB subnet group are listed.

**After a database is created, you can't change its VPC.**

**DB Subnet group** [Info](#)  
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

default-vpc-16af336e ▾

**Public access** [Info](#)

**Yes**  
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

**No**  
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

**VPC security group (firewall)** [Info](#)  
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

**Choose existing**  
Choose existing VPC security groups

**Create new**  
Create new VPC security group

**Existing VPC security groups**  
Choose one or more options ▾  
default X

**Availability Zone** [Info](#)  
No preference ▾

**RDS Proxy**  
RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.

**Create an RDS Proxy** [Info](#)  
RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see [Amazon RDS Proxy pricing](#).

► Additional configuration

h. Finally, Amazon RDS provides a number of additional configuration options to customize your deployment. You need to make one change in this area. Select the **Additional configuration** line to expand the options. Set the **Initial database name** to **wordpress**. This will ensure Amazon RDS creates the database in your MySQL instance upon initialization. You will use this database name when connecting to your database.

## ► Additional configuration

Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

## ▼ Additional configuration

Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

### Database options

Initial database name [Info](#)

wordpress 

If you do not specify a database name, Amazon RDS does not create a database.

DB parameter group [Info](#)

default.mysql8.0 

Option group [Info](#)

default:mysql-8-0 

- i. Retain the default settings for the remainder of options. At the bottom of the creation wizard, AWS will show you estimated monthly costs for your Amazon RDS database. If you are still eligible for the Amazon RDS Free Tier, you will see a note that the database will be free to you for up to 12 months.

Choose the **Create database** button to create your database.

### Estimated monthly costs

The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro, db.t3.micro or db.t4g.micro Instance.
- 20 GB of General Purpose Storage (SSD).
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

[Learn more about AWS Free Tier.](#) 

When your free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the [Amazon RDS Pricing page](#). 

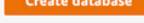
 You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

 **Create database**

- j. You should see a success notice indicating that your database is being created.

C Creating database **wordpress**  
Your database might take a few minutes to launch.  

RDS > Databases

Databases									
	DB identifier	Role	Engine	Region & AZ	Size	Status	CPU	Current act	Actions
 wordpress	Instance	MySQL Community	us-east-1d	db.t3.micro	 Creating	-			

In this module, you created a fully managed MySQL database using Amazon RDS. In the next module, you will create an Amazon EC2 instance for running your WordPress site.

## Crear una instancia EC2

### Implementation

#### Choosing an Amazon Machine Image



- a. To create your EC2 instance, go to [Amazon EC2 in the AWS Management Console](#). Choose the **Launch instance** button to open the instance creation wizard.

The screenshot shows the AWS Management Console EC2 Dashboard. On the left, there's a sidebar with various EC2-related options like Instances, Images, and Elastic Block Store. The main area is titled "Resources" and shows statistics for Instances (running: 0), Dedicated Hosts (0), Elastic IPs (0), Instances (0), Key pairs (1), Load balancers (0), Placement groups (0), Security groups (7), and Volumes (0). Below this, a callout box provides information about launching Microsoft SQL Server Always On availability groups. On the right, there are sections for "Launch instance" (with a prominent orange "Launch instance" button) and "Service health" (showing the service is operating normally).

- b. On the first page, enter *wordpress app* as your instance name.

The screenshot shows the "Launch an instance" wizard. It starts with a brief introduction to Amazon EC2. The next step, "Name and tags", has a "Name" field containing "wordpress app", which is highlighted with a red arrow. There's also a "Add additional tags" link.

- c. Next, choose an Amazon Machine Image (AMI). The AMI you choose will determine the base software that is installed on your new EC2 instance. This includes the operating system (Amazon Linux, Red Hat Enterprise Linux, Ubuntu, Microsoft Server, etc.), and the applications that are installed on the machine.

Many AMIs are general-purpose AMIs for running many different applications, but some are purpose-built for specific use cases, such as the Deep Learning AMI or various AWS Marketplace AMIs.

For this tutorial, choose the **Amazon Linux 2 AMI (HVM)** in the AMI selection view.

**NOTE:** Some commands in this tutorial will not work if Amazon Linux 2023 is selected.

## ▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

 *Search our full catalog including 1000s of application and OS images*

### Quick Start



Amazon Machine Image (AMI)



[Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community

#### Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type

ami-05fa00d4c63e32376 (64-bit (x86)) / ami-05f3141013eebdc12 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible



#### Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20220805.0 x86\_64 HVM gp2

#### Architecture

#### AMI ID

64-bit (x86)



ami-05fa00d4c63e32376

Verified provider

## Configuring an SSH key



You will see a details page on how to configure a key pair for your instance. You will use the key pair to SSH into your instance, which will give you the ability to run commands on your server.

- Open the **key pair (login)** section and choose **Create new key pair** for your instance.

▼ **Key pair (login)** Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

**Key pair name - required**

Select ▼ ↻ **Create new key pair**

- Give your key pair a name. Then choose the **Create key pair** button, which will download the .pem file to your machine. You will use this file in the next module.

### Create key pair

X

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn more ↗](#)

**Key pair name**

wordpress ←

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

**Key pair type**

RSA  
RSA encrypted private and public key pair

ED25519  
ED25519 encrypted private and public key pair (Not supported for Windows instances)

**Private key file format**

.pem  
For use with OpenSSH

.ppk  
For use with PuTTY

Cancel ➔ Create key pair

## Choosing an instance type



Scroll down to select an EC2 instance type. An instance type is a particular configuration of CPU, memory (RAM), storage, and network capacity.

AWS has a huge selection of [instance types](#) that cover many different workloads. Some are geared toward memory-intensive workloads, such as databases and caches, while others are aimed at compute-heavy workloads, such as image processing or video encoding.

Amazon EC2 allows you to run 750 hours per month of a t2.micro instance under the [AWS Free Tier](#). Select this option for this guide so that you won't incur any costs on your bill.

- a. Select the **t2.micro** instance.

▼ **Instance type** [Info](#)

Instance type

<b>t2.micro</b>	Free tier eligible	
Family: t2	1 vCPU	1 GiB Memory
On-Demand Linux pricing: 0.0116 USD per Hour		
On-Demand Windows pricing: 0.0162 USD per Hour		

[Compare instance types](#)

## Configuring an SSH key



You will see a details page on how to configure a key pair for your instance. You will use the key pair to SSH into your instance, which will give you the ability to run commands on your server.

- Open the **key pair (login)** section and choose **Create new key pair** for your instance.

▼ **Key pair (login)** Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

**Key pair name - required**

Select ▼ ↻ **Create new key pair**

- Give your key pair a name. Then choose the **Create key pair** button, which will download the .pem file to your machine. You will use this file in the next module.

### Create key pair

X

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn more ↗](#)

**Key pair name**

wordpress ←

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

**Key pair type**

RSA  
RSA encrypted private and public key pair

ED25519  
ED25519 encrypted private and public key pair (Not supported for Windows instances)

**Private key file format**

.pem  
For use with OpenSSH

.ppk  
For use with PuTTY

Cancel ➔ Create key pair

## Configuring a security group and launching your instance

You need to configure a security group before launching your instance. Security groups are networking rules that describe the kind of network traffic that is allowed to your EC2 instance. You want to allow two kinds of traffic to your instance:

- SSH traffic from your current IP address so you can use the SSH protocol to log in to your EC2 instance and configure WordPress
- HTTP traffic from all IP addresses so that users can view your WordPress site.

a. To configure this, select **Allow SSH traffic from My IP** and select **Allow HTTP traffic from the internet**.

