

# Debian 13 : Panoramica e obiettivi

## Introduzione

Iniziamo un percorso quotidiano su Debian 13, la nuova stable della distribuzione universale. L'obiettivo è analizzare con calma, passo passo, ogni componente essenziale: dall'installer al kernel, dai servizi di sistema alla sicurezza, fino agli strumenti per server e desktop.

## Cos'è Debian 13 e cosa cambia rispetto alle versioni precedenti

Debian 13 è la nuova release stable, focalizzata su stabilità, coerenza dell'ecosistema e sicurezza a lungo termine. Come tutte le stable, privilegia:

- **Pacchetti testati** e congelati per l'intero ciclo di vita;
- **Aggiornamenti di sicurezza** tempestivi, gestiti dal Security Team;
- **Compatibilità** con un'ampia gamma di architetture hardware.

È una base solida sia per server che per desktop, ideale quando la priorità è la prevedibilità del sistema.

## Struttura della serie: 20 capitoli tecnici su Debian 13

Questa serie è pensata come un **percorso di 20 capitoli**. Ogni articolo affronta un tema specifico:

1. Panoramica di Debian 13 e impostazione del percorso.
2. Installazione guidata (BIOS/UEFI, partizionamento, scelte minime).
3. Il kernel Linux in Debian 13.
4. Processi e servizi con `systemd`.
5. Struttura del filesystem (FHS) e layout Debian.
6. Gestione pacchetti con `apt` e `dpkg`.
7. Utenti, gruppi, permessi e `sudo`.
8. Rete: IP, DNS e strumenti base.
9. Logging: `journald` e `rsyslog`.
10. Storage: partizioni, LVM e montaggi.
11. Desktop: GNOME su Debian 13.
12. Driver grafici (Intel/AMD/NVIDIA).
13. Audio con PipeWire.
14. Virtualizzazione con KVM/QEMU.
15. Container su Debian.
16. Sicurezza e firewall.
17. Backup e ripristino.
18. Automazione con script, cron e timer.
19. Server web di base.
20. Comunità Debian, supporto e contributi.

L'idea è fornire una traccia modulare: ogni puntata è autonoma, ma l'insieme compone una mini-documentazione pratica su Debian 13.

## Prerequisiti minimi per seguire la serie

Per trarre il massimo da questa serie è utile avere:

- Una installazione (anche virtuale) di Debian 13;
- Accesso a un `terminal` con privilegi `sudo`;
- Familiarità di base con la riga di comando.

Tutti i comandi saranno mostrati in forma esplicita, ad esempio:

```
sudo apt update  
sudo apt upgrade
```

Gli esempi saranno sempre pensati per essere ripetibili su una Debian 13 “pulita”.



# 2 · Installazione: scelte, partizionamento e primo avvio

## Introduzione

Ci concentriamo sulle scelte critiche in fase di installazione di Debian 13. Una buona installazione è la base per un sistema stabile e semplice da mantenere.

## Preparazione del supporto di installazione

Per iniziare occorre scaricare l'immagine ufficiale di Debian 13 dal sito del progetto e creare una chiavetta avviabile. Da GNU/Linux è sufficiente usare `cp` o `dd` (con attenzione):

```
sudo cp debian-13-amd64-netinst.iso /dev/sdX
sync
```

Sostituisci `/dev/sdX` con il dispositivo corretto della tua chiavetta USB. In alternativa puoi usare strumenti come **Ventoy** o **Rufus** (da Windows).

## Schema di partizionamento: semplice ma flessibile

In ambienti di test o desktop generici, uno schema classico e robusto è:

- **EFI System Partition** (solo UEFI) · 512 MB;
- **/ (root)** · almeno 30–40 GB;
- **/home** separata · spazio restante;
- **swap** · in base alla RAM (2–4 GB per desktop tipico).

Per scenari più avanzati (server, VM, storage dinamico) vedremo in un giorno dedicato come usare **LVM** per una gestione più flessibile di volumi e snapshot.

## Selezione dei componenti di sistema

Durante l'installazione l'installer di Debian 13 propone la scelta di:

- **Ambiente desktop** (GNOME predefinito, ma disponibili KDE, Xfce, ecc.);
- **Stampanti**, strumenti di sistema, SSH server;
- **Mirror APT** da cui scaricare i pacchetti.

