Linux szerver

A Linux operációs rendszer

A Linux egy nyílt forráskódú, Unix-szerű operációs rendszer, amelyet Linus Torvalds hozott létre 1991-ben. Azóta számos fejlesztő közreműködésével folyamatosan fejlődik. A Linux magja a kernel, amely az operációs rendszer központi része, és az összes hardveres és szoftveres kommunikációt kezeli.

Főbb jellemzői:
- nyílt forráskódú - szabadon elérhető bárki számára
- stabil és biztonságos - gyakran használják szervereken és adatközpontokban
- közösségi támogatás - folyamatosan fejlesztik, hibajavítások, új funkciók
- disztribúciók – pl. Ubuntu, Arch, openSUSE, Debian stb.
- rugalmasság és testreszabhatóság - a felhasználók a saját igényeikre szabhatják

Alkalmazási területek:
- szerverek
- asztali gépek
- beágyazott rendszerek - okos eszközök, routerek, egyéb IoT eszközök
- fejlesztési környezet - szoftverfejlesztők számára ideális környezet a programok fejlesztéséhez és teszteléséhez

Linux szerver - hardverkövetelmény

Processzor: Legalább 2 GHz-es kétmagos processzor (64 bites, például Intel/AMD amd64 vagy ARM64

RAM: Legalább 1 GB RAM, de ajánlott legalább 3 GB vagy több

Tárhely: 5 GB az ISO telepítéséhez, de ajánlott 25 GB vagy több

Grafikus kártya – nem szükséges, de ajnánlott, ha a CLI fölé grafikus felületet is használni szeretnénk

A Linux – Unix hasonlóságok és különbségek

A Unix egy többfelhasználós, többfeladatos operációs rendszer, amelyet eredetileg az 1960-as évek végén fejlesztettek ki a Bell Laboratories-ban. Ez az OP rendszer az alapja a Linuxnak ill. a macOS-nek

A linux által átvett Unix filozófia (hasonlóságok):

- kis programok együttműködése - "egy program, egy feladat" elv. Rendszert alkot.
- szöveges adatok használata - olvasás, írás és feldolgozás
- szűrők és csővezeték – lehetővé teszi, hogy az egyik program kimenete egy másik program bemenete legyen – pl. ls | grep ".txt" | sort – sorba rendezi a szövegfileokat
- interaktív fejlesztést és tesztelés - a parancsok és szkriptek azonnal futtathatók

Különbségek:

- Amíg a Linux nyílt forráskódú, addig a Unix rendszerek zárt (Solaris) és nyílt forráskódúak (FreeBSD pl.).
- A Linuxnál többféle ablakkezelő van, addig az eredeti Unix parancssoros (+Gnome)
- Fájlrendszerek közti különbség (Linux: Ext4, Btrfs,Xfs, FAT32... addig a Unix: fs, hfs+)
- Processzorok (Linux: Intel, AMD, addig a Unix: Itanium (IA-64), ARM, SUN SPARC, PowerPC, aarch, x86...)

Az Ubuntu kiadások (distro-k)

Ubuntu – Gnome ablakkezelővel

Ubuntu Server – Szerver megoldás

Kubuntu - KDE Plasma desktop környezet (modern és elegáns)

Edubuntu – oktatási célú változat iskolák és diákok számára

Ubuntu Studio – multimédiás alkotók számára tervezett változat

Elementary OS – Pantheon desktop

POP!_OS – Feljesztők és játékosoknak

Lububtu – Gyors LXQt desktop környezet

A Linux összetevői

A Linux rendszer szerkezetileg több rétegből és komponensből áll, amelyek együttműködnek a rendszer megfelelő működése érdekében

Kernel réteg (rendszermag): Kezeli a processzort, memóriát, fáljrendszert, eszközöket – ez a rendszermag 97-ban % C-ben van megírva, 1% ASM, 1% C++, 1% többi.

System Call Interface (SCI): a kernel és a felhasználó közti réteg - rendszerhívások

Felhasználói tér: a kernel és a felhasználó közti réteg - rendszerhívások

Eszközök és alkalmazások: programok és fejlesztői eszközök

Parancsértelmező (Shell): parancssori felület - Bash

Hierarchikus fájlrendszer: minden fájl és könyvtár a gyökérkönyvtárból / ágazik el

Daemonok: folyamatosan futó háttérfolyamatok – pl. webszerver, adatbázis szerver stb.

Csomagkezelők: a szoftverek telepítésére, frissítésére és eltávolítására szolgálnak (apt, yum..)

Ablakkezelők: grafikus felületek a munka megkönnyítésére – pl. KDE, Gnome, XFCE

Hierarchikus fájlrendszer

/ - Gyökérkönyvtár – a fájlrendszer legfelsőbb szintje

/boot – rendszerindítás könyvtárai: pl. GRUB.

/bin - Bináris fájlok, alapvető rendszerparancsok – ls, cp, mv, rm, bash

/sbin – Rendszergazdai feladatok – shutdown, reboot ifconfig

/etc - Rendszerkonfigurációs fájlok – passwd, fstab, hosts

/home - Felhasználói könyvtárak … pl. /home/attila

/root - A root (rendszergazda) felhasználó otthoni könyvtára.

/lib ill. /lib64 - dinamikus könyvtármodulok – pl. libc.so.6

/usr – másodlagos csak olvasható felhasználói hierarchiakönyvtár

/var – változók, naplófájlok, nyomtatási sorok tárolása

/tmp - ideiglenes fájlok

/dev - hozzáférést biztosítanak a hardvereszközökhöz – pl. sda (merevlemez), tty (terminál)

/proc - futó folyamatokról információ – self, cpuinfo, meminfo

/sys - hardvereszközökről és eszközmeghajtókról szóló információkat tartalmaz – class, block, bus

/opt - opcionális szoftvereket tartalmazó könyvtár

/mnt - csatolási pontok könyvtára - /mnt/disk , /media/usb

A rendszerkönyvtárak és a megosztott könyvtárak fogalma

A rendszerkönyvtárak olyan alapvető könyvtárak, amelyek a rendszer indításához, alapműködéséhez és a rendszeradminisztrációhoz szükséges fájlokat és programokat tartalmazzák. Ezek a könyvtárak biztosítják, hogy a rendszer megfelelően működjön és a szükséges alapvető szolgáltatások elérhetőek legyenek.
/bin (felhasználói parancsok),
/sbin (rendszergazdai parancsok)
/etc (konfigurációs fájlok)
/dev (eszközök)
/proc (folyamatok)
/sys (kernel)

A megosztott könyvtárak olyan alapvető könyvtárak, amelyek különféle programok és szolgáltatások számára biztosítanak közös hozzáférést könyvtárfájlokhoz. Ezek a könyvtárak segítik a kód újrafelhasználását és csökkentik a redundanciát, mivel több program is használhatja ugyanazokat a könyvtárfájlokat.
/lib ill. /lib64 (alapvető működés)
/usr/lib (a felhasználói programokhoz szükséges megosztott könyvtárak)
/usr/share (megosztott adatfájlok - dokumentumok, ikonok, stb.)

A Superuser fogalma

A superuser (vagy root felhasználó) egy különleges felhasználói fiók a Unix-szerű operációs rendszereken (például Linux és macOS), amely teljes rendszergazdai jogosultságokkal rendelkezik. Ez a felhasználó képes mindenféle művelet végrehajtására, beleértve a rendszerkonfigurációkat, fájlok és felhasználói fiókok kezelését, valamint a rendszerkritikus fájlok szerkesztését és törlését.

A sudo (Superuser DO) egy Unix-szerű operációs rendszereken használt parancs, amely lehetővé teszi a rendszergazdai vagy root jogosultságokkal rendelkező felhasználók számára, hogy különböző rendszergazdai feladatokat végezzenek el. Ez a parancs ideiglenesen emelt szintű jogosultságokat ad a futtatott parancshoz, így a felhasználóknak nem kell állandóan root felhasználóként bejelentkezniük.

Linux szerver telepítése

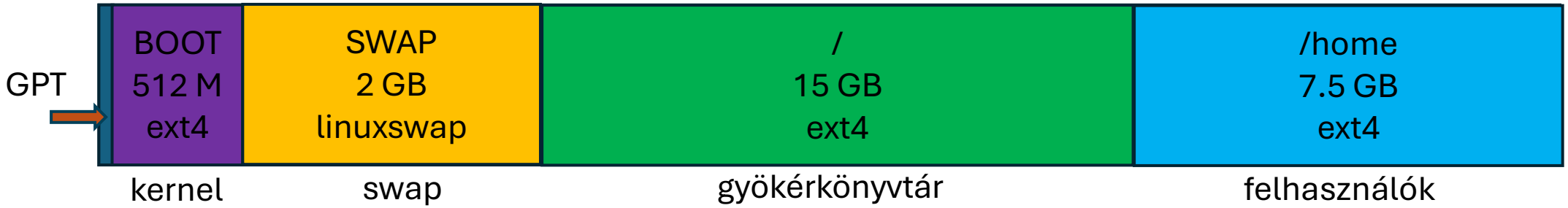
Kezdő telepítés: A telepítő automatikusan beállít mindent. Desktopra jó, szerverre nem

Haladó telepítés:
- Gyári telepítő expert módjában kézzel állítjuk be a partícionálást
- Konfigurálni kell, felosztani a lemezt kötetekre, méretezni kell őket

Profi telepítés:
- Nem használjuk a gyári telepítőt, parancssorból kell mindent telepíteni
- Meg tudunk valósítani speciális beállításokat pl. ZFS + RAID

A linux haladó telepítése – BIOS

Adott egy 25 GB-os VirtualBox lemez, 4 GB RAM-al hozárendelve, a következőképp telepítjük a linux rendszerünket gyári telepítő-varázslóval, alap ext4 filerendszerrel, BIOS rendszerinicializálással



1., Partíciós táblázat létrehozása a lemezre (GPT ajánlott, de lehet MBR is) – BIOS esetén kb. 1 MB a lemez elején

2., a szabad területen létrehozzuk a rendszerbetöltő partíciót 512 MB – erre kerül a kernel, fájlrendszerre beállítjuk az ext4-et, majd csatoljuk, mint /boot

3., a maradék szabad területen létrehozzuk a cserepartíciót (swap), beállítjuk a fáljrendszert linuxswap-ra. A RAM kiterjesztésére szolgál. Csatolási pontja nincs!

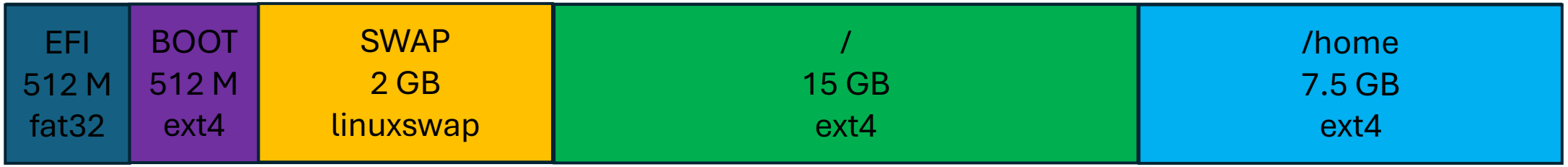
4., a maradék szabad területen létrehozzuk a gyökérkönyvtárat – ez a root 15 GB, ext4 , majd csatoljuk mint /

5., a fennmaradó szabad területen létrehozzuk a felhasználók mappáit – ez a “home” 7.5 GB, ext4 , majd csatoljuk mint /home

Megjegyzés: Nagyobb szervereknél eltolódik már az arány a /home ill. a SWAP irányába (nagyszámú felhasználó, memóriaigény stb.)

A linux haladó telepítése - UEFI

Adott egy 25 GB-os VirtualBox lemez, 4 GB RAM-al hozárendelve, a következőképp telepítjük a linux rendszerünket gyári telepítő-varázslóval, alap ext4 filerendszerrel, UEFI rendszerinicializálással



EFI kernel swap gyökérkönyvtár felhasználók

1., UEFI EFI bináris paríció létrehozása, FAT32, GRUB EFI, ami majd a /boot külön particiója lesz, csatolása /boot/efi. Ezt telepítéskor meg kell adni a GRUB-ra ! GPT/MBR partíciós séma

2., a szabad területen létrehozzuk a rendszerbetöltő partíciót 512 MB – erre kerül a kernel, fájlrendszerre beállítjuk az ext4-et, majd csatoljuk, mint /boot

3., a maradék szabad területen létrehozzuk a cserepartíciót (swap), beállítjuk a fáljrendszert linuxswap-ra. A RAM kiterjesztésére szolgál. Csatolási pontja nincs!

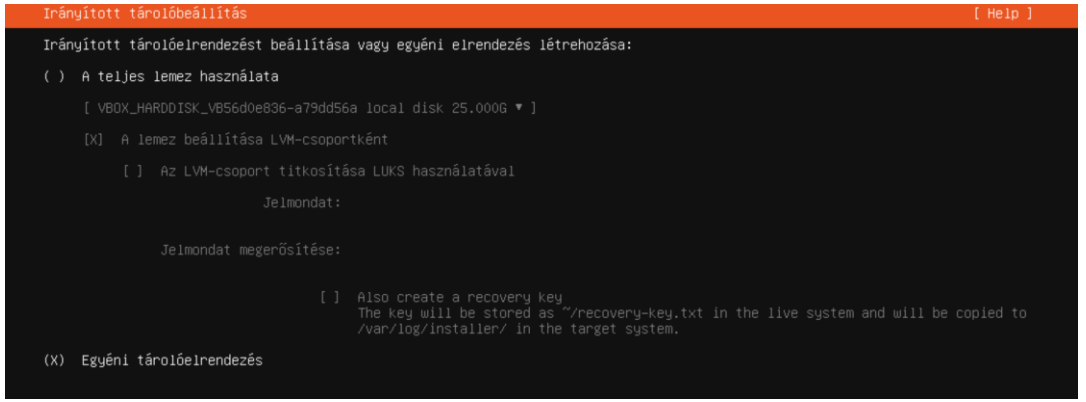
4., a maradék szabad területen létrehozzuk a gyökérkönyvtárat – ez a root 15 GB, ext4 , majd csatoljuk mint /

5., a fennmaradó szabad területen létrehozzuk a felhasználók mappáit – ez a “home” 7.5 GB, ext4 , majd csatoljuk mint /home

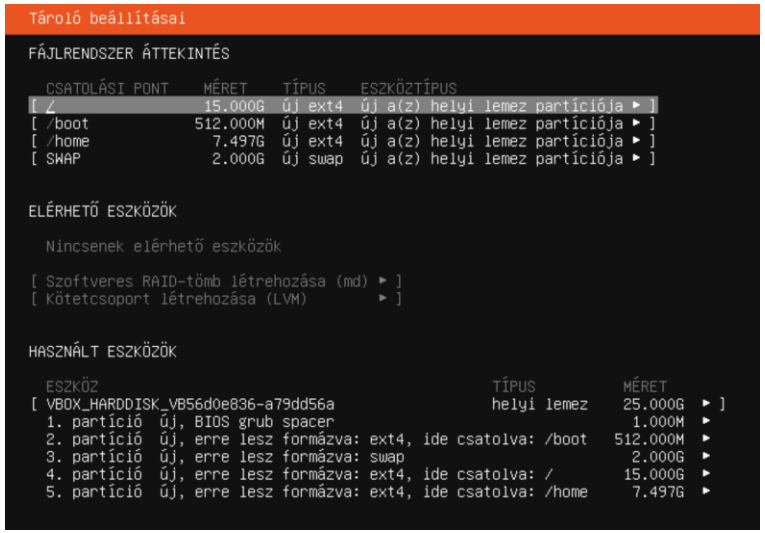
Megjegyzés: Nagyobb szervereknél eltolódik már az arány a /home ill. a SWAP irányába (nagyszámú felhasználó, memóriaigény stb.)

Ubuntu Server telepítés: BIOS+GPT

1. A telepítés folyamán a nyelv kiválasztása után a minimális Ubuntu Server verzsönt válasszuk, amelyet a telepítő varázsló felkínál.



2. Partícionálás folyamán kiválasztjuk az egyéni tárolóelrendezést



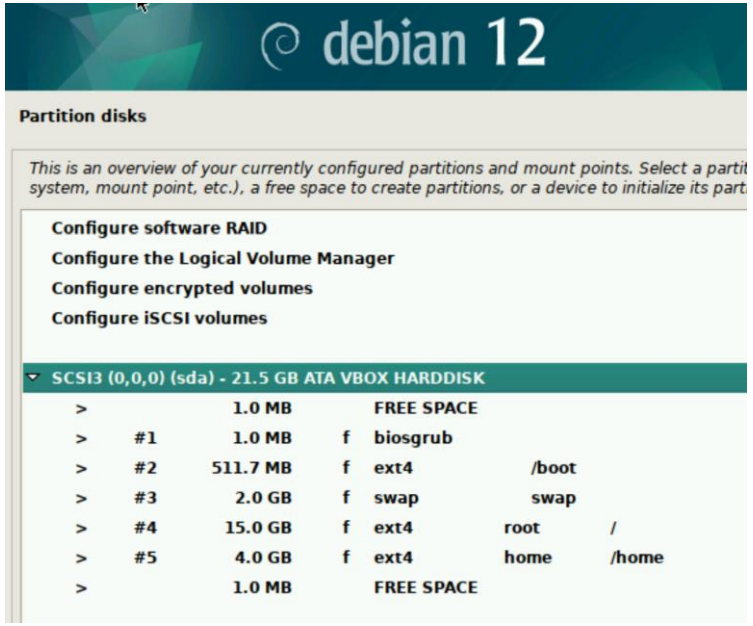
3. A korábbiakban leírtak alapján partícionálunk

1.Partíció (BIOS, GPT partíciós tábla) 1M - alapból megcsinálja a rendszerünk, nem kell vele foglalkoznunk
2.Partíció (kernel) – 512M, ext4, csatolás /boot
3.Partíció (cserepartíció)– 2 GB Swap, swap, nincs csatolás
4.Partíció (gyökérkönyvtár)– 15 GB, ext4 fájrendszer, csatolás /
5.Partíció (felhasználók) 7.497 GB, ext4, csatolás /home

4. A telepítés végén az Ubuntu Server Virtualbox ablaka alján található DVD-Lemez ikonnál kivesszük a telepítőlemezt, majd újraindítjuk

Debian haladó telepítés : BIOS + GPT

1. Választhatunk grafikus ill. sima telepítő között. A lényeg, hogy a Debian alapból az MBR partícionálist erölteti, és nem kínálja fel a GPT-t. Megnyomjuk a TAB-ot, majd az indítási parancsorhoz hozzá kell adni a következő paramétert:
d-i:partman-partitioning/default_label=gpt



2. kiválasztjuk a SCSI3 (0,0,0) (sda) VBOX lemezt , majd a következő lépsben a korábbiakban megfogalmazottak alapján telepítünk kézi partícionálással:

#1.Partíció (BIOS/GPT) – 1 MB, GRUB renszerbetöltő
#2.Partíció (kernel) – 511.7M, ext4, csatolás /boot
#3.Partíció (cserepartíció) – 2 GB Swap, swap, nincs csatolás
#4.Partíció (gyökérkönyvtár)– 15 GB, ext4, csatolás /
#5.Partíció (felhasználók) 7.87 GB, ext4, home, csatolás /home

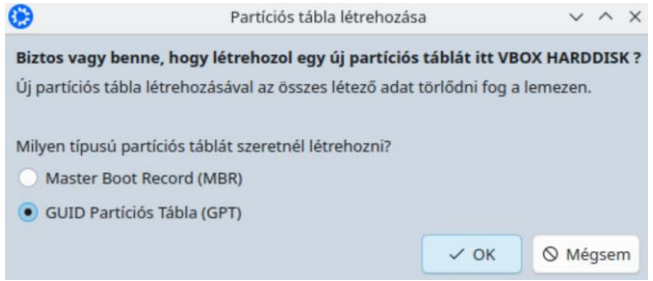
3. Az első #1 partíciónál (GPT, partíciós táblázat telepítése) figyelni kell, hogy biosgrub legyen a típus!

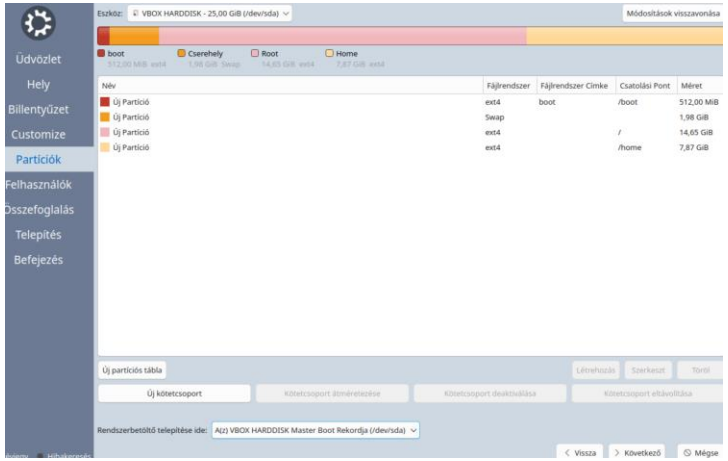
4. Elfogadjuk a partícionálást, és lefuttatjuk a Debian telepítőt, a tasksel-nél csak akkor rakunk grafikus felületet, ha kell, majd a végén GRUB rendszerbetöltőt rárakjuk az sda-ra (amit partícionáltunk).

Kubuntu haladó telepítés: BIOS + GPT

1. A telepítés folyamán a nyelv kiválasztása után kiválszjuk az egyéni partícionálást

2. Partícionálás folyamán kiválasztjuk a GPT partíciós táblázat telepítését





3. A korábbiakban leírtak alapján partícionálunk

1.Partíció – megcsináltuk az előző lépésben.
2.Partíció (kernel) – 512M, ext4, csatolás /boot
3.Partíció (cserepartíció) – 1.98 GB Swap, swap, nincs csatolás
4.Partíció (gyökérkönyvtár)– 14.65 GB,ext4, csatolás /
5.Partíció (felhasználók) 7.87 GB, ext4, csatolás /home

4. Még ugyanezen ablak alján kiválaszjuk a “Rendszerbetöltő telepítése ide” résznél “A(z) VBOX HARRDISK Master Boot Recordjára (/dev/sda)” opciót. Ez telepíti a lemezünk (sda) elejére a rendszerbetöltőt, amelyet a 2.lépésben kiválasztottunk.

5. Alapesetben a telepítő-varázsló lecsatolja a telepítési lemezt, úgyhogy csak újra kell indítanunk a befejezéshez.

Rendszerbetöltő (Boot manager)

A linux boot manager (rendszerbetöltő) elindul a számítógép bekapcsolásakor, és felelős az operációs rendszer betöltéséért. Megkeresi a rendszermag (kernel) helyét, betölti azt a memóriába, majd lefuttatja a kernel kódját.

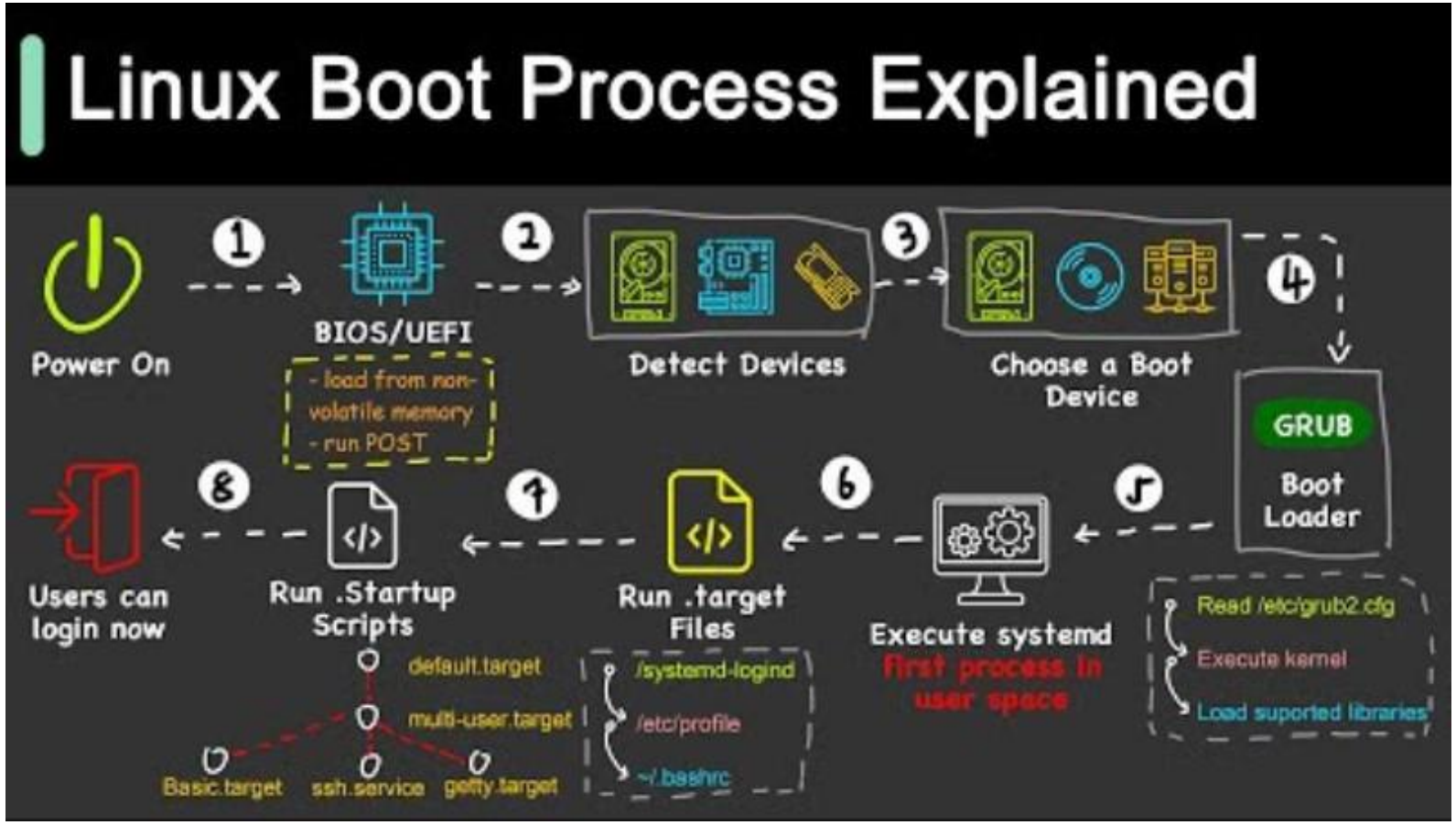
Feladatai: OP rendszer választás, rendszermag (kernel) megkerese, indítási paraméterek betöltése, a kernel betöltése a memóriába, végül magának a kernel kódjának a lefuttatása.

A Boot Manager két legelterjedtebb formája:

- GRUB (GRand Unified Bootloader): Az egyik legnépszerűbb boot manager a Linux rendszereken. Támogatja a különféle operációs rendszerek, például a Linux, Windows és más Unix-szerű rendszerek betöltését.

- LILO (Linux Loader): Egy régebbi boot manager, amelyet manapság ritkábban használnak, de korábban széles körben elterjedt volt.

A rendszerbetöltés folyamata Linux alatt



A GRUB telepítése és parancsai

1. Megnyitjuk a terminált, majd kiadjuk rendszerbetöltő (GRUB) telepítésére a parancsot: **sudo grub-install /dev/sda** .Érdekes lehet a partíciós séma kérdése is (MBR/GPT).

2. A GRUB konfigurációja: a fő konfigurációs fájl az /etc/default/grub mappában található, ezt szerkesztjük a nano szövegszerkesztővel:
sudo nano /etc/default/grub

Ha nincs telepítve, akkor a nano szövegszerkesztőt telepítjük **sudo apt install nano**

Magában a konfigurációs fájlban példák:
GRUB_DEFAULT=0 – az első bejegyzést teszi alapértelmezetté
GRUB_DEFAULT=1 - a második bejegyzést teszi alapértelmezetté
GRUB_TIMEOUT=10 – indítási időzítő
GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true – grub elrejtése, és alapindítása az első bejegyzéssel

3. A GRUB frissítése: **sudo update-grub**

4. Újraindítás (A GRUB érvénybe léptetése): **sudo reboot**

Az MBR lemezpartícinálási séma

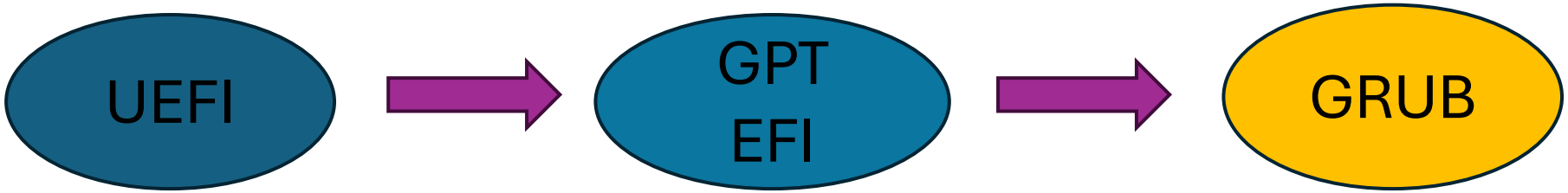
MBR (Master Boot Record) partíciós séma esetén a GRUB-ot a BIOS olyan lemezeken keresi, amelyik tartalmazza ezt. Az MBR három fő részből áll: bootstrap kód, partíciós táblázat, és bootstrap aláírás. Max. lemezméret 2 TB. Ha a lemezünk nem tartalmaz MBR-t, akkor a GRUB-bal együtt kell telepítenünk pl:

sudo apt update - Csomagfrissítés
sudo apt install mbr – Az MBR csomag telepítése
sudo fdisk /dev/sda – Az fdisk alkalmazása, partíciós tábla írása
“n” - új lemez, “p” - megjeleníti a partíciós táblát, “d” - partíció törlése, “w” – változtatások mentése
sudo install-mbr /dev/sda - Ha nincs MBR vagy sérült, akkor az MBR alkalmazása a lemezre
sudo grub-install /dev/sda – a GRUB rendszerbetöltő telepítése ugyanarra a lemezre.
sudo update-grub – a GRUB felfrissítése
sudo reboot - Újraindítás

A GPT lemezpartícionálási séma

A GPT (GUID Partition Table) egy új megközelítése a partíciós sémának. Max. méret 9.6 ZB.
Akár 128 partíciót is támogat. UEFI (Unified Extensible Firmware Interface) részeként jelent meg, és nincs szükség kiterjesztett vagy logikai partíciókra.

sudo apt update - Csomagfrissítés
sudo apt-get install gdisk – a gdisk telepítése
sudo gdisk /dev/sda – gdisk alkalmazása a lemezre, amit elindítva beállítjuk a következő érdékeket, úgy hogy létrehozunk egy ESP-t (EFI System Partition):
("o" parancs – törli a régi partíciókat szükség esetén)
"n" parancs: új lemez, First sector: (hagyd alapértelmezett értéken), Last sector: +512M , Hex code or GUID: EF00 – ez a szabványosított kód, amit olvasni tud az UEFI. "w" – változtatások mentése
sudo mkfs.vfat /dev/sda1 – az új 512MB partíciót FAT32-re formázzuk
sudo mkdir -p /mnt/efi majd a **sudo mount /dev/sda1 /mnt/efi** - EFI partícióként csatoljuk
sudo grub-install --target=x86_64-efi --efi-directory=/mnt/efi --bootloader-id=GRUB – GRUB telepítése
sudo update-grub – grub frissítése
sudo reboot - újraindítás



Lemezpartícionálási eszközök

1., **fdisk** - MBR lemezek kezelésére használják

2., **gdisk** – GPT lemezek partícionálására használjuk

3., **parted** – Rugalmas és hatékony eszköz mind MBR, mind GPT partíciók kezelésére

4., **cfdisk** - Interaktív és szöveges alapú felület az MBR lemezek partícionálásához használjuk

5., **gparted** - Grafikus felhasználói felülettel rendelkező partíciós menedzser, amely mind MBR, mind GPT lemezeket támogatja.

6., **lsblk** – Blokklistázás - áttekinthető és részletes információk a lemezekről és partíciókról.

A linux fájlrendszerek jellemzői

1., **ext4** – teljesítmény, gyorsaság, népszerű, max. 16 TB fájlméret, journaling, nincs snapshot

2., **XFS** – nagy teljesítmény és kapacitás, akár 8 EB méret, journaling, töredezettség

3., **Btrfs** – számos funkció, RAID, deduplikáció, 16 EB, journaling, instabilitás mivel fejlesztés alatt

4., **ZFS** – szerverek, tárolórendszerek, 256 EB, journaling, nagy memóriaigény

5., **F2FS** – SSD-kre és memória alapú kártyákra ideális, 16 TB-ig, journaling, nem optimális HDD-re

A linux fájlrendszerek integritásának fenntartása 1.

