UBUNTU 24.04 LTS VE SONRASI · APT · SNAP

Ubuntu El Kitabı

İleri kullanıcılar için

Sistemi içinden yönet. APT ve dpkg'nin derinlikleri, PPA'lar, sürüm yükseltme ve HWE, Netplan, systemd, AppArmor, ufw, Timeshift, NVIDIA sürücüleri ve açılmayan bir sistemi kurtarma.

BÖLÜMLER

Sarı eğik yazılar (`<paket>` gibi) senin dolduracağın yerlerdir. Terminalde yapıştırma Ctrl+Shift+V, kopyalama Ctrl+Shift+C'dir. `sudo` ile başlayan komutlar sistemi değiştirir; ne yaptığını okumadan çalıştırma.

APT derinlemesine

APT ayarları `/etc/apt/apt.conf.d/`, depo tanımları `/etc/apt/sources.list.d/`, sabitleme kuralları `/etc/apt/preferences.d/` altında durur.

```
$ apt policy <paket>
```

Kurulu sürüm, aday sürüm ve hangi depodan geldiği; öncelikleri gösterir.

```
$ apt-cache madison <paket>
```

Depolarda bulunan tüm sürümler.

```
$ sudo apt install <paket>=<sürüm>
```

Belirli bir sürümü kurar (düşürmek için de kullanılır).

```
$ sudo apt-mark hold <paket>
```

Paketi güncellemelere karşı sabitler. `unhold` çözer, `showhold` listeler.

```
$ apt-mark showmanual
```

Elle kurduğun paketler (bağımlılık olarak gelenler hariç).

```
$ apt-cache depends <paket>
```

Paketin bağımlılıkları. Tersine: `apt-cache rdepends --installed <paket>`.

```
$ sudo apt install --no-install-recommends <paket>
```

Önerilen (Recommends) ek paketleri kurmadan kurar.

```
$ sudo apt install apt-file && sudo apt-file update
```

Kurulu olmayan paketlerin dosyalarında arama yapmayı sağlar.

```
$ apt-file search <dosya-adı>
```

Bir dosyayı ya da komutu hangi paketin sağladığını bulur.

```
$ less /var/log/apt/history.log
```

APT işlem geçmişi: ne zaman, ne kuruldu/kaldırıldı. Eskiler için `zless history.log.1.gz`.

```
$ sudo apt clean
```

İndirilmiş paket önbelleğini (/var/cache/apt/archives) boşaltır. `autoclean` yalnızca eskimişleri siler.

```
$ sudo dpkg-reconfigure <paket>
```

Bir paketin kurulum sorularını yeniden sorar. Örnek: `tzdata`, `locales`, `keyboard-configuration`.

```
$ sudo apt install needrestart
```

Güncellemelerden sonra hangi servislerin yeniden başlatılması gerektiğini söyler.

```
$ sudo apt install unattended-upgrades &&
sudo dpkg-reconfigure -plow unattended-
upgrades
```

Güvenlik güncellemelerini otomatik kurar. Günlükler `/var/log/unattended-upgrades/` altında.

```
$ apt modernize-sources
```

Eski tek satırlık `sources.list` dosyalarını yeni deb822 (`.sources`) biçimine çevirir (APT 3.0 ve sonrası; sudo ile).

dpkg sorguları

`dpkg`, APT'nin altında çalışan düşük seviyeli paket aracıdır. Kurulum için APT kullan; dpkg sorgu ve onarım için idealdir.

```
$ dpkg -l | grep <kelime>
```

Kurulu paketleri durumlarıyla listeler (ii = kurulu, rc = kaldırılmış ama ayarları duruyor).

```
$ dpkg -L <paket>
```

Kurulu paketin dosyaları.

```
$ dpkg -S <dosya-yolu>
```

Bir dosyanın hangi pakete ait olduğunu söyler. Örnek: `dpkg -S /usr/bin/ls`.

```
$ dpkg -s <paket>
```

Paketin durumu, sürümü ve bağımlılıkları.

```
$ sudo dpkg --configure -a
```

Yarım kalmış kurulumları tamamlar (“dpkg was interrupted” hatası).

```
$ sudo apt --fix-broken install
```

Eksik bağımlılıkları kurarak bozuk durumu düzeltir.

```
$ dpkg -l | awk '/^rc/{print $2}' | xargs -r
sudo apt purge -y
```

Kaldırılmış paketlerden kalan ayar dosyalarını temizler.

`$ dpkg-query -W -f='${Installed-Size}\t${Package}\n' | sort -n | tail -20`	En çok yer kaplayan 20 paket (KB).
`$ sudo dpkg --add-architecture i386 && sudo apt update`	32 bit paketleri etkinleştirir (Steam, Wine için).
`$ sudo apt install debsums && sudo debsums -s`	Paket dosyalarının bozulup bozulmadığını denetler; yalnızca sorunluları yazar.