▼ **Network settings** [Info](#) [Edit](#)

Network [Info](#)  
vpc-722d0e08

Subnet [Info](#)  
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)  
Enable

**Firewall (security groups)** [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group     Select existing security group

We'll create a new security group called '**launch-wizard-2**' with the following rules:

<input checked="" type="checkbox"/> Allow SSH traffic from Helps you connect to your instance	Anywhere 0.0.0.0/0
<input type="checkbox"/> Allow HTTPS traffic from the internet To set up an endpoint, for example when c	Anywhere 0.0.0.0/0
<input type="checkbox"/> Allow HTTP traffic from the internet To set up an endpoint, for example when c	Custom My IP 54.240.198.35/32

**⚠** Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. X

## ▼ Network settings [Info](#)

[Edit](#)

Network [Info](#)

vpc-722d0e08

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

### Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group

Select existing security group

We'll create a new security group called '**launch-wizard-2**' with the following rules:

Allow SSH traffic from

Helps you connect to your instance

My IP

54.240.198.35/32

Allow HTTPS traffic from the internet

To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet

To set up an endpoint, for example when creating a web server

**⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.**

- b. In the **Network settings** section, choose the **Edit** button. Scroll down to **Firewall (security groups)** and enter *wordpress* for the **Security group name**.

## ▼ Network settings Info

Edit 

### Network Info

vpc-722d0e08

### Subnet Info

No preference (Default subnet in any availability zone)

### Auto-assign public IP Info

Enable

### Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group

Select existing security group

We'll create a new security group called '**launch-wizard-2**' with the following rules:

#### Allow SSH traffic from

Helps you connect to your instance

My IP

54.240.198.35/32

#### Allow HTTPS traffic from the internet

To set up an endpoint, for example when creating a web server

#### Allow HTTP traffic from the internet

To set up an endpoint, for example when creating a web server

 Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. 

## ▼ Network settings Info

### VPC - required Info

vpc-722d0e08  
172.31.0.0/16

(default) ▾



### Subnet Info

No preference

▼



Create new subnet



### Auto-assign public IP Info

Enable

▼

### Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group

Select existing security group

### Security group name - required

wordpress 

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and \_-:/()#@[]+=&;!\$\*

### Description - required Info

wordpress security group 

### Inbound security groups rules

▶ Security group rule 1 (TCP, 22, 54.240.198.35/32)

Remove

▶ Security group rule 2 (TCP, 80, 0.0.0.0/0)

Remove

## ▼ Summary

### Number of instances Info

1

### Software Image (AMI)

Amazon Linux 2 Kernel 5.10 AMI... [read more](#)  
ami-05fa00d4c63e32376

### Virtual server type (instance type)

t2.micro

### Firewall (security group)

New security group

### Storage (volumes)

1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable)

instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet. 

Cancel

Launch instance

## Launch

It is now time to launch your EC2 instance.

a. Choose the **Launch instance** button to create your EC2 instance.

**Network settings** [Info](#)

VPC - required [Info](#)  
vpc-722d0e08 (default) [Subnet](#)  
172.31.0.0/16

Subnet [Info](#)  
No preference [Create new subnet](#)

Auto-assign public IP [Info](#)  
Enable

Firewall (security groups) [Info](#)  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.  
 Create security group  Select existing security group

Security group name - required  
wordpress

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and \_:-:/()#@[]+=;&;\$^

Description - required [Info](#)  
wordpress security group

Inbound security groups rules

► Security group rule 1 (TCP, 22, 54.240.198.35/32) [Remove](#)

► Security group rule 2 (TCP, 80, 0.0.0.0/0) [Remove](#)

**Summary**

Number of instances [Info](#)  
1

Software Image (AMI)  
Amazon Linux 2 Kernel 5.10 AMI...[read more](#)  
ami-05fa00d4c63e52376

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
New security group

Storage (volumes)  
1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet. [X](#)

[Cancel](#) [Launch instance](#)

EC2 > Instances > Launch an instance

**Success**  
Successfully initiated launch of instance ([i-04210b3030238a333](#))

[Launch log](#)

**Next Steps**

**Get notified of estimated charges**  
[Create billing alerts](#) to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier)

**How to connect to your instance**  
Your instance is launching and it might be a few minutes until it is in the running state, when it will be ready for you to use  
Click [View Instances](#) to monitor your instance's status. Once your instance is in the 'running' state, you can connect to it from the Instances screen. Find out [how to connect to your instance](#)

[View more resources to get you started](#)

[View all instances](#)

You have successfully launched your EC2 instance. In the next module, we will configure your Amazon RDS database to work with your EC2 instance.

**Configurar tu base de datos RDS**

## Allow your EC2 instance to access your Amazon RDS database

First, you will modify your Amazon RDS database to allow network access from your EC2 instance.

In the previous module, you created security group rules to allow SSH and HTTP traffic to your WordPress EC2 instance. The same principle applies here. This time, you want to allow certain traffic from your EC2 instance into your Amazon RDS database.

- To configure this, go to the [Amazon RDS databases](#) page in the AWS console. Choose the MySQL database you created in the earlier module in this guide.

The screenshot shows the 'Databases' section of the AWS RDS console. A red arrow points to the 'DB identifier' column for the 'wordpress' database row. The database details shown are: Instance: MySQL Community, Region & AZ: us-east-1d, Size: db.t3.micro, Status: Available, CPU: 2.91%.

- Scroll to the **Connectivity & security** tab in the display and choose the security group listed in **VPC security groups**. The console will take you to the security group configured for your database.

The screenshot shows the 'Connectivity & security' tab for the 'wordpress' database. It displays the following details:

Endpoint & port	Networking	Security
Endpoint wordpress.cchtmzbwos.us-east-1.rds.amazonaws.com	Availability Zone us-east-1d	VPC security groups default (sg-86e6abd0) Active
Port 3306	VPC vpc-722d0e08	Publicly accessible No
	Subnet group default-vpc-722d0e08	Certificate authority rds-ca-2019
	Subnets subnet-7ce5be42 subnet-7730973a subnet-f92fe6f7 subnet-d60a1db1 subnet-7b5c4155 subnet-0e849452	Certificate authority date August 22, 2024, 10:08 (UTC-07:00)
	Network type IPv4	

- Select the **Inbound rules** tab, then choose the **Edit inbound rules** button to change the rules for your security group.

The screenshot shows the 'Inbound rules' tab for the 'sg-86e6abd0 - default' security group. A red arrow points to the 'Edit inbound rules' button in the top right corner of the table header. The table lists one rule:

Name	Security group ID	Security group name	VPC ID	Description	Owner	Inb
-	sg-86e6abd0	default	vpc-722d0e08	default VPC security gr...	921645009304	1 P

Below the table, there is a note: 'You can now check network connectivity with Reachability Analyzer' and a 'Run Reachability Analyzer' button. At the bottom, there is another 'Edit inbound rules' button in a red box.

<input checked="" type="checkbox"/>	-	sgr-04b0f69dfb05a106e	-	All traffic	All	All	sg-86
-------------------------------------	---	-----------------------	---	-------------	-----	-----	-------

d. The default security group has a rule that allows all inbound traffic from other instances in the default security group. However, since your WordPress EC2 instance is not in that security group, it will not have access to the Amazon RDS database.

Change the **Type** property to **MYSQL/Aurora**, which will update the **Protocol** and **Port range** to the proper values. Then, remove the current security group value configured for the **Source**.

EC2 > Security Groups > sg-86e6abd0 - default > Edit inbound rules

### Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
sgr-04b0f69dfb05a106e	MYSQL/Aurora 	TCP	3306	Custom 	<input type="text" value="sg-86e6abd0"/> 

[Add rule](#) [Cancel](#) [Preview changes](#) [Save rules](#)

EC2 > Security Groups > sg-86e6abd0 - default > Edit inbound rules

### Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
sgr-04b0f69dfb05a106e	MYSQL/Aurora 	TCP	3306	Custom 	<input type="text" value="sg-86e6abd0"/> 

[Add rule](#) [Cancel](#) [Preview changes](#) [Save rules](#)

e. For **Source**, enter **wordpress**. The console will show the available security groups that are configured. Choose the **wordpress** security group that you used for your EC2 instance.