Kritikus fontosságú a Linux rendszerek stabilitása és adatbiztonsága szempontjából.
Az integritás megőrzésének főbb lépései és eszközei:

1., Fájlrendszer ellenőrzése a partíción: **sudo fschk /dev/sda1**

2., Naplózás (journaling) – az ext4, XFS, és Btrfs nyilvántartást vezetnek a fájlműveletekről, ami segít a rendszer gyorsabb helyreállításában hiba esetén.
journalctl - a napló megnyitása
journalctl -u sshd – csak az sshd szolgáltatás naplóüzeneteinek megtekintése
journalctl -b – naplózás az utolsó rendszerbetöltés óta
journalctl --since="2024-12-25" – naplózás dátumtól megtekintése
journalctl -p warning – csak a figyelmeztetések megtekintése

3., Fájlrendszer csere hiba esetén – LVM, leállási idő nélkül

4., Snapshotok készítése (fő szubvoume: @):
sudo btrfs subvolume snapshot /mnt/@ /mnt/@_snapshot
Megjegyzés: a visszaállításkor lecsatoljuk a szubvolumot: **sudo umount /mnt/@** , majd felcsatoljuk a snapshotot az sda1-re pl.: **sudo mount -o subvol=@_snapshot /dev/sda1**

A linux fájlrendszerek integritásának fenntartása 2.

5., Rendszeres biztonsági mentés – pl. rsync, tar, és Bacula segítségével
Példa az felhasználói fiókom dokumentumairól másolatot készítek úgy, hogy részletesen archiválom (archive+verbose), és közben törlöm a forrsámappában már nem létező fájlokat a célkönyvtárból is.
sudo rsync -av --delete /home/attila/documents /home/attila/backups/

6., Fájlintegritás ellenőrzés – Az AIDE (Advanced Intrusion Detection Environment) és Tripwire eszközök használatával. Ezek egy adatbázist hoznak létre a rendszer fájljairól, amelyet a rendszer fájljainak integritásának ellenőrzésére és a rendszerszabályozások észlelésére. Számos hash algoritmust támogatnak (md5, sha1, sha256, sha512, rmd-160).

sudo apt install aide	- az aide telepítése
sudo aide --init	- az aide adatbázis inicializációja (elkészítése)
sudo aide --check	- az integritás ellenőrzése

A fájlrendszer létrehozása

1., A kiválasztott lemezen létrehozzuk az MBR partíciót **sudo fdisk /dev/sda** parancs segítségével
n – új partíció
Partition type: p - elsődleges
Partition number: 1 – első helyen
First sector: (alapértelmezett)
Last sector: +20G
w – változások mentése

2., létrehozzuk a filerendszert az mkfs paranccsal az új létrehozott sda1 partícióra – pl.
Ha XFS filerendszert: **sudo mkfs.xfs /dev/sda1**
Ha BtrFS filerendszert: **sudo mkfs.btrfs /dev/sda1**
Ha ext4 filerendszert: **sudo mkfs.ext4 /dev/sda1**
Ha F2FS filerendszert: **sudo mkfs.f2fs /dev/sda1**
Ha ZFS filerendszert: **sudo apt-get install zfsutils-linux**, majd **sudo mkfs.f2fs /dev/sda1**

Init-rendszerek (initialization systems)

Felelős a rendszer indításáért, leállításáért, futtatási szintek kezeléséért, valamint a különböző szolgáltatások és folyamatok elindításáért és kezeléséért. Az init rendszer az első folyamat, amelyet a kernel elindít a rendszer betöltésekor, és ez marad a gyökérfolyamat (PID 1), amelyből minden más folyamat származik.

SysV init - A hagyományos init rendszer, amely a futási szintek (runlevels) koncepcióját használja. A folyamatok sorosan, egymás után indulnak. A konfigurációs fileok a /etc/inittab ill. /etc/rc.d könyvtárban találhatóak

Upstart – Esemény alapú init rendszer, amelyet az Ubuntu fejlesztett ki. A rendszer eseményeire reagálva indítja és állítja le a szolgáltatásokat. A konfigurációs fájlok az /etc/init mappában találhatók.

systemd – Modern init rendszer, engedélyezi a folyamatok párhuzamos indítását, fejlett függőségkezelés, beépített naplózás. Az /etc/systemd/system (módosítások) és a /lib/systemd/system (disztribúció által biztosított) tartalmazza őket

Futtatási szintek (runlevels)

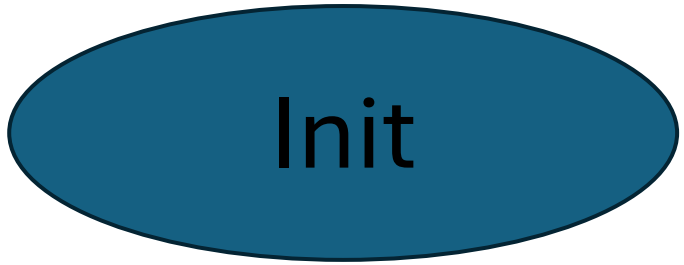
A futtatási szintek (runlevels) a Unix és Linux alapú operációs rendszerek egyik alapvető koncepciója, amely az operációs rendszer különböző állapotait és működési módjait határozza meg. Minden futtatási szint egy meghatározott szolgáltatás- és folyamatkészletet indít el vagy állít le. A futtatási szinteket a rendszer indításakor és működése közben állíthatjuk be. 0-tól 6-ig terjedő számokat használják a futtatási szintek megjelölésére.

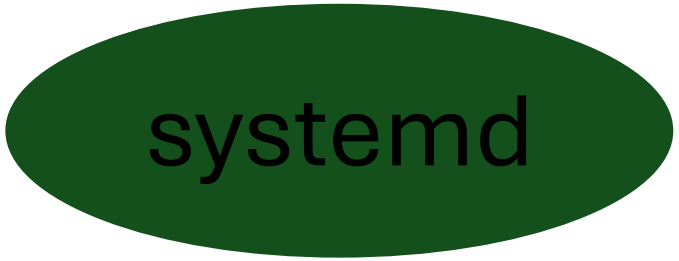
init 0 – shutdown	**poweroff.target**
init 1 – egyfelhasználós mód karbantartásra	**rescue.target**
init 2 – többfelhasználós mód	
init 3 – többfelhasználós mód hálózatokkal	**multi-user.target**
init 4 – nem használatos	
init 5 – többfelhasználós mód hálózat+ablakkezelő	**graphical.target**
init 6 – reboot	**reboot.target**

Init

systemd

Futtatási szintek alkalmazása





a futtatási szint ellenőrzése (nem szükséges a superuser):

runlevel **systemctl get-default**

többfelhasználós mód+hálózatokkal szintváltás:

sudo init 3 **sudo systemctl isolate multi-user.target**

ablakkezelő futtatási szintre váltás:

sudo init 5 **sudo systemctl isolate graphical.target**

ablakkezelő futtatási szint alapértelmezetté tétele:

sudo nano /etc/inittab
id:5:initdefault: - inittab file módosítása
sudo reboot

sudo systemctl set-default graphical.target
sudo reboot

A csomagkezelők

A csomagkezelők olyan eszközök és rendszerek, amelyek segítenek a szoftvercsomagok telepítésében, frissítésében, eltávolításában és kezelésében

A legelterjedtebb linuxos csomagkezelők az :

- APT (Advanced package tool) – Ubuntu alapú disztrok, Debian alapú disztrok (pl. Kali).
- RPM (RedHat Package Manager) – RedHat Ent.Linux RHEL / Fedora linux disztribúciók
- YUM (Yellowdog Updater, Modified) – RedHat Ent.Linux RHEL / Fedora linux disztribúciók
- Zypper – OpenSUSE, SUSE Linux Enterprise disztribúciók
- Portage – Gentoo disztribúció - emerge paracssal.

Egyéb csomagkezelők: DNF (Dandified YUM), Pacman, Flatpak, Snap

Műveletek APT csomagkezelővel

Az APT csomagkezelővel az alapműveletek a következők:

- Csomag telepítése : **sudo apt install csomagnév**
- Csomag eltávolítása: **sudo apt remove csomagnév**
- Csomag eltávolítása + konfigurációjával: **sudo apt purge csomagnév**
- Csomaglista frissítése: **sudo apt update**
- Telepített csomagok frissítése: **sudo apt upgrade**
- Elavult csomagok eltávolítása: **sudo apt autoremove**
- Csomagok keresése: **sudo apt search csomagnév**
- Csomaginformációk megtekintése **sudo apt show csomagnév**

Műveletek RPM csomagkezelővel

Az RPM csomagkezelő nem rendelkezik csomaglistával RedHat rendszereken, csak a YUM. Ezért az alapműveletek korlátozottak, és a következők:

- Csomag telepítése : **sudo rpm -i csomagnév.rpm**
- Csomag eltávolítása: **sudo rpm -e csomagnév.rpm**
- Telepített csomag frissítése: **sudo rpm -U csomagnév.rpm**
- Csomaginformációk megtekintése **sudo rpm -qi csomagnév**

Csomagból telepítés (*.deb)

A *.deb fájlok a Debian-alapú rendszerek, mint például az Ubuntu és a Debian, csomagformátuma. A csomag két archívumot tartalmaz: az egyik a szoftver fájljait tartalmazza, a másik pedig a telepítési információkat (például a vezérlőfájlt és a szkriptet).

sudo dpkg -i csomagnév.deb	- *.deb csomag telepítése
sudo apt-get install -f	- a csomag hiányzó függőségeinek telepítése
sudo dpkg -r csomagnév.deb	- *.deb csomag eltávolítása

Szkript-telepítés (*.run)

A *.run fájlok Linux vagy Unix rendszereken használt telepítő- vagy script fájlok, amelyek tartalmazzák a telepítési utasításokat és a szükséges fájlokat egy adott szoftverhez vagy alkalmazáshoz.

sudo chmod +x csomagnév.run - A csomag futtatásához szükséges végrehajtási jogosultság hozzáadása

./csomagnév.run - a telepítő futtatása

Terméktámogatás és verziófrissítés

Az Ubuntu rendszernek két fő típusú kiadása van: LTS (Long Term Support) és Interim kiadások. Mi az előzőt használjuk. Az LTS rendszer 5 év terméktámogatást jelent. Ez annyit jelent, hogy úgy tervezzük a rendszerünket, hogy ennyi időn belül verziófrissítésre lesz szükségünk !
Az Ubuntu rendszerek nem kínálnak SUSE alapú Tumbleweed rolling jellegű disztrokat.

Az Ubuntu rendszer alapfrissítése:
sudo apt update - lekéri a frissítési csomagokat az apt csomagkezelő számára
sudo apt upgrade - frissítés a lekért csomagokból
(Vagy kiadjuk a kettőt egyben alapelfogadással **sudo apt update && sudo apt upgrade -y**)

Az Ubuntu rendszer verzióléptetése (verziófrissítése) sorrendben:

sudo apt update	- lekéri a frissítési csomagokat
sudo apt upgrade	- frissítés a lekért csomagokból
sudo apt install update-manager-core	- frissítési mechanizmus meglétének ellenőrzése
sudo apt purge	- kitörli a telepített csomagokat
sudo do-release-upgrade	- verziófrissítés (jó azért ha évente megtörténik)

Automatikus frissítés

Az automatikus frissítések beállításához az Ubuntu Serveren a következőket kell csinálni (ha nem fut alapból már a folyamat):

1., Az unattended-upgrades csomag telepítése:
sudo apt install unattended-upgrades

2., Az unattended-upgrades enegdélyezése, elindítása, ellenőrzése:
sudo systemctl enable unattended-upgrades
sudo systemctl start unattended-upgrades
sudo systemctl status unattended-upgrades
sudo systemctl stop unattended-upgrades - az automatikus frissítések letiltása
sudo systemctl disable unattended-upgrades - az automatikus frissítések letiltása

3., Ha szükséges az unattended-upgrades konfiguráció, ahol alapértelmezés szerint csak a biztonsági frissítések vannak beállítva: **sudo nano /etc/apt/apt.conf.d/50unattended-upgrades**

Ezenfelül ha szeretnék E-mail értesítéseket is kapni róla, hogy mikor és mi frissült automatikusan, akkor még feltelepítjük mellé a notifiert: sudo apt install update-notifier-common

A tárház (repository)

Az Ubuntu rendszer a csomagkezelést az APT (Advanced Package Tool) segítségével végzi egy vagy több csomagforrásból azaz tárházból. A tárház (repo, repository) olyan hely, ahol a szoftvercsomagokat és frissítéseket tárolják és karbantartják. Innen a rendszer letölti és telepíti a szükséges csomagokat. A repository-k nagy előnye, hogy központi helyről biztosítanak frissítéseket és új szoftvereket, így a rendszer könnyen naprakész és biztonságos maradhat.

Az Ubuntu alap csomagforrásai a következőek:
1., Main – hivatalos csomagforrás
2., Restricted – nem nyílt forráskódú, de támogatja az Ubuntu
3., Universe – harmadik fél / közösség által karbantartott nyílt forráskódú szoftvercsomagok
4., Multiverse - szoftvercsomagok amelyeket az Ubuntu nem támogat

ls -laF /etc/apt/sources.list.d – a rendszerben futó tárházak(repo) megtekintése

Grafikus driverek tárházának hozzáadása és csomaglista frissítése
sudo add-apt-repository ppa:graphics-drivers/ppa
sudo apt update

Grafikus driverek tárházának eltávolítása és csomaglista frissítése:
sudo add-apt-repository --remove ppa:graphics-drivers/ppa
sudo apt update

Attribútomok

Linux rendszereken a fájlok és könyvtárak attribútumai olyan tulajdonságok, amelyek beállíthatók a fájlokra, hogy különféle viselkedést és jogosultságokat biztosítsanak. Ezek az attribútumok lehetővé teszik a fájlok speciális kezelési módjainak meghatározását, például írásvédettséget, törlésvédettséget stb. A hozzáadás és eltávolítás a “+” ill. “-” jelekkel történik.

A fájlok fő attribútomai:
i – írásvédett (immunable), nem módosítható
a – hozzáfűzött (append) csak hozzáfűzni lehet, módosítani és törölni nem

További attribútomok:
d – nem menthető (no dump)
A – nem frissül az elérési idő (no update time access)
Megjegyzés: A linux nem ismeri a Windows-os rejtett h attibútomot, fájlok neve előtt pont van, az jelenti, hogy a rejtett.

Példák:
sudo chattr +i kismajom.txt - a kismajom.txt legyen írásvédett
sudo chattr -a kismajom.txt – a kismajom.txt ne legyen hozzáfűzött, lehessen módosítani ill. törölni

Alap fájlkezelési műveletek

touch kismajom.txt – új üres kismajom.txt file létrehozása az adott mappában
touch .kismajom.txt - új üres rejtett kismajom.txt file létrehozása az adott mappában

mkdir kenguru – új üres kenguru mappa létrehozása az adott mappában
mkdir .kenguru – új üres rejtett kenguru mappa létrehozása az adott mappában

cp atti.txt kismajom.txt /home/attila/documents – a két szövegfájl másolása a célmappába
cp *.txt /home/attila/documents – az összes szövegfájl másolása a célmappába rekurzívan
cp *.* /home/attila/documents – az összes fájl másolása a célmappába
cp -r /home/attila/pictures /home/attila/documens – a képek mappa másolása a dokumentumokba rekurzívan (almappákkal és azok fájljaival)
cp -i kismajom.txt /home/attila/kismajom.txt – interaktív, tehát megkérdezi felülírás esetén
cp -u kismajom.txt /home/attila/kismajom.txt – update, csak akkor másolja a forrás fájlt, ha a cél fájl nem létezik, vagy régebbi, mint a forrás.

mv atti.txt /home/attila/documents – a szövegfájl áthelyezése a célmappába
mv cica.txt kandur.txt – a szövegfájl átnevezése

rm atti.txt – a szövegfájl eltávolítása
rm -rf egerek – az egerek mappa erőszakos eltávolítása (recursive + force).
rmdir attila – az attila mappa törlése

Egyéb fájlkezelési műveletek

ls -a – az összes fájl és mappa kislistázása az adott mappában (beleértve a rejtetteket is)

cat macska.txt – a macska.txt tartalmának megtekintése
more macska.txt – a macska.txt tartalmának megtekintése lapozva
less macska.txt – a macska.txt tartalmának megtekintése visszagörgetési lehetőséggel
head macska.txt – a macska.txt első 10 sorának megtekintésére
tail macska.txt – a macska.txt utolsó 10 sorának megtekintésére

find /home/attila/documents –name macska.txt – a macska.txt keresése az adott könyvtárban
grep “catwoman” macska.txt – a catwoman kifejezés megkeresése a macska.txt-ben
pwd – (Print Working Directory) megjeleníti a terminál aktuális munkamappáját
cd /home/attila – (Change Directory) mappaváltó útvonalra
.. visszatérés egy szinttel feljebb **~** gyökérmappa **-** visszatérés az előző könyvtárba

Nano és Midnight Commander

Linux szerveren az alapvető műveletekhez tanácsos egy fájlkezelő - Midnight Commander és egy szövegszerkesztő nano feltelepítése úgy, hogy:

1., frissítjük a rendszerünket és hozzáadjuk a universe repositoryt:

sudo apt update && sudo apt full-upgrade -y frissítés függőségekkel együtt

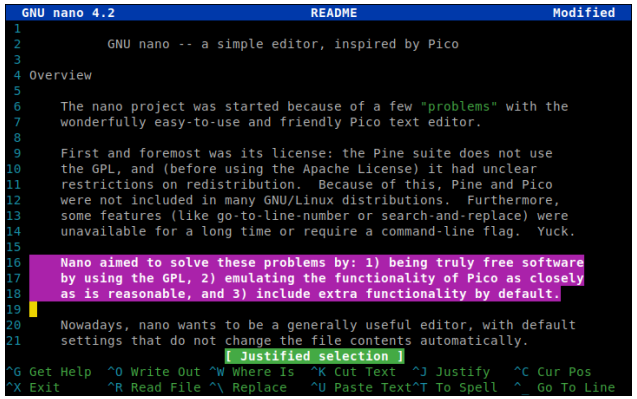
sudo add-apt-repository universe a universe repository hozzáadása az apt csomagkezelőhöz

2., Ha nincs alapból fent, akkor telepítjük a nano -t:

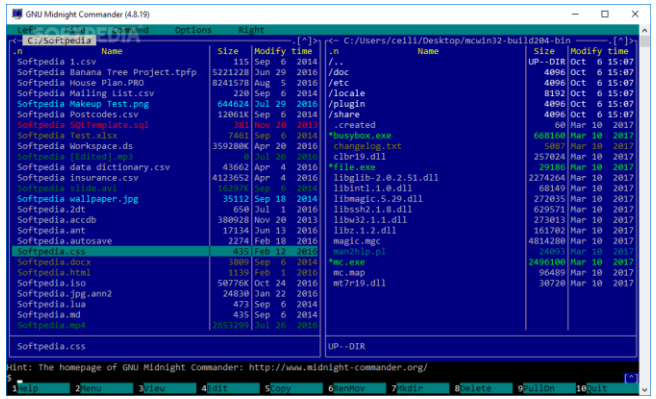
sudo apt-get install nano megjegyzés: az “apt-get install” a régi verziója az “apt install” parancsnak

3., A Midnight Commander telepítése: **sudo apt install mc**

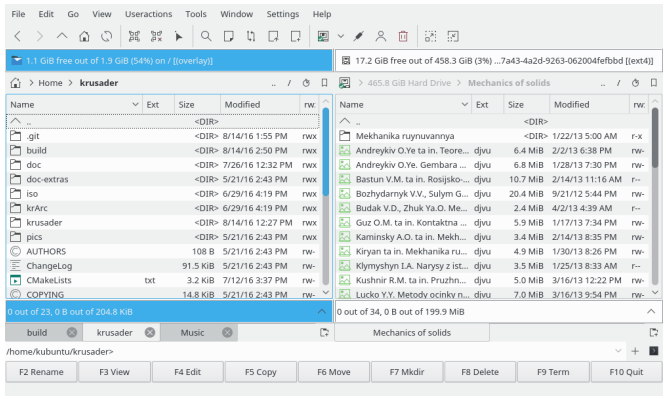
4., Ha grafikus ablakkezelők is van pl. KDE – telepítjük a krusader-t pl. a “kde-full” csomag alapból tartalmazza, de ha csak minimális KDE-nk van, akkor sudo apt install crusader



nano



Midnight Commander



krusader

Fájlok és könyvtárak tulajdonságai

Linux rendszerben a fájlok és a könyvtárak tulajdonságainál három dolog a mérvadó: jogosultság, tulajdonos, fájltípus.

Jogostulstági szintek:

r – olvasási jogosultság
w – írási jogosultság
x – végrehajtási jogosultság

Tulajdonlás:

u – user, tulaj
g – csoport
o – others, egyéb

Fájltípusok:

- – fájl
d – könyvtár (directory)
l – szimbolikus link (symlink)
b – blokk (block)
c – karakter (character)
p – cső (pipe)
s – csatlakozó (socket IP+port)

Jogosultságok és példák

Fájl jogosultságainak megtekintése az attila felhasználó mappájában: **ls -laF ~/papagájok.txt**

Adott a pelda.txt fájl, és ennek szeretnénk módosítani a jogosultságait úgy, hogy a tulajdonos mindent csinálhasson, a csoport tagjai csak olvashasság és végrehajthassák ill. mások csak olvashassák az állományt.

sudo chmod u+rwx,g+rx,o+r pelda.txt

Fájlok és könyvtárak tulajdonosának és csoportjának módosítására példa:
Adott a /home/jozsi/attilatolloptam mappa, és azt akarom, hogy én mint attila felhasználó a teachers csoportból birtokba vehessem

sudo chown -R alexovics:teachers /home/jozsi/attilatolloptam
Ahol a –R az a rekurzív módosítás, amely az összes al-fájlt és al-könyvtárat is érinti

Változtassuk meg az attila home mappájában található jegesmedve.txt csoportját medve nevezetűre:

sudo chgrp medve /home/attila/jegesmedve.txt

Jogosultságok egyéb példák

Adott a jupiter.txt fájl, és ennek szeretnénk módosítani a jogosultságait úgy, hogy a tulajdonos mindent csinálhasson, a csoport tagjai ill. mások csak olvashasság és végrehajthassák.

sudo chmod u=rwx jupiter.txt
sudo chmod go=rx jupiter.txt

Adott a szaturnusz mappa, és ennek szeretnénk módosítani a jogosultságait úgy, hogy a tulajdonos mindent csinálhasson, a csoport tagjai csak olvashasság és végrehajthassák, másoktól viszont minden jogosultságot megvonunk.

ls – lekérjük a mappánk tartalmát, hogy tartalmazza-e a szaturnusz-t mint almappát. Ha nem tartalmazza, akkor megkeressük **cd..**

sudo chmod u=rwx szaturnusz/
sudo chmod g=rx szaturnusz/
sudo chmod o=- szaturnusz/

Jogosultságok numerikus kódok

Ahhoz, hogy ne kelljen állandóan kiírni a jogosultság paramétereit (az előző példában az u+rwx,g+rx,o+r) bevezették a numerikus kódok használatát.

Minden számjegy az alábbi jogosultságok kombinációját képviseli:
r (read) = 4
w (write) = 2
x (execute) = 1

7 = 4 + 2 + 1 - rwx – olvasás, írás és végrehajtás
6 = 4 + 2 - rw – olvasás és írás
5 = 4 + 1 - rx – olvasás és végrehajtás
4 = 4 - r – csak olvasás
3 = 2 + 1 - wx – írás és végrehajtás
2 = 2 - w – csak írás
1 = 1 - x – csak végrehajtás
0 = - nincs jogosultság

chmod u+rwx,g+rx,o+r pelda.txt → **chmod 754 pelda.txt**

Az ACL (Access Control List)

Az ACL - részletes jogosultságok beállítására használják felhasználók és csoportok számára már létező fájlokon és könyvtárakon. Segítségével finomabb kontrollt biztosíthatunk a fájlok és könyvtárak hozzáférése felett lehetővé téve egyes felhasználók ill. csoportok jogosultságait.

getfacl /útovonal/fájl - Az ACL megtekintése fájlokon és könyvtárakon

setfacl -m u:attila:rw kismajom.txt - A felhasználó jogosultságainak beállítása
setfacl -m o:x kismajom.txt - A többi felhasználó jogosultságainak beállítása

setfacl -x g:teachers kismajom.txt - A csoport jogosultságainak törlése

setfacl -b kismajom.txt – Minden ACL törlése

setfacl -R -m u:attila:rw /home/attila – Rekurzív mód (az összes mappára és almappára vonatkoz.)

setfacl -m d:u:attila:rw kismajom.txt - A felhasználó jogosultságainak alapértelmezetté tétele

setfacl -k kismajom.txt – Alapértelmezett ACL-ek törlése

Felhasználók és csoportok

Felhasználó:

useradd	- hozzáadás
usermod	- módosítás
userdel	- törlés
passwd	- jelszó

Csoport:

groupadd	- hozzáadás
groupmod	- módosítás
groupdel	- törlés
gpasswd	- jelszó beállítása csoportoz,
gpasswd -a	- felhasználó hozzáadása csoprthoz
gpasswd -d	- felhasználó törlése csoportból

Alap példák:

sudo groupadd teachers - Az új teachers csoport létrehozása

sudo useradd -m -g teachers attila – Az attila hozzáadása a tanárokhoz, elsődleges cs. új mappával

sudo passwd attila – Az attila felhasználó jelszavának megadása

A Linux és az AD közti különbségek

Amíg a Windows Server Címtárszolgáltatója (Active Directory) fa struktúrájú hierarchia, addig a Linux elsődleges g és másodlagos (kiegészítő) csoportokat G definiál.

Minden felhasználónak kötelezően van egy elsődleges alapértelmezett csoportja, amely az /etc/passw fájlban van meghatározva.
Figyelem! Felhasználó létrehozásakor ha nem adjuk meg a -g paramétert (pl. **sudo useradd attila**) , akkor a linux rendszer automatikusan attila nevű elsődleges csoportot hoz létre attila felhasználónak!!
Érdemes ellenőrizni - ez a parancs kiiratja az elsődleges csoportot: **id -gn attila**

Egy felhasználó viszont több másodlagos csoporthoz is tartozhat. Ezek a csoportok további hozzáférési jogosultságokat biztosítanak.
Az elsődleges és másodlagos csoportjaim neveinek kiiratása: **id -Gn attila**

Példa: Hozd létre a veronika felhasználót új mappában, akinek az elsődleges csoportja wife, másodlagos csoportjai munka,makeup: **sudo useradd -m -g wife -G munka,makeup veronika**

Példa: Adjuk hozzá az előzőekben létrehozott felhasználót a cook másodlagos csoporthoz, majd távolítsuk el a munka másodlagos csoportjából:
sudo usermod -aG cook veronika - aG az appendgroup, hozzácsatol úgy, hogy nem veszíti el a másodlagos csoport jogosultságait
sudo gpasswd -d veronika cook – veronika felhasználó eltávolítása a cook csoportból
sudo usermod -g diakok tanulo – a tanulo felhasználó áthelyezése a diakok elsődleges csoportba

Lemezkvóták beállítása és ellenőrzése 1.

Többfelhasználós környezetben a lemezkvóták beállításával azt szabályozhatjuk, hogy az egyes felhasználók vagy csoportok mennyi területet használhatnak fel. Megakadályozhatjuk, hogy egyetlen felhasználó monopolizálja a lemezerőforrásokat. Ennek a menete a következő.

1., engedélyezzük a kvótákat: **sudo apt update** , majd **sudo apt install quota quotatool**

2., /etc/fstab szerkesztés, majd csatolás
sudo nano /etc/fstab –Pl. /dev/sda1 /home ext4 defaults,usrquota,grpquota 0 2 (dump – nincs fájlrendszer mentés, fscheck rendszerindításkor minden más fájlrendszerre a megadott sorrendben)
sudo mount -o remount,usrquota,grpquota /home – a /home újracsatolása

3., kvótafájlok létrehozása (inicializálása) és ellenőrzése:
sudo quotacheck -cum /home - a fájlrendszeren található kvótákat ellenőrzi és inicializálja.
sudo quotacheck -avugm - az összes kvótával rendelkező fájlrendszeren ellenőrzést végez
sudo quotaon -v /home – bekapcsolja a kvótákat

c - Új kvótafájlokat készít, ha még nem léteznek
u - Felhasználói kvótákat ellenőriz
m - Kikapcsolja a naplózás figyelmeztetéseit a futás közben
a - Minden fájlrendszeren futtatja a kvótaellenőrzést
v – verbose, részletes kimenet
g - Ellenőrzi a csoport kvótákat

Lemezkvóták beállítása és ellenőrzése 2.

4. A kvóta beállítása a felhasználó számára:
Példa: Be szeretnénk állítani egy kvótát attila felhasználó számára, amely megengedi hogy 10000 blokknyi adatot (kb.5 GB) írjon, és figyelmeztesse 12,000 blokknál.
Megjegyzés: Az ext4 fájlrendszernél 1 blokk az 4096 byte = 4 KB = cca. 0.004 MB

sudo edquota attila – erre a parancsra kijön egy táblázat, amit szerkeszteni kell

Disk quotas for user attila (uid 1001):

Filesystem	blocks	soft	hard	inodes	soft	hard
/dev/sda1	10000	12000	15000	0	0	0

blocks: Az aktuális lemezhasználat blokkban kifejezve
soft: A puha korlát, amely figyelmezteti a felhasználót
hard: A kemény korlát, amelynél a felhasználó nem tud további adatokat írni
inodes: Mappák és fájlok száma, és erre a soft és hard jellemző

5. Kvóták ellenőrzése - megjeleníti a kvóta használati adatokat minden felhasználóra és csoportra vonatkozóan: **sudo repquota /home**

Az Inode

Az inode (index node) egy adattárolási struktúra is egyben a fájlrendszerekben. amely egy fájl vagy könyvtár összes fontos információját tárolja, kivéve a fájl nevét és az aktuális adatokat. Az inode lehetővé teszi a fájlrendszer számára, hogy hatékonyan kezelje a fájlokat és könyvtárakat.
Mint szám jelenítődik meg – ez az inode szám – egy egyedi azonosító szám, nem összegszám!

Az inode az alábbi információkat tartalmazza:

- Fájl típusa (fájl, könyvtár, szimbolikus link, stb.)
- Hozzáférési jogok (olvasási, írási és végrehajtási jogosultságok)
- Tulajdonos felhasználó azonosítója (UID)
- Csoport azonosítója (GID)
- Létrehozási, módosítási és hozzáférési időbélyegek
- Hivatkozási számláló (az inode-ra mutató fájlok száma)
- Adattáblázatok (a fájl tényleges adataihoz vezető blokkok mutatói)

Példa: adott az attila.txt fájl. Mennyi inode-dal rendelkezik?
ls -i attila.txt – és megjelenik a fálj egyedi inode száma

Hardlink és Symlink

A hardlink (kemény hivatkozás) egy adott fájl közvetlen hivatkozása, amely ugyanazt az inode-ot használja. Mivel a hardlinkek ugyanazt az inode-ot osztják meg, bármilyen módosítás bármelyik hardlinken keresztül módosítja az eredeti fájlt is. Azonos inode számmal rendelkező hardlinkek ugyanarra a fájlra mutatnak – tehát, ugyan az az inode.
Figyelem, linuxnál nem működik a Windows mappa hardlinkelés (mklink /j), csak fájlokra

ln attila.txt hardlink.txt – hardlinkelés
ls –i attila.txt hardlink.txt – kiírja a két szövegfájl inodját (ha, egyezik akkor hardlinkelve vannak)
rm hardlink.txt – törli a hardlinket

A symlink (szimbolikus hivatkozás) egy symlink egy fájl vagy könyvtár hivatkozása, amely tartalmazza a cél fájl vagy könyvtár elérési útját. A symlink tulajdonképpen egy mutató, amely egy másik fájlra vagy könyvtárra mutat. Symlinkeknél az inode-ok segítenek a fájlrendszernek követni, hogy egy link milyen fájlra vagy könyvtárra mutat – tehát, külön inode
Itt már működik a mappák symlinkelése is, úgy ahogy Windowsban.

ln -s /home/user/attila/kutya /home/user/attila/linkek/kutya – szimbolikus hivatkozás két mappa közt
ls -l /home/user/attila/linkek/kutya – szimbolikus hivatkozás ellenőrzése, kiirja a hivatkozás mutatóját
rm /home/user/attila/linkek/kutya – szimbolikus link törlése

Programok telepítése forrásból

A Linux rendszerek egyik előnye, hogy ismert vagy nyílt forráskódból tudjuk telepíteni a programokat, nincs szükségünk “előre csomagolt” build-ekre, mint ahogy a Windows esetében, de akár mint a linuxos csomagkezelők által kínált csomagokra.