Depolar ve PPA

Ubuntu 24.04 ve sonrasında depolar deb822 biçimindeki `/etc/apt/sources.list.d/ubuntu.sources` dosyasında durur. PPA'lar aynı klasöre ayrı dosyalar olarak eklenir.

`$ sudo add-apt-repository universe`	Bir bileşeni açar (universe, multiverse, restricted).
`$ sudo add-apt-repository ppa:<kullanıcı>/<ppa>`	PPA ekler, anahtarını kurar ve paket listesini yeniler.
`$ sudo add-apt-repository --remove ppa:<kullanıcı>/<ppa>`	PPA'yı kaldırır; kurulu paketler olduğu gibi kalır.
`$ sudo apt install ppa-purge && sudo ppa-purge ppa:<kullanıcı>/<ppa>`	PPA'yı kaldırır ve paketleri Ubuntu'nun kendi sürümlerine geri düşürür.
`$ ls /etc/apt/sources.list.d/`	Ekli depo dosyaları.
`$ curl -fsSL <anahtar-URL> | sudo gpg --dearmor -o /etc/apt/keyrings/<ad>.gpg`	Üçüncü taraf deponun imza anahtarını ekler; depo dosyasında `Signed-By:` ile bu anahtarı göster.

İPUCU Sabitleme (pinning): `/etc/apt/preferences.d/` altına dosya yazıp bir depo ya da paketin önceliğini belirlersin. Örneğin Mozilla'nın APT deposundan Firefox kullanırken Snap sürümünün geri gelmesini böyle engellersin.

DİKKAT PPA'lar Canonical tarafından denetlenmez; güvenmediğin kişilerin PPA'larını ekleme. Sürüm yükseltmesinden önce PPA'ları kapatmak sorunları azaltır.

Sürüm yükseltme ve HWE

Ubuntu sürümler arasında yerinde yükseltmeyi destekler. LTS'ten bir sonraki LTS'e yükseltme, yeni LTS'in ilk ara sürümü (ör. 26.04.1) çıkınca açılır.

`$ sudo apt update && sudo apt full-upgrade`	Önce mevcut sürümü tamamen güncelle.
`$ sudo do-release-upgrade`	Bir sonraki desteklenen sürüme yükseltir.
`$ sudo do-release-upgrade -d`	LTS yükseltmesini ilk ara sürümü beklemeden başlatır (geliştirme sürümleri için de kullanılır).
`$ sudoedit /etc/update-manager/release-upgrades`	`Prompt=lts` yalnızca LTS'ler arası, `Prompt=normal` her sürüm için teklif alır.
`$ lsb_release -a`	Sürüm ve kod adı.
`$ sudo apt install --install-recommends linux-generic-hwe-24.04`	24.04 LTS'e daha yeni çekirdek (HWE) kurar. Sürüm numarasını kendi LTS'ine göre değiştir.
`$ pro status`	Ubuntu Pro servislerinin (ESM, Livepatch) durumu.
`$ sudo pro enable livepatch`	Güvenlik yamalarının çekirdeğe yeniden başlatmadan uygulanması.

Netplan

Ubuntu ağ yapılandırmasını Netplan ile yazar. Masaüstünde Netplan işi NetworkManager'a bırakır; sunucuda ise `/etc/netplan/*.yaml` dosyaları doğrudan kullanılır.

`$ ls /etc/netplan/`	Netplan yapılandırma dosyaları.

`$ sudo netplan get`	Birleştirilmiş geçerli yapılandırma.
`$ sudo netplan try`	Değişikliği uygular; 120 saniye içinde onaylamazsan eskisine döner (uzaktan çalışırken güvenli).
`$ sudo netplan apply`	Yapılandırmayı kalıcı olarak uygular.
`$ networkctl status`	systemd-networkd kullanılıyorsa arayüz durumları (sunucu kurulumları).

Güvenlik duvarı: ufw

Ubuntu'da ufw kurulu gelir ama kapalıdır. ufw (Uncomplicated Firewall), nftables üzerinde basit bir arayüzdür.