EC2 > Security Groups > sg-86e6abd0 - default > Edit inbound rules

### Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
sgr-04b0f69dfb05a106e	MYSQL/Aurora 	TCP	3306	Custom 	<input type="text" value="wordpress"/>  Security Groups <b>wordpress   sg-034f53b528a6795ea</b> 

[Add rule](#) [Cancel](#) [Preview changes](#) [Save rules](#)

f. After you choose the **wordpress** security group, the security group ID will be filled in. This rule will allow MySQL access to any EC2 instance with that security group configured.

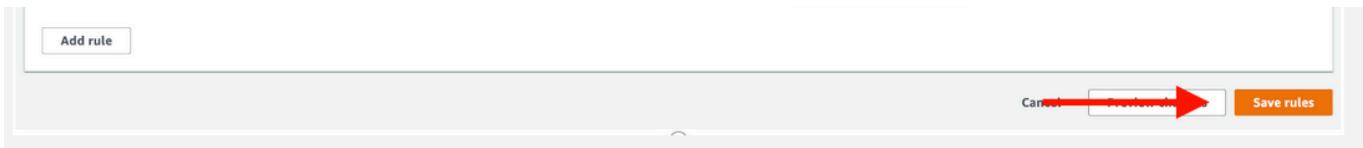
When you're finished, choose the **Save rules** button to save your changes.

EC2 > Security Groups > sg-86e6abd0 - default > Edit inbound rules

### Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
sgr-04b0f69dfb05a106e	MYSQL/Aurora 	TCP	3306	Custom 	<input type="text" value="sg-034f53b528a6795ea"/> 



## SSH into your EC2 instance

Now that your EC2 instance has access to your Amazon RDS database, you will use SSH to connect to your EC2 instance and run some configuration commands.

- Go to the [EC2 instances page](#) in the console. You should see the EC2 instance you created for the WordPress installation. Select it and you will see the Public IPv4 address and the Public IPv4 DNS in the instance description.

The screenshot shows the AWS EC2 Instances page. At the top, there is a search bar and several navigation buttons: 'Connect', 'Instance state', 'Actions', 'Launch instances', and a dropdown menu. Below the header, there is a table with columns: 'Name', 'Instance ID', 'Instance state', 'Instance type', 'Status check', 'Alarm status', and 'Availability'. A single row is selected, showing 'wordpress app' as the name, 'i-04210b3030238a333' as the Instance ID, 'Running' as the state, 't2.micro' as the type, '2/2 checks passed' as the status, 'No alarms' as the alarm status, and 'us-east-1b' as the availability zone. Below the table, the instance details are shown for 'i-04210b3030238a333 (wordpress app)'. The 'Details' tab is selected, followed by 'Security', 'Networking', 'Storage', 'Status checks', 'Monitoring', and 'Tags'. Under 'Instance summary', there are sections for 'Public IPv4 address' (52.87.182.77), 'Private IPv4 addresses' (172.31.87.110), 'Public IPv4 DNS' (ec2-52-87-182-77.compute-1.amazonaws.com), 'Instance state' (Running), 'Private IP DNS name (IPv4 only)' (ip-172-31-87-110.ec2.internal), 'Instance type' (t2.micro), 'Elastic IP addresses' (none), 'VPC ID' (vpc-722d0e08), and 'AWS Compute Optimizer finding' (No recommendations available for this instance). Red arrows point from the text 'Public IPv4 address' and 'Public IPv4 DNS' to their respective values in the instance details.

- Previously, you downloaded the .pem file for the key pair of your instance. Locate that file now. It will likely be in a Downloads folder on your desktop.

For Mac or Linux users:

Open a terminal window. If you are on a Mac, you can use the default Terminal program that is installed, or you can use your own terminal.

In your terminal, run the following commands to use SSH to connect to your instance. Replace the "<path/to/pem/file>" with the path to your file, e.g., "~/Downloads/wordpress.pem", and the "<publicIpAddress>" with the public IP address for your EC2 instance.

```
1 chmod 400 <path/to/pem/file>
2 ssh -i <path/to/pem/file> ec2-user@<public_IP_DNSAddress>
```

Copy

You should see the following in your terminal to indicate that you connected successfully:

```
user@host:~$ ssh -i /path/to/your.pem ec2-user@52.87.182.77
Amazon Linux 2 AMI
```

```
https://aws.amazon.com/amazon-linux-2/
8 package(s) needed for security, out of 13 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-87-142 ~]$
```

In your terminal, enter the following command to set an environment variable for your MySQL host. Be sure to replace "<your-endpoint>" with the hostname of your RDS instance.

```
1 export MYSQL_HOST=<your-endpoint>
```

 Copy

Next, run the following command in your terminal to connect to your MySQL database. Replace "<user>" and "<password>" with the master username and password you configured when creating your Amazon RDS database.

```
1 mysql --user=<user> --password=<password> wordpress
```

 Copy

Finally, create a database user for your WordPress application and give the user permission to access the **wordpress** database.

Run the following commands in your terminal:

```
1 CREATE USER 'wordpress' IDENTIFIED BY 'wordpress-pass';
2 GRANT ALL PRIVILEGES ON wordpress.* TO wordpress;
3 FLUSH PRIVILEGES;
4 Exit
```

 Copy

As a best practice, you should use a better password than *wordpress-pass* to secure your database.

Write down both the username and password that you configure, as they will be needed in the next module when setting up your WordPress installation.

In this module, you learned how to configure network and password security for your Amazon RDS database. Your EC2 instance now has network access to your Amazon RDS database. Further, you created a database user to be used by your WordPress application.

In the next module, you will configure your EC2 instance to run the WordPress application.

## Configurar WordPress en EC2

# Implementation

## Installing the Apache web server

To run WordPress, you need to run a web server on your EC2 instance. The open source [Apache web server](#) is the most popular web server used with WordPress.

To install Apache on your EC2 instance, run the following command in your terminal:

```
1 sudo yum install -y httpd
```

[Copy](#)

You should see some terminal output of the necessary packages being installed.

To start the Apache web server, run the following command in your terminal:

```
1 sudo service httpd start
```

[Copy](#)

You can see that your Apache web server is working and that your security groups are configured correctly by visiting the public DNS of your EC2 instance in your browser.

Go to the [EC2 Instances page](#) and find your instance. In the Description below, find the **Public IPv4 DNS** of your instance.

The screenshot shows the AWS Management Console interface for EC2 Instances. It displays a single instance named "wordpress app" with the ID "i-04210b3030238a333". The instance is listed as "Running" with an "t2.micro" instance type, "2/2 checks passed" status, and "No alarms" for "Alarm status". The "Availability" zone is "us-east-1b". There are buttons for "Connect", "Actions", and "Launch Instances". A search bar at the top is empty.

The screenshot shows the detailed view for the instance "i-04210b3030238a333". Under the "Details" tab, the "Instance summary" section is expanded, showing the following information:

Attribute	Value
Instance ID	i-04210b3030238a333 (wordpress app)
IPv6 address	-
Hostname type	IP name: ip-172-31-87-110.ec2.internal
Public IPv4 address	52.87.182.77
Instance state	Running
Private IP DNS name (IPv4 only)	ip-172-31-87-110.ec2.internal
Private IPv4 addresses	172.31.87.110
Public IPv4 DNS	ec2-52-87-182-77.compute-1.amazonaws.com

A red arrow points from the "Public IPv4 DNS" value to the "Test Page" screenshot below.

The screenshot shows a web browser window with the URL "Not Secure | ec2-52-87-182-77.compute-1.amazonaws.com". The page title is "Test Page". The content of the page is:

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

### If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

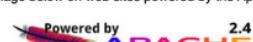
If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting [www.example.com](http://www.example.com), you should send e-mail to "[webmaster@example.com](mailto:webmaster@example.com)".

### If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the image below on web sites powered by the Apache HTTP Server:



## Download and configure WordPress

In this step, you will download the WordPress software and set up the configuration.