Előny: a kódnak csak azon komponenseiből építi meg a szoftvert, amelyre a gépünknek szüksége van
Hátrány: sok esetben nagyon időigényes a build folyamata

A forrásból telepítés menete a következő:

1., Az alapvető build fejlesztői eszközök telepítése: **sudo apt install build-essentials**

2., A program forráskódjának letöltése és kicsomagolása: direktben, git, stb.

3., Konfigurálás (a függőségek megoldása): a forrás mappa tartalmaz egy README vagy egy INSTALL file-t, amely tartalmazza a konfigurációs lépéseket, majd futtatjuk a vonatkoztatott mappából a **./configure** parancsot. A konfigurálás folyamán ha hiba van, akkor még van függőség, vagy ütközés, amelyet meg kell oldanunk. Ez az időigényes rész sok esetben

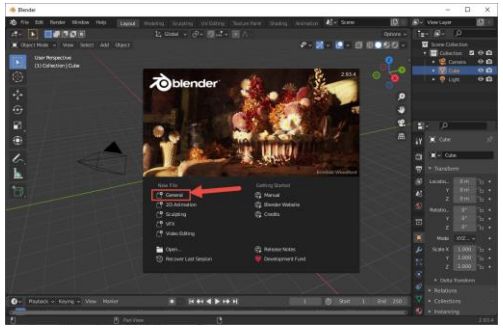
4., Fordítás és Telepítés – ha lefut sikeresen a ./configure parancs, akkor nekikezdhetünk a program fordításának és telepítésének a következő paranccsal: **make** , majd **sudo make install**

Programok telepítése forrásból példák

1., Alapvető programoknál ez a következőképp néz ki, ha a konfigurációt (függőségeket elrendeztük). Adott egy “kecskepapagáj” nevű program forráskódban, amely megnyit egy ablakot abban egy képet.

letöltjük, kicsomagoljuk a kódot, majd:
cd kecskepapagáj
./configure
make
sudo make install

2., Bonyoltultabb programoknál kész dokumentáció van a konfigurációra – pl. blender fordítása



- Átolvassuk a fejlesztők által kiadott dokumentációt (developer.blender.org), majd kiválasztjuk az OP renszert és azon belül az Ubuntu disztro családot.
- Fejlesztési keretrendszerek telepítése **sudo apt install python3 git git-lfs**
- Letöltés git-tel: **git clone** https://projects.blender.org/blender/blender.git
- Alapfüggőségek telepítése a dokumentáció szerint
- Ha van ./configure, vagy egyéb konfigurálási tanács megnézzük, processzorfüggőség, előre-fordított könyvtárak stb.
- **make update** majd, ha lefut **make** –el mehet a build

A Parancssor (shell)

A parancssor (más néven héj, terminál, konzol, shell) egy olyan réteg, amely a Linux kernel fölött helyezkedik el, és közvetlenül a felhasználóval kommunikál.
Másnéven: A shell felhasználói felület a kernelhez.
A bemenet és kimenet szöveges amely lehetővé teszi, hogy parancsokat adjanak ki az emberek a számítógépnek. Ezen parancsokat a gépek feldolgozzák, majd a kimeneti eredményeket szöveges formában visszaadják a felhasználóknak, akik ezeket értelmezik.

Működési elve:

1. Parancs Beírása: A felhasználó beír egy parancsot a shell promptba.

2. Parancs Értelmezése: A shell értelmezi a beírt parancsot és ellenőrzi a szintaxist

3. Parancs Végrehajtása: A shell a kernelhez továbbítja a parancsot, amely végrehajtja azt.

4. Eredmény Visszaadása: A kernel végrehajtja a parancsot és visszaadja az eredményt a shellnek, amely megjeleníti azt a felhasználónak.

Parancssori (shell) beállítások 1.

1., a. A default shell ellenőrzése (bash alapértelmezett) **echo $SHELL**

b. Egyéb shell kiválasztása (zsh, fish):
sudo apt install zsh - telepíti a zsh shell-t
chsh -s /bin/zsh attila - megváltoztatja a parancssort (change shell), s – shell, attila usernek

2., Álnevek (alias) használata - hosszabb parancsokat leegyszerűsít

alias update='sudo apt update && sudo apt upgrade -y'

3., Környezeti változók (Environment) beállítása:

a. **export PATH=$PATH:/home/attila/kepek/kutyak/jrt** - elérési útvonalak (PATH)

b. **export LANG=hu_HU.UTF-8** - nyelv (LANG)

c. **PS1='\u@\h:\w\$ ‘** - prompt string (PS1) – a shell kinézetéért és színéért felelős

Parancssori (shell) beállítások 2.

4. Színes shell beállítása

nano ~/.bashrc - megnyitjuk a bashscriptet nano-ban
force_color_prompt=yes - majd hozzáadjuk a színes shell parancsát

5. A Shell History jobb átláthatósága (pl. bash):

export HISTCONTROL=ignoredups:ignorespace - nem menti az egymásutáni ismétlődő parancsokat, sem amelyek szóközzel kezdődnek
export HISTSIZE=1000 - 1000 parancsot tároljon a parancstörténet a memóriában
export HISTFILESIZE=2000 - ~/.bash_history parancstörténeti fájlban tárolt parancsok mennyisége

Magát a Shell Historyt a következőképp indíthatom el:
history - listázza ki a már kiadott parancsokat
history 10 - listázza ki az utolsó10 parancsot
!42 - az n.edik parancs (ez esetben 42.) végrehajtása
!! - az utolsó parancs újrafuttatása
history –c - törölje a historyt

Napi munka Linux-szerveren

1., Csomagfrissítés és telepítés: **sudo apt update && sudo apt upgrade -y**

2., Rendszerkarbantartás: **sudo apt install htop vim curl wget git**

htop – rendszerfigyelés - valós idejű információk megjelenítése: CPU, memória, folyamatok
vim – (Vi Improved) fejlett szövegszerkesztő
curl – Fájlok letöltése és feltöltése, adatlekérdezések API-któl, amely URL-ek segítségével adatokat továbbít és kap különböző protokolokon keresztül (HTTP, HTTPS, FTP stb.)
wget – Nagyobb fájlok letöltése, webes tartalmak mentése és tükrözése
git - Elosztott verziókezelő rendszer, amelyet a forráskód fejlesztéséhez használnak, branchek stb.

3., Hálózatkezelés: **sudo apt install net-tools**
A net-tools tartalmazza az **ifconfig** (interfészek konfigurálására) és a **netstat** (hálózati statisztika) parancsokat

4., Szervernapló kezelés: **sudo apt install logrotate**

5., Feladatkezelés: **sudo apt install cron**
crontab -e - az ütemezett feladatok megjelenítése

6., Backup: **sudo apt install rsync**
rsync -avz /home/attila/ attila@rozsa.bimbo:/home/attila/ - mentés egy távoli szerverre
a – rekurzív másolás
v – verbose, részletes
z – tömörítve

Hasznos eszközök Linux-szerveren

1., Automatizálás és menedzsment (Ansible): Hatékony eszköz, amely automatizálja a konfigurációkezelést, a szoftvertelepítést és a feladatokat a számítógépes hálózatokon – pl. az nginx webszerver telepítése és konfigurálása több szerveren, konfigurációs fájlok másolása, docker konténerek kezelése: **sudo apt install ansible**

2., Webes adminisztráció (Apache): nyílt forráskódú Apache Software Foundation által fejlesztett és karbantartott szoftver, amely stabil és megbízható megoldást kínál a weboldalak és webalkalmazások kiszolgálására.
sudo apt install apache2
sudo systemctl start apache2
sudo systemctl enable apache2

3., Tűzfalkezelés (ufw): Uncomplicated Firewall, egy egyszerűsített tűzfalkezelő eszköz, amelyet az iptables fölé építettek az Ubuntu és Debian rendszereken
sudo apt install ufw - az ufw telepítése, ha nincs fent
sudo ufw allow OpenSSH - az SSH biztonsági komm.protokoll engedélyezése a 22-es porton
sudo ufw enable - ufw engedélyezése
sudo ufw status - a tűzfal ellenőrzése

Szövegfeldolgozás Linuxon: grep

A grep – az hatékony parancssori kereső.

Példa: **grep ’mérges’ kakadu.txt** - a mérges szó keresése a kakadu.txt-ben

Bemenet sorok:
A kakadu nagyon vidám.
A kakadu ma mérges volt.
A kakadu szereti a diót.

Kimenet:
A kakadu ma mérges volt.

Egyéb grep használat:

- rekurzív keresés: **grep -r ’szürke’ /home/attila/papagájok**

- csak a sorok számának megjelenítése: **grep -n ’szürke’ /home/attila/papagájok/jákó.txt**

Szövegfeldolgozás Linuxon: sed

sed - sorok szövegeinek helyettesítésére, sorok kiiratására használják:

Példa:
sed ’s/kakadu/kecskepapagáj/g’ kakadu.txt
s – helyettesítés (substitute)
g – az összes előfordulás

Bemenet sorok:
A kakadu nagyon vidám.
A kakadu ma mérges volt.
A kakadu szereti a diót.

Kimenet:
A kecskepapagáj nagyon vidám.
A kecskepapagáj ma mérges volt.
A kecskepapagáj szereti a diót.

További példák:
sed '/repül/d’ arapapagáj.txt - az összes sort törli amelyben előfordul a repül szó, de nem írja át a fájlt, csak megjeleníti, kimenetként.
sed -i '/repül/d’ arapapagáj.txt - átírja a fájlt (i - inplace) is az előzőekhez képest
sed '/repül/d’ arapapagáj.txt > kalitkában.txt - elmenti egy új szövegfájlba a szabványos kimenetet
(Az operátor > Alt+62, < Alt+60 a magyar billentyűzeten, ez meg a szabványos bemenet)
sed -n ’ 5p; 7p; 23p’ arapapagáj.csv > kivonat.csv – az 5.,7.,23. sor csv fájlba írása, n csendes mód
sed -n ’ 2,4p’ arapapagáj.csv > kivonat2.csv – a 2-4.sorok csv fájlba írása

Szövegfeldolgozás Linuxon: cut

cut - parancs egy hatékony eszköz, amely lehetővé teszi a szövegfájlok sorainak oszlopok vagy oszlopok mentén történő feldarabolását és kivágását. Ez különösen hasznos olyan feladatokhoz, mint a CSV fájlok kezelése, oszlopos adatok feldolgozása, és más hasonló szövegfeldolgozási műveletek.
Példa: **cut -d ',' -f 1,2 papagájaim.csv > kivonat.csv** – az első két oszlopot használjam egy CSV fájlból, majd mentsem el egy kivonat csv file-ba

Bemeneti CSV sorok:
1,Csiri,hullámospapagáj
2,Fülöp,szürkepapagáj
3,Baba,kecskepapagáj

Kimeneti CSV sorok:
1,Csiri
2,Fülöp
3,Baba

Vágjuk ki egy fájl minden sorának első két karakterét: **cut -c 1-2 majmok.txt**
Csak az 1. és 2. karaktert hagyjuk meg minden sorban: **cut -c 1-2 --complement majmok.txt**

c – karakter (character)
d – elválasztó (delimiter)
f – mező (field)
--complement - ellentétes

Szövegfeldolgozás Linuxon: tr

tr - (translate) parancs egy egyszerű és hasznos eszköz a karakterek átalakítására és cseréjére a szöveges adatfolyamokban. Elsősorban egyszerű karaktercserékre és törlésekre használják, és gyakran alkalmazzák adatfeldolgozásra és szövegszerkesztésre különböző szkriptekben. Példa:
tr 'a' 'A' < papagájaim.txt > viccesszöveg.txt
a papgájaim.txt szövegében felcseréli a kis a-betűket nagy A-ra, majd elmenti viccesszöveg.txt-nek.

Bemeneti szöveg:
Csiri egy hullámospapagáj
Fülöp meg egy szürkepapagáj
De a Baba már kecskepapagáj

Kimeneti szöveg:
Csiri egy hullámospApAgáj
Fülöp meg egy szürkepApAgáj
De A BAbA már kecskepApAgáj

További példák:
tr 'a-z' 'A-Z' < papagájaim.txt - az összes kisbetű átalakítása nagyra
tr -d 'a' < papagájaim.txt - az összes a betű eltávolítása a szövegből
tr -c 'a' 'X' < papagájaim.txt - komplementer csere a-betűn kívül minden X-re papagáj → XaXaXXX
tr -s 'a' 'X' < papagájaim.txt – az ismétlődő karaketerek összevonása (squeeze) papaaagáj → papagáj

Szövegfeldolgozás: kombinációk

Az életben sok esetben a grep, cut, sed, tr kombinációja, illetve ezek kimenetének további felhasználása és mentése történik meg.

Példa: a papgájaim.csv táblázatban cseréljük fel a pontosvesszőket vesszőkre, majd ennek kimenetét továbbhasználva javítsuk át az összes Jacky nevet Babára.

tr ';' ',' < papagájaim.csv | sed ’s/Jacky/Baba/g’ > javított.csv

| - a pipe (magyar billentyűzet AltGr + W) a kimenet további felhasználást jelenti.

Bemeneti csv:
1;Csiri;hullámospapagáj;kék
2;Fülöp;szürkepapagáj;szürke
3;Jacky;kecskepapagáj;zöld

Kimeneti csv:
1;Csiri;hullámospapagáj;kék
2;Fülöp;szürkepapagáj;szürke
3;Baba;kecskepapagáj;zöld

echo "A 70 Ronin" | tr –cd ’0-9’ - Csak a számokat írja ki a szöveges bemenetből. Kimenet: 70
echo -e "Első sor\nMásodik sor" – 2 sorba rakja a kimenetet. Megjegyzés: a \ backlash operátort, fordított perjelet, sorokkal való műveletekre használják szövegekben (magyar billentyűzeten az AltGr + Q)

Reguláris kifejezések

A reguláris kifejezések (regex vagy regexp) egy speciális szintaxis használatával definiált mintázatok, amelyek lehetővé teszik a szövegek keresését, illesztését és manipulálását. Ezek hatékony eszközök a szövegelemzéshez, szövegmódosításhoz és szövegkereséshez parancssori környezetekben. Az osztályozásuk a következőképp történik:

Pont (.): Bármely egy karakter . Pl. h.t - szövegben egyezik a “hat”, “hát”, “hét” szó.

Csillag (*): Karakter ismétlődés nullától. Pl. ha*t – szövegben “ht”, “hat”, “haaaaat”

Plusz (+): Karakter ismétlődés egytől. Pl. ha+t – szövegben “hat”, “haaaaat”

Kérdőjel (?): Karakter ismétlődés nullától egyig. Pl. ha?t – szövegben “ht”, “hat”

Szögletes zárójel ([]): Karakterosztály - bármely karaktert jelent a zárójelek között.pl: [aeiou]

Kapcsos zárójel ({}): Karakterosztály - Meghatározott számú ismétlődés o{2} megegyezik “oo”

Pipe (|) : Alternatívák használata- pl. kutya|macska megegyezik a vagy kutya vagy macska szóval

Caret (^) : A sor eleje - pl. ^szia – ha a sor elején van, akkor egyezés

Dollár ($) : A sor végét jelzik - pl. ámen$ – ha a sor végén van, akkor egyezés

Perjelek (/../) : Perjelek közti rész a keresett kifejezés - pl. /egér/ – ha van a szövegben, egyezés

Reguláris kifejezések példák

grep ‘m.ki’ kismajom.txt – ha van maki szó, akkor eredményt ad

Adott a papagájok.txt, amelyben előfordulnak a kecskepapagáj és az arapapagáj szavak. Azt szeretném, hogy a kimenetben írja át ezeket kakadura, majd az eredményt mentse el javított.txt-ként

A kecskepapagáj fehér.
A szürkepapagáj tarajos.
A kecskepapagáj okos.
A szürkepapagáj mérges.

A kakadu fehér.
A kakadu tarajos.
A kakadu okos.
A kakadu mérges.

sed 's/kecskepapagáj\|szürkepapagáj/kakadu/g’ papagájok.txt > javított.txt

s – helyettesítés
\| - Fordított perjel + pipe választó : a pipe karakter a reguláris kifejezésekben egy speciális karakter, és a sed parancs esetében escape-elni kell.

Az AWK feldolgozó

Egy sokoldalú parancssori eszköz és programozási nyelv. Szöveges adatfeldolgozásra, jelentéskészítésre és az adatmanipulációra terveztek. Különösen hasznos, ha strukturált szöveges adatokat kell feldolgozni, mint például CSV fájlokat vagy logfájlokat. A $ mezők tabulátorral vagy szóközzel vannak elválasztva.

awk '{print}' szerverlista.txt - a fájl kiiratása

awk '{print $1, $3}' szerverlisa.txt - kiírja az első és a harmadik mezőt minden sorból

awk '/192/ {print}' szerverlisa.txt - kiírja az összes sort, ahol előfordul a “192” minta

awk '{sum += $1} END {print sum}' sum.txt - összeadja az első mező összes értékét és kiírja az eredményt, amikor a fájl végére ér.

awk '{if ($1 > 100) print $1}' 100+.txt - kiírja azokat a sorokat, amelyek első mezőjének értéke nagyobb, mint 100.

Szabványos csatornák

A szabványos csatornákat – bemenet, (**< ,** stdin), kimenet (**> ,** stdout) és a hibakimenet, **2 ,** stderr) azért hívják szabványosnak, mert a programok és a parancsok alapértelmezettként használják a szöveg be- és kimenetének kezelésére. Ehhez társul még a csővezeték (**|** pipe), amely lehetővé teszi, hogy egy parancs kimenetét egy másik parancs bemenetéhez csatlakoztassuk.

< Szabványos bemenet fájlból - **tr 'a-z' 'A-Z' < eb.txt** (az összes kisbetű átalakítása nagyra)

> Szabványos kimenet átirányítása fájlba - **sed -n '5p' eb.txt > kivonat.txt** (az 5.sort fájlba)

>> Szabványos kimenet hozzáfűzése filehoz.

2> Sz. hibakimenet átirányítása új fájlba - **ls nemlétezőfájlok 2> error.log** (error log készítés)

2>> Szabványos hibakimenet átirányítása és hozzáfűzése meglévő fájlhoz –
ls nemlétezőfájlok 2>> error.log (log hozzáfűzés meglévő error.log-hoz)

&> Szabványos kimenet és hibakimenet és átirányítása új fájlba
ls nemlétezőfájlok &> outputanderror.log (kimenet és error log készítés & … AltGr+C)

&>> Szabványos kimenet és hibakimenet és átirányítása már meglévő fájlba
ls nemlétezőfájlok &>> outputanderror.log (kimenet és error log készítés meglévő fájlhoz)

A csővezeték (Pipeline)

A csővezeték (| pipe) lehetővé teszi, hogy egy parancs kimenetét egy másik parancs bemenetéhez csatlakoztassuk. Memória alapon kezeli az egyik folyamat kimenetét, a másik bemeneteként.
A puffer szinkonizálja az író és az olvasó folyamatokat. Lehetővé teszi a nagy adatmennyiségek továbbítását köztes fájlok nélkül.

A csővezeték - példa

1., A termés1.txt, termés2.txt fájlokban termények nevei vannak. Ezen fájlok tartalma duplikációt is tartalmaznak. Írassuk ki azon gyümölcsök és zöldségek neveit, amelyek az alma szót tartalmazó sorokat jelenítsük meg úgy, hogy az eredmény legyen ABC sorrendben, de ne legyen ismétlődés:

cat termés1.txt termés2.txt | sort | uniq | grep alma 

alma
birsalma

2., Az első szöveges fájlt rendezzétek ABC sorrendbe és írjátok ki az első két sorát

cat termés1.txt | sort | uniq | head -n 2 

alma
barack

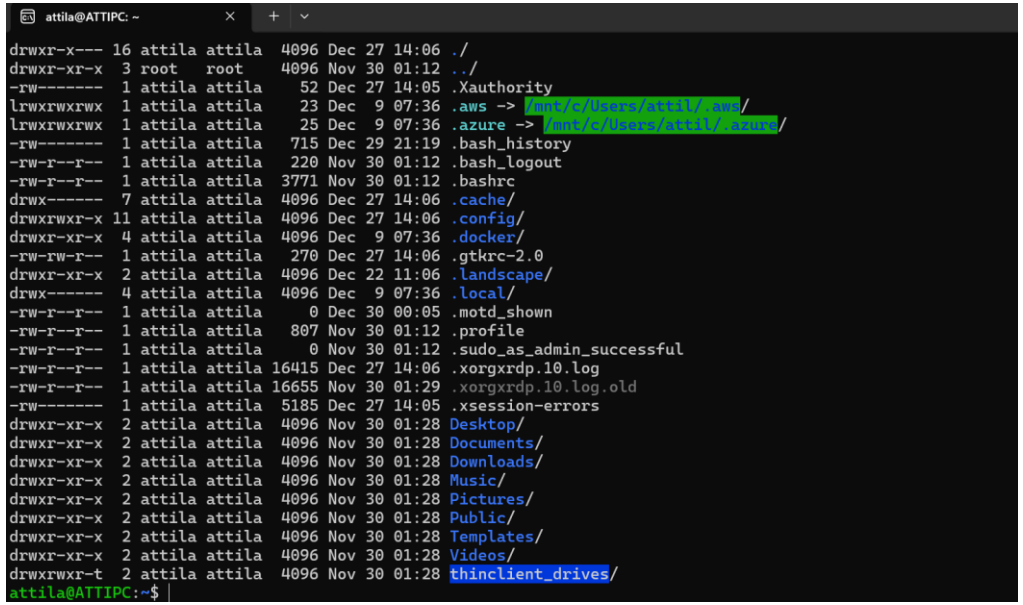
3., Minden paprika és paradicsom szót tartalmazó bejegyzést jelenítse meg a képernyőn, és egyúttal írja is bele a kimenet fájlba.

cat termés1.txt | sort | uniq | grep -E 'paprika | paradicsom' | tee kimenet.txt 

paprika
paradicsom

Listázás

A Listázás (ls) a könyvtárak tartalmának listázására szolgál.
ls -laF , ami az “1.linux varázsige “
l - hosszú formátumban listázza a fájlokat - jogosultságok, tulajdonos, méret és módosítás dátuma
a – minden file megjelenítése
F – fájltípus kiírása – / könyvtár, * végrehajtható file, @ symlink, | FIFO(pipe), = unix domain socket
R – rekurzív megjelenítés
t - Az utolsó módosítás időpontja alapján rendezi a fájlokat



A súgórendszer

1., A man parancs, azaz a manuál az egyes parancsokhoz részletes dokumentáció.
Egy beépített súgórendszer, amely segítségével megtekinthetjük a parancsok, programok és fájlok részletes leírását. Pl.:

man grep – megnyitja a grep parancs súgógát.
man -s 2 chmod - a chmod paracs súgója 2.szekciójának megnyitása (rendszerhívások)
man -k medve - megjeleníti az összes kézikönyvet, amely tartalmazza a medve kulcsszót

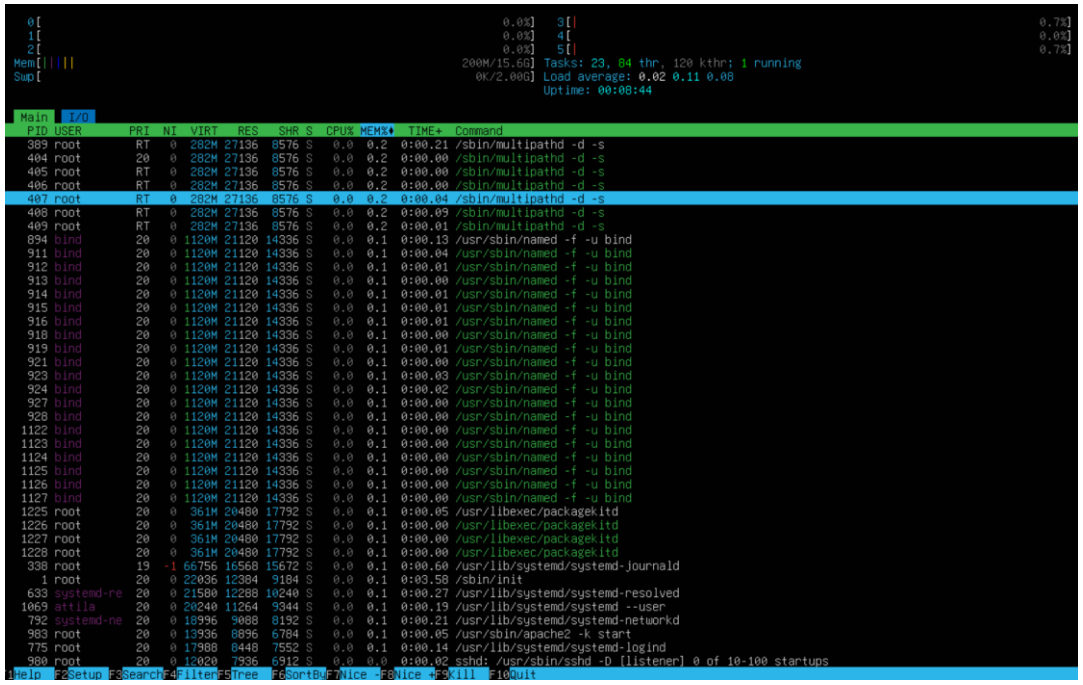
2., A **--help** kapcsoló(zászló,flag) pedig a parancsokhoz hozzárendelt gyors információlekérő súgó

Pl. **ls --help**

```
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).

Options:
  -a, --all                  do not ignore entries starting with .
  -A, --almost-all           do not list implied . and ..
  -b, --escape               print C-style escapes for nongraphic characte
  -c                         with -lt: sort by, and show, ctime (time of l
                             modification of file status information)
                             with -l: show ctime and sort by name
                             otherwise: sort by ctime
```

Erőforrás monitorozás

Linux rendszerek alatt az erőforrásokat a **top** parancs segítségével végezzük. Ennél részletesebb és használhatóbb listát a **htop** parancs ad, amelyet fel kell telepíteni repoból. Itt az egyes folyamatok prioritását is feljebb/lejjebb tudom állítani
(F7 - mínusz nice érték – fontos,
F8 + plusz nice érték - kevésbé fontos)



sudo apt update && sudo apt install htop
sudo htop

Internetes erőforrások megtekintésére a **nethogs** parancsot használjuk – megmutatja, melyik folyamat kommunikál az internettel, amolyan parancssoros wireshark.

Az **fuser** paranccsal azt nézzük meg, hogy egy adott fájt melyik folyamat használja. Az **lsof** (list open files) pedig megmutatja a sok megnyílt fájlt, amelyet használ a rendszerünk. Pl.
lsof -u attila | grep "/media/attila/" –a bedugott pendriveomon tart-e valaki nyitva fájlokat ?

Az **iotop**, amelyet szintén fel kell telepíteni, az pedig a folyamatok lemezhasználatát mutatja meg.

A **vmstat** gyors áttekintést ad a gépem állapotáról (folyamatok, memória, blokk I/O)

Futó folyamatok leállítása Linuxon

1., Futó folyamatok leállítása - formája a kill PID, ahol a futó folyamat azonosítóját PID-jét a ps –waux által kilistázott kimenetből kiolvassuk.
kill 172 - a 172 PID-es folyamat leállítása
kill 172 173 - a 172 és a 173-as folyamat leállítása

2., Az összes folyamat leállítása – ez a **killall** paranccsal történik

3., Folyamat névszerinti leállítása – itt ha ismerjük a folyamat nevét, aszerint is leállíthatjuk. Pl.:
pkill firefox - a firefox leállítása

4., Folyamat ablak szerinti leállítása – grafikus környezetben ha ki van nyitva egy ablakunk, amelyet ki akarunk kapcsolni, akkor kinyitjuk a terminált, és beírjuk, hogy **xkill** , majd arra az ablakra kattintunk, amelyiket le akarjuk lőni

5., Folyamatok leállítása jelzésküldéssel – a jelzésküldésből kétfélét ismerünk
a., erőszakos leállítás - SIGKILL, ennek a flagje 9. Pl.: **kill -9 124**
b., szelíd leállítás – SIGTERM, ennek a flagje 15. Pl.: **kill -15 175**

Megjegyzés: Sok futó folyamat esetén célszerű először azonosítani, megismerni mely PID-ek tartoznak a folyamathalmazhoz, vagy valamiféle maszk alapján. Erre a parancs:
ps -waux | grep folyamatnév

Folyamatok prioritásváltozása Linuxon

Az egyes folyamatok prioritását a nice és a renice parancsokkal módosíthatjuk Linux rendszereken. A prioritás meghatározása lehetővé teszi, hogy szabályozzuk, milyen mértékben használhatja a folyamat a rendszer erőforrásait.
Ezen parancsok értéktartománya -20-tól 19-ig vehet fel értékeket.

a., -20: Legmagasabb prioritás (legkevésbé "kedves" a rendszerrel, több CPU-t igényel).
b., 0 : A folyamatok alap prioritása
c., 19: Legalacsonyabb prioritás (leginkább "kedves" a rendszerrel, kevesebb CPU-t igényel).

1., A nice parancs egy új folyamatot indít prioritással az alapszinthez képest. Pl.:
nice -n -10 blender - magasabb prioritáson indítja a blender-t

2., A renice parancs egy már futó folyamatnak a prioritását változtatja meg. Pl.:
renice -10 blender - magasabb prioritásra rakja a blendert
renice 17 -p 124 - nagyon alacsony prioritásúra rakja a PID 124-es folyamatot

Naplózás Linuxon

A naplózás (loggolás) fontos szerepet játszik a rendszer adminisztrációban és hibakeresésben a Linux rendszereken. A naplófájlok segítségével nyomon követhetjük a rendszer, alkalmazások és szolgáltatások eseményeit. Amit érdemes tudni:

1. a syslog és a fejlettebb változata rsyslog - hagyományos naplózási szolgáltatás a Linux rendszereken. Különböző programok és szolgáltatások naplóüzeneteket küldhetnek a syslog démonnak. Megtekintése pl: **cat /var/log/syslog** vagy **tail -f /var/log/syslog** (itt a –f flag az a folyamatos nyomonkövetés a follow)

2., A gyakori naplófájlok a /var/log/ mappában találhatóak. Pl. syslog (általános), auth.log (hitelesítés), kern.log (kernel), dmesg (boot alatti események), daemon.log (démonok)

3., Egyéni naplózás: Az alkalmazások és szkriptek saját naplózására is van lehetőség a logger paranccsal, amely beleírja az üzenet a rendszer naplófájljaiba.
logger "Ez egy egyéni naplóüzenet"

4., A naplózási szolgáltatások konfigurációs fájlokkal rendelkeznek, amelyek segítségével testreszabhatod a naplózás viselkedését. Pl.: rsyslog esetén a /etc/rsyslog.conf fájl

Naplózási szintek

- DEBUG: Hibakeresési információk
- INFO: Általános információk
- NOTICE: Normális, de fontos események
- WARNING: Figyelmeztetések
- ERR: Hibák
- CRIT: Kritikus hibák
- ALERT: Azonnali figyelmet igénylő problémák
- EMERG: Rendszerszintű vészhelyzetek

Naplózás hálózaton keresztül

1. Az rsyslog telepítése szerverre és kliensre: **sudo apt update && sudo apt install rsyslog -y**

2. Naplószerver konfiguráció a szerver gépen: **sudo nano /var/log/rsyslog.conf** ,
majd itt engedélyezzük a TCP és UDP portokat és mentjük:

```
module(load="imtcp")
input(type="imtcp" port="514")
module(load="imudp")
input(type="imudp" port="514")
```

3. Naplószerver újraindítása: **sudo systemctl restart rsyslog**

4. Naplókliens konfigurálása a kliensgépen: **sudo nano /var/log/rsyslog.conf** , majd itt hozzáadjuk, hogy a naplóüzeneteket a központi szerverre küldje (TCP port @@, UDP port @):
. @@szerver_ip_címe:514

5. Naplókliens újraindítása: **sudo systemctl restart rsyslog**

6. A naplózást úgy ellenőrizzük: hogy a kliens gépen kiadjuk, hogy: **logger "Ez egy egyéni naplóüzenet"** , majd a szerveren megtekintjük a napló aljáát pl: **sudo tail -f /var/log/syslog**

Feladatok ütemezése – cron

A cron egy démon (daemon, háttérfolyamat), amely időzített ismétlődő feladatokat hajt végre meghatározott időközönként. Beállítása a crontab (cron table) fájlban találhatók, amely a feladatokat és azok időzítését definiálja.
crontab -e - a cronetab-hoz a bejegyzés hozzáadása, illetve már meglévő ütemezésnél annak módosítására szolgáló parancs

* * * * * parancs
| | | | |
| | | | +-- A hét napja (0-7) (0 és 7 = vasárnap)
| | | +---- A hónap napja (1-31)
| | +------ Hónap (1-12)
| +-------- Óra (0-23)
+---------- Perc (0-59)

Például, a papagájok programot minden héten csütörtök pont délben szeretném lefuttatni, akkor
0 12 * * 4 /home/attila/programok/papagájok sort beillesztem a crontab-ba

crontab -l - megjeleníti a felhasználó összes aktuális cron feladatát
crontab -r - törli a felhasználó aktuális crontab fájlját

Feladatok ütemezése - at

Az at parancs egyszeri feladatok ütemezésére szolgál egy meghatározott időpontban. Nem ismétlődő feladatokra használatos, ellentétben a cron-nal. Az at paracs végrehajtása után kiírja a visszaigazolást. Példák az at-re:

Ütemezett szövegkiiratás holnap du.2 óra 15 percre
echo "A kecskepapagájom már vörös szemekkel rikácsol" | at 2:15 PM tomorrow

4 óra múlva induljon el a cirmoscica.sh program
echo "sh /home/attila/cirmoscica.sh" | at now + 4 hours

Írjunk ki Újév előtt két órával egy figyelmeztetést
echo "Két óra múlva Újév!" | at 10pm Dec 31

Megjegyzés: az at-nél alapból nem működik a parancssoros másodperces pontosság, viszont a date parancsal kell formázni a dátumot az adott parancsra
echo "A kecskepapagájom már vörös szemekkel rikácsol" | at -t $(date -d "tomorrow 14:15:16" +"%Y%m%d%H%M.%S")

atq – az ütemezett at feladatok listázása, és hozzárendel feladatazonosítót
atrm feladatazonosító – az ütemezett at feladatok törlése

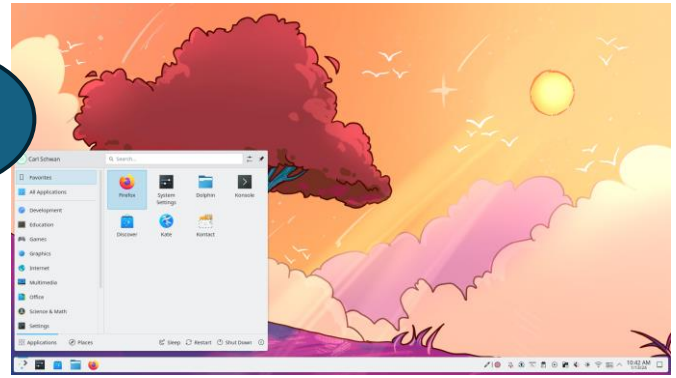
Grafikus felületek

A grafikus felületek (GUI - Graphical User Interface) a felhasználók közötti interakció, és a napi munka megkönnyítése érdekében jöttek létre. Csak akkor telepítsünk fel és futtassunk grafikus szintet szerverre, ha szükségünk van rá. Vannak szép felhasználóbarát GUI-k (pl. KDE), de vannak pehelysúlyú GUI-k is (pl. XFace).