`$ sudo ufw status verbose`	Durum ve kurallar.
`$ sudo ufw default deny incoming && sudo ufw default allow outgoing`	Gelen bağlantıları kapat, gidenlere izin ver (masaüstü için makul varsayılan).
`$ sudo ufw enable`	Güvenlik duvarını açar ve açılışta etkin kalmasını sağlar.
`$ sudo ufw allow ssh`	SSH'ye (22) izin verir. `sudo ufw limit ssh` kaba kuvvet denemelerini yavaşlatır.
`$ sudo ufw allow 8080/tcp`	Bir portu açar.
`$ sudo ufw allow from 192.168.1.0/24 to any port 445`	Yalnızca yerel ağdan belirli bir porta izin verir.
`$ sudo ufw status numbered`	Kuralları numaralı listeler.
`$ sudo ufw delete <numara>`	Numarasıyla kural siler.

```
$ sudo ufw app list
```

Paketlerin tanımladığı hazır uygulama profilleri.

```
$ sudo apt install gufw
```

Grafik arayüz.

> **DİKKAT** Docker kendi iptables/nftables kurallarını yazdığı için yayımladığı portlar ufw kurallarını atlar. Docker kullanıyorsan portları `127.0.0.1:8080:80` biçiminde yerel adrese bağla.

AppArmor

Ubuntu, programların neye erişebileceğini AppArmor profilleriyle sınırlar. Bir program "Permission denied" verip dosya izinleri doğruysa neden AppArmor olabilir.

```
$ sudo aa-status
```

Yüklü profiller ve hangi modda oldukları (enforce / complain).

```
$ sudo journalctl -k --grep=apparmor
```

AppArmor'un engellediği işlemler ("DENIED" satırları).

```
$ sudo apt install apparmor-utils
```

aa-complain, aa-enforce, aa-logprof gibi araçları kurar.

```
$ sudo aa-complain /etc/apparmor.d/<profil>
```

Profili yalnızca kayıt tutan moda alır (teşhis için).

```
$ sudo aa-enforce /etc/apparmor.d/<profil>
```

Profili yeniden zorlayıcı moda alır.

```
$ sudo aa-logprof
```

Günlükteki engellemelere bakıp profile kural eklemeyi etkileşimli olarak önerir.

```
$ sudo apparmor_parser -r
/etc/apparmor.d/<profil>
```

Düzenlenen profili yeniden yükler. Yerel eklemeler `/etc/apparmor.d/local/` altına yazılır.

İPUCU Ubuntu 24.04'ten beri yetkisiz kullanıcı ad alanları (unprivileged user namespaces) AppArmor ile kısıtlıdır. Bazı AppImage ve Electron uygulamaları bu yüzden "sandbox" hatası verir. Doğru çözüm, uygulama için `/etc/apparmor.d/` altında `userns,` izni veren bir profil yazmaktır; kısıtlamayı tümden kapatmak (`kernel.apparmor_restrict_unprivileged_userns=0`) güvenliği azaltır.

systemd servisleri

Servisleri (arka plan hizmetleri), zamanlayıcıları ve açılış sürecini systemd yönetir.

`$ systemctl status <servis>`	Durumu, son günlük satırlarını ve PID'i gösterir.
`$ sudo systemctl start|stop|restart <servis>`	Servisi başlatır / durdurur / yeniden başlatır.
`$ sudo systemctl enable --now <servis>`	Açılışta başlamasını sağlar ve hemen başlatır.
`$ sudo systemctl disable --now <servis>`	Açılışta başlamasını engeller ve durdurur.
`$ sudo systemctl mask <servis>`	Servisin hiçbir şekilde başlatılmasını engeller. `unmask` geri açar.
`$ systemctl --failed`	Başarısız olan birimleri listeler.
`$ systemctl list-units --type=service --state=running`	Çalışan servisler.
`$ systemctl list-unit-files --state=enabled`	Açılışta başlayan birimler.
`$ systemctl list-timers`	Zamanlanmış görevler (cron'un systemd karşılığı).
`$ systemctl cat <servis>`	Birim dosyasının içeriğini ve ek parçaları gösterir.
`$ sudo systemctl edit <servis>`	Paket dosyasına dokunmadan override (drop-in) oluşturur.

`$ sudo systemctl daemon-reload`	Birim dosyalarını değiştirdikten sonra systemd'nin yeniden okumasını sağlar.
`$ systemctl --user enable --now <servis>`	Kullanıcı servisleri; tanımlar `~/.config/systemd/user/` altında durur.
`$ loginctl enable-linger $USER`	Kullanıcı servislerinin oturum açılmadan da çalışmasını sağlar.
`$ systemd-analyze blame`	Açılışta hangi servisin ne kadar sürdüğünü sıralar.
`$ systemd-analyze critical-chain`	Açılıştaki kritik yolu gösterir.

Günlükler: journalctl

Sistem günlükleri systemd journal'da tutulur. Ubuntu'da rsyslog da kuruludur; aynı mesajlar `/var/log/syslog` dosyasına da yazılır.