First, download and uncompress the software by running the following commands in your terminal:

```
1 wget https://wordpress.org/latest.tar.gz
2 tar -xzf latest.tar.gz
```

 Copy

If you run `ls` to view the contents of your directory, you will see a tar file and a directory called `wordpress` with the uncompressed contents.

```
1 $ ls
```

 Copy

The output should look like the following:

```
[ec2-user@~]$ ls
latest.tar.gz wordpress
```

Change the directory to the `wordpress` directory and create a copy of the default config file using the following commands:

```
1 cd wordpress
2 cp wp-config-sample.php wp-config.php
```

 Copy

Then, open the `wp-config.php` file using the [nano](#) editor by running the following command.

```
1 nano wp-config.php
```

 Copy

You need to edit two areas of configuration.

First, edit the database configuration by changing the following lines:

```
1 // ** MySQL settings - You can get this info from your web host ** //
2 /** The name of the database for WordPress */
3 define( 'DB_NAME', 'database_name_here' );
4
5 /** MySQL database username */
6 define( 'DB_USER', 'username_here' );
7
8 /** MySQL database password */
9 define( 'DB_PASSWORD', 'password_here' );
10
11 /** MySQL hostname */
12 define( 'DB_HOST', 'localhost' );
```

 Copy

The values should be:

- **DB\_NAME**: "wordpress"
- **DB\_USER**: The name of the user you created in the database in the previous module
- **DB\_PASSWORD**: The password for the user you created in the previous module
- **DB\_HOST**: The hostname of the database that you found in the previous module

The second configuration section you need to configure is the **Authentication Unique Keys and Salts**. It looks as follows in the configuration file:

OJO: DB\_HOST se refiere al Endpoint:

# Connectivity & security

## Endpoint & port

### Endpoint

[wordpress.cu6ytj2mieha.us-east-1.rds.amazonaws.com](#)

```
1  /**#@+
2  * Authentication Unique Keys and Salts.
3  *
4  * Change these to different unique phrases!
5  * You can generate these using the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key se
6  * You can change these at any point in time to invalidate all existing cookies. This will force all users to have tc
7  *
8  * @since 2.6.0
9  */
10 define( 'AUTH_KEY',         'put your unique phrase here' );
11 define( 'SECURE_AUTH_KEY',  'put your unique phrase here' );
12 define( 'LOGGED_IN_KEY',    'put your unique phrase here' );
13 define( 'NONCE_KEY',        'put your unique phrase here' );
14 define( 'AUTH_SALT',        'put your unique phrase here' );
15 define( 'SECURE_AUTH_SALT', 'put your unique phrase here' );
16 define( 'LOGGED_IN_SALT',   'put your unique phrase here' );
17 define( 'NONCE_SALT',       'put your unique phrase here' );
```

 Copy

Go to [this link](#) to generate values for this configuration section. You can replace the entire content in that section with the content from the link.

You can save and exit from nano by entering **CTRL+O [ENTER]** followed by **CTRL+X**.

With the configuration updated, you are almost ready to deploy your WordPress site. In the next step, you will make your WordPress site live.

## Deploying WordPress

In this step, you will make your Apache web server handle requests for WordPress.

First, install the application dependencies you need for WordPress. In your terminal, run the following command.

```
1 sudo amazon-linux-extras install -y mariadb10.5 php8.2
```

 Copy

Second, change to the proper directory by running the following command:

```
1 cd /home/ec2-user
```

 Copy

Then, copy your WordPress application files into the `/var/www/html` directory used by Apache.

```
1 sudo cp -r wordpress/* /var/www/html/
```

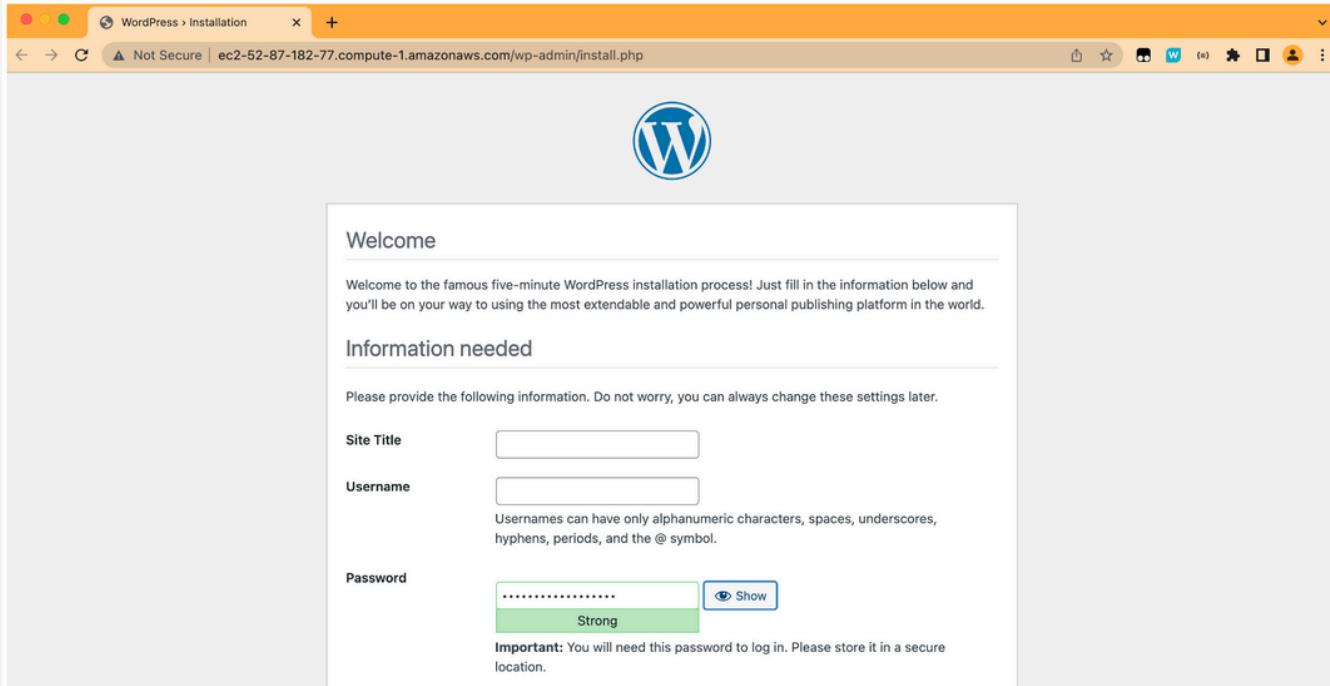
 Copy

Finally, restart the Apache web server to pick up the changes.

```
1 sudo service httpd restart
```

 Copy

You should see the WordPress welcome page and the five-minute installation process.



That's it. You have a live, publicly accessible WordPress installation using a fully managed MySQL database on Amazon RDS.

In the next module, you will clean up your resources and see some next steps for your WordPress installation.

## Crear una página web simplona sin DataBase

Pregunta del 23-24 q2

1. Ves a la web de AWS awsacademy.instructure.com
2. Inicia sesión y lánchela el "Learner Lab"

3. Haz una nueva instancia de EC2 con Amazon linux
4. Selecciona la instance type que te diga el enunciado.
5. Todo default
6. Llámale como quieras
7. Crea un nuevo security group, y añade estas Inbound Rules:
  - Permitir tráfico HTTP (80), source: la que te diga el enunciado
  - Permitir tráfico SSH (22), source: tu IP
8. Crea tu key pair y descárgalo
9. Dale a launch
10. Abre la terminal de Linux del ordenador y haz:

```
chmod 400 Descarregues/clau-examen.pem
ssh -i Descarregues/clau-examen.pem ec2-user@<ip-Publica-ec2>
```

*La IP pública la puedes encontrar haciendo click sobre esta instancia en 2nda col*

11. Deberías estar en la terminal de la instancia. Ejecuta las siguientes comandadas:

```
# Actualiza paquetes
sudo yum update -y
# Instala apache
sudo yum install -y httpd
# Inicia apache
sudo systemctl start httpd
sudo systemctl enable httpd
```