Per un ambiente minimalista è possibile deselegionare l'ambiente desktop e installarlo in un secondo momento:

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

Questa scelta è utile sui server o sulle VM che nascono “server-only”.

## 3 • Il cuore del sistema: il kernel Linux

Il kernel è il nucleo di Debian 13: gestisce processi, memoria, I/O, rete e interfaccia con l'hardware. Capire come è gestito in Debian aiuta a intervenire in modo consapevole su prestazioni e compatibilità.

### Verificare versione e configurazione di base

Per verificare la versione del kernel in esecuzione:

```
uname -r
```

```
uname -a
```

Su Debian 13 i kernel disponibili sono pacchettizzati come `linux-image-*`. Puoi elencarli con:

```
apt search ^linux-image
```

Un sistema tipico desktop usa il metapacchetto `linux-image-amd64`, che punta sempre alla versione stabile più recente disponibile per l'architettura.

### Moduli caricabili e gestione driver

I driver sono gestiti come moduli kernel. Per vedere quelli attualmente caricati:

```
lsmod | less
```

Per ottenere dettagli su un modulo specifico:

```
modinfo nome_modulo
```

La configurazione di quali moduli caricare automaticamente all'avvio è gestita tramite file in `/etc/modules` e in `/etc/modprobe.d/`. In molti casi Debian rileva automaticamente l'hardware e carica i moduli necessari senza interventi manuali.

### Aggiornare o cambiare kernel in sicurezza

Il modo raccomandato per aggiornare il kernel è tramite APT:

```
sudo apt update
```

```
sudo apt full-upgrade
```

Se desideri mantenere più versioni di kernel affiancate (per test o rollback), Debian lo consente: ogni `linux-image-*` installa un proprio set di file in `/boot`. Il bootloader (tipicamente GRUB) permette di scegliere quale kernel avviare dal menu avanzato.

# 4 · Processi, servizi e systemd: come Debian avvia e gestisce il sistema

In Debian 13 il sistema di init predefinito è systemd. È responsabile dell'avvio dei servizi, della gestione dei processi e di molte funzionalità di basso livello come i log, i timer e i target di sistema.

## Capire i target di systemd e lo stato del sistema

Per vedere lo stato generale di systemd:

```
systemctl status
```

Il sistema è organizzato in *target* (equivalenti ai vecchi runlevel). Ad esempio:

- `multi-user.target`: sistema multiutente senza interfaccia grafica;
- `graphical.target`: sistema con display manager e desktop;
- `rescue.target`: modalità di manutenzione minimale.

## Gestire servizi: avvio, arresto e abilitazione

Il comando principale è `systemctl`. Alcuni esempi:

```
# Verificare lo stato di un servizio
```

```
systemctl status ssh
```

```
# Avviare / fermare
```

```
sudo systemctl start ssh
```

```
sudo systemctl stop ssh
```

```
# Abilitare / disabilitare all'avvio
```

```
sudo systemctl enable ssh
```

```
sudo systemctl disable ssh
```

Debian integra in systemd la gestione dei servizi di sistema, di quelli di rete e di molti demoni di sistema. Conoscere `systemctl` è fondamentale per amministrare Debian 13.

## Creare un semplice servizio personalizzato

Come esempio, ecco un servizio minimale che esegue uno script in background:

```
# /etc/systemd/system/mio-servizio.service
```

```
[Unit]
```

```
Description=Esempio di servizio personalizzato
```

```
[Service]
```

```
ExecStart=/usr/local/bin/mio-script.sh
```

Restart=on-failure

[Install]

WantedBy=multi-user.target

Dopo aver creato il file e reso eseguibile lo script, ricarica la configurazione di systemd e abilita il servizio:

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable --now mio-servizio.service
```

# 5 • Filesystem e struttura di Debian: FHS, /etc, /var, /usr, /home

Debian segue lo standard FHS (Filesystem Hierarchy Standard), che definisce dove vanno collocati file di configurazione, binari, librerie e dati variabili. Capire questa struttura è essenziale per orientarsi nel sistema.

## Panoramica delle directory principali

Le directory chiave da conoscere sono:

- `/`: radice del filesystem;
- `/etc`: configurazioni di sistema;
- `/var`: dati variabili (log, spool, cache);
- `/usr`: programmi e librerie gestiti dal sistema;
- `/home`: dati degli utenti;
- `/opt`: applicazioni opzionali, solitamente non gestite da APT.

Un semplice comando per dare un'occhiata è:

```
ls -F /
```

## Configurazioni in /etc e dati variabili in /var

`/etc` contiene la maggior parte delle configurazioni testuali di sistema. Alcuni esempi:

- `/etc/apt/sources.list` per i repository APT;
- `/etc/network/interfaces.d/` (se usato) per la rete;
- `/etc/ssh/sshd_config` per il server SSH.

`/var` contiene invece dati che cambiano nel tempo:

- `/var/log/`: log di sistema e dei servizi;
- `/var/lib/`: database e stati interni dei demoni;
- `/var/spool/`: code di stampa, mail, ecc.

## /usr e /home: sistema vs utenti

`/usr` ospita la maggior parte dei binari, librerie e file condivisi forniti dai pacchetti Debian. Alcuni path importanti:

- `/usr/bin`: programmi per tutti gli utenti;
- `/usr/sbin`: programmi amministrativi;
- `/usr/share`: dati architettura-indipendenti (documentazione, icone, ecc.).

`/home` contiene invece le directory personali degli utenti. Ogni utente ha i propri file e configurazioni (in genere dotfile come `.bashrc`, `.config/`, ecc.). Separare `/home` su una partizione dedicata facilita reinstallazioni e backup mirati.

# 6 • Gestione pacchetti con APT e dpkg: installare, aggiornare, rimuovere

APT e dpkg sono il cuore della gestione software in Debian 13. Sapere come usarli in modo corretto permette di mantenere il sistema coerente e pulito.

## APT: il front-end ad alto livello

I comandi più usati di APT sono:

```
sudo apt update          # aggiorna l'elenco pacchetti
sudo apt upgrade        # aggiorna i pacchetti installati
sudo apt full-upgrade   # gestisce cambi di dipendenze
sudo apt install pkg    # installa un pacchetto
sudo apt remove pkg     # rimuove un pacchetto
sudo apt autoremove     # rimuove dipendenze non più necessarie
```

Per cercare un pacchetto:

```
apt search nome_pacchetto
```

## dpkg: il livello basso

dpkg lavora direttamente sui pacchetti `.deb`:

```
sudo dpkg -i file.deb    # installa un pacchetto locale
dpkg -l | grep nome     # elenca pacchetti installati
dpkg -L nome_pacchetto  # elenca i file installati da un pacchetto
```

In caso di dipendenze mancanti dopo `dpkg -i` puoi sempre sistemare con:

```
sudo apt -f install
```

## Repository, componenti e pinning

Le sorgenti dei pacchetti sono definite in `/etc/apt/sources.list` e in `/etc/apt/sources.list.d/`. Le sezioni principali sono:

- `main`: software libero supportato da Debian;
- `contrib`: dipende da componenti non-liberi;
- `non-free / non-free-firmware`: software o firmware non libero.

Per casi avanzati, il *pinning* permette di definire priorità diverse per pacchetti provenienti da repository differenti (ad es. `stable` e `backports`) tramite file in `/etc/apt/preferences.d/`.

# 7 · Utenti, gruppi, permessi e sudo: controllare chi può fare cosa

La sicurezza di base di Debian 13 passa dalla corretta gestione di utenti, gruppi e permessi. In questo giorno vediamo come funzionano e come sfruttare sudo in modo sicuro.

## Gestire utenti e gruppi

Per creare un nuovo utente con directory home e shell:

```
sudo adduser nomeutente
```

Per aggiungere un utente a un gruppo (ad esempio sudo):

```
sudo usermod -aG sudo nomeutente
```

I gruppi sono definiti in `/etc/group`, mentre gli utenti in `/etc/passwd`.

## Permessi su file e directory

I permessi classici UNIX si gestiscono con `chmod`, `chown` e `chgrp`. Esempio:

```
# leggere i permessi
```

```
ls -l file
```

```
# cambiare proprietario e gruppo
```

```
sudo chown utente:gruppo file
```

```
# impostare permessi (rwxr-x---)
```

```
chmod 750 file
```

Per scenari più avanzati Debian supporta anche ACL (Access Control List) tramite `setfacl` e `getfacl`.

## Configurare sudo in modo sicuro

`sudo` permette a utenti non root di eseguire comandi con privilegi elevati. La configurazione principale è in `/etc/sudoers` (modificare solo tramite `visudo`) e in `/etc/sudoers.d/`.

Esempio minimale in un file in `/etc/sudoers.d/mioadmin`:

```
mioadmin ALL=(ALL:ALL) ALL
```

È possibile restringere i comandi consentiti per aumentare ulteriormente la sicurezza.

# 8 · Rete di base: configurare IP, DNS e strumenti di diagnostica

Una configurazione di rete corretta è fondamentale per qualsiasi sistema. In Debian 13 possiamo usare NetworkManager, systemd-networkd o configurazioni statiche a seconda del contesto.

## Verificare configurazione e interfacce

Per vedere le interfacce di rete e il loro stato:

```
ip addr
```

```
ip route
```

Per testare la connettività di base:

```
ping -c 4 debian.org
```

## Configurazione tramite NetworkManager

Su sistemi desktop Debian 13 usa tipicamente NetworkManager, gestibile via interfaccia grafica o con `nmcli`:

```
nmcli device status
```

```
nmcli connection show
```

È possibile creare connessioni statiche, Wi-Fi, VPN, ecc. direttamente da `nm-connection-editor` o da `nmcli`.

## DNS, /etc/hosts e strumenti di diagnostica

I DNS sono solitamente configurati dal gestore di rete (NetworkManager o systemd-resolved). Per controllare quale DNS viene usato:

```
resolvectl status
```

Strumenti utili di diagnostica:

- `traceroute` per analizzare il percorso dei pacchetti;
- `dig` o `drill` per interrogare i DNS;
- `ss` per vedere le porte in ascolto.

# 9 • Logging di sistema: journalctl, journald e rsyslog

Il logging è fondamentale per la diagnosi dei problemi. Debian 13 usa systemd-journald di default, spesso in combinazione con rsyslog sui sistemi server.

## journalctl: consultare i log di systemd

Per vedere i log più recenti:

```
journalctl -xe
```

Per filtrare per unità di servizio:

```
journalctl -u ssh.service
```

Per visualizzare solo i log dell'avvio corrente:

```
journalctl -b
```

## Persistenza dei log e integrazione con rsyslog

Per abilitare la persistenza dei log di journald, verifica che esista la directory `/var/log/journal/`. In caso contrario:

```
sudo mkdir -p /var/log/journal
```

```
sudo systemctl restart systemd-journald
```

rsyslog continua a gestire in molti casi i log tradizionali in file di testo sotto `/var/log/` (ad esempio `/var/log/syslog`, `/var/log/auth.log`).

## Rotazione e gestione dello spazio

La rotazione dei log in `/var/log` è gestita da `logrotate`, con configurazione in `/etc/logrotate.conf` e `/etc/logrotate.d/`. Per forzare una rotazione manuale:

```
sudo logrotate -f /etc/logrotate.conf
```

Per limitare lo spazio usato dal journal, si possono configurare limiti in `/etc/systemd/journal.conf` (opzioni come `SystemMaxUse=`).

# 10 • Storage e LVM: partizioni, volumi logici e /etc/fstab

Gestire correttamente dischi, partizioni e volumi è essenziale per scalabilità e affidabilità. Debian 13 integra strumenti classici come parted, fdisk e LVM2.

## Partizioni e filesystem di base

Per elencare i dischi e le partizioni:

```
lsblk
```

```
sudo fdisk -l
```

Per creare un filesystem ext4 su una nuova partizione:

```
sudo mkfs.ext4 /dev/sdX1
```

## Introduzione a LVM (Logical Volume Manager)

LVM permette di creare volumi logici flessibili sopra i dischi fisici. Flusso tipico:

```
sudo pvcreate /dev/sdX2
```

```
sudo vgcreate vg_dati /dev/sdX2
```

```
sudo lvcreate -n lv_backup -L 100G vg_dati
```

```
sudo mkfs.ext4 /dev/vg_dati/lv_backup
```

In questo modo è possibile ridimensionare i volumi logici e spostarli tra dischi con meno vincoli rispetto alle partizioni tradizionali.

## Montaggi e /etc/fstab

Per montare automaticamente volumi all'avvio si utilizza /etc/fstab. È buona pratica usare gli UUID dei filesystem:

```
blkid /dev/vg_dati/lv_backup
```

Esempio di riga in /etc/fstab:

```
UUID=... /backup ext4 defaults 0 2
```

Dopo aver modificato /etc/fstab è possibile testare con:

```
sudo mount -a
```

# 11 · Desktop: GNOME, Wayland e Xorg

Per l'ambiente desktop predefinito Debian 13 propone GNOME, tipicamente su Wayland. Vediamo le implicazioni pratiche e come passare eventualmente a Xorg.

## GNOME come task predefinito

Se hai selezionato l'ambiente desktop in fase di installazione, probabilmente stai usando GNOME. In caso contrario puoi installarlo con:

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

GNOME fornisce un'esperienza integrata, con gestione notifiche, impostazioni di sistema e integrazione con NetworkManager, PipeWire, ecc.

## Wayland vs Xorg su Debian 13

Per impostazione predefinita GNOME viene avviato su Wayland se supportato dall'hardware. Dalla schermata di login (GDM) puoi scegliere la sessione "GNOME su Xorg" dal menu a forma di ingranaggio.

Wayland offre vantaggi in termini di sicurezza e gestione compositing, ma alcuni programmi legacy potrebbero richiedere Xorg. Debian 13 supporta entrambi.

## Strumenti utili per il desktop

Alcuni strumenti tipici per un desktop Debian produttivo:

- `gnome-tweaks` per personalizzazioni avanzate;
- `gnome-extensions-app` per gestire estensioni GNOME;
- tool come `flatpak` per installare applicazioni sandboxed (opzionale).

Installazione rapida:

```
sudo apt install gnome-tweaks gnome-shell-extensions
```

# 12 · Driver grafici: Intel, AMD e NVIDIA su Debian 13

Una corretta configurazione della grafica è fondamentale per performance e stabilità. Debian 13 integra driver open source maturi per Intel e AMD, mentre per NVIDIA spesso si ricorre ai driver proprietari.

## Verificare l'hardware grafico

Per identificare la GPU:

```
lspci | grep -i vga
```

```
lspci | grep -i 3d
```

Per controllare quali driver sono in uso puoi usare `lshw` o `glxinfo` (pacchetto `mesa-utils`).

## Driver Intel e AMD (Mesa)

Su Intel e AMD i driver open source forniti da Mesa sono in genere la scelta migliore. Pacchetti tipici:

```
sudo apt install firmware-misc-nonfree mesa-utils
```

In molti casi Debian configura automaticamente questi driver, senza richiedere interventi manuali.

## Driver NVIDIA proprietari

Per GPU NVIDIA moderne può essere necessario installare i driver proprietari:

```
sudo apt install nvidia-driver
```

Questo installerà kernel module, librerie e utilità come `nvidia-smi`. Dopo l'installazione è consigliato riavviare. In ambienti produttivi è importante verificare la compatibilità tra versione del driver e versione del kernel.

# 13 · Audio su Debian 13: PipeWire, ALSA e PulseAudio

Debian 13 adotta PipeWire come backend audio moderno su molte installazioni desktop, integrandosi con ALSA e sostituendo gradualmente PulseAudio.

## Strato hardware: ALSA

ALSA (Advanced Linux Sound Architecture) comunica direttamente con l'hardware audio. Per elencare i dispositivi:

```
aplay -l
```

```
arecord -l
```

## PipeWire e compatibilità con PulseAudio

PipeWire funge da gestore audio (e video) a livello utente, fornendo compatibilità con app progettate per PulseAudio. Per verificare che PipeWire sia attivo:

```
systemctl --user status pipewire
```

Strumenti come `pavucontrol` restano utili per regolare volumi e routing.

## Diagnostica problemi audio

In caso di problemi audio:

- Verifica muto/volume da interfaccia grafica;
- Controlla i log dell'utente con `journalctl --user -u pipewire`;
- Assicurati che l'utente appartenga ai gruppi opportuni (tipicamente non necessario intervenire su Debian recente).

# 14 • Virtualizzazione con KVM/QEMU e virt-manager

Debian 13 è un'ottima base per la virtualizzazione tramite KVM/QEMU, con gestione semplificata da virt-manager.

## Verificare supporto hardware alla virtualizzazione

Per controllare se la CPU supporta virtualizzazione hardware:

```
egrep -c '(vmx|svm)' /proc/cpuinfo
```

Se il risultato è  $> 0$ , la CPU supporta VT-x (Intel) o AMD-V.

## Installare KVM, libvirt e virt-manager

Installazione dei pacchetti principali:

```
sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients virt-manager
```

Aggiungi il tuo utente al gruppo libvirt:

```
sudo usermod -aG libvirt $USER
```

## Gestione delle VM

Con virt-manager puoi creare, clonare e gestire macchine virtuali tramite interfaccia grafica. In alternativa, la CLI virsh permette automazioni più spinte:

```
virsh list --all
```

```
virsh start nome-vm
```

```
virsh shutdown nome-vm
```

# 15 • Container su Debian 13: Docker, Podman e LXC

I container permettono di isolare applicazioni e ambienti in modo leggero. Su Debian 13 possiamo usare Docker, Podman o LXC a seconda delle esigenze.

## Docker su Debian 13

Docker è molto diffuso nell'ecosistema dei container. Sui sistemi Debian è consigliabile usare il repository ufficiale Docker per versioni aggiornate, ma esiste anche un pacchetto nei repository Debian.

Un ciclo minimale:

```
sudo docker pull debian:13
```

```
sudo docker run -it --rm debian:13 bash
```

## Podman come alternativa rootless

Podman implementa gran parte dell'API Docker ma con un approccio *rootless*. Installazione:

```
sudo apt install podman
```

Esempio di esecuzione:

```
podman run --rm -it debian:13 bash
```

## LXC/LXD per container di sistema

LXC è più vicino a una virtualizzazione a livello di sistema, ideale per creare ambienti isolati simili a VM ma più leggeri. Debian fornisce i pacchetti:

```
sudo apt install lxc
```

L'uso di LXC richiede una configurazione più approfondita, ma si integra bene in scenari multi-tenant.

# 16 · Sicurezza e firewall: nftables, ufw e buone pratiche

La sicurezza di Debian 13 si basa su un buon modello di permessi, aggiornamenti regolari e un firewall configurato in modo coerente.

## Aggiornamenti e hardening di base

La prima buona pratica è mantenere il sistema aggiornato:

```
sudo apt update
```

```
sudo apt full-upgrade
```

Pacchetti utili di hardening:

- `sudo apt install unattended-upgrades` per aggiornamenti automatici;
- `sudo apt install fail2ban` per mitigare brute force su servizi esposti.

## nftables come firewall moderno

Debian 13 adotta `nftables` come backend moderno per il firewall. Un esempio minimale per permettere solo SSH e traffico già stabilito:

```
table inet filter {{
  chain input {{
    type filter hook input priority 0;
    policy drop;
    ct state established,related accept
    iif lo accept
    tcp dport 22 accept
  }}
}}
```

La configurazione persistente può essere salvata in `/etc/nftables.conf` e attivata con:

```
sudo systemctl enable --now nftables
```

## ufw come frontend semplificato

Per chi preferisce un'interfaccia più semplice, `ufw` è disponibile anche su Debian:

```
sudo apt install ufw
```

```
sudo ufw default deny incoming
```

```
sudo ufw default allow outgoing
```

```
sudo ufw allow 22/tcp
```

```
sudo ufw enable
```

# 17 • Backup e ripristino: strategie e strumenti

Un sistema senza backup affidabili è un sistema a rischio. Debian 13 offre molti strumenti per backup locali, remoti e incrementali.

## Strategia prima dello strumento

Prima di scegliere lo strumento è importante definire:

- Quali dati salvare (configurazioni, database, file utente...);
- Con quale frequenza (giornaliera, settimanale);
- Dove conservare i backup (disco esterno, NAS, remoto).

## rsync per backup semplici

rsync è uno strumento flessibile per backup incrementali:

```
rsync -avh --delete /home/utente/ /mnt/backup/home-utente/
```

Può essere schedulato con cron o con timer systemd.

## Strumenti avanzati: BorgBackup, Restic

Per backup deduplicati e cifrati sono molto usati:

- `borgbackup` (pacchetto `borgbackup`);
- `restic` (pacchetto `restic`).

Entrambi permettono repository locali o remoti, snapshot, verifica di integrità e restore selettivo.

# 18 · Automazione: shell script, cron e systemd timer

Automatizzare compiti ripetitivi rende l'amministrazione di Debian 13 più efficiente e meno soggetta a errori.

## Shell script come unità di automazione

Un semplice script di manutenzione:

```
#!/bin/bash
set -e
```

```
apt update
apt -y upgrade
apt -y autoremove
```

Salvalo in `/usr/local/sbin/manutenzione.sh` e rendilo eseguibile:

```
sudo chmod +x /usr/local/sbin/manutenzione.sh
```

## Cron: schedulazione tradizionale

Per eseguire lo script ogni notte alle 3:00:

```
sudo crontab -e
```

Aggiungi:

```
0 3 * * * /usr/local/sbin/manutenzione.sh >/var/log/manutenzione.log 2>&1
```

## Timer di systemd

In alternativa è possibile usare i timer systemd, che offrono una integrazione più stretta con il sistema. Struttura tipica:

- Unità servizio in `/etc/systemd/system/mio-task.service`;
- Timer associato in `/etc/systemd/system/mio-task.timer`.

Il timer può essere attivato con `systemctl enable --now mio-task.timer`.

# 19 • Server web di base: Nginx/Apache su Debian 13

Debian 13 è spesso scelta come base per server web grazie alla stabilità dei pacchetti e alla lunga tradizione di integrazione con Apache e Nginx.

## Installare un web server

Per installare Nginx:

```
sudo apt install nginx
```

Per installare Apache:

```
sudo apt install apache2
```

Dopo l'installazione il servizio viene avviato automaticamente.

## Verificare che il server risponda

Controlla lo stato del servizio:

```
systemctl status nginx
```

```
# oppure
```

```
systemctl status apache2
```

Da un browser o con curl:

```
curl http://localhost
```

## Struttura di base della configurazione

Per Nginx, i virtual host sono tipicamente in `/etc/nginx/sites-available/` con symlink in `sites-enabled/`. Per Apache, i virtual host sono in `/etc/apache2/sites-available/` e attivati con `a2ensite`.

Dopo ogni modifica alla configurazione, ricorda di testare e ricaricare:

```
sudo nginx -t && sudo systemctl reload nginx
```

```
# oppure
```

```
sudo apache2ctl configtest && sudo systemctl reload apache2
```

# 20 · Comunità Debian, documentazione e come contribuire

Chiudiamo la serie guardando alla comunità che sta dietro Debian 13 e alle risorse per continuare a imparare e, se vuoi, contribuire direttamente al progetto.

## Documentazione ufficiale e wiki

Le risorse principali sono:

- Il **Debian Administrator's Handbook** (aggiornato per le nuove release);
- Il sito **www.debian.org** con sezioni dedicate a installazione, FAQ, sicurezza;
- Il **wiki Debian**, ricco di guide pratiche.

## Canali di supporto

Per supporto e confronto con altri utenti:

- Mailing list ufficiali;
- Canali IRC/Matrix dedicati a Debian;
- Forum e community locali (LUG, gruppi utenti).

## Contribuire: bug report, traduzioni, pacchetti

Contribuire a Debian non significa solo scrivere codice. Puoi:

- Segnalare bug e fornire report dettagliati;
- Aiutare nelle traduzioni della documentazione;
- Mantenere pacchetti o collaborare con i maintainer esistenti.

Questa serie è solo un punto di partenza: l'ecosistema Debian è vasto e in continua evoluzione.