

Operacijski sustavi (OS)

Nositelj: doc. dr. sc. Ivan Lorencin

Asistent: Luka Blašković, mag. inf.

Ustanova: Sveučilište Jurja Dobrile u Puli, Fakultet informatike u Puli



Fakultet informatike u Puli

(4) Rad na Virtualnom stroju: Uvod

#4

OS

Virtualni stroj (eng. *Virtual Machine*) je programski proizvod koji emulira ponašanje određenog računalnog sustava, a pritom koristi arhitekturu i funkcije stvarnog računala. Virtualni strojevi često pružaju cijelu sistemsku platformu koja omogućava izvođenje kompletног operacijskog sustava (OS). Računalo koje pokreće virtualni stroj naziva se domaćin (eng. *host*), dok se virtualni stroj naziva gost (eng. *guest*). Studenti će u ovoj skripti naučiti kako instalirati i koristiti virtualni stroj, pritom koristeći vlastito računalo kao domaćina.

Posljednje ažurirano: 19.5.2025.

Sadržaj

- [Operacijski sustavi \(OS\)](#)
- [\(4\) Rad na Virtualnom stroju: Uvod](#)
 - [Sadržaj](#)
- [1. Virtualizacija](#)
- [2. Virtualni stroj](#)
 - [2.1 Instalacija VirtualBoxa](#)
 - [2.2 Ubuntu Server](#)
 - [2.3 Izrada virtualnog stroja](#)
 - [2.4 Instalacija Ubuntu Servera](#)
- [3. Ubuntu Server](#)
 - [3.1 APT \(Advanced Package Tool\)](#)
 - [3.2 SSH \(Secure Shell\)](#)
 - [3.2.1 Instalacija OpenSSH poslužitelja](#)

- [Alat systemctl](#)
- [Alat lsof](#)
- [3.2.2. Povezivanje na SSH poslužitelj](#)
 - [Alat ip](#)
 - [Alat ss](#)
- [4. Zadaci za Vježbu 4](#)

1. Virtualizacija

Virtualizacija je tehnologija koja omogućava stvaranje virtualnih reprezentacija poslužitelja, pohrane, mreže i drugih fizičkih resursa. Alati za virtualizaciju oponašaju funkcije fizičkog hardvera i omogućuju pokretanje više virtualnih strojeva na jednom fizičkom računalu.

Korištenjem virtualizacije možemo fleksibilnije upravljati bilo kojim hardverskim resursom. Fizički poslužitelji su stvarni, opipljivi uređaji koji moraju biti smješteni negdje u prostoru - poput podatkovnog centra. Oni stalno troše električnu energiju i zahtijevaju održavanje (fizičke provjere, nadogradnje dijelova, popravci u slučaju kvara i sl.). Osim toga, zauzimaju prostor, što može biti skupo, posebno u velikim organizacijama.

Pristup fizičkom poslužitelju često nije jednostavan, što zbog fizičke udaljenosti, što zbog sigurnosnih protokola i ograničenom pristupu putem mrežne infrastrukture (sigurnosne postavke, vatrozidi i sl.). Virtualizacija uklanja navedena ograničena **apstrahiranjem funkcionalnosti fizičkog hardvera u softver**.

Primjer virtualizacije: Zamislite tvrtku koja treba poslužitelje za tri funkcije:

1. sigurno pohranjivanje poslovne e-pošte
2. pokretanje aplikacije za krajnje korisnike
3. pokretanje internih poslovnih aplikacija

Recimo da svaka od tih funkcija ima različite zahtjeve za konfiguracijom:

- Aplikacija za e-poštu zahtijeva veću pohranu i Windows operacijski sustav
- Aplikacija za krajnje korisnike zahtijeva Linux operacijski sustav i veliku procesorsku snagu kako bi podnijela veliki broj korisnika na web stranici
- interna poslovna aplikacija zahtijeva iOS i više radne memorije (RAM)

U fizičkom svijetu, tvrtka bi trebala kupiti tri fizička poslužitelja, svaki s različitim specifikacijama. Navedeno zahtjeva **visoke inicijalne troškove**, a održavanje i nadogradnje se provode za svaki stroj pojedinačno. Također, tvrtka ne možemo optimalno koristiti resurse, budući da samim posjedovanjem fizičkih poslužitelja troši električnu energiju i prostor, plaća 100% troškova održavanja, ako ustvari im je potreban samo dio njihove pohrane i procesorske snage.