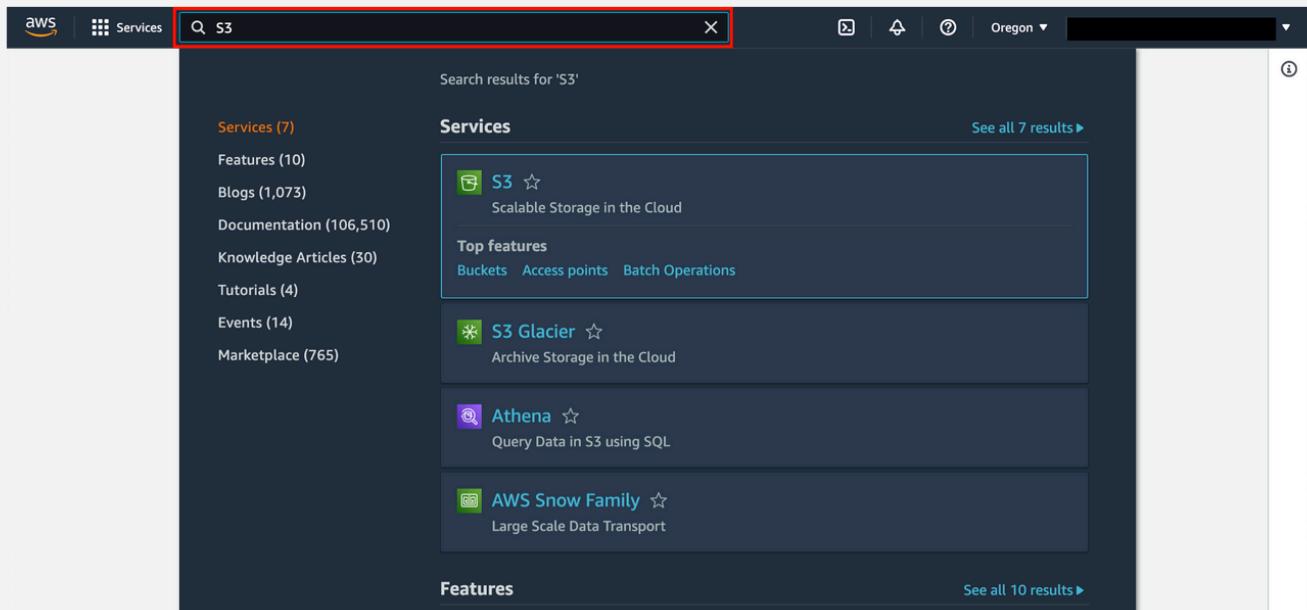
12. Con el navegador, escribe la IP pública en la barra y mira si está activo.
13. Si está activo crea el archivo html y pega el código que te dan en el enunciado.

```
sudo nano /var/www/index.html
```

## Amazon S3 (Amazon Simple Storage Service)\*

### Acceso a la consola de Amazon S3

Haga clic en la página principal de la Consola de administración de AWS para abrirla en una nueva ventana del navegador para poder tener abierta esta guía paso a paso. Cuando la pantalla se cargue, ingrese su nombre de usuario y contraseña para comenzar. A continuación, escriba S3 en la barra de búsqueda y seleccione S3 para abrir la consola.



## Cree un bucket de S3

En este paso, creará un *bucket* de Amazon S3. Un bucket es el contenedor en el que almacena los archivos.

- En el panel de S3, haga clic en **Crear bucket**.

Si es la primera vez que crea un bucket, verá una pantalla similar a la imagen que se muestra aquí.

Si ya creó buckets de S3 con anterioridad, el panel de S3 enumerará todos los buckets creados.

The screenshot shows the AWS S3 service dashboard. On the left, there's a sidebar with options like 'Buckets', 'Access Points', 'Object Lambda Access Points', etc. The main area is titled 'Amazon S3 > Buckets'. It features an 'Account snapshot' section with metrics: Total storage (8.3 KB), Object count (1), and Avg. object size (8.3 KB). A note says you can enable advanced metrics in the 'default-account-dashboard' configuration. Below this is a table titled 'Buckets (1) Info' with one item listed:

Name	AWS Region	Access	Creation date
cf-templates-1gq6gjg8me6n9-us-west-2	US West (Oregon) us-west-2	Objects can be public	May 10, 2022, 22:57:18 (UTC-07:00)

A red box highlights the 'Create bucket' button at the top of the table.

- Introduzca el nombre del bucket. Los nombres de los buckets no pueden repetirse en Amazon S3. Para esta guía, hemos utilizado *mysuperawsbucket*, pero debe elegir un nombre que sea pertinente y único para su caso. También existen otras [restricciones acerca de los nombres de los buckets de S3](#). Una vez que haya seleccionado un nombre, seleccione la región en la que creará el bucket.

The screenshot shows the 'Create bucket' page in the AWS Management Console. The 'Bucket name' field contains 'mysuperawsbucket', and the 'AWS Region' dropdown is set to 'US West (Oregon) us-west-2'. A note at the bottom left says 'Copy settings from existing bucket - optional'.

c. Tiene la posibilidad de definir la configuración de los permisos para su bucket de S3. Deje los valores predeterminados y seleccione **Next (Siguiente)**.

The screenshot shows the continuation of the 'Create bucket' process. Under 'Object Ownership', 'ACLs disabled (recommended)' is selected. Under 'Block Public Access settings for this bucket', 'Block all public access' is checked, along with several sub-options related to ACLs and policies.

**Object Ownership**

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

**ACLs disabled (recommended)**  
All objects in this bucket are owned by this account.  
Access to this bucket and its objects is specified using only policies.

**ACLs enabled**  
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

**Object Ownership**  
Bucket owner enforced

**Block Public Access settings for this bucket**

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

**Block all public access**  
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through new access control lists (ACLs)**  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**  
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

d. Existen muchas opciones útiles para el bucket de S3, como el [control de versiones](#), las [etiquetas](#), el [cifrado predeterminado](#) y el [bloqueo de objetos](#). No las activaremos en este tutorial.

Seleccione **Create bucket** (Crear bucket).

**Bucket Versioning**  
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

**Bucket Versioning**  
 Disable  
 Enable

**Tags (0) - optional**  
Track storage cost or other criteria by tagging your bucket. [Learn more](#)

No tags associated with this bucket.  
[Add tag](#)

**Default encryption**  
Automatically encrypt new objects stored in this bucket. [Learn more](#)

**Server-side encryption**  
 Disable  
 Enable

**▼ Advanced settings**

**Object Lock**  
Store objects using a write-once-read-many (WORM) model to help you prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. [Learn more](#)

Disable  
 Enable  
Permanently allows objects in this bucket to be locked. Additional Object Lock configuration is required in bucket details after bucket creation to protect objects in this bucket from being deleted or overwritten.

**Object Lock works only in versioned buckets. Enabling Object Lock automatically enables Bucket Versioning.**

**After creating the bucket you can upload files and folders to the bucket, and configure additional bucket settings.**

[Cancel](#) [Create bucket](#)

## Carga de un archivo



En este paso, cargará un archivo al nuevo bucket de Amazon S3.

- a. Verá el nuevo bucket en la consola de S3. Haga clic en el nombre del bucket para ir hasta este.

The screenshot shows the AWS S3 Buckets page. At the top, a green banner indicates that a bucket named "mysuperawsbucket" has been successfully created. Below the banner, the "Account snapshot" section provides an overview of storage usage. The main area displays a list of buckets, with "mysuperawsbucket" being the second item in the list. This item is highlighted with a red box. The table columns include Name, AWS Region, Access, and Creation date. The "mysuperawsbucket" row shows it was created in US West (Oregon) region, has "Bucket and objects not public" access, and was created on May 12, 2022, at 17:34:04 UTC-07:00.

Name	AWS Region	Access	Creation date
cf-templates-1gq6gjg8me6n9-us-west-2	US West (Oregon) us-west-2	Objects can be public	May 10, 2022, 22:57:18 (UTC-07:00)
mysuperawsbucket	US West (Oregon) us-west-2	Bucket and objects not public	May 12, 2022, 17:34:04 (UTC-07:00)

- b. Se encuentra en la página de inicio del bucket. Seleccione Cargar.