`$ journalctl -b`	Bu açılıştan beri tüm günlük.
`$ journalctl -b -1 -p err`	Bir önceki açılıştaki hatalar. Çökme sonrası ilk bakılacak yer.
`$ journalctl -u <servis> -f`	Bir servisin günlüğünü canlı izler.
`$ journalctl -p warning --since "1 hour ago"`	Son bir saatteki uyarı ve üstü mesajlar.
`$ journalctl --since today --grep "<kelime>"`	Bugünkü günlükte metin arar.
`$ journalctl -k`	Yalnızca çekirdek mesajları (dmesg karşılığı).
`$ journalctl --list-boots`	Kayıtlı açılışları listeler.
`$ journalctl --disk-usage`	Günlüklerin kapladığı alan.

`$ sudo journalctl --vacuum-time=2weeks`	İki haftadan eski günlükleri siler. `--vacuum-size=500M` de kullanılabilir.
`$ coredumpctl list`	Çöken programların kayıtları (systemd-coredump kuruluysa). `coredumpctl info <PID>` ayrıntı verir.
`$ sudo dmesg -w`	Çekirdek halkasını canlı izler (USB takıp çıkarınca ne olduğunu görmek için).

Ağ: nmcli, ip, ss

Masaüstü kurulumlarında ağı NetworkManager yönetir; komut satırı aracı `nmcli`, metin arayüzü `nmtui`.

`$ nmcli device status`	Ağ aygıtları ve durumları.
`$ nmcli device wifi list`	Görünen Wi-Fi ağları, sinyal gücüyle.
`$ nmcli device wifi connect "<SSID>" password "<parola>"`	Wi-Fi ağına bağlanır.
`$ nmcli connection show`	Kayıtlı bağlantılar.
`$ nmcli connection up|down "<ad>"`	Bağlantıyı açar / kapatır.
`$ sudo nmcli connection modify "<ad>" ipv4.method manual ipv4.addresses 192.168.1.50/24 ipv4.gateway 192.168.1.1 ipv4.dns "1.1.1.1 9.9.9.9"`	Statik IP tanımlar. Sonra `nmcli connection up "<ad>"`.
`$ nmtui`	Menülü metin arayüzü; grafik ortam yokken işe yarar.
`$ sudo hostnamectl set-hostname <yeni-ad>`	Bilgisayarın adını değiştirir.
`$ ip -br a`	Arayüzler ve IP adresleri, kısa biçimde.

`$ ip route`	Yönlendirme tablosu ve varsayılan ağ geçidi.
`$ resolvectl status`	DNS sunucuları (systemd-resolved kullanılıyorsa). Aksi halde `cat /etc/resolv.conf`.
`$ ss -tulpn`	Dinleyen TCP/UDP portları ve hangi program olduğu (programı görmek için sudo).
`$ ping -c 4 <adres>`	Erişilebilirlik testi.
`$ tracepath <adres>`	Paketlerin izlediği yol.
`$ curl -I https://<site>`	Bir sitenin HTTP başlıklarını getirir.
`$ dig <alan-adı>`	DNS sorgusu (bind9-dnsutils paketi).

Kullanıcılar ve izinler

Bu sistemde sudo yetkisi `sudo` grubuna üyelikle verilir.

`$ id`	Kullanıcı kimliğin ve grupların.
`$ sudo useradd -m -G sudo -s /bin/bash <kullanıcı>`	Ev klasörlü, sudo yetkili yeni kullanıcı. Ardından `sudo passwd <kullanıcı>`.
`$ sudo usermod -aG <grup> <kullanıcı>`	Kullanıcıyı gruba ekler (`-a` olmadan diğer gruplardan çıkarır). Oturumu yeniden açınca etkinleşir.
`$ sudo userdel -r <kullanıcı>`	Kullanıcıyı ev klasörüyle siler.
`$ passwd`	Kendi parolanı değiştirir.
`$ chmod +x <dosya>`	Dosyayı çalıştırılabilir yapar.

`$ chmod 644 <dosya>`	Sahibi okur/yazar, diğerleri okur. 755: klasörler ve betikler için tipik.
`$ chmod -R u+rwX,go-rwx <klasör>`	Klasörü yalnızca sahibine açar.
`$ sudo chown -R <kullanıcı>:<grup> <yol>`	Sahibini değiştirir.
`$ setfacl -m u:<kullanıcı>:rw <dosya>`	Belirli bir kullanıcıya ek izin verir (ACL). `getfacl` görüntüler.
`$ sudo -i`	Root kabuğu açar. İşin bitince exit.
`$ sudo visudo -f /etc/sudoers.d/<ad>`	Sudo kurallarını sözdizimi denetimiyle düzenler.