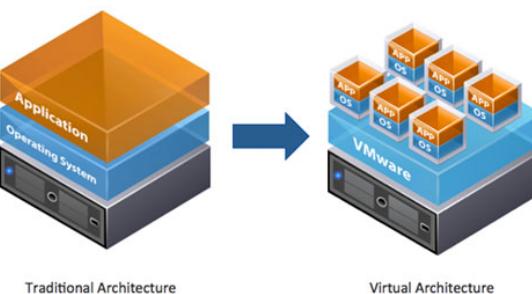
Prednosti virtualizacije:

- **Učinkovitije korištenje resursa:** Koristeći virtualizaciju, tvrtka može pokrenuti tri virtualna poslužitelja (ili virtualne strojeve) na jednom fizičkom poslužitelju. Svaki virtualni stroj može imati svoj operacijski sustav i aplikacije, a svi dijele resurse fizičkog poslužitelja. Na taj način, naša zamišljena tvrtka sada mora brinuti samo o jednom fizičkom poslužitelju, a ne o tri.

- **Smanjenje troškova:** Virtualizacija smanjuje troškove hardvera, energije i prostora. Tvrta može smanjiti broj fizičkih poslužitelja koje posjeduje, čime se smanjuju troškovi održavanja i električne energije.
- **Fleksibilnost:** Virtualni strojevi se mogu lako premještati između fizičkih poslužitelja, što omogućava lakše upravljanje resursima i optimizaciju performansi. Ako jedan fizički poslužitelj postane preopterećen, virtualni stroj se može premjestiti na drugi fizički poslužitelj s više slobodnih resursa.
- **Disaster recovery:** Virtualizacija omogućava brže oporavak od potencijalnih katastrofa (npr. prirodne nepogode, kibernetički napadi i sl.). Ako fizički poslužitelj otkaže, virtualni stroj se može lako premjestiti na drugi fizički poslužitelj bez gubitka podataka ili funkcionalnosti.

Virtualization Defined

For those more visually inclined...



Ilustracija tradicionalne računalne arhitekture vs. Virtualna arhitektura

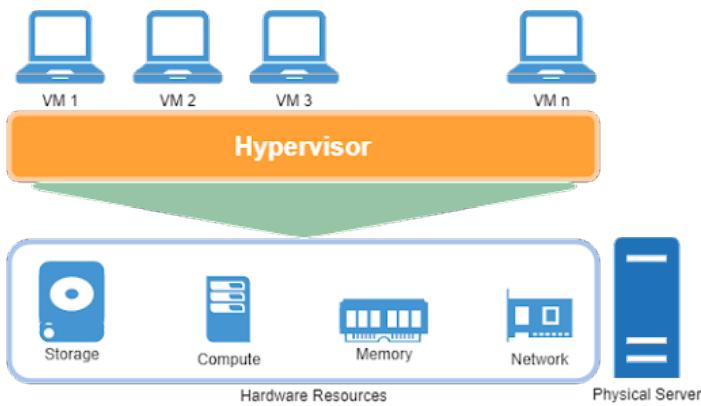
2. Virtualni stroj

Virtualni stroj (*eng. Virtual machine* ili VM) je softverska simulacija fizičkog računala koja omogućava pokretanje operacijskih sustava i aplikacija unutar virtualnog okruženja. Virtualno okruženje koristi resurse fizičkog računala kojeg često nazivamo **domaćin** (*eng. host*), dok se virtualni stroj naziva **gost** (*eng. guest*).

Više virtualnih strojeva može raditi na istom fizičkom računalu, svaki s vlastitim operacijskim sustavom i aplikacijama - kao što smo već prikazali u ranijem primjeru. Softver koji omogućuje virtualizaciju i upravljanje virtualnim strojevima naziva se **hipervizor** (*eng. hypervisor*).