The screenshot shows the AWS S3 Objects page for the "mysuperawsbucket" bucket. The top navigation bar shows the path "Amazon S3 > Buckets > mysuperawsbucket". The main content area is titled "mysuperawsbucket" and includes tabs for Objects, Properties, Permissions, Metrics, Management, and Access Points. The "Objects" tab is selected. A table below shows zero objects. The "Actions" menu is expanded, and the "Upload" button is highlighted with a red box. The "Upload" button is located at the bottom of the table's header row.

c. Para seleccionar y cargar un archivo, haga clic en **Add files** (Aregar archivos) o **Add folder** (Aregar carpeta) y seleccione los archivos de muestra que deseé almacenar o arrastre y suelte un archivo en el cuadro de carga. Los archivos se mostrarán una vez seleccionados los archivos que quiere cargar.

The screenshot shows the AWS S3 'Upload' interface. At the top, there's a navigation bar with 'aws', 'Services', a search bar, and 'Global'. Below it, the path 'Amazon S3 > Buckets > mysuperawebucket > Upload' is shown. The main area has a title 'Upload' with a 'Info' link. A large red box highlights a dashed blue border area where files can be dragged and dropped. Below this, a table lists 'Files and folders (1 Total, 66.5 KB)'. The table has columns: Name, Folder, Type, and Size. A single row is visible: 'kittens.png' (image/png, 66.5 KB). To the right of the table are 'Remove', 'Add files' (which is highlighted with a red box), and 'Add folder' buttons. A search bar labeled 'Find by name' is also present.

d. Puede revisar los detalles y permisos del destino. Para este tutorial, deje los valores predeterminados.

The screenshot shows the 'Destination' tab of the AWS S3 Bucket Properties. It includes sections for 'Destination' (set to 's3://mysuperawebucket'), 'Destination details' (Bucket Versioning is disabled), 'Default encryption' (disabled), and 'Object Lock' (disabled). A warning message at the bottom left encourages enabling Bucket Versioning. The 'Permissions' section notes that the bucket owner has enforced object ownership and provides a link to learn more. A note at the bottom states that the bucket has the 'bucket owner enforced' setting applied for object ownership.

e. Puede configurar propiedades como la clase de almacenamiento, el cifrado del lado del servidor, las sumas de comprobaciones adicionales, las etiquetas y los metadatos con el objeto. Deje los valores predeterminados y seleccione **Upload** (Cargar).

## ▼ Properties

Specify storage class, encryption settings, tags, and more.

### Storage class

Amazon S3 offers a range of storage classes designed for different use cases. [Learn more](#) or see [Amazon S3 pricing](#)

Storage class	Designed for	Availability Zones	Min storage duration	Max storage duration
<input checked="" type="radio"/> Standard	Frequently accessed data (more than once a month) with milliseconds access	≥ 3	-	-
<input type="radio"/> Intelligent-Tiering	Data with changing or unknown access patterns	≥ 3	-	-
<input type="radio"/> Standard-IA	Infrequently accessed data (once a month) with milliseconds access	≥ 3	30 days	1 year
<input type="radio"/> One Zone-IA	Recreatable, infrequently accessed data (once a month) stored in a single Availability Zone with milliseconds access	1	30 days	1 year
<input type="radio"/> Glacier Instant Retrieval	Long-lived archive data accessed once a quarter with instant retrieval in milliseconds	≥ 3	90 days	1 year
<input type="radio"/> Glacier Flexible Retrieval (formerly Glacier)	Long-lived archive data accessed once a year with retrieval of minutes to hours	≥ 3	90 days	-
<input type="radio"/> Glacier Deep Archive	Long-lived archive data accessed less than once a year with retrieval of hours	≥ 3	180 days	-
<input type="radio"/> Reduced redundancy	Noncritical, frequently accessed data with milliseconds access (not recommended as S3 Standard is more cost effective)	≥ 3	-	-

### Server-side encryption settings

Server-side encryption protects data at rest. [Learn more](#)

#### Server-side encryption

- Do not specify an encryption key
- Specify an encryption key

⚠ If your bucket policy requires encrypted uploads, you must specify an encryption key or your upload will fail.

ⓘ Since default encryption is disabled for this bucket, no encryption settings will be applied to the objects when storing them in Amazon S3.

**Additional checksums**

Checksum functions are used for additional data integrity verification of new objects. [Learn more](#)

**Additional checksums**

**Off**  
Amazon S3 will use a combination of MD5 checksums and Etags to verify data integrity.

**On**  
Specify a checksum function for additional data integrity validation.

**Tags - optional**

Track storage cost or other criteria by tagging your objects. [Learn more](#)

No tags associated with this resource.

[Add tag](#)

**Metadata - optional**

Metadata is optional information provided as a name-value (key-value) pair. [Learn more](#)

No metadata associated with this resource.

[Add metadata](#)

[Cancel](#) [Upload](#)

Verá el objeto en la pantalla de inicio del bucket.

Services  [Option+S] [Global ▾](#)

**Upload succeeded**  
View details below.

**Upload: status** [Close](#)

**Summary**

Destination	Succeeded	Failed
s3://mysuperawsbucket	1 file, 66.5 KB (100.00%)	0 files, 0 B (0%)

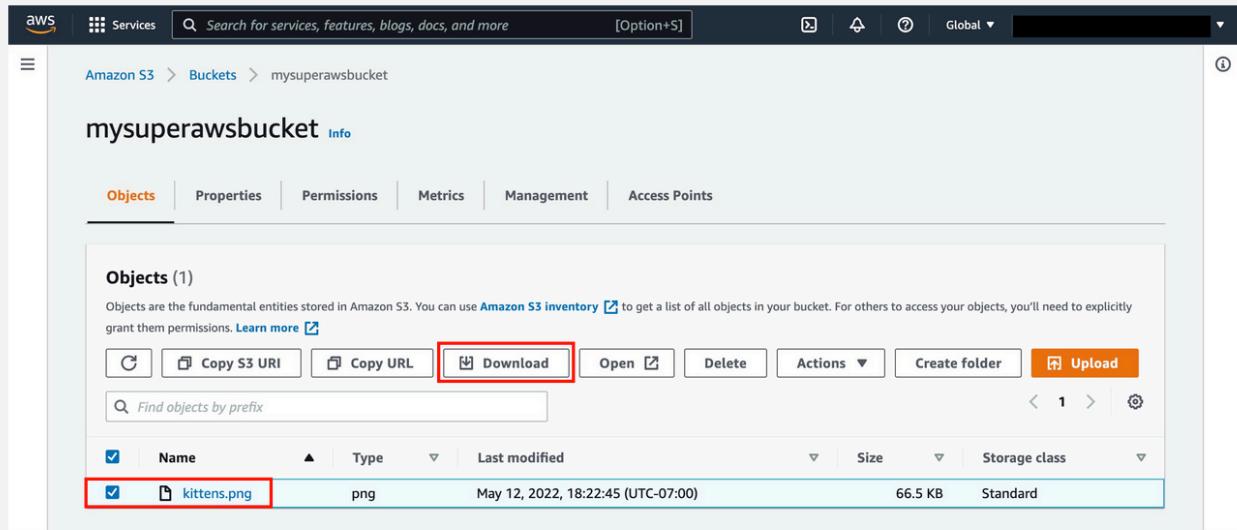
[Files and folders](#) [Configuration](#)

**Files and folders (1 Total, 66.5 KB)**

Name	Folder	Type	Size	Status	Error
kittens.png	-	image/png	66.5 KB	Succeeded	-

En este paso, descargará el archivo del bucket de Amazon S3.

- a. Seleccione la casilla de verificación ubicada al lado del archivo que desea descargar y a continuación seleccione **Descargar**.