İPUCU Ubuntu'da root hesabının parolası yoktur (kilitlidir); root kabuğu için `sudo -i` kullan.

SSH

`$ sudo apt install openssh-server`	SSH sunucusunu kurar; servis (ssh) kendiliğinden başlar. Durum: `systemctl status ssh`.
`$ ssh-keygen -t ed25519 -C "<etiket>"`	Anahtar çifti oluşturur (`~/.ssh/id_ed25519`).
`$ ssh-copy-id <kullanıcı>@<sunucu>`	Açık anahtarını sunucuya yükler; sonra parolasız girersin.
`$ ssh <kullanıcı>@<sunucu> -p <port>`	Bağlanır.
`$ ssh -L 8080:localhost:80 <kullanıcı>@<sunucu>`	Yerel 8080 portunu sunucunun 80 portuna tünel yapar.
`$ scp <dosya> <kullanıcı>@<sunucu>:~/`	Dosyayı sunucuya kopyalar. Klasör için `-r`.

İPUCU `~/.ssh/config` ile kısa adlar tanımla:
`Host sunucum · HostName 192.168.1.10 · User kullanici · IdentityFile ~/.ssh/id_ed25519`
Sonra yalnızca `ssh sunucum` .

İPUCU Parolayla girişi kapatmak için `/etc/ssh/sshd_config.d/10-yerel.conf` dosyasına `PasswordAuthentication no` yaz ve SSH servisini yeniden başlat.

Diskler, takas ve anlık görüntüler

Ubuntu kurulumda varsayılan olarak ext4 kullanır. Sistem anlık görüntüleri için en yaygın araç Timeshift'tir: ext4'te rsync modunda, Ubuntu tarzı `@` / `@home` alt birimli Btrfs'te btrfs modunda çalışır.

`$ sudo apt install timeshift`	Timeshift'i kurar; ilk açılışta sihirbaz hedef diski ve sıklığı sorar.
`$ sudo timeshift --create --comments "<açıklama>"`	Elle anlık görüntü alır. Büyük bir güncellemeden önce iyi alışkanlık.
`$ sudo timeshift --list`	Anlık görüntüleri listeler.
`$ sudo timeshift --restore`	Etkileşimli geri yükleme. Sistem açılmıyorsa canlı USB'den de çalıştırılabilir.
`$ lsblk -f && findmnt /`	Diskler, dosya sistemleri ve kökün nereye bağlı olduğu.
`$ sudoedit /etc/fstab`	Açılışta bağlanan diskler. UUID'leri `sudo blkid` verir. Hata açılışı bozar; sonra `sudo mount -a` ile dene.
`$ swapon --show`	Etkin takas alanları.
`$ sudo fallocate -l 4G /swapfile && sudo chmod 600 /swapfile && sudo mkswap /swapfile && sudo swapon /swapfile`	4 GB takas dosyası oluşturur. Kalıcı olması için fstab'a `/swapfile none swap sw 0 0` ekle.
`$ sudo apt install systemd-zram-generator`	RAM'de sıkıştırılmış takas (zram); ayarı `/etc/systemd/zram-generator.conf` .

```
$ sudo apt install smartmontools && sudo smartctl -a /dev/<sda>
```

Disk sağlık raporu (SMART). NVMe için `/dev/nvme0`.

```
$ sudo tune2fs -l /dev/<sda2> | grep -i -E 'state|mount count'
```

ext4 bölümünün durumu.

> **DİKKAT** `fsck`'yi bağlı bir bölümde çalıştırma. Kök bölümü denetlemek için canlı USB'den aç ya da GRUB'da çekirdek satırına geçici olarak `fsck.mode=force` ekleyerek başlat.

Çekirdek, GRUB ve önyükleme

GRUB ayarları `/etc/default/grub` dosyasındadır; değişiklikten sonra `update-grub` çalıştırılır. Çekirdek meta paketi `linux-generic`, başlık dosyaları `linux-headers-generic`'dir.

```
$ dpkg --list 'linux-image*' | grep ^ii
```

Kurulu çekirdekler.

```
$ uname -r
```

Çalışan çekirdek.

```
$ sudoedit /etc/default/grub
```

GRUB ayarları. Menüyü her açılışta göstermek için `GRUB_TIMEOUT_STYLE=menu` ve `GRUB_TIMEOUT=5`.

```
$ sudo update-grub
```

GRUB yapılandırmasını yeniden üretir.