U grubo, hipervizor se može podijeliti u dvije kategorije:

- **Tip 1 ("bare metal"):** Hipervizor radi izravno na fizičkom hardveru i upravlja virtualnim strojevima bez potrebe za operacijskim sustavom domaćina. Primjeri uključuju *VMware ESXi*, *Microsoft Hyper-V* i *Xen*.
- **Tip 2 ("hosted"):** Hipervizor radi unutar operacijskog sustava domaćina i koristi resurse fizičkog računala putem operacijskog sustava. Primjeri uključuju *VMware Workstation*, *Oracle VirtualBox* i *Parallels Desktop*.



 Ilustracija predstavlja hijerarhiju virtualne arhitekture: fizičko računalo → hipervizor → VM1, VM2, VM3 ...

Naša zamišljena tvrtka iz ranijeg primjera može ići još jedan korak naprijed i u potpunosti virtualizirati svoje poslovanje. Naime, umjesto da kupuje fizičko računalo, tvrtka može unajmiti virtualno računalo od nekog pružatelja cloud usluga (eng. *cloud service provider*) i tamo podignuti sve svoje virtualne strojeve. Na taj način, tvrtka uopće ne mora brinuti o fizičkom hardveru, troškovima održavanja, slučajnim kvarovima i sl. Također, na ovaj način tvrtka može lako povećati ili smanjiti svoje resurse prema potrebama, čime se troškovi poslovanja prilagođavaju trenutnim potrebama. Ovakav oblik usluge nazive se **IaaS** (eng. *Infrastructure as a Service*).

Ipak, na ovim vježbama se nećemo baviti cloud uslugama, već ćemo se fokusirati na virtualizaciju na fizičkom - našem računalu. Kako bismo to postigli, koristit ćemo **Oracle VirtualBox** - besplatni hipervizor koji radi na svim popularnim operacijskim sustavima (Windows, Linux, macOS). U nastavku ćemo proći kroz osnovne korake instalacije i korištenja VirtualBoxa.

2.1 Instalacija VirtualBoxa

Oracle VirtualBox je besplatni hipervizor koji omogućava virtualizaciju na svim popularnim operacijskim sustavima. Razvijen je 2007. godine, a izdaje se pod GNU General Public License (GPL) licencom - kao i bash shell.



 Oracle VirtualBox logo - <https://www.virtualbox.org/>

VirtualBox spada u kategoriju hipervizora tip 2 (hosted), što znači da radi unutar operacijskog sustava domaćina i može se koristiti paralelno uz ostale aplikacije na domaćinu.

VirtualBox možete preuzeti preko sljedeće poveznice: <https://www.virtualbox.org/wiki/Downloads>

- Preuzmite verziju za svoj operacijski sustav (Windows, Linux, macOS)
- Ako koristite Windows, instalaciju pokrenite kao administrator, potencijalno će vas tražiti instalaciju dodatnih zavisnih komponenti (Python Core / win32api), instalirajte ih

Svi moderni procesori podržavaju virtualizaciju putem hipervizora tipa 2, no operacijski sustav domaćina će ponekad ovu opciju imati onemogućenu. Kako biste mogli učinkovito koristiti virtualizaciju, morat ćete omogućiti virtualizaciju u **BIOS/UEFI** postavkama računala.

Ovisno o operacijskom sustavu, provjeru možete napraviti na sljedeći način:

- **Windows:** Otvorite "Task Manager" (`Ctrl + Shift + Esc`) → `Performance` → `CPU` → provjerite je li opcija "Virtualization" uključena

Utilisation	Speed	Base speed:	3.70 GHz
9%	4.46 GHz	Sockets:	1
Processes	Threads	Cores:	6
281	3858	Logical processors:	12
Up time		Virtualisation:	Enabled
		L1 cache:	384 KB
		L2 cache:	3.0 MB
		L3 cache:	32.0 MB
	0:00:38:42		