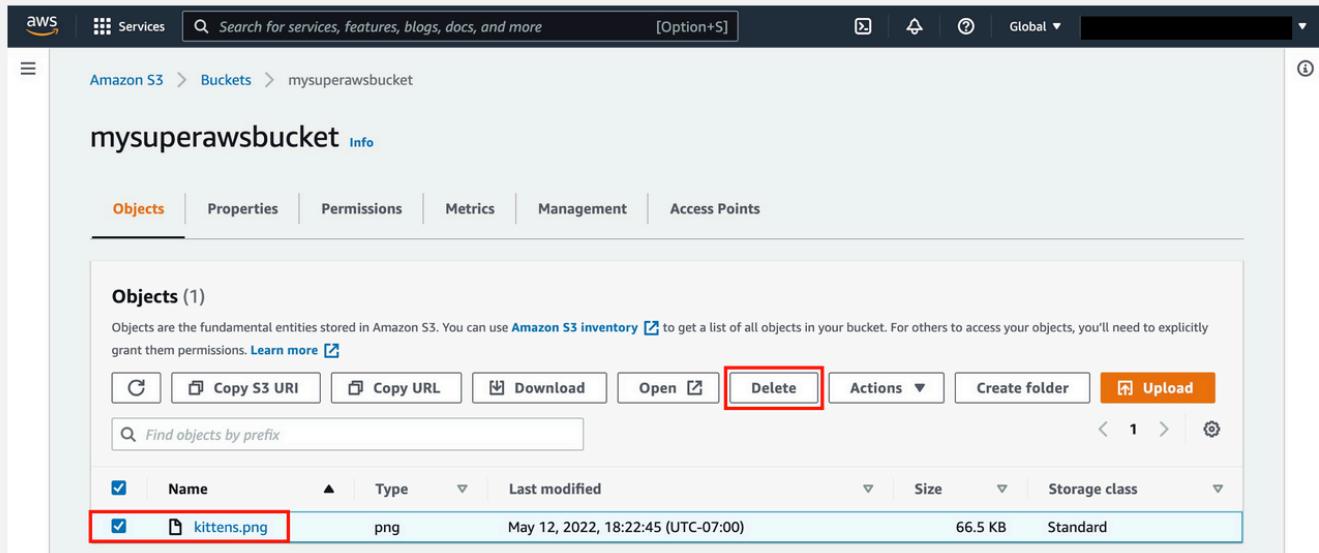


The screenshot shows the AWS S3 console interface. The user is in the 'Objects' tab of the 'mysuperawsbucket'. A single object, 'kittens.png', is listed. The 'Download' button in the toolbar above the list is highlighted with a red box. The file details are shown in the table below:

Name	Type	Last modified	Size	Storage class
kittens.png	png	May 12, 2022, 18:22:45 (UTC-07:00)	66.5 KB	Standard

Puede eliminar fácilmente el objeto y el bucket de la consola de Amazon S3. De hecho, una práctica recomendada consiste en eliminar los recursos que ya no utiliza para que no le sigan cobrando por ellos.

- a. Primero deberá eliminar el objeto. Seleccione la casilla ubicada junto al archivo que quiera eliminar y seleccione **Delete (Eliminar)**.



The screenshot shows the AWS S3 console interface. The user is in the 'Objects' tab of the 'mysuperawsbucket'. A single object, 'kittens.png', is listed. The 'Delete' button in the toolbar above the list is highlighted with a red box. The file details are shown in the table below:

Name	Type	Last modified	Size	Storage class
kittens.png	png	May 12, 2022, 18:22:45 (UTC-07:00)	66.5 KB	Standard

- b. Revise e ingrese **permanently delete** (eliminar permanentemente) en el campo de entrada de texto para confirmar la eliminación. Haga clic en **Delete objects** (Eliminar objetos).

The screenshot shows the 'Delete objects' dialog box in the AWS S3 console. It displays a single object named 'kittens.png' with a size of 66.5 KB. A warning message at the top states: 'Deleting the specified objects can't be undone.' Below the table, a section titled 'Permanently delete objects?' contains a text input field with the placeholder 'To confirm deletion, type *permanently delete* in the text input field.' The input field contains the text 'permanently delete', which is highlighted with a red box. At the bottom right are two buttons: 'Cancel' and 'Delete objects', with 'Delete objects' being the primary action button.

c. Haga clic en Amazon S3 > Buckets (Buckets) para ver todos los buckets de una región.

The screenshot shows the 'Buckets' page in the AWS S3 console. On the left, there is a navigation sidebar with sections like 'Buckets', 'Storage Lens', and 'Feature spotlight'. The main content area shows an 'Account snapshot' with total storage of 8.3 KB, one object, and an average object size of 8.3 KB. Below this is a table titled 'Buckets (2)' showing two buckets: 'cf-templates-1gg6gjg8me6n9-us-west-2' and 'mysuperawsbucket'. The 'Create bucket' button is visible at the top of the bucket list table.

Name	AWS Region	Access	Creation date
cf-templates-1gg6gjg8me6n9-us-west-2	US West (Oregon) us-west-2	Objects can be public	May 10, 2022, 22:57:18 (UTC-07:00)
mysuperawsbucket	US West (Oregon) us-west-2	Bucket and objects not public	May 12, 2022, 17:34:04 (UTC-07:00)

d. Seleccione el botón de opción a la izquierda del bucket que creó y, luego, seleccione **Delete**.

The screenshot shows the AWS S3 'Buckets' page. At the top, there's an 'Account snapshot' section with storage details: Total storage 8.3 KB, Object count 1, Avg. object size 8.3 KB. To the right, a note says 'You can enable advanced metrics in the "default-account-dashboard" configuration.' Below this is a table titled 'Buckets (2)'. The first row is for 'cf-templates-1gg6gjg8me6n9-us-west-2'. The second row, for 'mysuperawsbucket', has its radio button selected and is highlighted with a red box. In the top right of the table header, there are buttons for 'Copy ARN', 'Empty', 'Delete' (which is also highlighted with a red box), and 'Create bucket'.

e. Para confirmar la eliminación, ingrese el nombre del bucket en el campo de entrada de texto y seleccione **Delete bucket** (Eliminar bucket).

This screenshot shows the 'Delete bucket' confirmation dialog. It starts with a warning message: '⚠ Deleting a bucket cannot be undone.' followed by 'Bucket names are unique. If you delete a bucket, another AWS user can use the name.' Below this is a section titled 'Delete bucket "mysuperawsbucket"?'. It contains a text input field with the value 'mysuperawsbucket', which is highlighted with a red box. At the bottom right of the dialog, there are 'Cancel' and 'Delete bucket' buttons, with 'Delete bucket' also highlighted with a red box.

## Lambda functions

Servicio en la nube basado en funciones y que elimina la necesidad de levantar una infraestructura compleja.

## Create a Lambda function with the console

In this example, your function takes a JSON object containing two integer values labeled "length" and "width". The function multiplies these values to calculate an area and returns this as a JSON string.

Your function also prints the calculated area, along with the name of its CloudWatch log group. Later in the tutorial, you'll learn to use [CloudWatch Logs](#) to view records of your functions' invocation.

### To create a Hello world Lambda function with the console

1. Open the [Functions page](#) of the Lambda console.
2. Choose **Create function**.
3. Select **Author from scratch**.
4. In the **Basic information** pane, for **Function name**, enter **myLambdaFunction**.
5. For **Runtime**, choose either **Node.js 22.x** or **Python 3.13**.
6. Leave **architecture** set to **x86\_64**, and then choose **Create function**.

In addition to a simple function that returns the message `Hello from Lambda!`, Lambda also creates an [execution role](#) for your function. An execution role is an AWS Identity and Access Management (IAM) role that grants a Lambda function permission to access AWS services and resources. For your function, the role that Lambda creates grants basic permissions to write to CloudWatch Logs.