A grafikus felület részei:

- Asztali környezet (Desktop Environment): KDE Plasma, Gnome, Xface, LXQt
- Ablakkezelő (Window Manager): Kwin (KDE), Mutter (Gnome), Openbox, i3 stb.
- Panel és menürendszer: Plasma Panel (KDE), Dash (Gnome), Whisker Menu (Xface)
- Fájlkezelő: Dolphin (KDE), Nautilus (Gnome)
- Terminál: Konsole (KDE), Gnome Terminal
- Bejelentkezés kezelő (login manager): SDDM (KDE), GDM (Gnome)

KDE



gnome

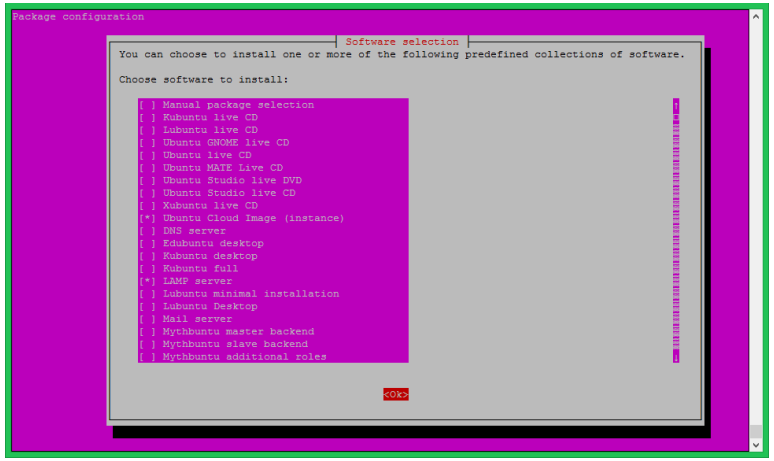
Ctrl+Alt+F3 – ha lassú a KDE, akkor átugrik init 3-ba, Ctrl+Alt+F2 pedig visszaugrik init 5-be

A KDE grafikus felület telepítése

Az Ubuntu családban a Kubuntu alapból a KDE grafikus felületet húzza fel, mint alapértelmezett GUI.
Viszont Ubuntu Serveren mivel nincs alapból fel kell telepíteni.

a., telepítés a tasksel segítségével:
itt választhatunk grafikus felületek között

sudo apt update && sudo apt install tasksel -y
sudo tasksel



, majd a telepítés végén újraindítunk:
sudo systemctl isolate reboot.target

b., vagy telepítés parancssorban pl. KDE: **sudo apt install kde-full**, majd újraindítunk
sudo systemctl isolate reboot.target

2., Az SDDM telepítése, ha még nincs fent – “Display manager”
sudo apt update && sudo apt install sddm -y - telepítés
sudo systemctl enable sddm - alapértelmezetté tétel
sudo systemctl isolate reboot.target - újraindítás

Portok és a TCP, UDP protokollok

A port a hálózatokban egy logikai kapcsolódási pont, amelyen keresztül az adatok beérkeznek vagy távoznak egy hálózati eszközön. Ezeket a portokat hálózati protokollok használják az adatok célállomásra történő irányításához. Szervereknél a kapcsolatlétrehozás típusáért felelő protokollok a TCP és UDP.

TCP (Transmission Control Protocol)
Kapcsolat-orientált: A TCP kapcsolatot hoz létre a küldő és a fogadó között az adatátvitel megkezdése előtt. Ez biztosítja, hogy az adatok megbízhatóan és sorrendben érkezzenek meg.
Hibajavítás: A TCP ellenőrzi, hogy az adatok sértetlenek-e, és újraküldi őket, ha hibát talál.
Teljesítmény: A TCP megbízhatósága miatt némi késleltetést okozhat, de biztosítja, hogy az adatok pontosan és sorrendben érkezzenek meg.
Felhasználási területek: Web böngészés (HTTP/HTTPS), e-mailek (SMTP), fájlátvitel (FTP) stb.

UDP (User Datagram Protocol)
Kapcsolatmentes: Az UDP nem hoz létre kapcsolatot a küldő és a fogadó között, mielőtt elküldi az adatokat. Az adatok egyszerűen elküldésre kerülnek, és nincs garancia arra, hogy megérkeznek vagy sorrendben érkeznek meg. Gyorsabb, de nincs hibajavítás.

További hálózati protokollok: HTTP, FTP, HTTPS, FTPS, SSH, DNS, DHCP, IMAP, SNMP, ICMP

A hálózati címfordítás (NAT)

A hálózati címfordítás (Network Address Translation, NAT) egy hálózati technika, egy olyan közvetítő, amely lehetővé teszi a belső hálózati eszközök számára, hogy kommunikáljanak az internettel anélkül, hogy mindegyiküknek egyedi külső IP-címe lenne. A NAT modernebb megjelenése a PAT.
Jellemzői:

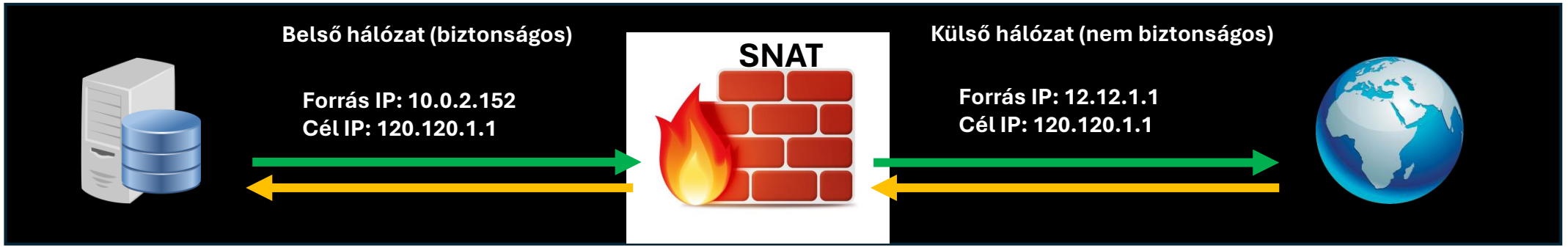
1., IP-cím megtakarítása: Az internetre csatlakozó eszközök mindegyikének egyedi IP-címre van szüksége. A NAT lehetővé teszi, hogy egy vállalat vagy otthon belső hálózata csak egy vagy néhány külső IP-címet használjon, így csökkentve az IP-címek iránti igényt.

2., Biztonság növelése: A NAT elrejti a belső hálózaton lévő eszközök valódi IP-címét a külső hálózatok (pl. az internet) elől, ezáltal növelve a hálózati biztonságot.

3., Címfordítás: A NAT közvetlenül fordítja le a belső hálózat IP-címeit és portjait a külső hálózat (pl. az internet) IP-címeire és portjaira, lehetővé téve a hálózaton belüli eszközök számára az internetes kommunikációt.

A forrás-IP hálózati címfordítás (SNAT)

A Source Network Address Translation (SNAT) a hálózati címfordítás egy olyan típusa, amely a forrás IP-címeket fordítja le. Lehetővé teszi, hogy több belső hálózati eszköz egyetlen, vagy néhány külső IP-címet használjon az internetes kommunikáció során. Az IP-címek megtakarítását segíti elő, növeli a biztonságot is, mert a belső hálózat eszközei el vannak rejtve.



1., Csomagküldés: mikor egy belső hálózatban lévő eszköz (például egy számítógép) adatcsomagot küld az internetre, a forrás IP-címe az adott eszköz belső IP-címe lesz

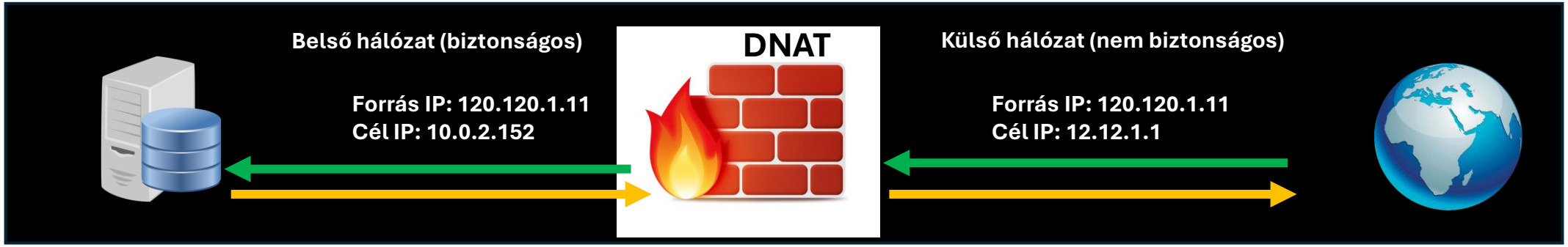
2., Címfordítás: SNAT az eszköz forrás IP-címét egy külső, publikus IP-címre cseréli le. Ez lehet egy előre meghatározott IP-cím vagy egy adott címkészletből választott cím

3., A csomag továbbítása: Az átalakított csomagot ezután továbbítják az internetre, ahol a forrás IP-cím már a külső, publikus IP-cím lesz.

4., Válaszfogadás: Amikor az interneten lévő célpont válaszol, az adott válaszcsomag visszaérkezik a SNAT eszközhöz, amely visszafordítja az IP-címet az eredeti belső IP-címre, és továbbítja a válaszcsomagot a megfelelő belső eszközre.

A cél-IP hálózati címfordítás (DNAT)

A Destination Network Address Translation (DNAT) olyan hálózati címfordítás, amely a cél IP-címeket fordítja le. Lehetővé teszi, hogy egy külső hálózatról érkező csomagot egy belső hálózati eszközhöz irányítsunk anélkül, hogy a külső eszközök ismernék a belső hálózat IP-címeit. Javítja a hálózati biztonságot és lehetővé teszi a belső hálózat eszközeinek az internetes elérhetőségét.



1., Csomag érkezése: Amikor egy külső hálózatról (például az internetről) érkező adatcsomag megérkezik a DNAT eszközhöz, a csomag cél IP-címe egy külső, publikus IP-cím lesz.

2., Címfordítás: A DNAT az eszköz cél IP-címét egy belső, privát IP-címre cseréli le. Ez lehet egy előre meghatározott belső IP-cím, amely egy adott eszközhöz tartozik a belső hálózatban.

3., A csomag továbbítása: Az átalakított csomagot ezután továbbítják a belső hálózati eszközhöz, ahol a cél IP-cím már a belső, privát IP-cím lesz.

4., Válaszküldés: Amikor a belső hálózati eszköz válaszol, a válaszcsomag ismét áthalad a DNAT eszközön, amely visszafordítja a cél IP-címet az eredeti külső IP-címre, mielőtt a csomagot továbbítja a külső hálózatra.

SNAT és DNAT Ubuntu Serveren

Ha a routerünk helyett a saját szerverünkön tűzfalán (ufw) akarjuk megvalósítani a hálózati címfordítást (SNAT + DNAT), akkor a következő a teendő:
Példaként az előző ábra alapján a file szerverünkre alkalmazzuk, a következő a teendő:

1., A rendszerkonfigurációban engedélyezzük az IP továbbítást (packet forwarding)
sudo nano /etc/sysctl.conf - megnyitjuk a rendszerkonfigurációs file-t, és itt hozzáadjuk
net.ipv4.ip_forward=1 - hozzáadjuk az bejegyzést, engedélyezzük az IP-cím továbbítást
sudo sysctl -p - a rendszerkonfiguráció újraindítása

2., Engedélyezzük a tűzfalat, majd szerkesztjük a NAT szabálytáblázatát
sudo ufw enable - engedélyezzük a tűzfalat a szerveren
sudo nano /etc/ufw/before.rules - NAT szabálytáblázat szerkesztése a (fájl elejére)

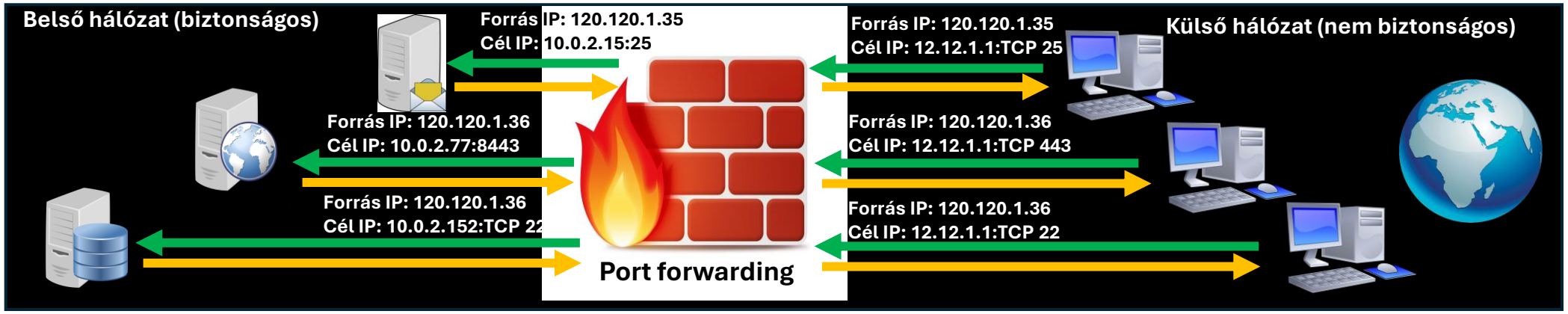
```
*nat
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -s 10.0.2.0/24 -o enps03 -j SNAT --to-source 12.12.1.1        # NAT szabály
COMMIT
*nat
:PREROUTING ACCEPT [0:0]
-A PREROUTING -p tcp --dport 2222 -j DNAT --to-destination 10.0.2.152:2222   # port forward szab.
COMMIT
```

3., A packet forwarding engedélyezése a tűzfalon: **sudo nano /etc/default/ufw**
DEFAULT_FORWARD_POLICY="ACCEPT"

4., Újraindítjuk a tűzfalat: **sudo ufw reload**

Port továbbítás (Port Forwarding)

A port forwarding, vagy port továbbítás egy olyan hálózati technika, amely lehetővé teszi, hogy a belső hálózati eszközökön futó szolgáltatásokat elérjük a külső hálózatról. Ezzel lehetővé válik például, hogy egy otthoni hálózaton futó weboldalt, játékszervert stb elérjünk az internetről.



1., Csomag Érkezése: Egy külső hálózatról érkező csomag megérkezik a routerhez, az adott csomagnak egy cél IP-címe és portja van. Például egy webkérés esetén a cél IP-cím a router külső IP-címe, a port pedig általában a 80-as vagy 443-as port (HTTP vagy HTTPS).

2., A port továbbítása: A router beállításai alapján meghatározza, hogy az adott port melyik belső hálózati eszközhöz és porthoz van társítva. Ezután a router átírja a csomag cél IP-címét és szükség esetén a portját a belső hálózati eszköz IP-címére és megfelelő portjára. Például egy belső webkiszolgáló esetén a csomag továbbítása a belső IP-címre és a 80-as portjára történi

3., A csomag továbbítása: Az átalakított csomagot ezután továbbítják a megfelelő belső hálózati eszközhöz, ahol a cél IP-cím és port már a belső hálózat IP-címe és portja lesz.

4., Válaszküldés: Amikor a belső hálózati eszköz válaszol, a válaszcsomag ismét áthalad a routeren, amely visszafordítja a cél IP-címet és portot az eredeti külső IP-címre és portra

Port Forwarding Ubuntu Serveren

Ha a routerünk helyett a saját szerverünkön tűzfalán (ufw) akarjuk megvalósítani a SNAT-ot,
Példaként az előző ábra alapján a file szerverünkre alkalmazzuk, a következő a teendő:

1., A rendszerkonfigurációban engedélyezzük az IP továbbítást (packet forwarding)
sudo nano /etc/sysctl.conf - megnyitjuk a rendszerkonfigurációs file-t, és itt hozzáadjuk
net.ipv4.ip_forward=1 - hozzáadjuk az bejegyzést, engedélyezzük az IP-cím továbbítást
sudo sysctl -p - a rendszerkonfiguráció újraindítása

2., Engedélyezzük a tűzfalat, majd szerkesztjük a NAT szabálytáblázatát
sudo ufw enable - engedélyezzük a tűzfalat a szerveren
sudo nano /etc/ufw/before.rules - NAT szabálytáblázat szerkesztése a (fájl elejére)
*nat
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -s 10.0.2.0/24 -o enps03 -j SNAT --to-source 12.12.1.1 # NAT szabály
COMMIT
*nat
:PREROUTING ACCEPT [0:0]
-A PREROUTING -p tcp --dport 22 -j DNAT --to-destination 10.0.2.152:2222 # port forward szab.
COMMIT

3., A packet forwarding engedélyezése a tűzfalon: **sudo nano /etc/default/ufw**
DEFAULT_FORWARD_POLICY="ACCEPT"

4., Újraindítjuk a tűzfalat: **sudo ufw reload**

Biztonságos adat továbítás
SSH szerver

Az SSH (Secure Shell) egy titkosított protokoll, amely lehetővé teszi a táveltérés és a távoli rendszerek kezelését biztonságos csatornán keresztül. Amikor egy SSH kliens csatlakozik egy SSH szerverhez, a szerver az állandóan használt nyilvános kulcsát küldi el a kliensnek az autentikáció és az adatcsatorna biztonságának megkezdése érdekében. Ez a kulcs a szerver egyedi azonosítója. Ubuntu Serveren a beállítása a következőképp történik:

1., Feltelepítjük a Secure Shell szervert és a tűzfalat: **sudo apt install openssh-server ufw -y**

2., Elindítjuk és alapértelmezetté tesszük: **sudo systemctl enable ssh** majd **sudo systemctl start ssh**

3., Szerkesztük a konfigurációját **sudo nano /etc/ssh/sshd_config**, és engedélyezzük a következőket úgy, hogy átjavítjuk a # sorokat, ill. hozzáadjuk ami hiányzik.

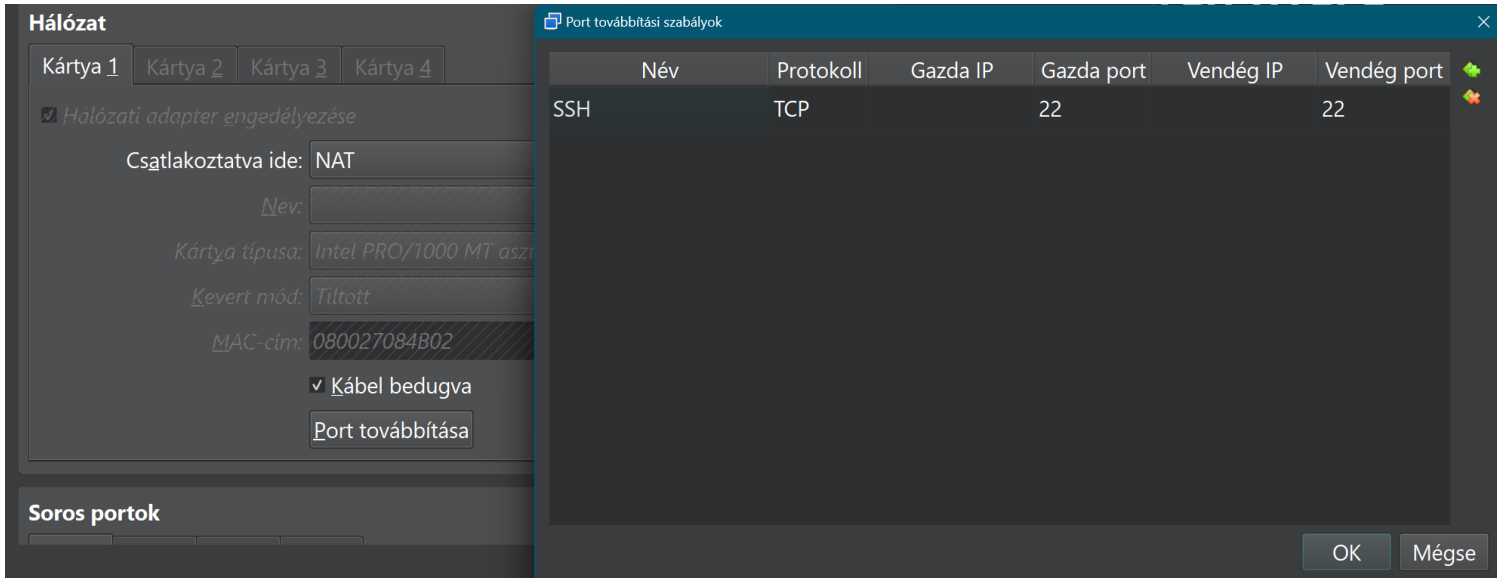
```
PermitRootLogin yes
PasswordAuthentication yes
Port 22
AllowUsers attila
```

majd érvényesítjük a szabályokat: **sudo systemctl restart ssh**

4., Tűzfalon engedélyezzük:
sudo ufw enable
sudo ufw allow ssh
sudo ufw status

Biztonságos adattovábítás
SSH kapcsolat - VirtualBox

A Virtualboxban két lehetőségünk van. A könnyebbik, hogy a NAT-ot Bridge-re állítjuk és minden flottul megy, viszont maradjunk NAT-ban, hogy értelmet nyerjen. Az Ubuntu Server hálózati beállításainál, Port forwardingolunk (a gazdagép 22-es portját a virtuális gép 22-es portjához rendeljük hozzá)
Ezt a Port továbbítása gombnál tehetjük meg az ábrán látható beállításokkal.



Szimmetrikus és aszimmetrikus titkosítás

A szimmetrikus titkosítás egy olyan módszer, ahol ugyanazt a kulcsot használják az adatok titkosítására és visszafejtésére – pl. AES, DES, 3DES, Blowfish. Előnye az egyszerűség és a gyorsaság, viszont hátránya a biztonság.

Megjegyzés: Napjaink másik megoldása a (TOTP - Time-based One-Time Password), ahol szimmetrikus kulcsos kódolást használnak idő alapú jelszavakkal, ahol a szolgáltató, mind az alkalmazás ugyanazt a titkos kulcsot ismeri. Itt a biztonságot a rövid idő jelenti – pl. Authenticatorok (Microsoft, Google… Ügyfélkapu+)

Az asszimmetrikus titkosítás két különböző kulcsot használ: egy nyilvános kulcsot a titkosításhoz és egy privát kulcsot a visszafejtéshez. Jellemzői a kulcspárok - a nyilvános kulcs bárki által használható az adatok titkosítására, míg a privát kulcsot csak a címzett ismeri. Biztonságosabb, mint a szimmetrikus titkosítás mivel a privát kulcs nem kerül megosztásra, így nehezebb kompromittálni – RSA, DSA, ECC, EIGamal, Diffle-Hellman (SSH)

Az SSH működése

1., Mind a kliens, mind a szerver generál egy-egy kulcspárt, amely egy nyilvános és egy privát kulcsból áll

2., A kliens megosztja a nyilvános kulcsát a szerverrel, és a szerver is megosztja a nyilvános kulcsát a klienssel

3., “A közös titok” - az SSH protokoll általában a Diffie-Hellman kulcscserét használ a közös titok létrehozásához. Ez a közös titok lesz az alapja a szimmetrikus titkosítási kulcsnak. Mindkét fél a saját privát kulcsát és a másik fél nyilvános kulcsát használva létrehozza ugyanazt a közös titkot. Ezt az eljárást úgy tervezték, hogy a közös titok azonos legyen, de harmadik fél ne tudja könnyen kiszámítani, még akkor sem, ha ismeri a nyilvános kulcsokat.

4., Szimmetrikus titkosítás – a közös titkot a továbbiakban szimmetrikus kulcsok generálására használják, amelyeket aztán a tényleges adatátvitel titkosítására használnak

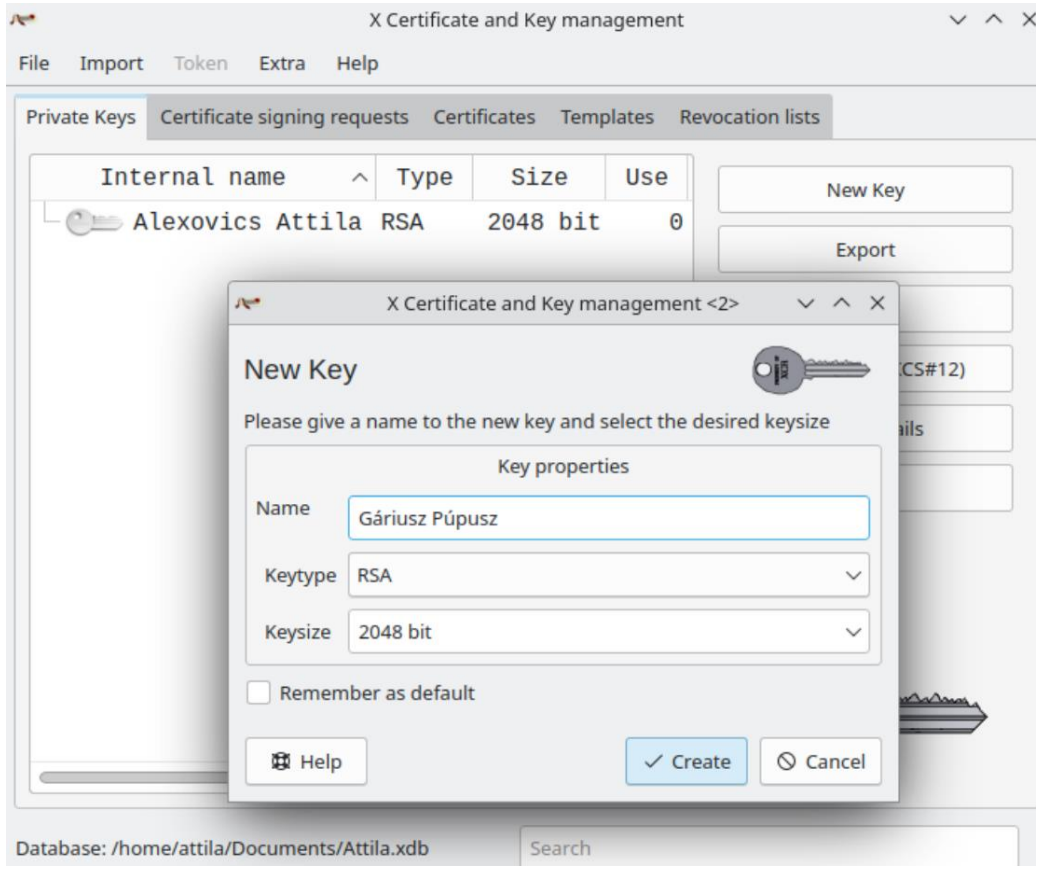
Biztonságos adattovábítás
XCA – OpenSSH kulcsmenedzser

Ha már telepítettünk egy grafikus felületet (pl. KDE) és nem akarunk szenvedni a kulcsgenerálás hosszú paramatéter-sorával, akkor erre megoldás az XCA

1. Létrehozzuk a kulcsadatbázist: Attila.xcb

2. Létrehozzuk a kulcsokat, ahol megadhatjuk a kódolás típusát és a kulcs méretét:
pl. Alexovics Attila, RSA, 2048 bit

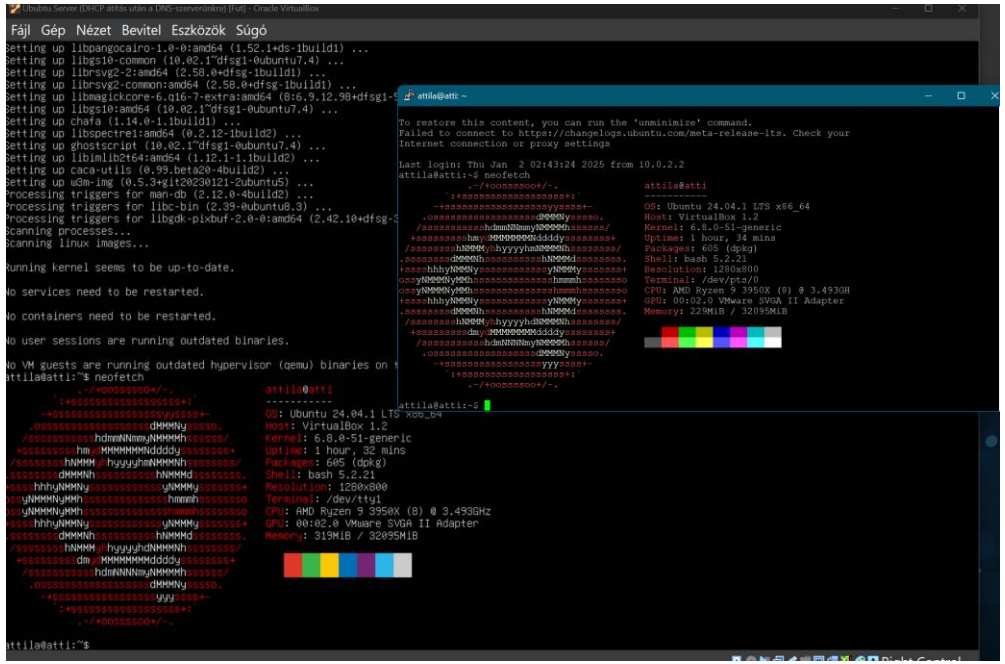
3. További műveletek kulcsokkal és tanusítványokkal (lásd menü)