> **İPUCU** Çekirdek parametresi eklemek için `/etc/default/grub` içindeki `GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"` satırına ekle, sonra `sudo update-grub`.

```
$ cat /proc/cmdline
```

Çalışan çekirdeğin parametreleri.

```
$ sudo update-initramfs -u
```

Çalışan çekirdek için initramfs'i yeniden oluşturur. Tümü için `-u -k all`.

```
$ sudo apt install linux-headers-generic
```

Sürücü derlemek (DKMS) için çekirdek başlıkları.

```
$ dkms status
```

DKMS ile derlenen modüller (NVIDIA, VirtualBox, Wi-Fi) ve hangi çekirdek için derlendikleri.

`$ mokutil --sb-state`	Secure Boot açık mı.
`$ sudo mokutil --import /var/lib/dkms/mok.pub`	DKMS imzalama anahtarını MOK'a kaydeder. Bir parola belirle; yeniden başlatınca mavi ekranda “Enroll MOK” seç.
`$ lsmod | grep <modül>`	Yüklü çekirdek modülleri. `modinfo` ayrıntı, `sudo modprobe` yükler.
`$ sudo apt install --install-recommends linux-generic-hwe-24.04`	LTS'e daha yeni HWE çekirdeği kurar; numarayı kendi LTS sürümüne göre değiştir.

Sürücüler: NVIDIA ve donanım hızlandırma

Ubuntu özel sürücüleri `ubuntu-drivers` aracıyla yönetir; Secure Boot için imzalama işini de kurulum sırasında yapar.

`$ ubuntu-drivers list`	Donanımın için mevcut sürücüler.
`$ sudo ubuntu-drivers install`	Önerilen sürücüyü kurar.
`$ sudo ubuntu-drivers install nvidia:<sürüm>`	Belirli bir NVIDIA sürücü dalını kurar (ör. `nvidia:570`).
`$ nvidia-smi`	GPU durumu ve kullanımı.
`$ prime-select query`	Çift GPU'lu dizüstünde etkin mod (nvidia / intel / on-demand).
`$ sudo apt install intel-media-va-driver-non-free`	Intel donanım video çözme (multiverse).
`$ sudo apt install vainfo && vainfo`	VA-API donanım hızlandırma desteğini doğrular.
`$ lspci -k | grep -A 3 -E "VGA|3D"`	Ekran kartı ve kullanılan çekirdek sürücüsü.

```
$ sudo apt install steam-installer
```

Steam (multiverse; 32 bit mimariyi kendisi ekler).

> **DİKKAT** NVIDIA sürücüsünü nvidia.com'daki .run dosyasıyla kurma; çekirdek güncellemelerinde bozulur ve paket yöneticisiyle çakışır.

Podman, Docker, Distrobox

Ana sistemi kirletmeden başka dağıtımların araçlarını kullanmak için Distrobox, servisler için Podman ya da Docker kullanılır. Podman, Docker komutlarıyla büyük ölçüde uyumludur ve root gerektirmez.

```
$ sudo apt install podman distrobox
```

Podman ve Distrobox'ı kurar.

```
$ distrobox create -i registry.fedoraproject.org/fedora:latest -n <ad>
```

Başka bir dağıtımın konteynerini oluşturur (ör. `-i archlinux`, `-i ubuntu:24.04`).

```
$ distrobox enter <ad>
```

İçine girer; ev klasörün paylaşılır. İçeride `distrobox-export --app <uygulama>` uygulamayı ana menüye ekler.

```
$ podman run -it --rm debian bash
```

Geçici bir konteyner başlatır.

```
$ podman ps -a
```

Tüm konteynerler.

```
$ podman images
```

Yerel imajlar. `podman system prune -a` kullanılmayanları siler.

```
$ podman run -d -p 8080:80 -v ~/site:/usr/share/nginx/html docker.io/library/nginx
```

Klasör bağlayarak servis çalıştırır.

```
$ sudo apt install docker.io docker-compose-v2
```

Docker'ı Ubuntu deposundan kurar (Docker'ın kendi deposu daha günceldir).

```
$ sudo usermod -aG docker $USER
```

Docker'ı sudo'suz kullanmak için (oturumu yeniden aç). Bu grup root'a eşdeğer yetki verir.

İPUCU Podman konteynerini systemd servisi yapmak için Quadlet kullan: `~/.config/containers/systemd/` altına `.container` dosyası yaz, `systemctl --user daemon-reload` ve başlat.