 Provjera je li virtualizacija uključena kroz Task Manager na Windows OS-u

Ako virtualizacija nije omogućena, slijedite sljedeće korake:

1. Ponovno pokrenite računalo i uđite u BIOS/UEFI postavke (obično pritiskom na tipku **F2**, **Del** ili **Esc** prilikom pokretanja računala)
 2. Potražite opciju **Virtualization Technology**, **Intel VT-x** ili **SVM**, **AMD-V** kod AMD procesora - često se nalazi u izbornicima **Advanced**, **CPU Configuration** ili **Security**
 3. Spremite promjene i izadžite iz BIOS/UEFI postavki
 4. Ponovno pokrenite računalo

Za Intel računala, tipične označke su **VT-x**, **VT-d**, **Intel Virtualization Technology**, dok su za AMD računala to **AMD-v**, **SVM Mode** ili **Secure Virtual Machine**. Ako ne možete pronaći opciju, provjerite dokumentaciju svojeg prijenosnog računala ili matične ploče.



💡 Hint: Googlaj: How to enable virtualization in BIOS on <yaša matična> motherboard.

- **macOS:** Ovisno koristite li Apple Silicon ili Intel procesor, postavke će se razlikovati. Međutim, Apple Silicon i Intel procesori će u većini slučajeva imati virtualizaciju omogućenu prema zadanim postavkama.

Kod Apple Silicon procesora, provjeru možete izvršiti pomoću sljedećih naredbi:

```
→ sysctl -a | grep hv
```

Ako je postavka `machdep.hv support: 1` onda je virtualizacija omogućena.

Kod Intel procesora na macOS-u, provjeru možete izvršiti pomoću slijedećih naredbi:

```
→ sysctl -a | grep machdep.cpu.features
```

Ako je postavka `vmx` uključena, onda je virtualizacija omogućena.

- **Linux:** Provjeru možete izvršiti pomoću sljedeće naredbe:

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

Ako je rezultat 0, virtualizacija nije omogućena.

Ako je rezultat 1 ili više, virtualizacija je omogućena.

Prisjetimo se, vmx označava Intel procesore, dok svm označava AMD procesore.

Možete provjeriti i pomoću kvm-ok alata:

```
→ sudo apt install cpu-checker  
→ kvm-ok
```

Ako je virtualizacija isključena, možete ju omogućiti u BIOS/UEFI postavkama računala, kao što je opisano ranije.

2.2 Ubuntu Server

Jednom kada ste instalirali VirtualBox i omogućili virtualizaciju, sljedeći korak je preuzimanje ISO datoteke operacijskog sustava koji želite instalirati na virtualni stroj.

Slika diska (eng. *disk image*) ili **ISO** je posebna vrsta datoteke koja sadrži sve instalacijske podatke koji bi se nalazili na stvarnom fizičkom disku (kao što je CD, DVD...) ili USB uređaju - radi se o **digitalnoj kopiji fizičkog medija**.

ISO datoteke se najčešće koriste za distribuciju operacijskih sustava, premda se mogu koristiti i za druge oblike softvera.

Ako koristite Windows kao operacijski sustav domaćina, tehnički je moguće instalirati i Windows kao gost - i to bilo koju verziju. Primjerice, moguće je instalirati Windows 10 kao gosta unutar Windows 11 domaćina. Sve što trebate je preuzeti [odgovarajuću ISO datoteku s Microsoftove web stranice](#).

Iako možemo ovo izvesti i na taj način se upoznati s virtualizacijom, mi ćemo za potrebe ovih vježbi koristiti **Ubuntu Server** za našeg gosta.

Ubuntu Server je jedna od najpopularnijih Linux poslužiteljskih distribucija koja se koristi u raznim okruženjima, uključujući web hosting, cloud računarstvo (AWS, Azure, gcloud), IoT uređaji i dr.

Ubuntu Server je besplatan i predstavlja idealno rješenje za učenje koncepata poput CLI-a, mrežnog upravljanja (networking), upravljanja korisničkim ovlastima, instalacija paketa i sl.