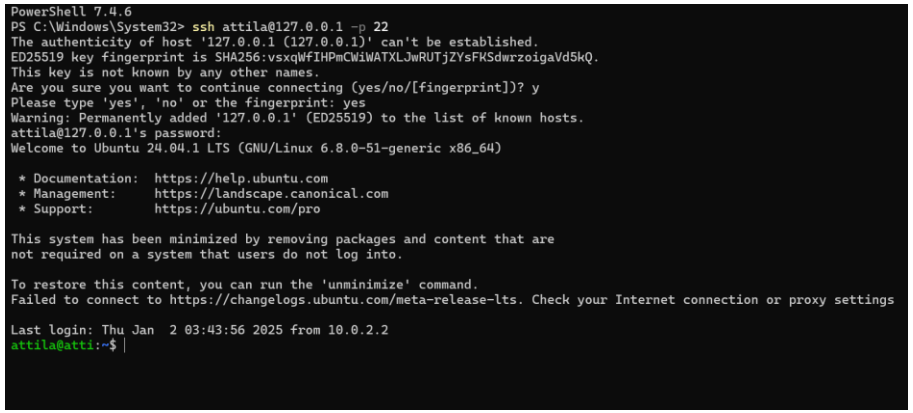


Biztonságos adattovábítás
SSH kliens oldal

1., Ha a gazdagépünk Linux, akkor telepítenünk kell az SSH klinest: **sudo apt install openssh-client**, majd itt csatlakozunk: **ssh attila@127.0.0.1 -p 22**

2., Ha a gazdagépünk Windows, akkor a tűzfal bejövő és kimenő szabályainál át kell engednünk a 22-es portot, majd telepítenünk kell egy SSH klinest, pl. Putty, ahol kiadjuk a loopback-re a parancsot: 127.0.0.1, 22-es porton



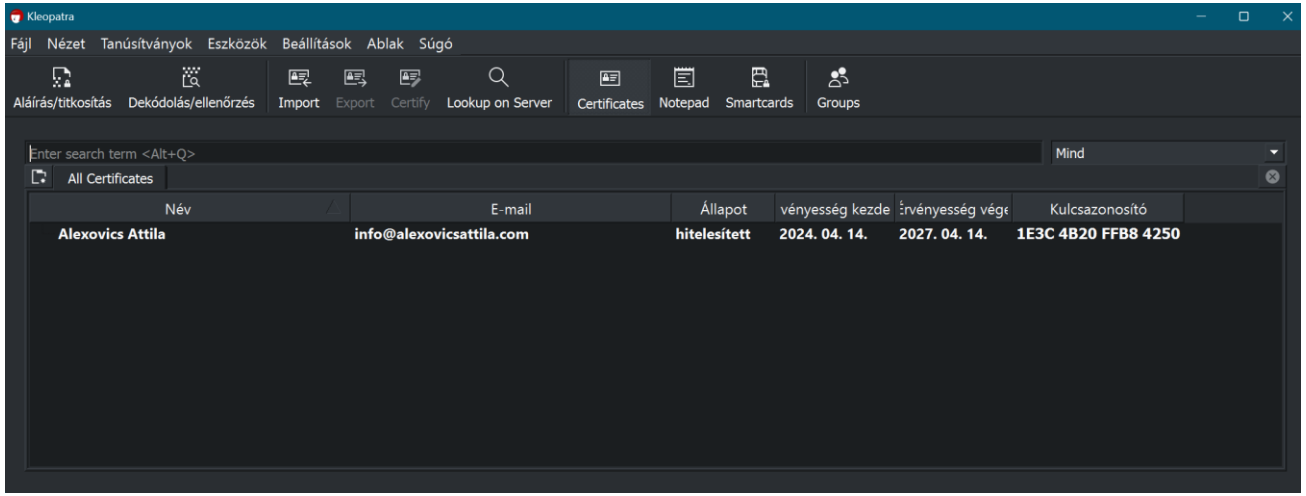


Vagy pedig WSL (+szintén openssh-client telepítés majd a parancs), vagy Powershell-ben: **ssh attila@127.0.0.1 -p 22**

Biztonságos adattovábítás
GPG szerver

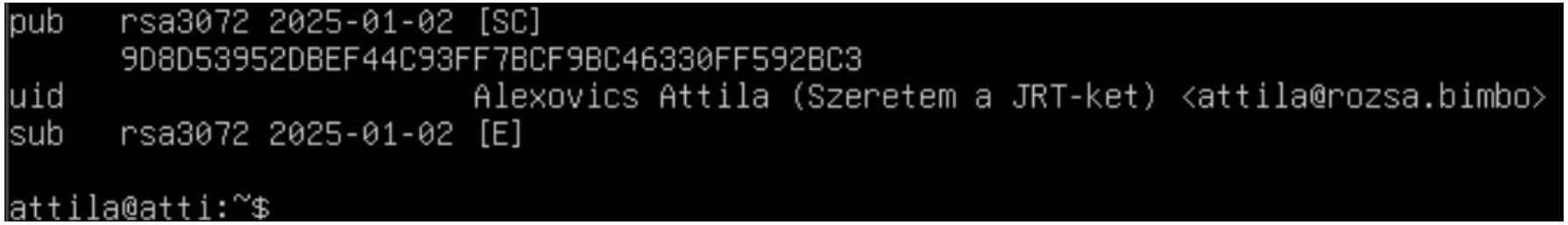
A GPG lehetővé teszi a titkosítást és az aláírást nyílt kulcsú kriptográfia segítségével. Alapvetően a PGP (Pretty Good Privacy) nyílt forráskódú alternatívája, és számos hasonló funkciót kínál.
Alkotóelemei: Nyilvános és privát kulcsok, kulcsok kezelése (létrehozás, import, export, aláírás és visszavonás), titkosítás és visszafejtés (encryption and decription). Szerver oldali telepítés:
sudo apt update sudo apt install gnupg – aki nem akar szenvedni és van grafikus felülete, az telepítse a Kleopatra-t sudo apt install kleopatra

1., Szerver oldali GPG telepítés:
sudo apt update majd **sudo apt install gnupg2**, aki rendszeresen dolgozik kulcsokkal, az telepítse a Kleopatra-t **sudo apt install kleopatra** (grafikus felület, sok függőséggel)



Biztonságos adattovábítás
GPG szerver - csatlakozás menete

2., kigeneráljuk a kulcspárt, azaz a nyilvános (public.key) és a privát kulcsot (private.key) **gpg --full-generate-key** . A parancs kiadása után megadjuk a kódolás tipusát, a nevünket, jelszavaunkat, e-mail (pl. attila@rozsa.bimbo), lejárati időt stb., majd hagyjuk a kulcspárt kigenerálódni



3., miután megvan, ellenőrizzük a kulcspárt: **gpg --list-keys** , amire ugyanezt a kiiírást kapjuk, mint az előzőekben.

4.,exportáljuk a nyilvános kulcsot, majd hozzárendeljük a felhasználó kulcsaihoz

5.,engedélyezzük a GPG számára az SSH-t a gpg-agent-nál: enable-ssh-support

6., Virtualboxot beállitjuk port forwardinggal 22/22 es portokra az Ubuntu Serverhez, és csatlakozunk hozzá klines oldalról a 22-es porton: Linux (ssh kliens), Windows (PuTTY, Filezilla,Powershell stb.)

Biztonságos adattovábítás
X11 Tunnels

X11 Tunnels (Az X11 forwarding) lehetővé teszi, hogy grafikus alkalmazásokat futthatunk egy távoli szerverről egy grafikus környezetben (például egy Windows gazdagépről)

1. Szerver oldalon telepítenünk kell az X autorizációs csomagokat (hálózati protokoll a grafikus felületeknek) **sudo apt updade && sudo apt install xauth x11-utils -y**

2. Az SSH konfigurációjában **sudo nano /etc/ssh/sshd_config** hozzáadjuk:
X11Forwarding yes
X11DisplayOffset 10
X11UseLocalhost yes

3. Újraindítijuk az ssh szervert **sudo systemctl restart sshd**

4. Windows alá telepitük az X kiszolgálót (Xming, vagy Vcvcxsrv, vagy a store-ból a fizetőst)

5. PuTTY-ból csatlakozunk ugyanúgy a 22-es porton keresztül, csak a beállitásoknűl beállitjuk az X11-nél hogy engedélyezzük a port forwardingját az X11-nek , csatlakozunk majd elindithatunk egy programot, amely már ablakot kíván: pl. gedit

6. PowerShell-nél még egyszerűbb a helyzet. Ott a –X flag jelenti az X szerver támogatást:
ssh -X attila@127.0.0.1 -p 22

Biztonságos adattovábítás
X11 Tunnels – tutorial video

Az útvonalvizsgáló - Traceroute

A traceroute parancs segít megvizsgálni, hogyan jut el egy adatcsomag a számítógépedről a célpont hálózati eszközig. Különösen hasznos hálózati problémák diagnosztizálására. A parancs megmutatja az útvonalat, amelyen keresztül az adatcsomagok haladnak, és a köztes állomásokat (azaz az útvonalon lévő routereket) is megjeleníti. Emellett az egyes állomások közötti válaszidőket is figyelemmel kíséri.

Az útvonal megjelenítésére át kell kapcsolnunk vagy Bridge módba (NAT nem látja a gazdagép hálózatát), vagy egy WSL Ubuntu / natív Linux rendszeren megnézzük:

Példa: megvizsgáljuk, hogyan jutunk el a Google-ig a gépünkről.
1-4., A szolgáltatónál pattog, *** -valószinűleg rejtett szervert jelent (Pick-Up Kft.)-nél
5-7., Bekerül a retn.net rendszerébe (The Eurasian Network)
8 *** -rejtett szerver (gondolom az NSA, FBI, CIA stb., mielőtt az USA-ba érkezik)
9., 209.85.255.242 – megérkezik a szignál a Google-re, Mountan View CA, USA.

```
attila@ATTIPC:~$ traceroute google.com
traceroute to google.com (142.250.180.206), 30 hops max, 60 byte packets
 1  172.29.144.1 (172.29.144.1)  0.273 ms  0.252 ms  0.292 ms
 2  192.168.34.1 (192.168.34.1)  5.966 ms  5.956 ms  5.958 ms
 3  * * *
 4  89.251.32.1 (89.251.32.1)  5.995 ms  6.082 ms  6.074 ms
 5  ae4-330.rt.bix.bud.hu.retn.net (87.245.243.180)  9.598 ms  9.586 ms  8.984 ms
 6  ae6-8.rt.dpx.bud.hu.retn.net (87.245.232.158)  9.566 ms  12.258 ms  12.226 ms
 7  gw-as15169.retn.net (87.245.244.5)  12.215 ms gw-as15169.retn.net (87.245.243.183)
7.245.244.5)  20.967 ms
 8  * * *
 9  209.85.255.242 (209.85.255.242)  21.470 ms 209.85.244.146 (209.85.244.146)  21.317
.250.180.206)  21.623 ms
attila@ATTIPC:~$
```

Forgalomátirányítás - quagga

A Quagga egy szoftvercsomag, amely lehetővé teszi, hogy egy Linux rendszeret valódi routerré alakítsuk. Támogatja a legfontosabb IPv4 és IPv6 routing-protokollokat, mint például a RIP, OSPF és BGP

1., Telepítjük a quagga csomagot: **sudo apt updade && sudo apt install quagga -y**

2., Mintafájlok másolása és konfigurálás, naplófájlok, tulajdonjogok, stb.

3., Indítás és engedélyezés (zebra és bgpd)

4., IPv4 és IPv6 Unicast Forwarding engedélyezése

5., GNS3–ban teszteljük a Quagga-t

A szerver hálózati címzése

A szerver hálózati címzése az IP-cím beállítását és a megfelelő hálózati konfigurációkat foglalja. Statikus IP-címet kell kiosztani azért, hogy mindig elérhetők legyenek ugyanazon a címen.
Ez a következőképp történik:

1., Telepítjük a net-tools-t: **sudo apt update && sudo apt install net-tools -y**

2., az Ethernet interfészek azonosítása: kiadjuk az **ip a** parancsot (vagy az vagy **ifconfig** -ot). Megjelenik egy lista az elérhető ethernet csatlakozókról. Az ”eth*” interfészek a valódi csatlakozók, az alsó interface pedig a “lo”, ami a loopback interfész. Ez hozzárendeli a 127.0.0.1 – et a localhost-hoz. A loopback interfészt gyakran használják hálózati szoftverek fejlesztésénél és tesztelésénél, mivel lehetővé teszi, hogy a hálózati forgalmat a helyi rendszeren belül teszteld anélkül, hogy tényleges hálózati kapcsolatra lenne szükség.

3., a hálózati csatlakozók konfigurációs fájljait itt találjuk – ezeket kell majd későbbiekben szerkeszteni.

Ubuntu Server-en a hálózati csatlakozók konfigurációs fájlja az /etc/netplan mappában található yaml fájl. Alapból DHCP-re van állítva a 50-cloud-init.yaml

Debianon, Ubuntun és Kubuntun a hálózati csatlakozók konfigurációs fájlja az /etc/network/interfaces fájl

Hálózati címzés - egyszeri címzések

Ubuntu Serveren ha konfigurációs fájl nélkül szeretnél egyszeri címezést beállítani, használhatjuk az **ip a** parancsot az interfész, ill. az **ip route** parancsot az átjáró kiolvasására.

Ezek a parancsok azonnal érvénybe lépnek, de újraindítás után elvesznek, mivel nem kerülnek be a konfigurációs fájlba. Ezért ezeket az egyszeri beállításokhoz használhatod, például tesztelési célokra.

Megnézzük a hálózati interfészünket, majd aztán (ip a, ip route), és ha üres akkor kitöltjük:

sudo ip a add 10.2.0.15/24 dev enp0s3	- IP-cím és almaszk hozzáadása az interfészünkhöz (a - addr rövidítése)
sudo ip l set enp0s3 up	- Interfészen aktiválás (l - link rövidítése)
sudo ip route add default via 10.0.2.2	- hozzáadjuk az alapértelmezett átjárót
sudo ip a del	- IP-cím törlése az interfészről
sudo ip route del	- átjáró törlése

Ubuntu Server - hálózati címzés
NETPLAN - Google névszerverrel

1., Telepítjük a net-tools-t és a ping-et: **sudo apt update && sudo apt install net-tools iputils-ping**

2., az IP és az alnet maszk: kiadjuk az **ip a** (vagy ipconfig) parancsot, és megnézzünk az interfészünkön az ip címünket és az alnet maszkot.

3., az átjárónk címét a **ip route** parancs segítségével tudjuk meg. (vagy a route -n parancsal megkeressük az UG flaggal jelölt sort)

4. A konfigurációs file szerkesztése **sudo nano /etc/netplan/50-cloud-init.yaml** , ha nincs DNS kiszolgálónk felrakva, a nameservert tegyük ideiglenesen google-re (8.8.8.8), Az IP cím lehet a megfelelő 10.0.2.xxx. Figyelem, a YAML fájl bekezdésérzékeny – ügyelni a helyes tördelésre!!

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no
      addresses:
      - 10.0.2.15/24
      gateway4: 10.0.2.2
      nameservers:
        addresses:
        - 8.8.8.8
        - 8.8.4.4
```

5., frissítjük a netplan-t **sudo netplan apply** . Egészen addig próbálkozunk, amíg csak a 3 db warning marad, de azt nem kell figyelembe venni.

6., snapshotoljuk Virtualboxban, ha jó, újraindítunk
sudo systemctl isolate reboot.target

7., újraindítás után ellenőrizzük a yaml-t fájlt megvan-e a változtatás, ill. megpingetjük a googlet (8.8.8.8)

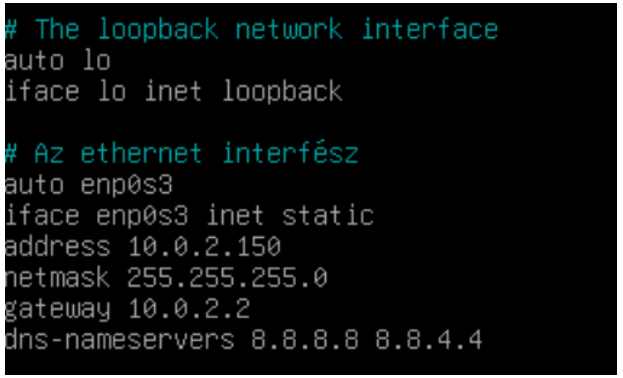
Debian - hálózati címzés
NETWORK INTERFACES - Google NS

1., Sudo helyett belépünk Superuser módba: **su**

2., Telepítjük a nano, net-tools, iputils-ping-et, ha nincsenek fent: **apt install nano net-tools iputils-ping**

3., kiadjuk az **ip a** parancsot, és megnézzünk az interfészünkön az ip címünket és az alnet maszkot. Utána kiadjuk az **ip route** (vagy ip route show, vagy route -n) parancsot és megnézzük az átjárót (router)

4., a konfigurációs file szerkesztése: **nano /etc/network/interfaces**
Itt megadjuk a loopback alatt az interfacet,
kiválasztjuk, hogy statikus IP, majd beírjuk a kívánt IP-címet,
az almaszkot, az átjárót, és a nameservert
(egyenlőre google 8.8.8.8 stb.)



5. Újraindítjuk a hálózati szolgáltatást : **systemctl restart networking**

6. Megpingeljük a google-t : **ping 8.8.8.8**

7. Kilépünk Superuser módból : **exit**

DNS szerver telepítése

1., A Bind (bind9) és a DNS Utils (dnsutils) csomagok telepítése:
sudo apt update && sudo apt install bind9 dnsutils -y

2., A Bind konfigurációs fájljában hozzáadjuk a továbbítókat (forwarders) az options blokkba :
sudo nano /etc/bind/named.conf.options (az egyszerűség kedvéért)

```
options {
   directory "/var/cache/bind";
   forwarders {
      10.0.2.2; # a routerünk mint forwarder
   };
   dnssec-validation auto; # A DNSSEC az biztonsági protokoll, aláírások, integritás
   auth-nxdomain no; # a szerver nem ad válaszokat az NXDOMAIN eredményekre (biztonság)
   listen-on { 10.0.2.15; }; # ezeken az IP-címeken fogad bejövő kéréseket a szerver
   listen-on-v6 { any; }; # a DNS szerver az összes IPv6 címén fogad bejövő kéréseket
};
```

3. Elindítjuk majd érvénybe léptetjük a DNS-szervert, hogy automatikusan induljon:
sudo systemctl start bind9
sudo systemctl enable bind9
ha nem megy, akkor szerver oldali flush dns .... **sudo rm /var/cache/bind/managed-keys.bind***,
végezetül **sudo systemctl restart bind9**

A zóna konfigurációja

A DNS zóna a DNS szerver által kezelt domain nevek és IP-címek egy csoportja. Egy zóna tartalmazza az összes szükséges információt a domain nevekről és azokhoz tartozó erőforrásokról, például IP-címekről, aliasokról, mail szerverekről stb.

1. A zónák létrehozásánál először szerkesztjük **sudo nano /etc/bind/named.conf.local** konfigurációs fájlt. Esetünkben a mindenki által “megkedvelt” rozsa.bimbo –t

```
zone "rozsa.bimbo" {
   type master;
   file "/etc/bind/db.rozsa.bimbo";
};
```

A zónafájl kitöltése

Létrehozzuk a rozsa.bimbo zónabázist: **sudo nano /etc/bind/db.rozsa.bimbo**, majd kitöltjük:

```
$TTL    604800
@       IN      SOA     ns.rozsa.bimbo. admin.rozsa.bimbo. (
                        2024123101      ; Serial
                        604800          ; Refresh
                        86400           ; Retry
                        2419200         ; Expire
                        604800 )        ; Negative Cache TTL
;
@       IN      NS      ns.rozsa.bimbo.
ns      IN      A       10.0.2.15
www     IN      A       10.0.2.77
ftp             IN      A       10.0.2.88
```

a., TTL – time to live másodpercben
b., SOA rekord, névszerver, admin e-mail
- serial dátum,módosításszám
- frissítés (7 nap)
- újrapróbálkozás (minden nap)
- lejárat (28 nap)
- negatív válaszok tárolása (7 nap)

- Fő zónagyökértartomány megadása
- DNS névszerver: esetünkben
- Web szerver, ha későbbiekben…77-re
- FTP szerver, ha későbbiekben…88-ra

2. Ellenőrizzük a kitöltést, hogy ne legyen error-os
sudo named-checkconf
sudo named-checkzone rozsa.bimbo /etc/bind/db.rozsa.bimbo

3. A gyorsítótár törlése majd a zóna érvénybe léptetése
sudo rm /var/cache/bind/managed-keys.bind

sudo systemctl restart bind9

Hálózati címzés
a DNS szerverünkre - NETPLAN

1. A netplan konfigurációs file szerkesztése **sudo nano /etc/netplan/50-cloud-init.yaml** , majd a nameservert tegyük a DNS-szerverünk IP-címére: 10.0.2.15
Figyelem, a YAML fájl bekezdésérzékeny – ügyelni a helyes tördelésre!!

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no
      addresses:
      - 10.0.2.15/24
      gateway4: 10.0.2.2
      nameservers:
        addresses:
        - 10.0.2.15
```

2., frissítjük a netplan-t **sudo netplan apply** . Egészen addig próbálkozunk, amíg csak a warning marad, de azt nem kell figyelembe venni. Ha valamiért nincs telepítve a netplan, akkor a csomag neve netplan.io

3., snapshotoljuk Virtualboxban, ha jó, majd kikapcs/bekapcs **sudo systemctl isolate poweroff.target**

4., újraindítás után ellenőrizzük a yaml-t fájlt megvan-e a változtatás, majd újraellenőrizzük a DNS szerverünet a DNS szerver ellenőrzése részben leírtak szerint (netstat, ip addr, ping, nslookup)

Névfeloldás

1., Az Ubuntu Serveren a névfeloldás a DNS (Domain Name System) segítségével történik, amelyet a rendszer a systemd segítségével történik: **sudo nano /etc/systemd/resolved.conf**
(régebbi ubuntu ill. más rendszeknél: sudo nano /etc/resolved.conf). Jól kitöltött zónafájl esetében nem kötelező, de nagyon ajánlott. Hasznos lehet a lokális DNS cache-elés és lekérdezések kezelésére

[Resolve]
DNS=10.0.2.15
Domains=rozsa.bimbo

Megjegyzés, a systemd resolved.conf bővebb manuált ad, ha átolvassuk.

2., Újraindítjuk a systemd-resolved szolgáltatást

sudo systemctl restart systemd-resolved

A DNS szerver ellenőrzése

1. Netstat: **sudo netstat -tuln | grep :53 | grep 10.0.2.15**
Itt a TCP-nek figyelniük kell (listen), UDP süket. A figyelés annyit jelet, hogy azokhoz a portokhoz lehet kapcsolódni

2. Hálózati interfész: **ip a**

3. Megpingeljük: **ping 10.0.2.15**

4. A Netcat szerint open állapotban van-e a port**: nc -zv 10.0.2.15 53** (ha nincs netcat fel kell telepíteni **sudo apt install netcat-traditional**)

5. Végül az nslookup segítségével megnézzük a névfeloldás(oka)t: Itt a kívánt nevet rendeljük a DNS szerverünkhöz (10.0.2.15), amely megmutatja a DNS szervert, portot, nevet, és a hozzárendelt IP-címet

nslookup ns.rozsa.bimbo 10.0.2.15

```
attila@atti:~$ nslookup ns.rozsa.bimbo 10.0.2.15
Server:         10.0.2.15
Address:        10.0.2.15#53

Name:   ns.rozsa.bimbo
Address: 10.0.2.15
```

nslookup www.rozsa.bimbo 10.0.2.77

```
attila@atti:~$ nslookup www.rozsa.bimbo 10.0.2.15
Server:         10.0.2.15
Address:        10.0.2.15#53

Name:   www.rozsa.bimbo
Address: 10.0.2.77
```

DHCP szerver telepítése

1., Ubuntu Serveren a DHCP szerver telepítése a isc-dhcp-server csomag telepítésével kezdődik:
sudo apt update && sudo apt install isc-dhcp-server -y

2., A DHCP konfigurációs fájljának szerkesztése: **sudo nano /etc/dhcp/dhcpd.conf** . A helyes kitöltéshez megnézzük az alapinformációkat (ip a, ip route show). Az én esetemben VBox alatt:

```
subnet 10.0.2.0 netmask 255.255.255.0 {
   range 10.0.2.100 10.0.2.150;
   option subnet-mask 255.255.255.0;
   option routers 10.0.2.2;
   option broadcast-address 10.0.2.255;
   option domain-name-servers 10.0.2.15;  # A rozsa.bimbo DNS-szervere
}
```

3. Megadjuk a vonatkoztatott hálózati interfészt: **sudo nano /etc/default/isc-dhcp-server**
INTERFACESv4="enp0s3"

4. Indítás és engedélyezés:
sudo systemctl start isc-dhcp-server
sudo systemctl enable isc-dhcp-server

Megjegyzés: a systemctl-nál a szolgáltatást stop-pal leállitjuk, disable-nem indul el következő rendszerindításkor

DHCP kliens telepítése

1., Kliensgépeken a DHCP-kliens telepítése a isc-dhcp-client csomag telepítésével kezdődik:
sudo apt update && sudo apt install isc-dhcp-client -y

(2., Linux esetében a DHCP kliens konfigurációs fájljának szerkesztésével nincs semmi teendőnk, de ha mégis, akkor itt megtehetjük: **sudo nano /etc/dhcp/dhcpclient.conf**)

3. Indítás és engedélyezés Linux alapú hálózati gépeken:
sudo systemctl start isc-dhcp-client
sudo systemctl enable isc-dhcp-client

4. Új IP-cím kérése a szerverhez:
sudo dhclient -r - újraindítja (release) az IP-cím kezelés folyamatát
sudo dhclient - elindítja az új IP-cím kérést a DHCP szervertől

Megjegyzés: a systemctl-nál a szolgáltatást stop-pal leállitjuk, disable-nem indul el következő rendszerindításkor

DNS + DHCP telepítési összefoglaló

Alapparancsok telepítése: nano, net-tools, iputils-ping, isc-dhcp-server, bind9, dnsutils stb.

Hálózati címzés próbára a Google (8.8.8.8)-vel

DNS-szerver telepítése: esetemben a 10.0.2.15 –re

Zóna meghatározása: miből fog állni a rozsa.bimbo .. dns-szerver, webszerver stb.

Névfeloldás: DNS-szerver hozzárendelése a tartományhoz (10.0.2.15 a rozsa.bimbo-hoz)

DNS-szerver ellenőrzése: 4 eszközzel

DHCP szerver telepítés a DNS-szerver IP-címmel

DHCP kliensek telepítése a hálózati linuxos gépekre

Webszerver telepítés - Apache

Ubuntu Serveren kétféle webszerver telepítés van. Az egyik az Apache, másik az Nginx. Az utóbbi az újabb, viszont az apache-t fogjuk megcsinálni.

1., Előfeltételek: DNS zónafájlját kitöltöttük az előzőekben (lásd a zónakonfigurációs részt a prezentációban). Felvettük a tartományt – rozsa.bimbo, és hozzárendeltük a www-hez a 10.0.2.77-es IP-t. Telepítettük a KDE-t is az SDDM-el (lásd grafikus felületek részt). Ez a böngészőnk grafikus felülete végett lesz mérvadó (Konqueror, Firefox stb.)

2., telepítjük az apache webszervert: **sudo apt update && sudo apt install apache2 -y**

3., Az apache alapkonfiguráció:

a., **sudo nano /etc/apache2/sites-available/000-default.conf**

```
<VirtualHost 10.0.2.77:80>
        ServerName www.rozsa.bimbo
        DocumentRoot /var/www/html
</VirtualHost>
```

b., **sudo nano /etc/apache2/apache2.conf** – itt most nem törlünk, csak a file végére beillesztjük:

```
ServerName www.rozsa.bimbo
```

4., Lefuttatjuk a konfigurációs ellenőrzést **sudo apache2ctl configtest**, itt Syntax OK.

WWW – Tűzfal, Host, IP, Netplan, portok

5., Tűzfalon engedélyezzük a 80-as portot - **sudo ufw allow 80/tcp**
Utána pedig érvénybe léptetjük - **sudo ufw enable**
Majd lekérdezzük - **sudo ufw status** – itt látnunk kell a 80-as portot, hogy nyitva

6., A host fájlban megcsináljuk a hozzárendelést**: sudo nano /etc/hosts**
10.0.2.77 www.rozsa.bimbo

7., A Netplan-ban frissítjük a táblázatot: **sudo nano /etc/netplan/50-cloud-init.yaml** – hozzáadjuk a virtuális interfészünknél a "- 10.0.2.77/24-es" kívánt webszerver IP-t (hogy a következő újraindításnál megjegyezze), majd érvénybe léptetjük **sudo netplan apply**

8., Apache – "Listen port" – végül konfiguráljuk az Apache portját:
sudo nano /etc/apache2/ports.conf
Listen 10.0.2.77:80 – hozzáadjuk a webszerverünk kívánt IP-címét, és a figyelő-portot, amelyet keresztülengedtünk a tűzfalon

9., Végső config teszt - **sudo apache2ctl configtest** .... Syntax OK-t kell kapjunk

10., Az apache frissítése, érvénybe léptetése, majd VirtualBox snapshot:
sudo systemctl restart apache2
sudo systemctl enable apache2

Webszerver – Próba

```
attila@atti:~$ nslookup www.rozsa.bimbo
Server:         127.0.0.53
Address:        127.0.0.53#53

Name:   www.rozsa.bimbo
Address: 10.0.2.77

attila@atti:~$ nslookup ns.rozsa.bimbo
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   ns.rozsa.bimbo
Address: 10.0.2.15
```

nslookup

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no
      addresses:
      - 10.0.2.15/24
      - 10.0.2.77/24
      gateway4: 10.0.2.2
      nameservers:
        addresses:
        - 10.0.2.15
```

Netplan (50-cloud-init.yaml)

```
attila@atti:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
80/tcp                     ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
80/tcp (v6)                ALLOW       Anywhere (v6)

attila@atti:~$
```

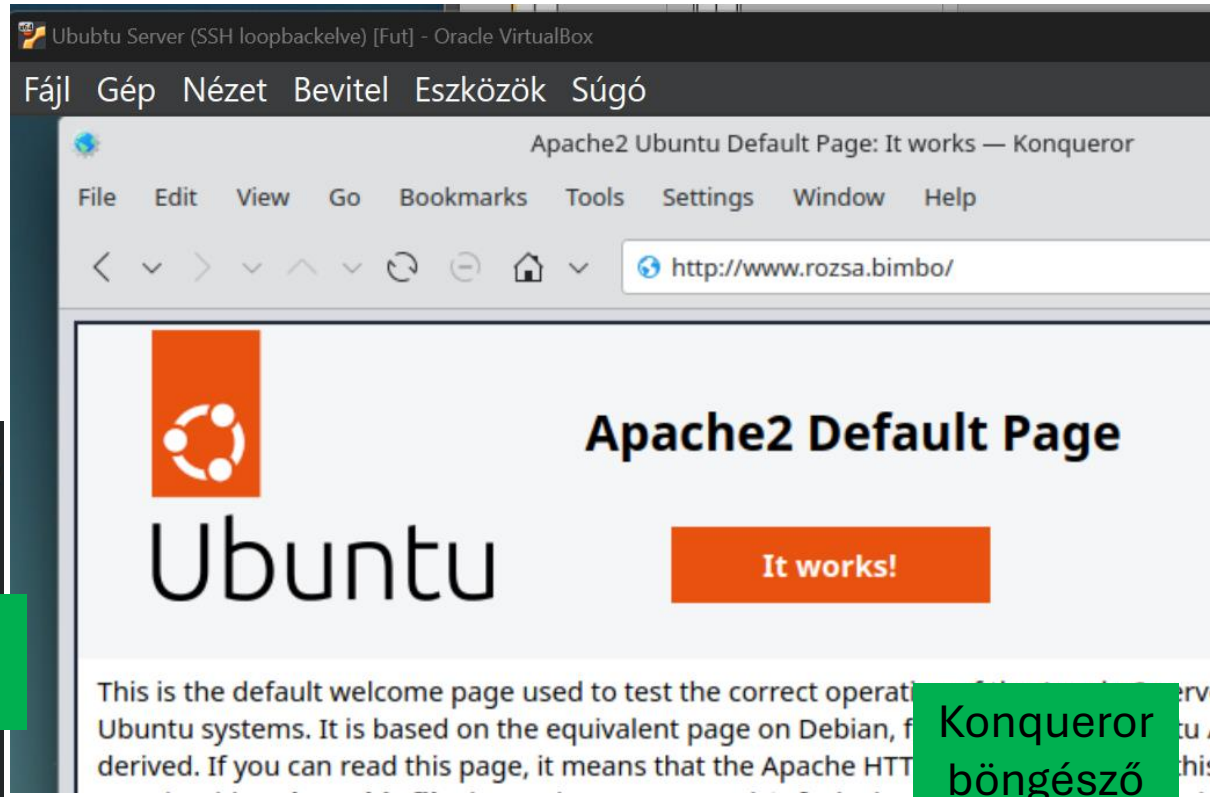
tűzfal

```
attila@atti:~$ sudo apache2ctl configtest
[sudo] password for attila:
Syntax OK
attila@atti:~$
```

configtest