Metin işleme ve boru hatları

`|` bir komutun çıktısını diğerine verir. `>` dosyaya yazar (üstüne), `>>` ekler, `2>&1` hataları da aynı yere yönlendirir. `&&` önceki başarılıysa, `||` başarısızsa çalışır.

```
$ grep -rni "<metin>" <klasör>
```
Klasörde büyük/küçük harf duyarsız, satır numaralı yinelemeli arama.

```
$ grep -v "^#" <dosya> | grep -v "^$"
```
Yorum ve boş satırları atar; yapılandırma dosyalarını okumak için.

```
$ sed -i 's/<eski>/<yeni>/g' <dosya>
```
Dosyada bul-değiştir. Önce `-i` olmadan deneyip çıktıya bak.

```
$ awk '{print $1, $3}' <dosya>
```
Satırların 1. ve 3. sütununu yazar.

```
$ sort <dosya> | uniq -c | sort -rn | head
```
En sık tekrar eden satırlar.

```
$ wc -l <dosya>
```
Satır sayısı.

```
$ cut -d: -f1 /etc/passwd
```
İki nokta ile ayrılmış 1. alan (kullanıcı adları).

```
$ find . -name "*.log" -mtime +7 -delete
```
7 günden eski .log dosyalarını siler.

```
$ find . -type f -name "*.jpg" -print0 |
xargs -0 -I{} cp {} ~/Resimler/
```
Bulunan dosyaları tek tek kopyalar (boşluklu adlarda güvenli).

```
$ <komut> | tee <dosya>
```
Çıktıyı hem ekrana hem dosyaya yazar.

```
$ echo "<satır>" | sudo tee -a /etc/<dosya>
```
Root dosyasına ekleme yapar (`sudo echo >>` çalışmaz).

```
$ diff -u <eski> <yeni>
```
İki dosyanın farkı.

`$ watch -n 2 <komut>`	Komutu 2 saniyede bir tekrarlar.
`$ curl -s <URL> | jq '.'`	JSON çıktısını biçimlendirir (jq paketi).

Arşiv, yedek, senkronizasyon

`$ tar -czvf <arşiv>.tar.gz <klasör>`	Klasörü gzip ile arşivler.
`$ tar --zstd -cvf <arşiv>.tar.zst <klasör>`	zstd ile arşivler; gzip'ten hızlı ve daha iyi sıkıştırır.
`$ tar -xvf <arşiv>`	Her tür tar arşivini açar. Hedef klasör için `-C <klasör>`.
`$ tar -tvf <arşiv>`	Açmadan içeriği listeler.
`$ zip -r <arşiv>.zip <klasör>`	Zip oluşturur. Açmak için `unzip <arşiv>.zip`.
`$ 7z x <arşiv>.7z`	7z / rar gibi biçimleri açar (7zip paketi).
`$ rsync -avh --progress <kaynak>/ <hedef>/`	Klasörü eşitler; yalnızca değişenleri kopyalar. Sondaki / "içeriği" anlamına gelir.
`$ rsync -avh --delete --dry-run <kaynak>/ <hedef>/`	Hedefte kaynakta olmayanları silerek aynala; önce --dry-run ile ne olacağını gör.
`$ rsync -avz -e ssh <kaynak>/ <kullanıcı>@<sunucu>:<yol>/`	SSH üzerinden uzak yedek.
`$ rclone config`	Google Drive, OneDrive, S3 gibi bulut hedefleri tanımlar; `rclone sync` ile eşitler.

Kabuğu kişiselleştirme

Bash ayarları ev klasöründeki `~/.bashrc` dosyasındadır. Takma adları ve değişkenleri dosyanın sonuna

ekle.

`$ nano ~/.bashrc`	Kabuk ayar dosyasını açar.
`alias guncelle='sudo apt update && sudo apt upgrade && sudo snap refresh'`	Tek kelimeyle sistem güncellemesi. Dosyaya yazıp kaydet.
`alias ll='ls -lah --group-directories-first'`	Kullanışlı ayrıntılı liste.
`$ source ~/.bashrc`	Değişiklikleri yeni terminal açmadan yükler.
`export PATH="$HOME/.local/bin:$PATH"`	Kendi betiklerinin klasörünü PATH'e ekler.
`$ echo $SHELL`	Kullandığın kabuk.
`$ sudo apt install zsh && chsh -s /usr/bin/zsh`	Zsh kurar ve varsayılan yapar (oturumu yeniden aç). Fish için aynı yöntem.
`$ type <komut>`	Bir adın takma ad mı, fonksiyon mu, dosya mı olduğunu gösterir.

İPUCU Betik yazarken ilk satıra `#!/usr/bin/env bash` , ikinci satıra `set -euo pipefail` koy; hatada durur, tanımsız değişkeni yakalar. Sonra `chmod +x betik.sh` .

gsettings ile gizli GNOME ayarları

GNOME ayarlarının hepsi dconf veritabanında durur. Ayarlar uygulamasında olmayan seçeneklere gsettings ile ulaşırsın.