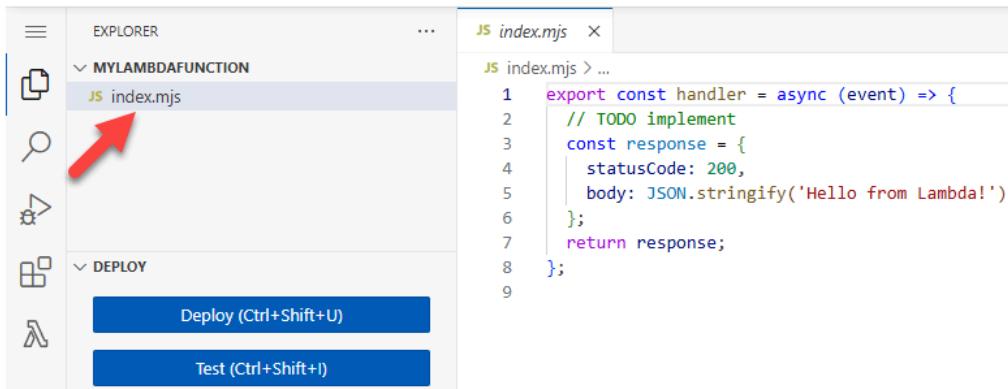
Use the console's built-in code editor to replace the Hello world code that Lambda created with your own function code.

[Node.js](#)    Python

### To modify the code in the console

1. Choose the **Code** tab.

In the console's built-in code editor, you should see the function code that Lambda created. If you don't see the **index.mjs** tab in the code editor, select **index.mjs** in the file explorer as shown on the following diagram.



2. Paste the following code into the **index.mjs** tab, replacing the code that Lambda created.

```
export const handler = async (event, context) => {

  const length = event.length;
  const width = event.width;
  let area = calculateArea(length, width);
  console.log(`The area is ${area}`);

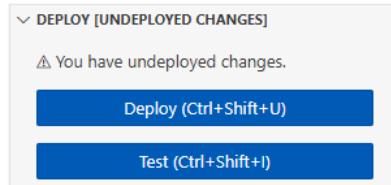
  console.log('CloudWatch log group: ', context.logGroupName);

  let data = {
    "area": area,
  };
  return JSON.stringify(data);

  function calculateArea(length, width) {
    return length * width;
}
```

```
 }  
};
```

3. In the **DEPLOY** section, choose **Deploy** to update your function's code:



### Understanding your function code

Before you move to the next step, let's take a moment to look at the function code and understand some key Lambda concepts.

- **The Lambda handler:**

Your Lambda function contains a Node.js function named `handler`. A Lambda function in Node.js can contain more than one Node.js function, but the `handler` function is always the entry point to your code. When your function is invoked, Lambda runs this method.

When you created your Hello world function using the console, Lambda automatically set the name of the handler method for your function to `handler`. Be sure not to edit the name of this Node.js function. If you do, Lambda won't be able to run your code when you invoke your function.

To learn more about the Lambda handler in Node.js, see [Define Lambda function handler in Node.js](#).

- **The Lambda event object:**

The function `handler` takes two arguments, `event` and `context`. An `event` in Lambda is a JSON formatted document that contains data for your function to process.

If your function is invoked by another AWS service, the event object contains information about the event that caused the invocation. For example, if your function is invoked when an object is uploaded to an Amazon Simple Storage Service (Amazon S3) bucket, the event contains the name of the bucket and the object key.

In this example, you'll create an event in the console by entering a JSON formatted document with two key-value pairs.

- **The Lambda context object:**

The second argument that your function takes is `context`. Lambda passes the `context object` to your function automatically. The context object contains information about the function invocation and execution environment.

You can use the context object to output information about your function's invocation for monitoring purposes. In this example, your function uses the `logGroupName` parameter to output the name of its CloudWatch log group.

To learn more about the Lambda context object in Node.js, see [Using the Lambda context object to retrieve Node.js function information](#).

- **Logging in Lambda:**

With Node.js, you can use `console.log` and `console.error` to send information to your function's log. The example code uses `console.log` statements to output the calculated area and the name of the function's CloudWatch Logs group. You can also use any logging library that writes to `stdout` or `stderr`.

To learn more, see [Log and monitor Node.js Lambda functions](#). To learn about logging in other runtimes, see the 'Building with' pages for the runtimes you're interested in.

## Invoke the Lambda function using the console code editor

To invoke your function using the Lambda console code editor, create a test event to send to your function. The event is a JSON formatted document containing two key-value pairs with the keys "length" and "width".

### To create the test event

1. In the **TEST EVENTS** section of the console code editor, choose **Create test event**.



2. For **Event Name**, enter **myTestEvent**.
3. In the **Event JSON** section, replace the default JSON with the following:

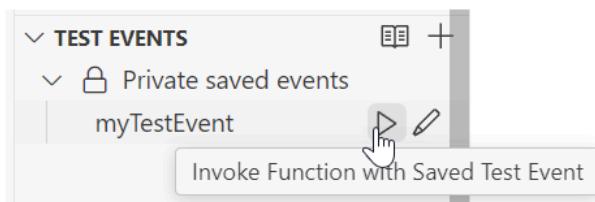
```
{  
  "length": 6,  
  "width": 7  
}
```



4. Choose **Save**.

### To test your function and view invocation records

In the **TEST EVENTS** section of the console code editor, choose the run icon next to your test event:



When your function finishes running, the response and function logs are displayed in the **OUTPUT** tab. You should see results similar to the following:

Node.js    Python

Status: Succeeded  
Test Event Name: myTestEvent

Response  
"{"area":42}"

Function Logs

```
START RequestId: 5c012b0a-18f7-4805-b2f6-40912935034a Version: $LATEST
2024-08-31T23:39:45.313Z      5c012b0a-18f7-4805-b2f6-40912935034a      INFO      The area
2024-08-31T23:39:45.331Z      5c012b0a-18f7-4805-b2f6-40912935034a      INFO      CloudW
END RequestId: 5c012b0a-18f7-4805-b2f6-40912935034a
REPORT RequestId: 5c012b0a-18f7-4805-b2f6-40912935034a Duration: 20.67 ms      Billed
```

Request ID  
5c012b0a-18f7-4805-b2f6-40912935034a

When you invoke your function outside of the Lambda console, you must use CloudWatch Logs to view your function's execution results.

#### To view your function's invocation records in CloudWatch Logs

1. Open the [Log groups](#) page of the CloudWatch console.
2. Choose the log group for your function (`/aws/lambda/myLambdaFunction`). This is the log group name that your function printed to the console.
3. Scroll down and choose the **Log stream** for the function invocations you want to look at.

Log streams (14)		<input type="button" value="C"/>	Delete	Create log stream	Search all log streams
<input type="text"/>	Filter log streams or try prefix search	<input type="checkbox"/> Exact match	<input type="checkbox"/> Show expired	<a href="#">Info</a>	< 1 >
<input type="checkbox"/> Log stream	Last event time	▼	▼		
<a href="#">2024/04/30/[\$LATEST]e0fa</a>	2024-04-30 17:24:16 (UTC)				
<a href="#">2024/04/19/[\$LATEST]e9a</a>	2024-04-19 20:59:06 (UTC)				
<a href="#">2024/02/22/[\$LATEST]cf0</a>	2024-02-22 18:38:41 (UTC)				
<a href="#">2024/02/21/[1]d132c4d</a>	2024-02-21 21:37:01 (UTC)				
<a href="#">2024/02/21/[1]5ad</a>	2024-02-21 21:37:01 (UTC)				



You should see output similar to the following:

Node.js	Python
<pre>INIT_START Runtime Version: nodejs:22.v13      Runtime Version ARN: arn:aws:lambda:us- START RequestId: aba6c0fc-cf99-49d7-a77d-26d805dacd20 Version: \$LATEST 2024-08-23T22:04:15.809Z      5c012b0a-18f7-4805-b2f6-40912935034a INFO  The area is 2024-08-23T22:04:15.810Z      aba6c0fc-cf99-49d7-a77d-26d805dacd20 INFO  CloudWatch END RequestId: aba6c0fc-cf99-49d7-a77d-26d805dacd20 REPORT RequestId: aba6c0fc-cf99-49d7-a77d-26d805dacd20 Duration: 17.77 ms      Bill</pre>	