```
attila@atti:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Thu 2025-01-02 21:17:43 UTC; 1h 5min ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 5724 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 5727 (apache2)
      Tasks: 55 (limit: 38430)
     Memory: 5.7M (peak: 6.4M)
        CPU: 614ms
     CGroup: /system.slice/apache2.service
             ├─5727 /usr/sbin/apache2 -k start
             ├─5728 /usr/sbin/apache2 -k start
             └─5730 /usr/sbin/apache2 -k start

Jan 02 21:17:43 atti systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jan 02 21:17:43 atti systemd[1]: Started apache2.service - The Apache HTTP Server.
attila@atti:~$
```

apache2 status



Konqueror böngésző

Webszerver - háttéradatbázis telepítése PHP támogatással

A háttéradatbázis kiszolgáló (MySQL vagy a MariaDB) kulcsszerepet játszik a dinamikus weboldalak és webalkalmazások működtetésében. Fő célja az adatkezelés, felhasználói menedzsment, biztonság, adatintegritás és skálázhatóság. Ubuntu Serveren a háttéradatbázis telepítéséhez PHP-re és PHP támogató Apache modulra és a MariaDB-re (nyílt forráskódú MySQL kiszolgáló) lesz szükségünk.

1., Előfeltételek: futó apache2 webszerver

2., PHP telepítése: **sudo apt update && sudo apt install php libapache2-mod-php -y**

3., MariaDB telepítése: **sudo apt install mariadb-server mariadb-client -y**

4., Adatbázis inicializáció, root jelszó: **sudo mysql_secure_installation**

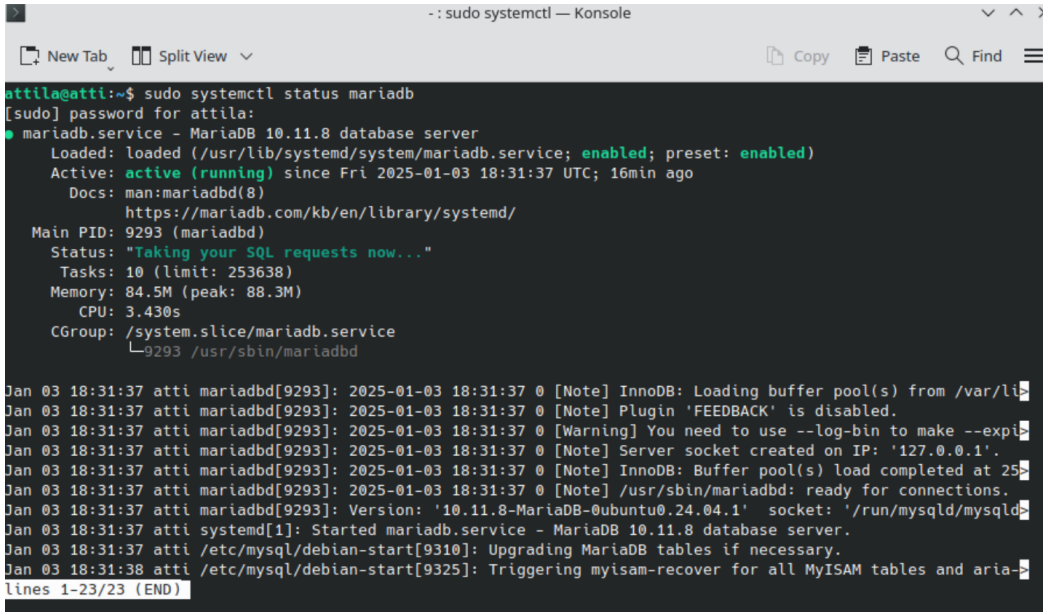
5., Webszerver újraindítás: **sudo systemctl restart apache2.service**

6., PHP fájl létrehozása: **sudo nano /var/www/html/phpinfo.php**

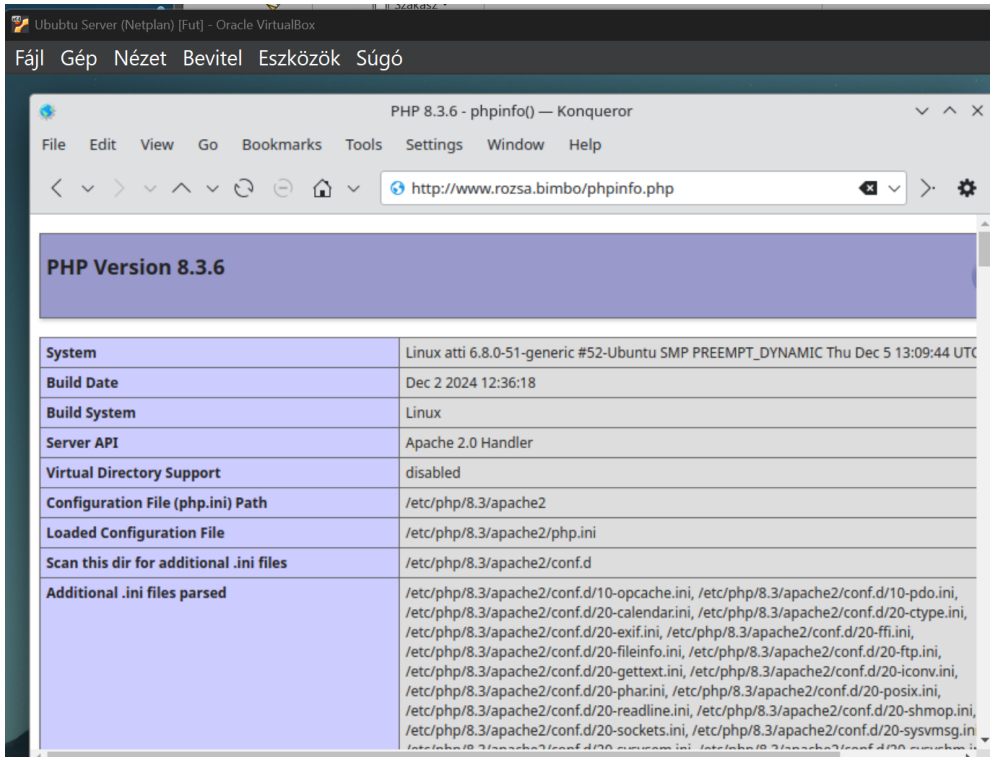
```
<?php
phpinfo();
?>
```

Webszerver - háttéradatbázis telepítése PHP támogatással - próba

7., MariaDB folyamatlekérdezés



8., Próba böngészőben:
http://10.0.2.77/phpinfo.php
(....vagy http://www.rozsa.bimbo/phpinfo.php)



Adatbázis szerverek
MySQL, MariaDB, PostgreSQL

1., MySQL – hasonló a menet, mint a MariaDB-nél, csak mysql-server csomagot telepítjük a mariadb-server helyett.

2., MariaDB – megcsináltuk, lényegében egy MySQL fork, fejlettebb

2., PostgreSQL – objektum-relációs adatbázis-kezelő rendszer (RDBMS).
Telepítése, adatbázis létrehozás, és PHP integráció:

1., feltelepítjük a csomagokat: **sudo apt install postgresql postgresql-contrib php-pgsql –y**
2., elindítjuk a szolgáltatást - **sudo systemctl start postgresql**
3. érvénybe léptetjük - **sudo systemctl enable postgresql**
4. Alapbeállítások - **sudo -i -u postgres**
5. Elindítjuk - **psql**
6. PSQL-ben felvétel - CREATE USER attila WITH PASSWORD 'JrT123456’;
   CREATE DATABASE papagájok OWNER attila;
   \q
7. Érvénybe léptetés - **sudo systemctl restart apache2**

LAMP

A LAMP egy népszerű nyílt forráskódú szoftvercsomag, amelyet dinamikus weboldalak és webalkalmazások fejlesztésére használnak. A neve négy összetevő betűszavából ered: Linux (operációs rendszer), Apache (webszerver), MySQL vagy MariaDB (adatbázis-kezelő rendszer), és PHP, Perl vagy Python (programozási nyelv). Ezek együttműködve biztosítanak egy stabil és rugalmas környezetet a webfejlesztők számára.

A XAMPP a windows alatti régi megoldás volt, ahol X-a cross platformot jelentette, és az utolsó P a rövidítésben az a Perl volt. Jelenleg a WAMP használatos e téren

Egyszerűbb megoldás tehát telepíteni a LAMP-ot, de csak első ránézésre. Különösen hátrányos akkor, ha egyedi szervereket telepítenél külön-külön. A külön telepített szerverek előnye a LAMP csomaggal szemben a rugalmasság, testreszabhatóság, teljesítmény, inkompatibilitás – továbbá, ha külön telepítem, akkor a moduláris frissítések kevesebb rendszerszintű beavatkozást jelentenek.

Adatbázis szerverek
MS SQL (Microsoft)

1., Letölteni a Microsoft SQL Sever nyilvános GPG kulcsát:
curl -fsSL https://packages.microsoft.com/keys/microsoft.asc | sudo gpg --dearmor -o /usr/share/keyrings/microsoft-prod.gpg

2., MS SQL Server Ubuntu repository letöltése:
curl -fsSL https://packages.microsoft.com/config/ubuntu/22.04/mssql-server-2022.list | sudo tee /etc/apt/sources.list.d/mssql-server-2022.list

3., MS SQL telepítése:
sudo apt update && sudo apt install mssql-server -y

4., Adatbázis konfiguráció
sudo /opt/mssql/bin/mssql-conf setup

5., Kiszolgáló újraindítása
sudo systemctl restart mssql-server

6., Ha szükséges a távelérés – itt az alapértelmezett a 1433/TCP port
sudo ufw allow 1433/tcp
sudo ufw enable

FTP-szerver telepítés, konfiguráció

Az előzőekben kitaláltuk, hogy a 10.0.2.88-as IP-címünk legyen az FTP-szerver, és a szerver neve ftp legyen

1., Letöltjük és telepítjük a proftpd csomagot: **sudo apt update && sudo apt install proftpd -y**

2., Elindítjuk: **sudo systemctl start proftpd**

3., Szerkesztjük a proftpd konfigurációs fájlját, megadjuk a szerver nevét:
sudo nano /etc/proftpd/proftpd.conf
ServerName "ftp"

4., A tűzfalon átengedjük a 21-es portot, majd érvénybe léptetjük: **sudo ufw allow 21/tcp**
sudo ufw enable

5., Újraindítjuk a proftpd kiszolgálót: **sudo systemctl restart proftpd**

6., Lekérdezzük az állapotát: **sudo systemctl status proftpd**

FTP-szerver beállítás IP-re

7., A Netplan-ban frissítjük a táblázatot: **sudo nano /etc/netplan/50-cloud-init.yaml** – hozzáadjuk a virtuális interfészünknél a “- 10.0.2.88/24-es” kívánt FTP-szerver IP-t (hogy a következő újraindításnál megjegyezze), majd érvénybe léptetjük **sudo netplan apply**

8., A zónabázisban megnézzük, hogy felvettük-e már korábban a zónába a végére:
(esetünkben a zóna rozsa.bimbo): **sudo nano /etc/bind/db.rozsa.bimbo**
ftp IN A 10.0.2.88 - ha nem, akkor hozzáadjuk, mentjük

9., A helyi zónában megnézzük, hogy master-ra (elsődlegesre) van-e állítva a zónabázis:
sudo nano /etc/bind/named.conf.local

```
zone "rozsa.bimbo" {
   type master;
   file "/etc/bind/db.rozsa.bimbo";
};
```

10., Újraindítjuk a DNS-szervert: **sudo systemctl restart bind9**

FTP-szerver próba

```
attila@atti:/etc/bind$ nslookup ftp.rozsa.bimbo
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   ftp.rozsa.bimbo
Address: 10.0.2.88

attila@atti:/etc/bind$
```

nslookup ftp.rozsa.bimbo

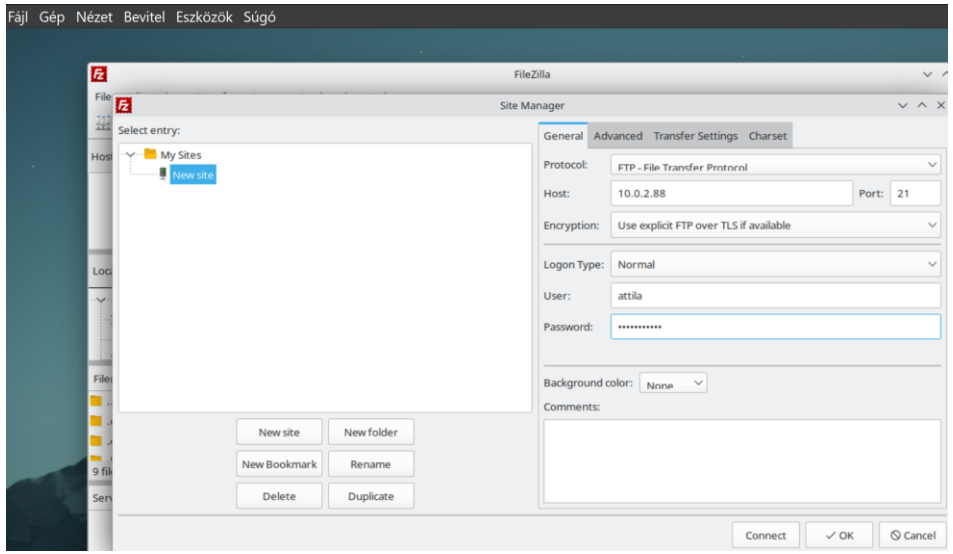
```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no
      addresses:
      - 10.0.2.15/24
      - 10.0.2.77/24
      - 10.0.2.88/24
      gateway4: 10.0.2.2
      nameservers:
        addresses:
        - 10.0.2.15
attila@atti:~$
```

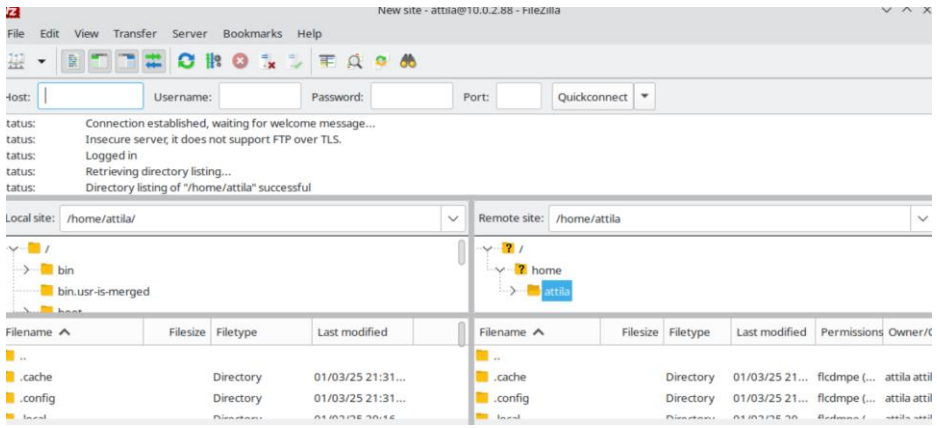
Netplan (50-cloud-init.yaml)

```
attila@atti:~$ sudo systemctl status proftpd
● proftpd.service - ProFTPD FTP Server
     Loaded: loaded (/usr/lib/systemd/system/proftpd.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-01-03 21:18:31 UTC; 20min ago
       Docs: man:proftpd(8)
    Process: 3525 ExecStartPre=/usr/sbin/proftpd --configtest -c $CONFIG_FILE $OPTIONS (code=ex
    Process: 3527 ExecStart=/usr/sbin/proftpd -c $CONFIG_FILE $OPTIONS (code=exited, status=0/S
   Main PID: 3528 (proftpd)
      Tasks: 1 (limit: 38430)
     Memory: 2.0M (peak: 3.7M)
        CPU: 218ms
     CGroup: /system.slice/proftpd.service
             └─3528 "proftpd: (accepting connections)"

Jan 03 21:18:31 atti systemd[1]: Starting proftpd.service - ProFTPD FTP Server...
Jan 03 21:18:31 atti proftpd[3525]: Checking syntax of configuration file
Jan 03 21:18:31 atti systemd[1]: proftpd.service: Failed to parse PID from file /run/proftpd.pi
                                 - ProFTPD FTP Server.
                                 ion): session opened for user attila(u
                                 ion): session closed for user attila
```

Proftpd - állapotlekérdezés

Filezilla (de lehet más ftp kliens is):
sudo apt install filezilla





HTTPS és FTPS Ubuntu Serveren

Ahhoz, hogy el tudjuk érni a kívánt biztonsági szintet a következőkre van szükség:

1., Az openssl csomag telepítése: **sudo apt update && sudo apt install openssl -y**

2., SSL/TLS tanusítvány létrehozása pl. Apache-hoz, amit nem CA adott ki, hanem „önaláírt”: **sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt**

3., SSL/TLS modul engedélyezése – esetünkben Apache-n.: **sudo a2enmod ssl**, majd **sudo systemctl restart apache2** –al frissítjük a webszevert.

4., Virtuális HOST konfiguráció – **sudo nano /etc/apache2/sites-available/default-ssl.conf**

<VirtualHost 10.0.2.77:443>
ServerAdmin attila@rozsa.bimbo
DocumentRoot /var/www/html

SSLEngine on
SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
</VirtualHost>

5., Érvénybe léptetjük a 443-as porthoz rendelt hostot: **sudo a2ensite default-ssl.conf**, majd **sudo systemctl restart apache2** –al frissítjük a webszevert.

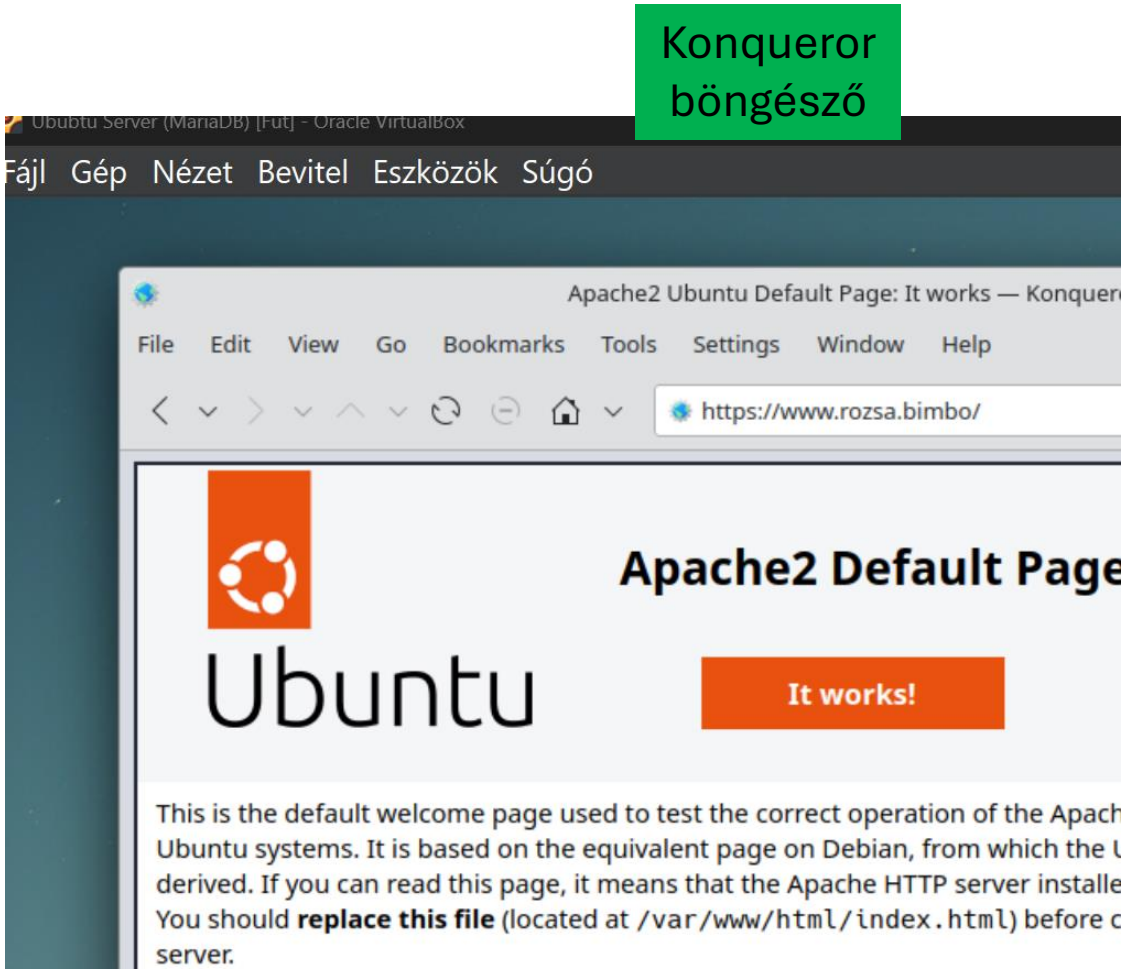
6., tűzfal konfiguráció a 443-as portra: **sudo ufw allow 443/tcp**, majd **sudo ufw enable**

HTTPS és FTPS próba

tűzfal

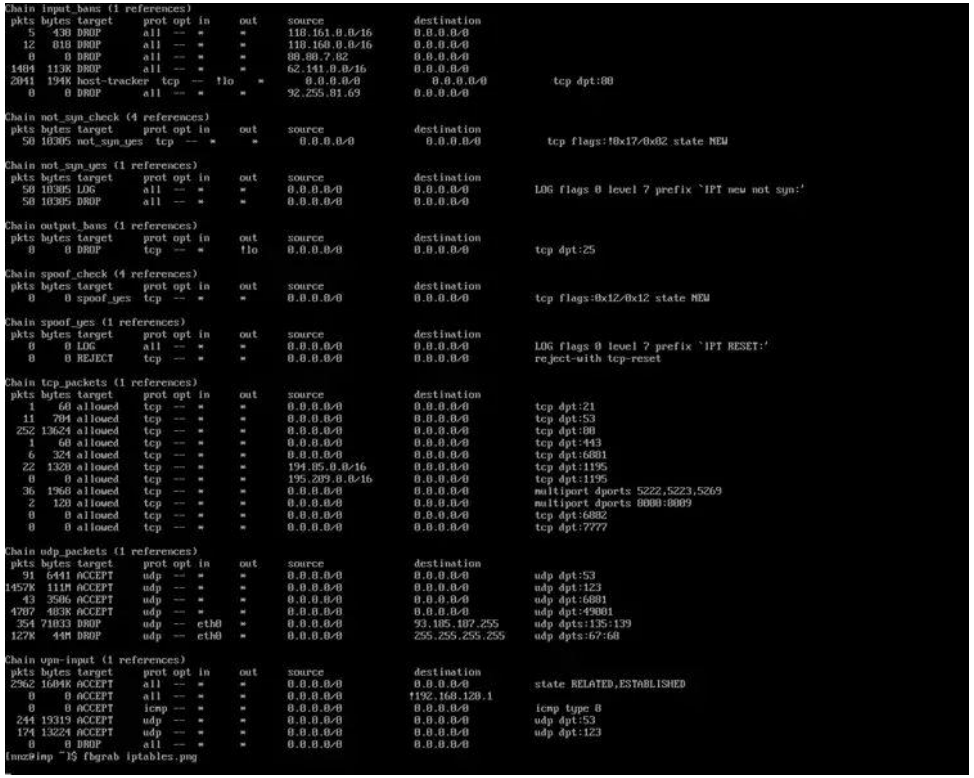
```
attila@atti:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
80/tcp                     ALLOW       Anywhere
21/tcp                     ALLOW       Anywhere
443/tcp                    ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
80/tcp (v6)                ALLOW       Anywhere (v6)
21/tcp (v6)                ALLOW       Anywhere (v6)
443/tcp (v6)               ALLOW       Anywhere (v6)

attila@atti:~$
```

Konqueror böngésző



Tűzfalszabályok – iptables

Az iptables lehetővé teszi a rendszergazda számára, hogy konfigurálja a Linux kernel tűzfalának IP-csomagszűrési szabályait, amelyeket különböző Netfilter modulokként valósítanak meg. A szűrők táblázatokba vannak rendezve, amelyek szabályláncokat (INPUT, OUTPUT, FORWARD) tartalmaznak a hálózati forgalom csomagjainak kezeléséhez. Jelenleg különböző kernelmodulokat és programokat használnak a különböző protokollokhoz; *Az iptables* IPv4-re, *az ip6tables* IPv6-ra, *az ebtables* Ehernetes keretekre vonatkozik.



Tűzfalszabályok megvalósítása

1., Tűzfalszabály hozzáadása - hozzáadhatunk szabályokat a különböző szabályláncokhoz.
Pl: a bejövő láncban engedélyezi a 80-as portot (http default):
sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT

2., Tűzfalszabály eltávolítása - eltávolíthatunk szabályokat a különböző szabályláncokból.
Pl: a kimenő láncból eltávolítjuk a 21-as portot (ftp default):
sudo iptables -D OUTPUT -p tcp --dport 21 -j ACCEPT

3., Tűzfalszabály eltávolítása folyamatazonosítóval:
Pl: a továbbított láncból eltávolítjuk a 17-es folyamatotOUTPUT
sudo iptables -D FORWARD 17

4., Tűzfalszabályok listázása - **sudo iptables -L**

Proxy konfiguráció – squid

A proxy egy közvetítő szerver, amely átirányítja és anonimizálja a webes forgalmat, de nem titkosítja azt. A VPN egy titkosított csatornát hoz létre a felhasználó és az internet között, teljes adatvédelmet és biztonságot biztosítva minden online tevékenységhez. Míg a proxy gyorsabb lehet, a VPN átfogóbb biztonságot nyújt. Ubuntu Serveren a proxy megvalósítása a squid-del történik pl.

1., Squid telepítése: **sudo apt update && sudo apt install squid -y**

2., Squid konfigurálása: **sudo nano /etc/squid/squid.conf**
Pl1. http_port 8888 – port átirányítás pl. HTTP-t a 8888-ra
Pl2. acl my_network dstdomain .rozsa.bimbo – a hálózati HTTP forgalom csak a zónámnak
http_access allow my_network van engedélyezve, Squid Access List-tel.
Pl3.acl hu_domains dstdomain .hu - az összes magyar domainre engedély
http_access allow hu_domains

3., Áirányítás a Squid proxynkra: pl.:
sudo iptables -t nat -A PREROUTING -p tcp -d .hu --dport 80 -j REDIRECT --to-port 8888

4., Érvénybe léptetjük a proxy szervert: **sudo systemctl restart squid**

Levelezőszerverek

Ubuntu Serveren a levelezőszerverek megvalósításához a következő protokollok a felelősek:

SMTP - a leggyakrabban használt protokoll az e-mailek küldésére. Felelősek az üzenetek átviteléért az egyik szervertől a másikig.

Postfix **-** nyílt forráskódú, nagy teljesítményű és biztonságos Mail Transfer Agent (MTA), amelyet széles körben használnak az e-mail továbbítására.

Sendmail - régebbi és hagyományos UNIX rendszereken gyakran használt MTA szoftver. Bár nagy teljesítményű, a konfigurációja bonyolultabb lehet.

Exim egy rendkívül rugalmas és testreszabható MTA, amelyet széles körben használnak. Könnyen konfigurálható és karbantartható.

POP3 - (Post Office 3) egy protokoll, amely lehetővé teszi az e-mailek letöltését a szerverről a felhasználó számítógépére. Miután az e-maileket letöltötték, azok törlődnek a szerverről.

IMAP (Internet Message Access Protocol) lehetővé teszi az e-mailek kezelését közvetlenül a szerveren. Az e-mailek szinkronizálva maradnak a különböző eszközök között, így minden eszközön ugyanaz az e-mail állapot látható.

IMAP levelezőszerver

Ubuntu Serveren a levelezőszerverek megvalósításához a következő protokollok a felelősek:

1., Dovecot telepítése.: **sudo apt update && sudo apt install dovecot-core dovecot-imapd -y**

2., Dovecot konfiguráció: **sudo nano /etc/dovecot/dovecot.conf**
protocols = imap

3., Postafiókok helyének konfigurációja - **sudo nano /etc/dovecot/conf.d/10-auth.conf**
mail_location = maildir:~/mailmappa – a /home/attila/mailmappa lesz a postafiókok helye

4., Autentikáció beállítása - **sudo nano /etc/dovecot/conf.d/10-mail.conf**
disable_plaintext_auth = no – ne legyen kommentelve a sor a configban
auth_mechanisms = plain login – az azonosítás engedélyezése

IMAP levelezőszerver SSL/TLS

5., SSL kulcsok generálása Dovecatnak - **sudo openssl req -new -x509 -days 365 -nodes -out /etc/ssl/certs/dovecot.pem -keyout /etc/ssl/private/dovecot.pem**

6., SSL/TLS engedélyezése - **sudo nano /etc/dovecot/conf.d/10-ssl.conf**

ssl = required
ssl_cert = </etc/ssl/certs/dovecot.pem
ssl_key = </etc/ssl/private/dovecot.pem

7., A tűzflalon való átengedés 143 (IMAP), 993 (IMAP SSL)
sudo ufw allow 143/tcp
sudo ufw allow 993/tcp

8., A Dovecot elindítása:
sudo systemctl start dovecot
sudo systemctl enable dovecot – érvénybe léptetés
sudo systemctl status dovecot – állapotlekérzdés

```
attila@atti:~$ sudo systemctl status dovecot
● dovecot.service - Dovecot IMAP/POP3 email server
     Loaded: loaded (/usr/lib/systemd/system/dovecot.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-01-04 00:28:09 UTC; 7min ago
       Docs: man:dovecot(1)
             https://doc.dovecot.org/
   Main PID: 5848 (dovecot)
     Status: "v2.3.21 (47349e2482) running"
      Tasks: 4 (limit: 38430)
     Memory: 3.2M (peak: 3.6M)
        CPU: 77ms
     CGroup: /system.slice/dovecot.service
             ├─5848 /usr/sbin/dovecot -F
             ├─5849 dovecot/anvil
             ├─5850 dovecot/log
             └─5851 dovecot/config

Jan 04 00:28:09 atti systemd[1]: Starting dovecot.service - Dovecot IMAP/POP3 email server...
Jan 04 00:28:09 atti dovecot[5848]: master: Dovecot v2.3.21 (47349e2482) starting up for imap
Jan 04 00:28:09 atti systemd[1]: Started dovecot.service - Dovecot IMAP/POP3 email server.
```

Shell-szriptek

A shell scriptek olyan szöveges fájlok, amelyek parancsokat és utasításokat tartalmaznak a shell (például Bash, Zsh, vagy Ksh) számára. Ezek a parancsok és utasítások automatizálják a feladatokat, amelyek különben kézi beavatkozást igényelnének. A shell scriptek nagyban megkönnyítik a rendszeradminisztrációs feladatokat és a mindennapi munkafolyamatokat.

Alapvető példák:

Létrehozzuk a hellovilag szkriptet: **nano hellovilag.sh**
#!/bin/bash
echo "Hello, világ!„
date

Miután elmentettük megadjuk neki a végrehajtási jogosultságot:
chmod +x hellovilag.sh

Ezután futtatjuk a scriptet **./hellovilag.sh**

Shell-szriptek – vezérlési szerkezetek

A shell-szkriptek vezérlési szerkezetei:

1., If-Else

```
#!/bin/bash
if [ $1 -gt 100 ]; then
  echo "A szám nagyobb, mint 100."
else
  echo "A szám kisebb vagy egyenlő, mint 100."
fi
```

2., For ciklus (C-stílus)

```
#!/bin/bash
for ((i = 1; i <= 5; i++)); do
  echo "A szám: $i"
done
```

3., While ciklus

```
#!/bin/bash
counter=1
while [ $counter -le 5 ]; do
  echo "A számláló értéke: $counter"
  ((counter++))
done
```

4., Függvények