`$ gsettings set org.gnome.desktop.interface color-scheme 'prefer-dark'`	Koyu tema (açık için 'default').
`$ gsettings set org.gnome.desktop.interface clock-show-seconds true`	Saatte saniyeleri gösterir.
`$ gsettings set org.gnome.desktop.interface clock-show-weekday true`	Saatin yanında gün adını gösterir.

`$ gsettings set org.gnome.mutter center-new-windows true`	Yeni pencereleri ekran ortasında açar.
`$ gsettings set org.gnome.desktop.peripherals.touchpad tap-to-click true`	Dokunarak tıklama.
`$ gsettings set org.gnome.shell disable-user-extensions true`	Tüm eklentileri tek seferde kapatır (sorun giderme). false ile geri açılır.
`$ gsettings list-recursively org.gnome.desktop.interface`	Bir şemadaki tüm anahtarlar ve değerleri.
`$ gsettings reset <şema> <anahtar>`	Bir ayarı varsayılana döndürür.
`$ dconf dump / > ~/gnome-ayarlar.ini`	Tüm GNOME ayarlarını dosyaya yedekler.
`$ dconf load / < ~/gnome-ayarlar.ini`	Yedeği geri yükler; yeni kurulumda ayarları taşımak için.
`$ sudo apt install dconf-editor`	Tüm anahtarlara açıklamalarıyla göz atabileceğin grafik araç.

Performans ve güç

GNOME güç profillerini power-profiles-daemon üzerinden yönetir.

`$ powerprofilesctl list`	Güç profilleri ve etkin olan (power-profiles-daemon).
`$ powerprofilesctl set performance`	Profil değiştirir: power-saver / balanced / performance.
`$ sudo apt install lm-sensors`	İşlemci ve disk sıcaklıkları; kurduktan sonra `sensors`.
`$ sudo apt install btop nvtop`	Gelişmiş sistem ve GPU izleyici.

`$ sudo iotop -o`	Diske en çok yazan süreçler (iotop paketi).
`$ systemd-analyze`	Toplam açılış süresi (firmware, yükleyici, çekirdek, kullanıcı alanı).
`$ sudo fstrim -av`	SSD'de TRIM çalıştırır. Haftalık otomatik yapmak için `sudo systemctl enable --now fstrim.timer`.
`$ upower -i $(upower -e | grep BAT)`	Pil sağlığı, kapasite ve şarj bilgisi.
`$ free -h && swapon --show`	RAM ve takas alanları.

Sorun giderme ve kurtarma

Sistem açılmadığında ya da bir güncelleme bir şeyi bozduğunda sırasıyla denenecekler.

İPUCU **1. Eski çekirdekle aç.** GRUB'da "Advanced options for Ubuntu" altından bir önceki çekirdeği seç. Menü görünmüyorsa açılırken Esc'e (UEFI) ya da Shift'e (BIOS) bas.

İPUCU **2. Metin konsoluna geç.** Ctrl+Alt+F3 (F2–F6). Giriş yapıp `journalctl -b -p err` ile hatalara bak.

`$ sudo systemctl restart display-manager`	Grafik oturum yöneticisini (GDM, SDDM, LightDM) yeniden başlatır; açık oturum kapanır.
`$ sudo dpkg --configure -a && sudo apt --fix-broken install`	Yarım kalmış güncellemeleri tamamlar.

İPUCU **3. Kurtarma modu.** GRUB > Advanced options > "(recovery mode)" girdisi. Menüden "root" ile kabuğa düş; kök salt okunur açılırsa `mount -o remount,rw /`.

İPUCU **4. Canlı USB'den chroot.** Canlı USB ile aç, terminalde aşağıdakileri sırayla çalıştır. Disk adlarını `lsblk` ile kendi sistemine göre değiştir.

`$ sudo mount /dev/<sda2> /mnt && sudo mount /dev/<sda1> /mnt/boot/efi`	Kök ve EFI bölümlerini bağlar (Btrfs ise `-o subvol=@` ekle).
`$ for d in dev proc sys run; do sudo mount --rbind /$d /mnt/$d; done && sudo chroot /mnt`	Kurulu sisteme geçer; içeride apt, update-grub, update-initramfs kullanılabilir.
`$ grub-install && update-grub`	chroot içinde önyükleyiciyi yeniden kurar (UEFI sistemlerde).
`$ sudo timeshift --restore`	Timeshift kuruluysa canlı USB'den son sağlam anlık görüntüye dönülebilir.

Komutlar Ubuntu 24.04 LTS ve sonrası (GNOME, APT, Snap) için yazıldı. Menü adları Türkçe arayüze göre verildi; İngilizce arayüzde parantez içindeki karşılıklara bak. **sudo** ile başlayan komutlar sistemi değiştirir, ne yaptığını okumadan çalıştırma.