```
#!/bin/bash
function intro {
  echo "Szia, a fajtám, $1!"
}
intro "kecskepapagáj"
greet "szürkepapagáj"
```

Lemezek csatolása Ubuntu Serverhez

VirtualBox-ban létrehoztunk még 2 db virtuális lemezt a következő módon:

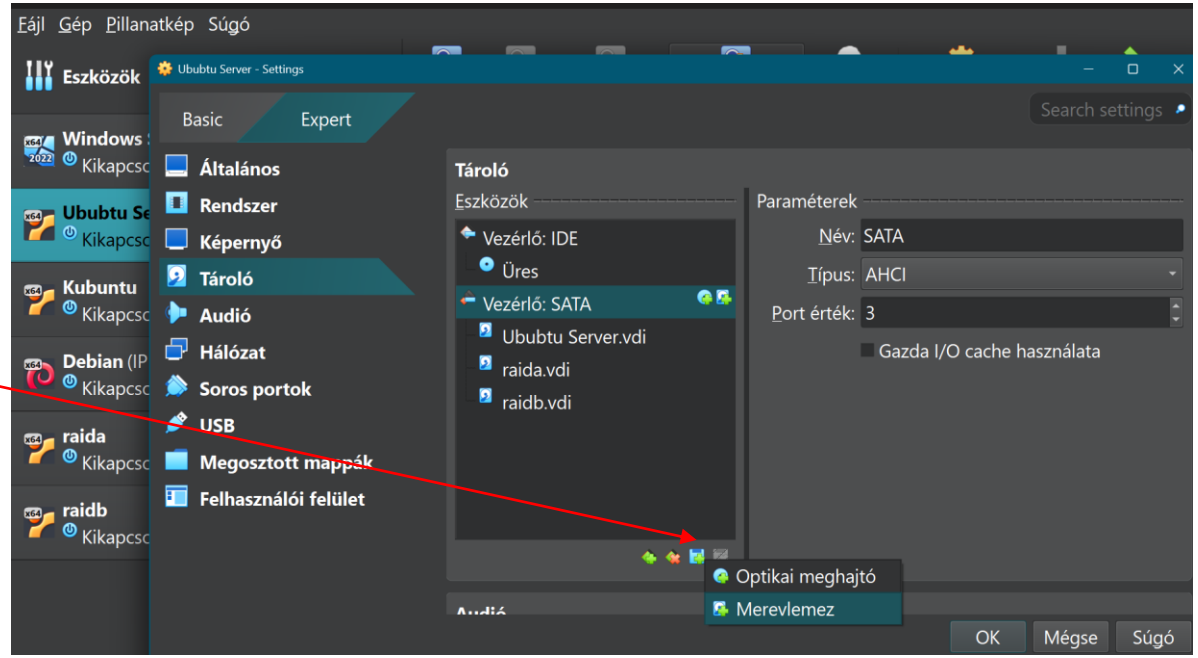
1., Eszközök ➡ Új ➡ Név: pl. raida ➡ Tipus: Linux, altipus Ubuntu, verzió Ubuntu (64-bit)

2., Megadjuk a Hardverparamétereket, majd a Harddisk-nél pl.10 GB, és bekapcsoljuk a “Pre-allocate Full Size” opciót, hogy előre foglalja le az összes területet, hogy linuxban ne irogassa dinamikusan a RAID diszkeket (nyilván ez most szentségtörés)

3., Az első pontban megfogalmazottak alapján létrehozzuk a 2.lemez is, amelynek a neve raidb, ugyanzokkal a paraméterekkel, mint az első lemezt! (fontos)

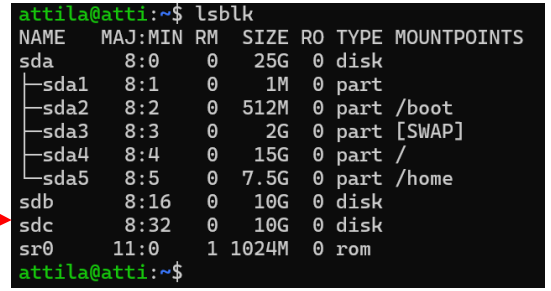
4., VirtualBoxban egyesével csatoljuk a 2 db új virtuális lemezünket az Ubuntu Server beállításainál:
Tároló: Vezérlő SATA, itt a 3.ikon a “Merevlemez”

5., Elindítjuk az Ubuntu Server-t



Partícionálás Ubuntu Serveren

1, Parancssorban kiadjuk az **lsblk** parancsot,
ahol látnunk kell a két VirtualBoxban létrehozott
10 GB-os lemezt sdb, sdc név alatt



2, megcsináljuk az MBR alapú partícionálást, partíciós táblázat NÉLKÜL!! (RAID-hez nem kell, amúgy o-billentyű az MBR PT létrehozása): **sudo fdisk /dev/sdb** , majd itt:

a., n (új partíció) ➡ 1 (elsődleges) ➡ megadjuk a méretet / szektorméretet

b., t (partíció típusa) ➡ 83 (Linux)

c., w (elmenti a beállításokat és kilép a partícionálóból)

3, Ellenőrizzük a partíciót: **sudo fdisk -l**

4, Elnevezzük a frissen elkészült partíciónkat „angyalok”-ra majd ext4 fájlrendszert írunk rá
sudo mkfs.ext4 -L angyalok /dev/sdb1

5, Létrehozzuk a csatolási könyvtárat: **sudo mkdir /mnt/angyalok**

6, Csatoljuk a partíciót: **sudo mount /dev/sdb1 /mnt/angyalok**

7, Ellenőrizzük, hogy sikerült-e a csatolás: **df -h**

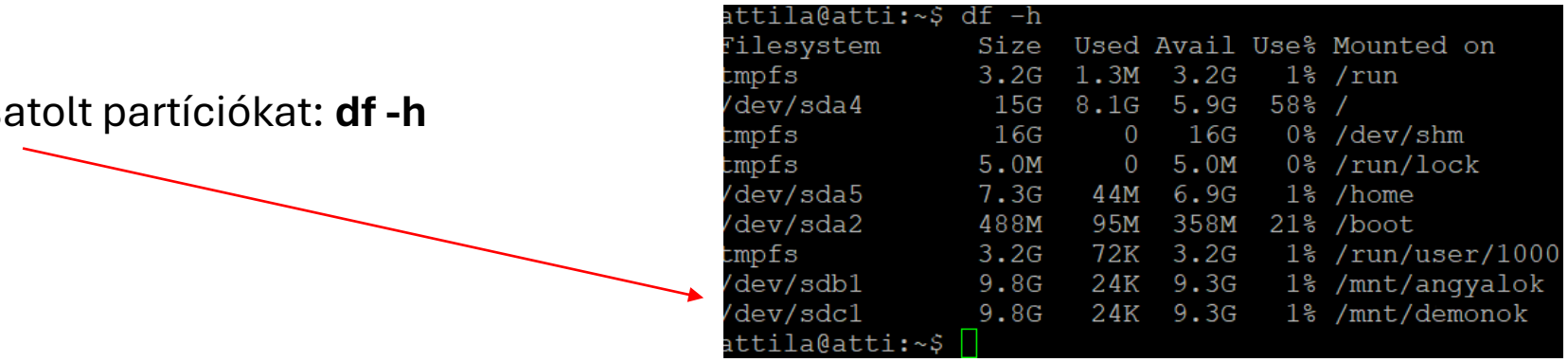
8, Opcionálisan ha szeretnénk minden rendszerindításkor automatikusan csatolni, akkor az fstab-ban hozzáadjuk a bejegyzést: **sudo nano /etc/fstab** (később a **sudo reboot**-tal érvénybe léptethetjük)
/dev/sdb1 /mnt/angyalok ext4 defaults 0 2

RAID 1 - feltételek

1, Ismételjük meg ugyanezeket a lépéseket a másik - sdc virtuális lemezre is, és címkézzük meg mint ”demonok”. (Partícionálás Ubuntu Serveren fejezet):

sudo fdisk /dev/sdc - sdc partícionálás
sudo mkfs.ext4 -L demonok /dev/sdc1 - sdc1 elsődleges partíción fájlrendszer
sudo mkdir /mnt/demonok - csatolási pont létrehozás
sudo mount /dev/sdc1 /mnt/demonok - csatolás

2, ellenőrizzük a felcsatolt partíciókat: **df -h**



RAID 1-es tükrözés létrehozása

1, Lecsatoljuk a létrehozott két partíciót:
sudo umount /mnt/angyalok
sudo umount /mnt/démonok

2, Telepítjük a RAID tükrözéshez szükséges mdadm csomagot (ha nincs még fent...)
sudo apt update && sudo apt install mdadm -y

3, Létrehozzuk a RAID 1 típusú tükrözést 2 lemezre, amelynek a címkéje angyalokesdemonok
sudo mdadm --create /dev/md/angyalokesdemonok -l 1 -n 2 /dev/sdb1 /dev/sdc1
yes-sel jóváhagyjuk. Az mdadm parancsnál ki kell írnunk, hogy --create, nincs flag opció
-l – a tükrözés típusa –esetünkben 1-es
-n – lemezek száma –esetünkben 2-db

4, A RAID tömb állapotának ellenőrzése: **cat /proc/mdstat és a sudo systemctl status mdmonitor**

```
attila@atti:~$ cat /proc/mdstat
Personalities : [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md127 : active raid1 sdc1[1] sdb1[0]
      10475520 blocks super 1.2 [2/2] [UU]

unused devices: <none>
attila@atti:~$
```

```
attila@atti:~$ sudo systemctl status mdmonitor
● mdmonitor.service - MD array monitor
     Loaded: loaded (/usr/lib/systemd/system/mdmonitor.service; static)
     Active: active (running) since Sat 2025-01-04 12:35:27 UTC; 4min 18s
       Docs: man:mdadm(8)
   Main PID: 3538 (mdadm)
      Tasks: 1 (limit: 38430)
     Memory: 396.0K (peak: 656.0K)
        CPU: 36ms
     CGroup: /system.slice/mdmonitor.service
             └─3538 /sbin/mdadm --monitor --scan

Jan 04 12:35:27 atti systemd[1]: Started mdmonitor.service - MD array moni
Jan 04 12:35:27 atti mdadm[3538]: mdadm: NewArray event detected on md dev
Jan 04 12:35:27 atti mdadm[3538]: mdadm: RebuildStarted event detected on
Jan 04 12:36:27 atti mdadm[3538]: mdadm: Rebuild76 event detected on md de
Jan 04 12:37:27 atti mdadm[3538]: mdadm: RebuildFinished event detected or
```

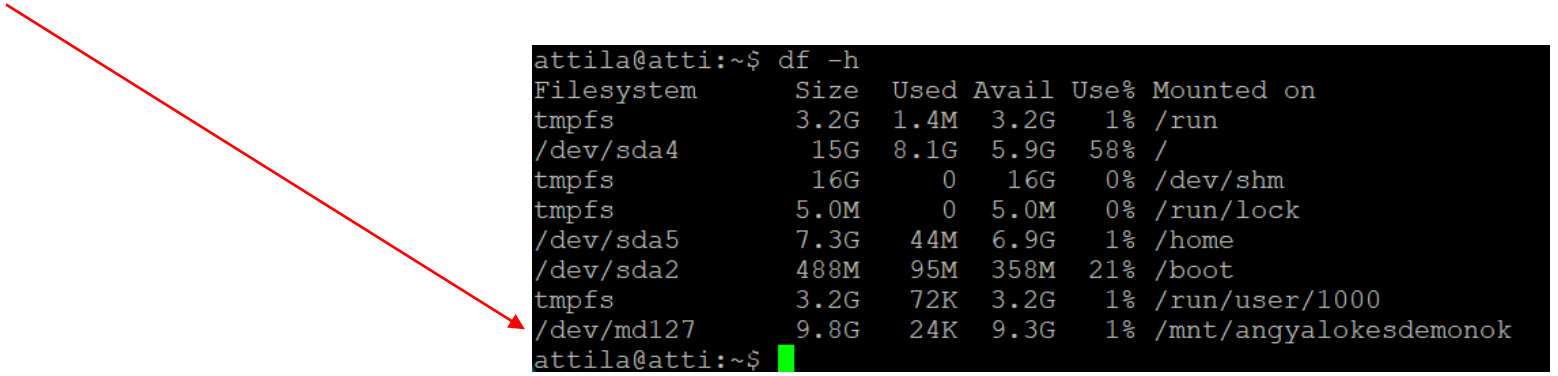
5, A RAID formázása: **sudo mkfs.ext4 /dev/md/angyalokesdemonok**

RAID 1-es tükrözés csatolása

1, Létrehozzuk a csatolási pontot: **sudo mkdir /mnt/angyalokesdemonok**

2, Csatoljuk a RAID-et: **sudo mount /dev/md/angyalokesdemonok /mnt/angyalokesdemonok**

3, Ellenőrizzük, hogy sikerült-e a csatolás: **df -h**



4, Opcionálisan ha szeretnénk minden rendszerindításkor automatikusan csatolni, akkor az fstab-ban hozzáadjuk a bejegyzést: **sudo nano /etc/fstab** (ezt pl. **sudo reboot** paranccsal érvénybe léptethetjük)
/dev/md/angyalokesdemonok /mnt/angyalokesdemonok ext4 defaults 0 0

5, A RAID automatikus aktiválásához rendszerindításkor kiadjuk:
sudo sh -c 'mdadm --detail --scan >> /etc/mdadm/mdadm.conf' – RAID scan, hozzáfűzés a confighoz
sudo update-initramfs -u - initframs ideiglenes gyökérrendszer frissítés

RAID - kiegészítő gondolatok

A RAID konfigurálásakor nem szükséges a lemez elején MBR-t létrehozni. Általában RAID környezetben a lemezeket közvetlenül használják anélkül, hogy különálló partíciótáblát (mint az MBR) hoznának létre – fdisk-nél nem kell megnyomni az “o” -t

Biztonsági mentés a rendszer és felhasználói adatokról

Ububtu Serveren az rsync egy hatékony és sokoldalú eszköz a fájlok és könyvtárak szinkronizálásához és biztonsági mentéséhez.

Feltelepítjük: **sudo apt update && sudo apt install rsync -y** a megoldási lehetőségek a következők:

1., Helyi biztonsági mentés: **sudo rsync -a /home/attila/ /backup/attila/**

2., Távoli biztonsági mentés rozsa.bimbo szerverre:
sudo rsync -a -e ssh /home/attila/ attila@rozsa.bimbo:/backup/attila/

3., Csak új fájlok mentése:
sudo rsync -a --delete /home/attila/ /backup/attila/

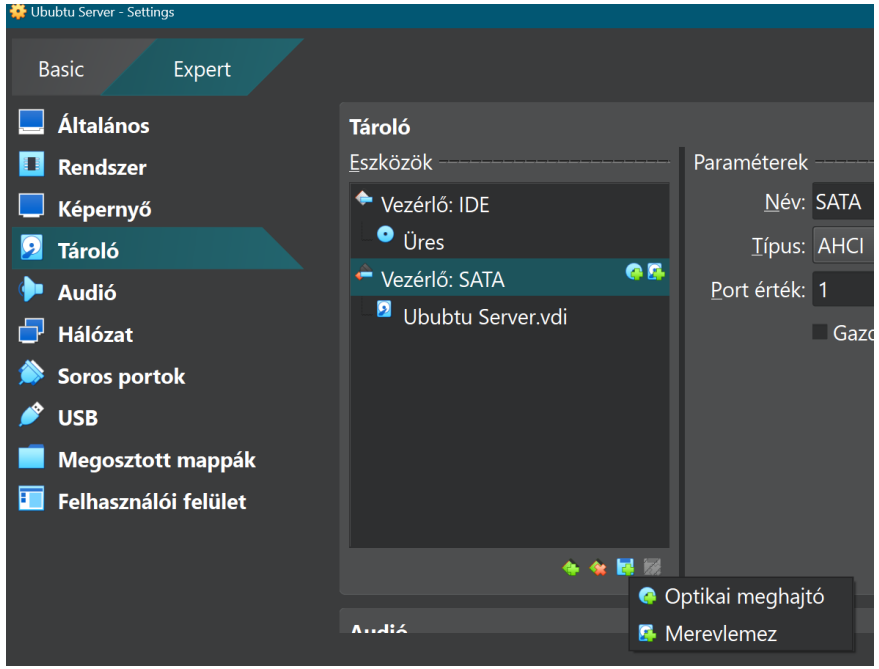
4., Rendszermentés:
sudo rsync -aAXv / /backup/system-backup/

5., Időzített rendszermentés január 3-án éjjel 2 órakor:
sudo crontab -e - az időzítő szerkesztése
0 2 3 1 * sudo rsync -aAXv / /backup/system-backup/

Biztonsági mentés visszaállítása

Adott a biztonsági mentésünk a /backup/system-backup/ helyen. Innen a visszaállítás a következőképp történik.

1., Virtualboxon egy új virtuális gépet hozunk létre, majd csatoljuk a régi rendszerünket is a Tárolónál.



2., Ubuntu Live környezetben csatoljuk a mentett virtuális gép lemezét, majd a root particionkat:
sudo mount /dev/sda1 /mnt

3., Visszaállítunk: **sudo rsync -a /backup/system-backup/ /mnt/**

4., Partíció lecsatolása : **sudo umount /mnt**

5., Lemez ki, majd újraindítunk : **sudo systemctl isolate reboot.target**

AD megoldások Linuxon

LDAP (Lightweight Directory Access Protocol) - egy protokoll, amely lehetővé teszi a hierarchikus adatstruktúrák kezelését, hasonlóan az Active Directoryhoz. OpenLDAP egy nyílt forráskódú megvalósítás, amelyet használhatunk Linux rendszereken. Az LDAP segítségével létrehozhatunk felhasználókat, csoportokat és hierarchiákat, valamint kezelhetjük a hozzáférési jogosultságokat.
Telepítése: **sudo apt install slapd ldap-utils**

Samba – egy nyílt forráskódú szoftver, amely lehetővé teszi a Windows és Linux rendszerek közötti fájlmegosztást és nyomtatásmegosztást. Lehetőséget nyújt az Active Directory megvalósítására, így felhasználókat és csoportokat kezelhetünk, és integrálhatjuk őket a Windows környezetekkel.
Telepítése: **sudo apt install samba**

LDAP – Címtárszolgáltás telepítése

1., Ubuntu Serveren telepítjük a csomagjait: **sudo apt update && sudo apt install slapd ldap-utils**

2., A DNS zónafájlba felvesszük a 10.0.2.93-at mint ldap bejegyzést, majd újraindítjuk a DNS-t:
sudo nano /etc/bind/db.rozsa.bimbo
ldap IN A 10.0.2.93
sudo systemctl restart bind9

3., Beállítjuk a host nevét, majd szerkesztjük a hosts file-t:
sudo hostnamectl set-hostname ldap.rozsa.bimbo
sudo nano /etc/hosts , ahol hozzáadjuk 10.0.2.93 ldap.rozsa.bimbo

4., Felvesszük a Netplan-ba is
sudo nano /etc/netplan/50-cloud-init.yaml az interfészünk címjeihez: - 10.0.2.93/24
sudo netplan apply - frissítjük

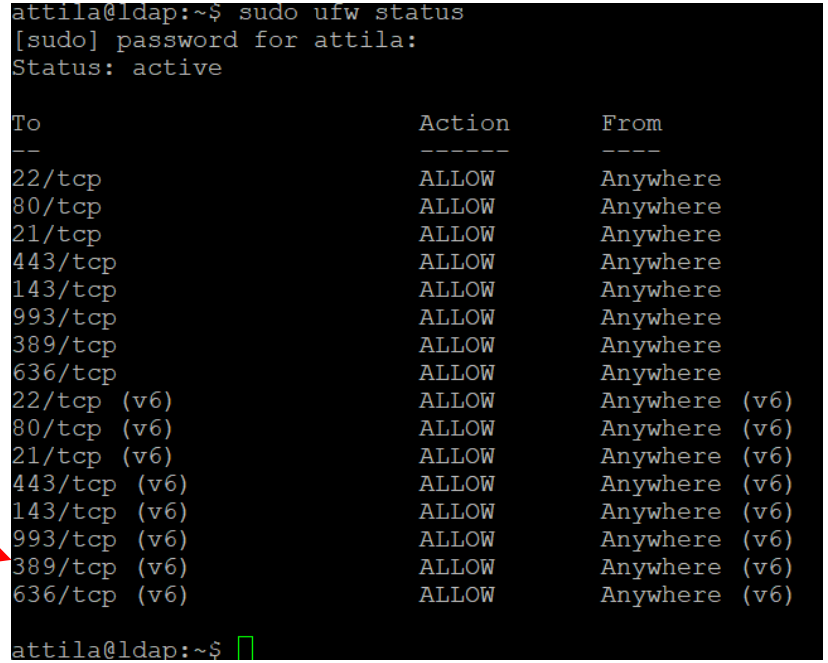
5., OpenLDAP (re)konfiguráció:
sudo systemctl stop slapd - LDAP leállítása
sudo rm -rf /var/lib/ldap/* - már meglévő/default konfigurációs fájlok törlése
sudo rm -rf /etc/ldap/slapd.d/*
sudo dpkg-reconfigure slapd - itt omit: no, szerver: ldap.rozsa.bimbo, szixi, tanárok, purge:no, rédi adatbázi áthelyezés:yes
sudo systemctl restart slapd - LDAP újraindítása

LDAP szolgáltatás beállítása

1., Felvesszük a zónánkat az LDAP szolgáltatásba az ldap.conf fájl segítségével
sudo nano /etc/ldap/ldap.conf
BASE dc=rozsa,dc=bimbo
URI ldap://ldap.rozsa.bimbo

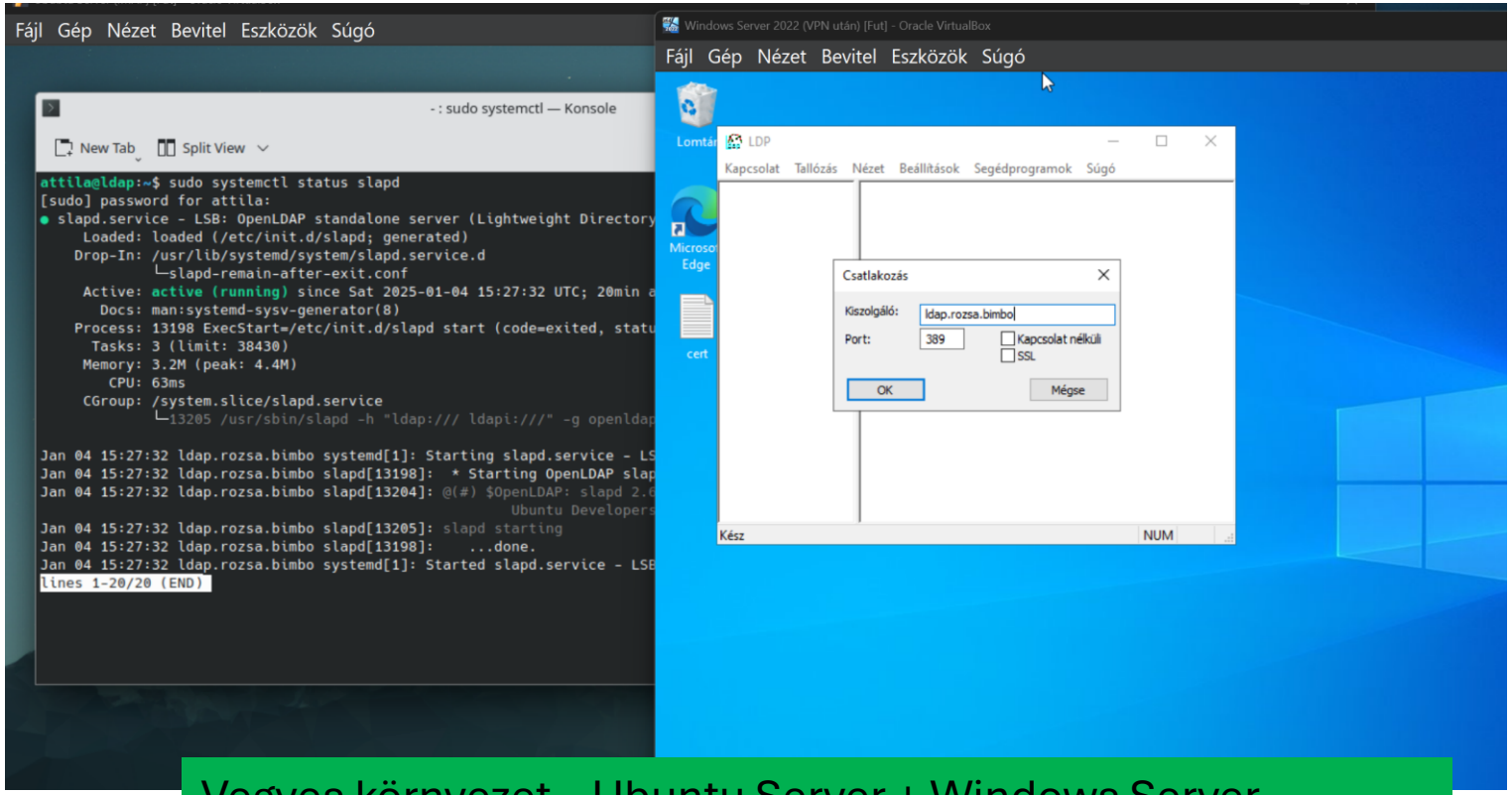
2., **sudo systemctl restart slapd** - LDAP újraindítása

3., Tűzfalkonfiguráció
sudo ufw allow 389/tcp – megadjuk a 389/TCP portot
sudo ufw enable - érvénybe léptetjük
sudo ufw status - lekérdezzük



LDAP – Csatlakozás Windows Serveren

Windows Serveren elinditjuk az LDP-t (ldp.exe), majd csatlakozunk az Ubuntu Serveren futó ldap.rozsa.bimbo-hoz. Megjegyzés : VirtualBox alatt nem fog működni alapból – a NAT végett. Az így létrehozott új felhasználók elérhetőek lesznek mind az Ubuntu, mind a Windows rendszereken, feltéve, hogy ezek a rendszerek ugyanazt az Active Directory-t használják hitelesítéshez és címtárkezeléshez.



Vegyes környezet – Ubuntu Server + Windows Server

Fájlszerver vegyes környezetben - Samba

1., A Samba-szerver telepítése a következő: **sudo apt update && sudo apt install samba -y**

2., Megosztási mappa felvétele: **sudo mkdir -p /home/attila/sambashare**

3., Samba-konfiguráció: **sudo nano /etc/samba/smb.conf**
[sambashare]
comment = Samba megosztás
path = /home/attila/sambashare
read only = no
browsable = yes

4., Samba kiszolgáló újraindítása **sudo systemctl restart smbd.service**

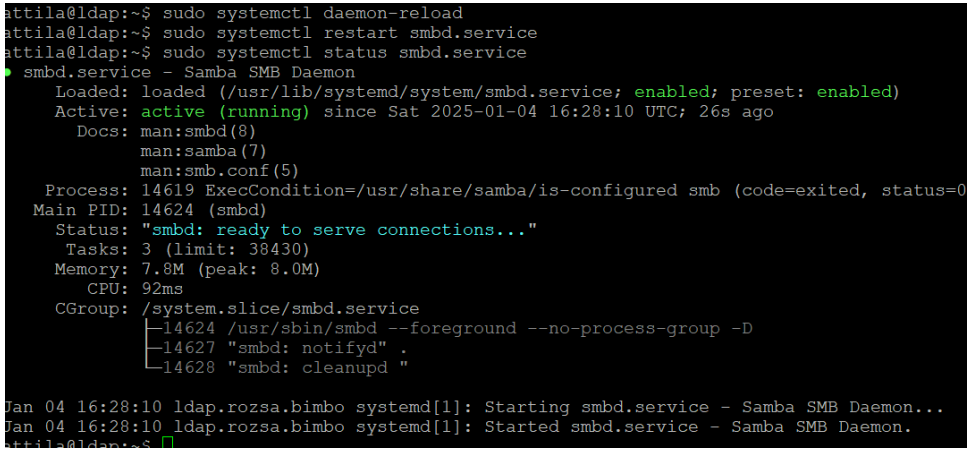
5., Felhasználói fiók létrehozása (mivel a Samba nem használja a rendszerfiók jelszavát, létre kell hozni egy Samba jelszót a felhasználói fiókhoz:) **sudo smbpasswd -a attila**. Ha hozzá akarunk még adni felhasználókat, akkor sudo adduser felhasználó, majd sudo smbpasswd -a felhasználó

6., Az egyszerűség kedvéért nem rendeltem hozzá külön IP-t és nevet, hanem a fő elérési címen hagytam, **smb://10.0.2.15/sambashare**

Samba, tűzfal és Windows Server

1., A Samba fájl szerkesztése és környezeti változók hozzáadása: **sudo nano /etc/default/samba**
SMBDOPTIONS="-D"
NMBDOPTIONS="-D"
WINBINDOPTIONS="--no-process-group"

2., A Samba szolgáltatás újratöltése:
sudo systemctl daemon-reload
sudo systemctl restart smbd.service



3., Tűzfalon átengedjük a Samba és a NetBIOS portokat (445/TCP, 139/TCP ,137/UDP, 138/UDP), majd érvénybe léptetjük őket
sudo ufw allow 445/tcp
sudo ufw allow 139/tcp
sudo ufw allow 137/udp
sudo ufw allow 138/udp
sudo ufw enable - érvénybe léptetjük
sudo ufw status - lekérdezzük

4., Az egyszerűség kedvéért nem rendeltem hozzá külön IP-t és nevet, Windows Serveren az elérése ezért: **smb://10.0.2.15/sambashare** lesz

Szerverbiztonság

nslookup -query=hinfo chess.com - egy adott site gazdagépéről közöl infot (DNS)

whois lichess.org – megnézi hol van regisztrálva

nmap lichess.org – port szkenner, feltérképezi a hálózatot, megnézi mi hol van és mit csinál a hálózaton, azonosítja a futó szolgáltatásokat és az operációs rendszert a célgépen.

arp -a – a helyi hálózat feltérképezésére szolgál, IP címek és MAC címek

sudo wireshark – hálózatelemző szoftver, amely a forgalmat elemzi és rögzíti, protokollismeret – több mint 100 protokollt ismer, hibaelemzés, hibaelhárítés, és biztonsági auditálás – biztonsági rések keresése

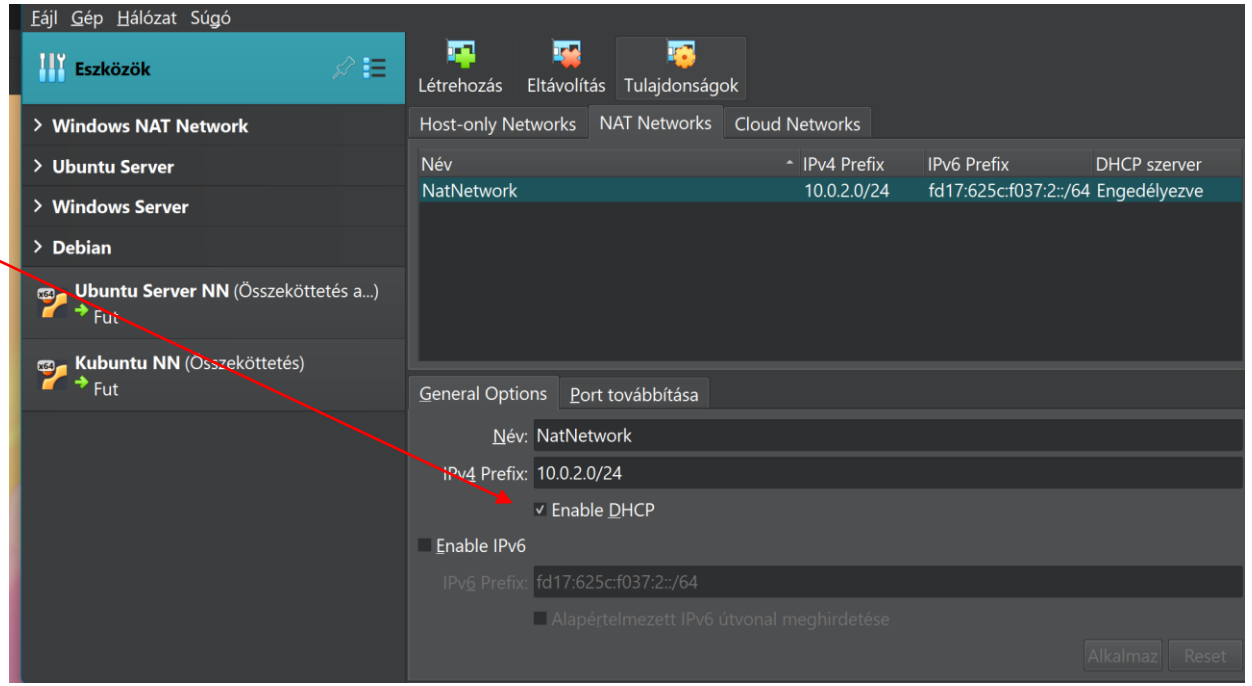
Etikus hekkelés:
1., megcélozzuk a kívánt oldalt: **nmap** -v -p 1-1024 kismajom.hu (vagy nmap -Pn kismajom.hu)

2., megnézzük melyik port van nyitva a szkennelés után. Ha olyan van nyitva ahol nincs SSL, tehát mondjuk a 21-es az FTP-szerver, azonnal támadható

3., Wiresharkkal nyilvános Wi-Fi hálózatról rögzítem a csomagját, mivel a csomag nem titkosított, tartalmazza a bejelentkezési kísérletet, és megvan a username, password.

Ubuntu Server + Kubuntu
NAT Network kapcsolata 1.

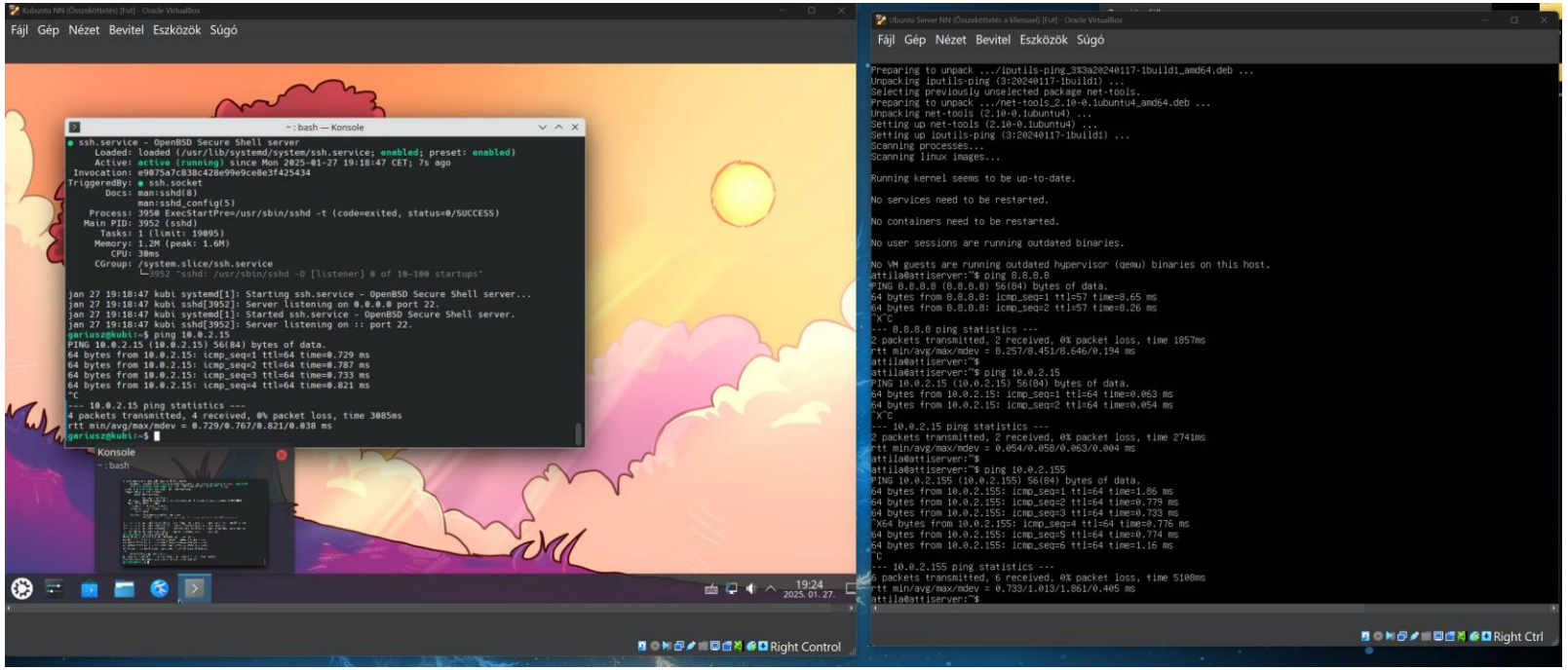
0., Előfeltételek. NAT Network módban nem tanácsos DHCP szerver-kliens alapú kiosztás, mivel a VirtualBox a saját elosztását erölteti és nem a DHCP szerver “range” tartományát. Alapból a sorban következő IP-címet osztja ki a kliensnek. Ezt a módot szintén ki lehet kapcsolni VirtualBoxban. Mivel most statikusan címeztem meg mindent bennehagytam.



Ubuntu Server + Kubuntu
NAT Network kapcsolata 2.

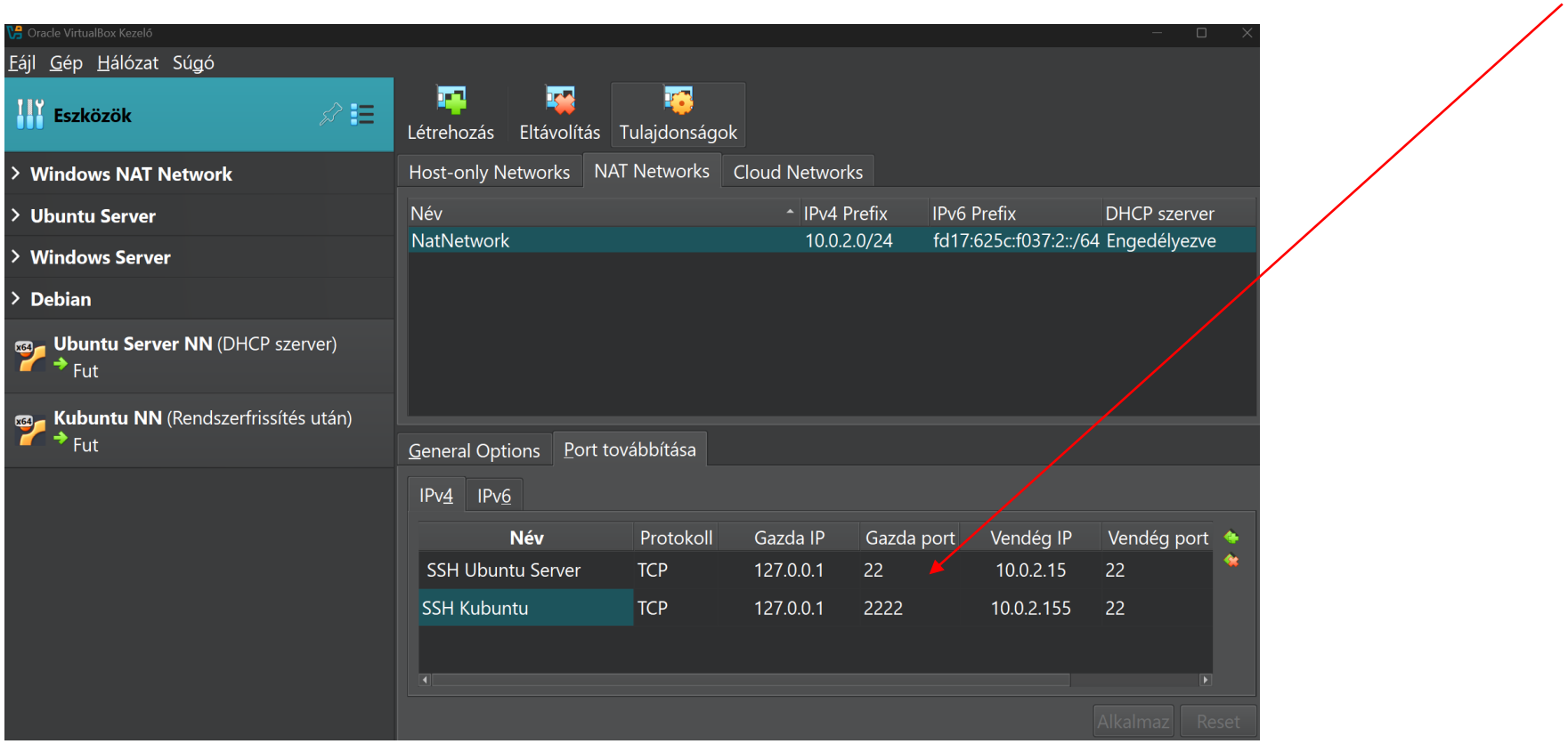
1., Adott a befuttatott konfigurált Ubuntu Server, SSH és DNS-kiszolgálóval, SSH klienssel, és statikus 10.0.2.15 IP-címmel, attila felhasználóval.

2., Adott a befuttatott konfigurált Kubuntu kliens, statikus 10.0.2.155 IP-címmel, gariusz felhasználóval. Szintén telepítésre került az SSH kliens és a kiszolgáló is.



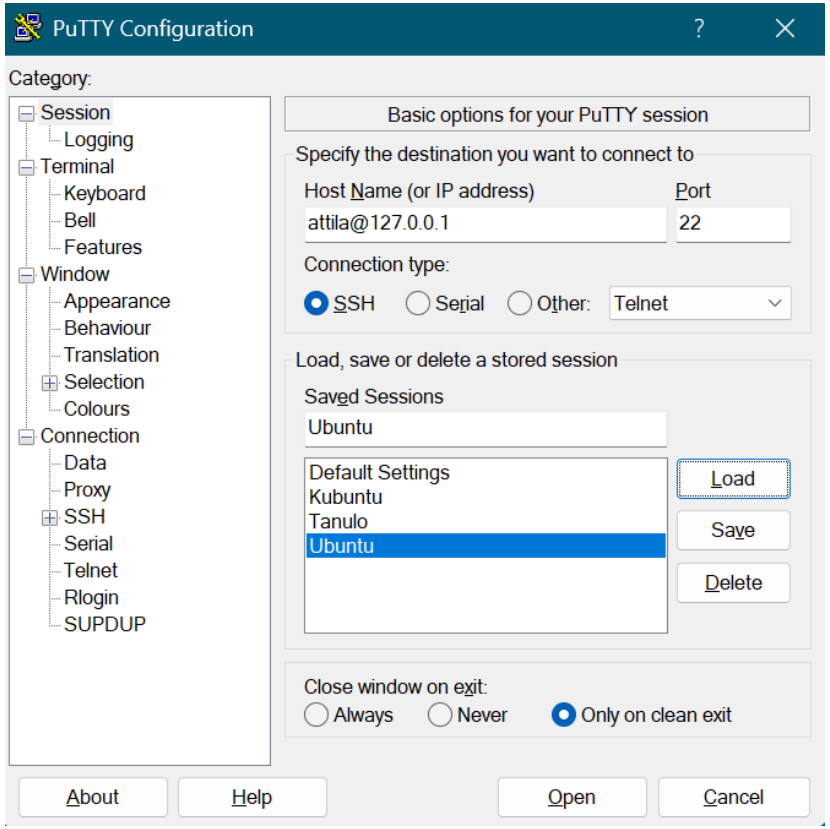
Ubuntu Server + Kubuntu
NAT Network kapcsolata 3.

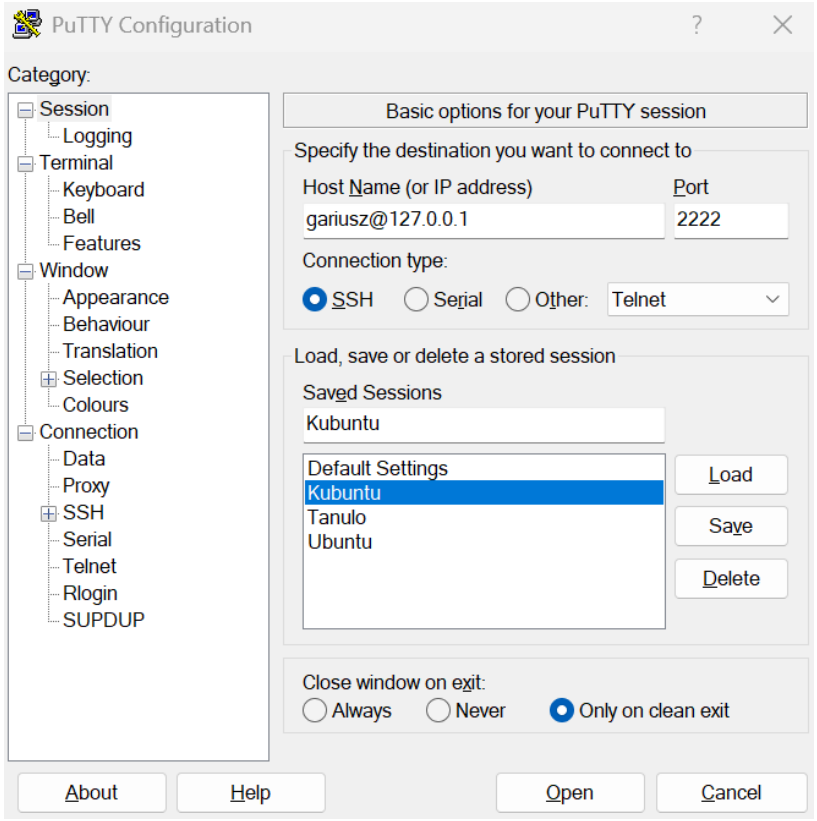
3., A VirtualBox-ban az Ubuntu Servert és a Kubuntut úgy Port Forwardoljuk, hogy a gazdaport különböző legyen, amíg az vendég port (Putty konzol pl.) ugyan az maradjon (22-es).



Ubuntu Server + Kubuntu
NAT Network kapcsolata 4.

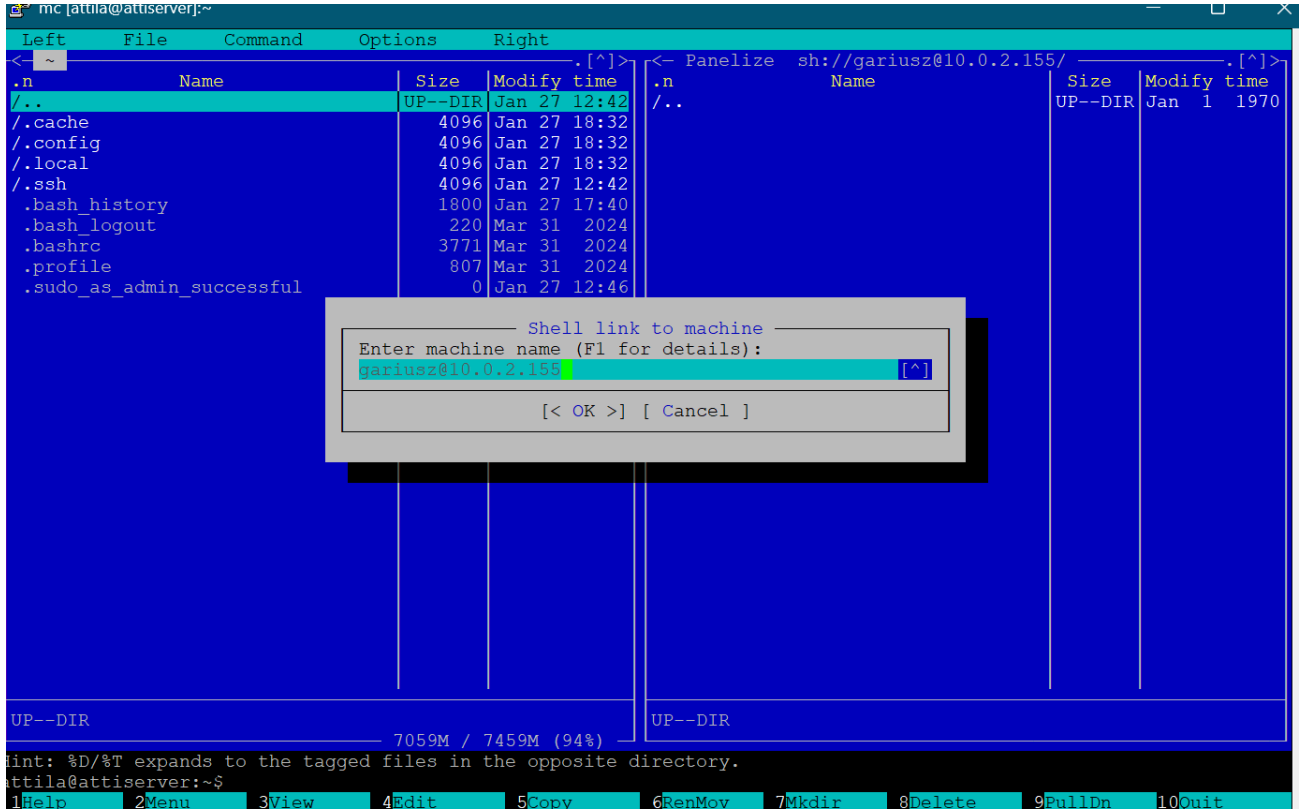
4., A külső secure shell konzolba (pl. Putty) megadjuk a helyes portokat és a belépéseket és tetszés szerint csatlakozhatunk:



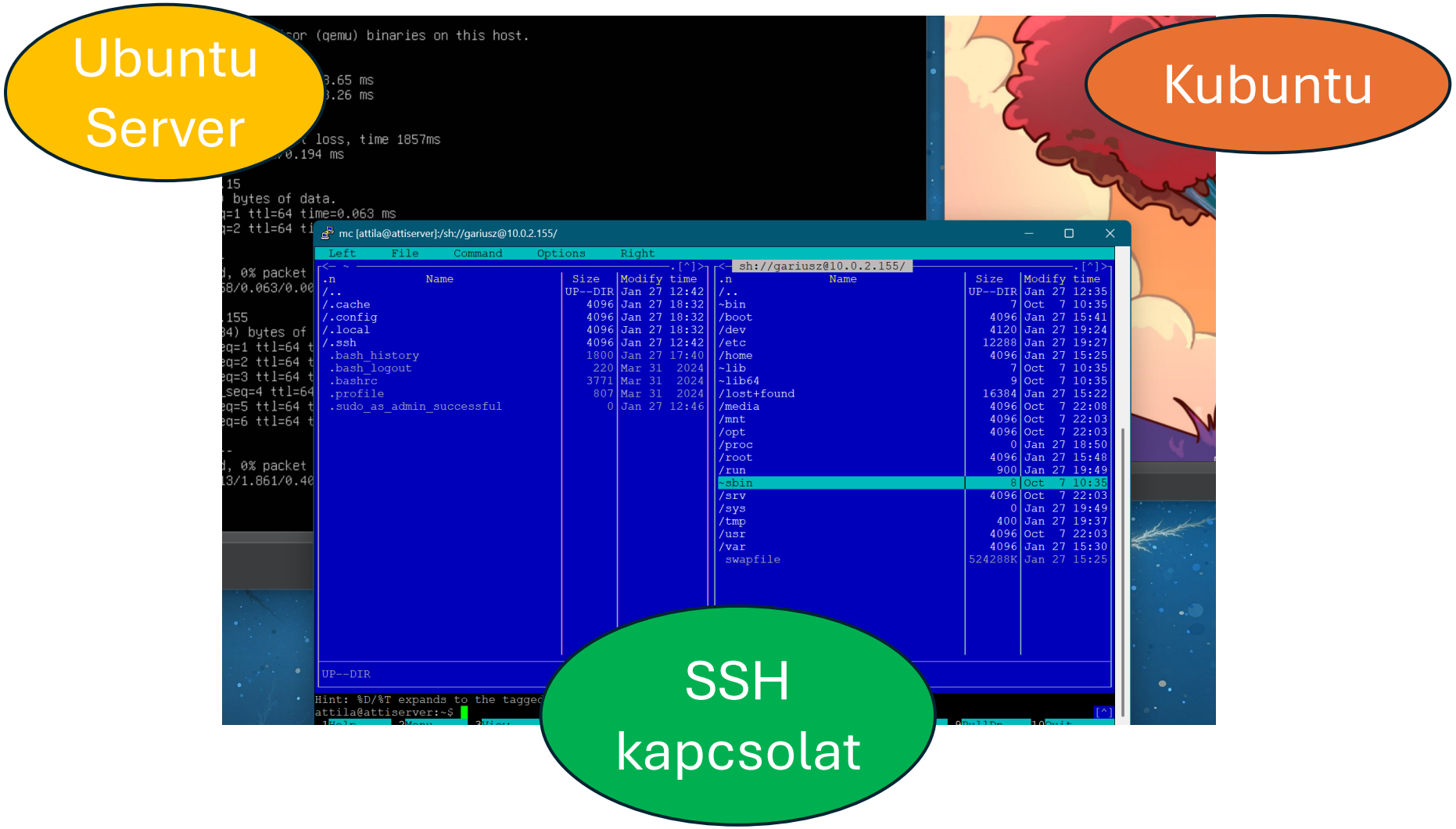


Ubuntu Server + Kubuntu
NAT Network kapcsolata 5.

5., Ubuntu Serveren Midnight Commanderben csatlakozunk a következőképp Gáriusz Púpusz Kubuntujához úgy, hogy nyomunk egy F9-et, kijön a jobb menü, majd Shell link...
(természetesen a kapcsolat visszafele is működik szerverről kliensre)



Ubuntu Server + Kubuntu NAT Network kapcsolata 6.



Ubuntu Server + Kubuntu
NAT Network: Webszerver hálózati elérés 1

1., Ubuntu Serveren telepítettük az Apache2 webszervert a prezentáció “Webszerver telepítés – Apache” része szerint, konfigurálva a zónánkat és a webszerver nevét www.rozsa.bimbo -ra amely a 10.0.2.77 IP-címen elérhető

2., A Kubuntu kliens gépen:

a., feltelepítjük a böngészőt: **sudo apt update && sudo apt install konqueror -y**

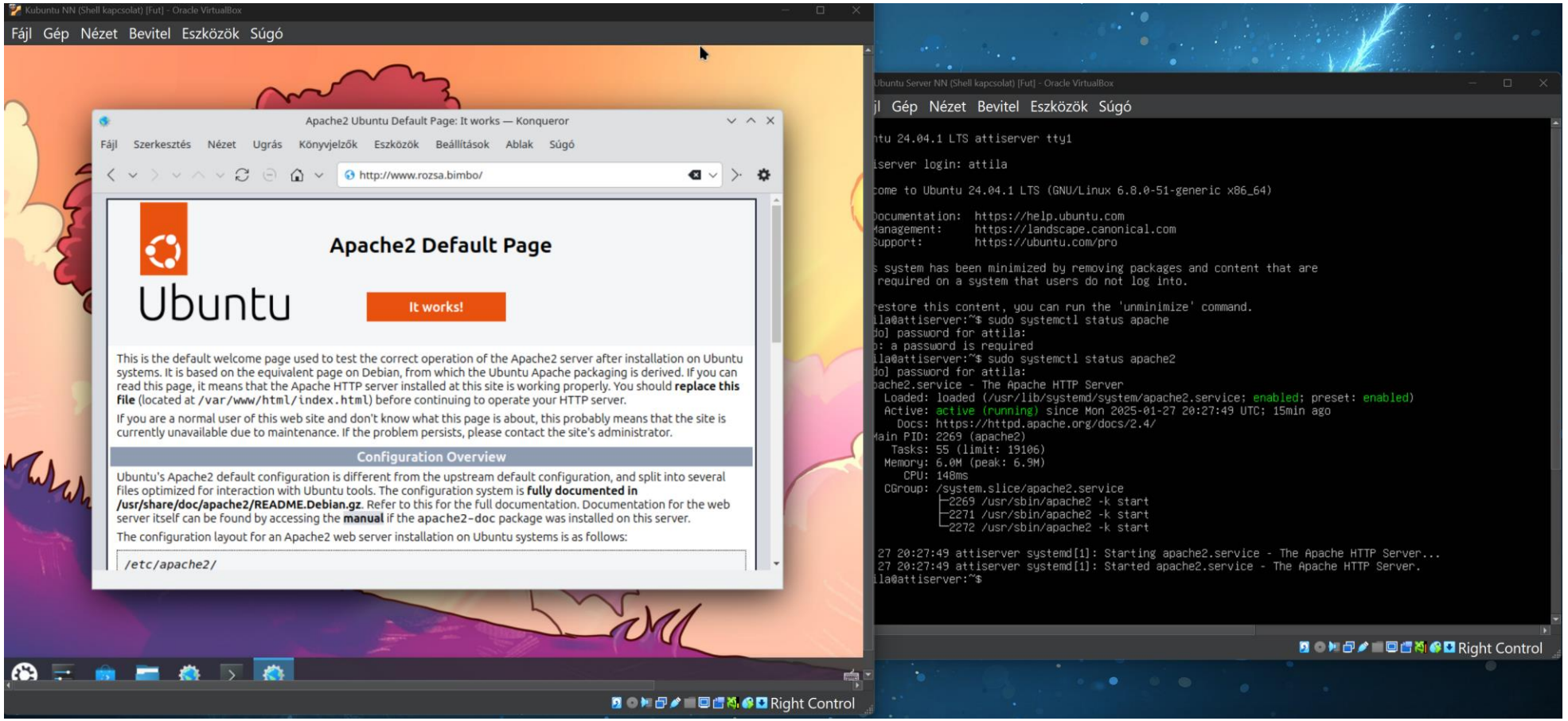
b., szerkesztjük a host filet: **sudo nano /etc/hosts**

10.0.2.77

www.rozsa.bimbo - felvesszük a névfeloldást

c., a böngészőbe beírjuk: http://10.0.2.77 , vagy http://www.rozsa.bimbo

Ubuntu Server + Kubuntu
NAT Network: Webszerver hálózati elérés 2

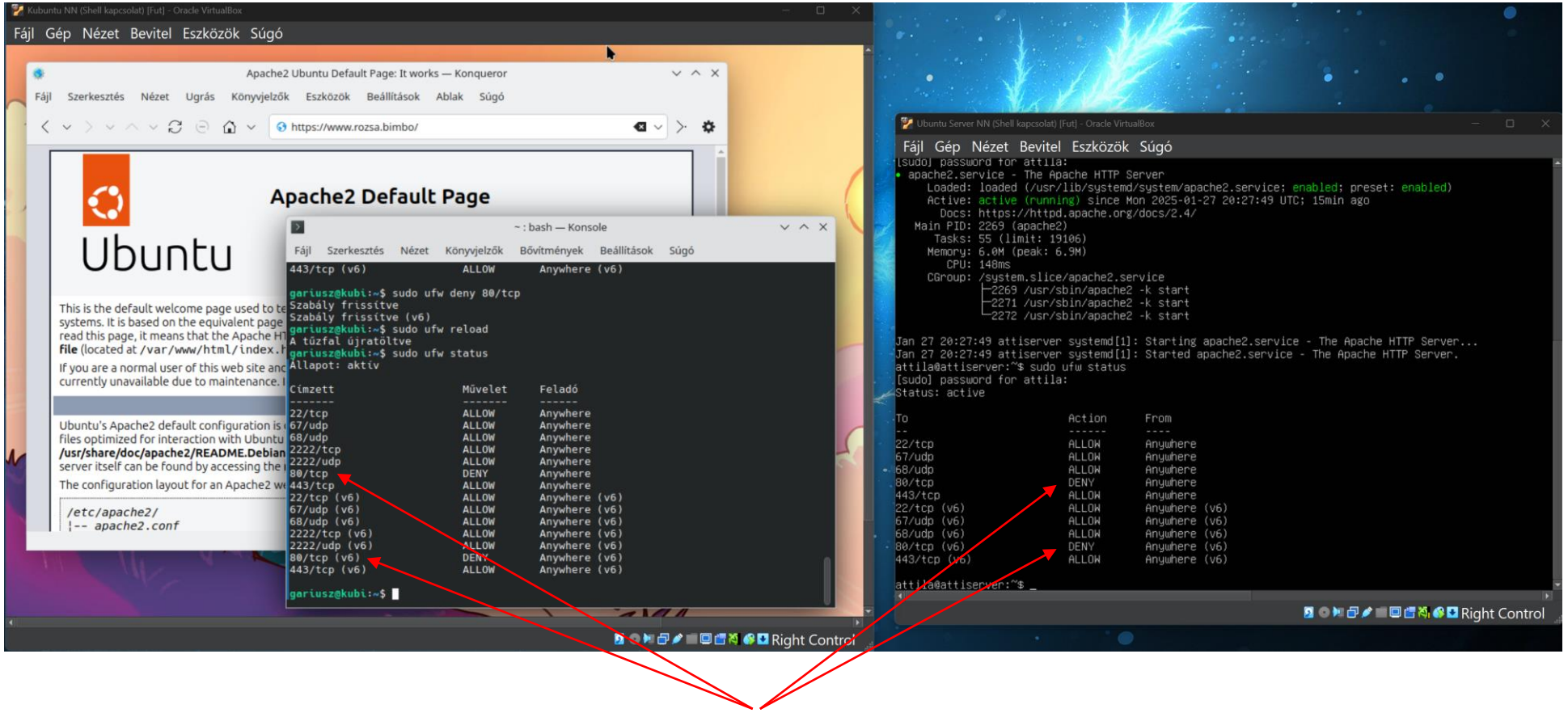


Ubuntu Server + Kubuntu
NAT Network: HTTPS hálózati elérés 1

1., Ubuntu Serveren végrehajtjuk az SSL/TLS-hez kapcsolódó lépéseket a tananyag “HTTPS és FTPS Ubuntu Serveren” c. fejezete alapján.
a., OpenSSL csomag telepítése
b., SSL/TLS tanusítvány létrehozása (CA-általi, önaláírt)
c., SSL/TLS apache modul engedélyezése (**sudo a2enmod ssl**), majd újraindítjuk az apache-t
d., Virtuális HOST konfiguráció – **sudo nano /etc/apache2/sites-available/default-ssl.conf**
e., Érvénybe léptetjük a 443-as porthoz rendelt hostot: **sudo a2ensite default-ssl.conf**
f., Újraindítjuk az apache-t
g., Tűzfalon engedélyezzük a 443-as portot, és letiltjuk a 80-as portot (**sudo allow ufw 443/tcp** és letiltás **sudo ufw deny 80/tcp**)

2., A Kubuntu kliens gépen:
a., Tűzfalon engedélyezzük a 443-as portot, és letiltjuk a 80-as portot (**sudo allow ufw 443tcp** és letiltás **sudo ufw deny 80/tcp**)
b., a böngészőbe beírjuk: https://10.0.2.77 , vagy https://www.rozsa.bimbo
c., elfogadjuk az Ubuntu szerver által kigenerált SSL/TLS tanusítványt

Ubuntu Server + Kubuntu
NAT Network: HTTPS hálózati elérés 2



Hálózatbiztonsági szempontból tanácsos a HTTP 80-as (és az FTP 21-es) portokat tiltani, mert támadhatóak - Brute Force, DDoS, ManInTheMiddle, SQL Injection, Phishing stb.

Egyéb programok és segédprogramok

1., Oneko – egér megváltoztatása (kutya, macska, sakura) - **sudo apt install oneko**

2., Kate – KDE szövegszerkesztő - **sudo apt install kate**

3., Codium – egy integrált fejlesztői környezet (IDE), amely szolgáltatja a VS Code funkcióit, mivel annak a forkja, de nélkülözi a Microsoft által beépített nyomon követési (telemetry) funkciókat
A snap csomagtelepítővel lehet telepíteni (tegyük a repoban rendbe ha nem megy, ellenőrizzük stb)
sudo snap install codium --classic

4., A libreoffice KDE-s kiadása **sudo apt install libreoffice-kde**

5., KDEnLive - ingyenes nagytudású videovágó: **sudo apt install kdenlive**

6., Krita - Ingyenes nagytudású grafikus/festőprogram: **sudo apt install krita**

AI és Ubuntu Server

1., MS Copilot kliens Ubuntu alá: **sudo snap install copilot-client**

2., DeepSeek szerver telepítése Ubuntu Server alatt:

sudo apt install curl	- a curl adatátvivő eszköz telepítése
sudo curl -fsSL https://ollama.com/install.sh | sh	- az ollama (AI keretrendszer) telepítése
sudo ollama --version	- az ollama verzióellenőrzése
sudo systemctl is-active ollama.service	- az ollama futásának ellenőrzése
sudo ollama pull deepseek-r1:7b	- a deepseek nyelvi modelljének letöltése
ollama run deepseek-r1:7b	- a deepseek futtatása

az Open WebUI – kommunikációs kliens feltelepítése és konfigurációja kliens (ill. szerver) oldalon

Model	Parameters	Disk Space Required	Minimum RAM	Recommended GPU VRAM	Performance
`deepseek-r1:1.5b`	1.5 Billion	3 GB	8 GB	4 GB	Fastest, low memory usage, less accuracy
`deepseek-r1:7b`	7 Billion	15 GB	16 GB	8 GB	Good balance of speed and accuracy
`deepseek-r1:8b`	8 Billion	17 GB	24 GB	10 GB	Similar to 7B, slightly improved reasoning
`deepseek-r1:14b`	14 Billion	30 GB	32 GB	16 GB	Better understanding, needs more RAM
`deepseek-r1:32b`	32 Billion	70 GB	64 GB	24 GB	High accuracy, slower response times
`deepseek-r1:70b`	70 Billion	160 GB	128 GB	48 GB	Very accurate, slow inference speed
`deepseek-r1:671b`	671 Billion	1.5 TB	512 GB+	Multiple GPUs, 100 GB+ VRAM	Cutting-edge accuracy, extremely slow