 Ubuntu Server logo - <https://ubuntu.com/download/server>

Ubuntu Server nema grafičko korisničko sučelje (GUI) kao što je to slučaj s [Desktop verzijom](#), već se koristi isključivo putem CLI-a. Kao ljudsku, Ubuntu Server koristi **bash** koji smo do sada radili na vježbama.

Ubuntu Server može imati puno različitih namjena, a najčešće se koristi za:

- **Web poslužitelje** (Apache, Nginx): npr. za hosting web stranica
- **Izvršavanje aplikacija kroz runtime okruženja** (Java, Python, Node.js)
- **Izvršavanje aplikacija u kontejnerima** (Docker, Kubernetes)
- **Upravljanje bazama podataka** (MySQL, PostgreSQL, MongoDB)
- **Uloga datotečnog ili e-mail poslužitelja** (Samba, Postfix)

ISO sliku za Ubuntu Server možete preuzeti sa sljedeće poveznice: <https://ubuntu.com/download/server>

- Odaberite verziju koju želite preuzeti (preporuka LTS - *Long Term Support*)

VAŽNO:

Ako koristite Windows (Intel, AMD), Linux (Intel, AMD) ili macOS (Intel), preuzmite [verziju za x86-64 arhitekturu](#) - `amd64`.

Ako koristite Apple Silicon (M serija) procesor, preuzmite [verziju za ARM64 arhitekturu](#) - `arm64`.

2.3 Izrada virtualnog stroja

Jednom kad ste preuzeli ISO datoteku, otvorite VirtualBox i slijedite sljedeće korake:

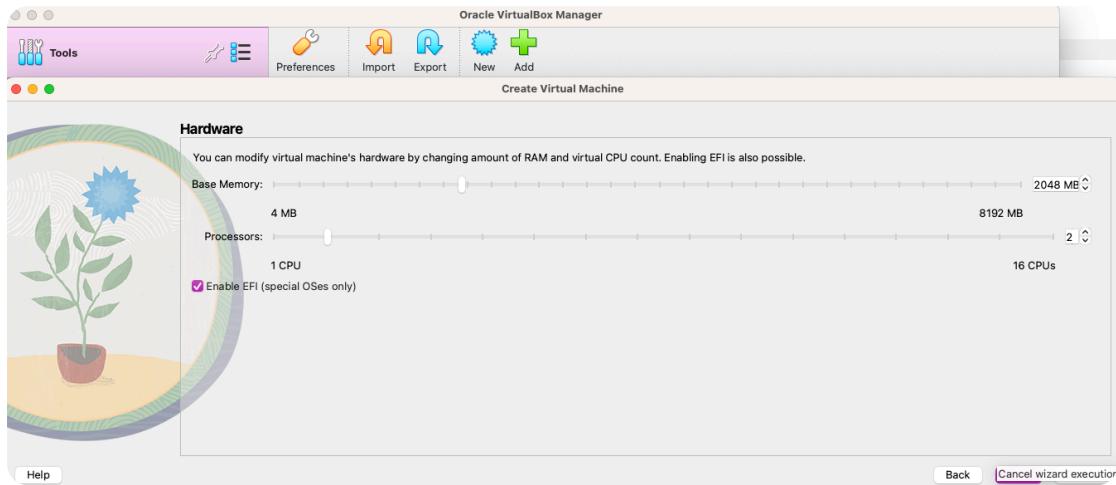
1. Kliknite na `New` gumb putem sučelja i unesite naziv virtualnog stroja (npr. "Ubuntu Server")
2. Odaberite ISO datoteku koju ste preuzeli
3. Type postavite na "Linux", a Subtype na Ubuntu



VirtualBox: Izrada novog virtualnog stroja

U sljedećem prozoru će vas tražiti da odredite količinu radne memorije i procesorskih jezgri koje želite dodijeliti virtualnom stroju.

Za naše potrebe, možemo dodijeliti 2 GB RAM-a i 2 procesorske jezgre. Opciju "Enable EFI" možete ostaviti uključenu, no nije nužno.



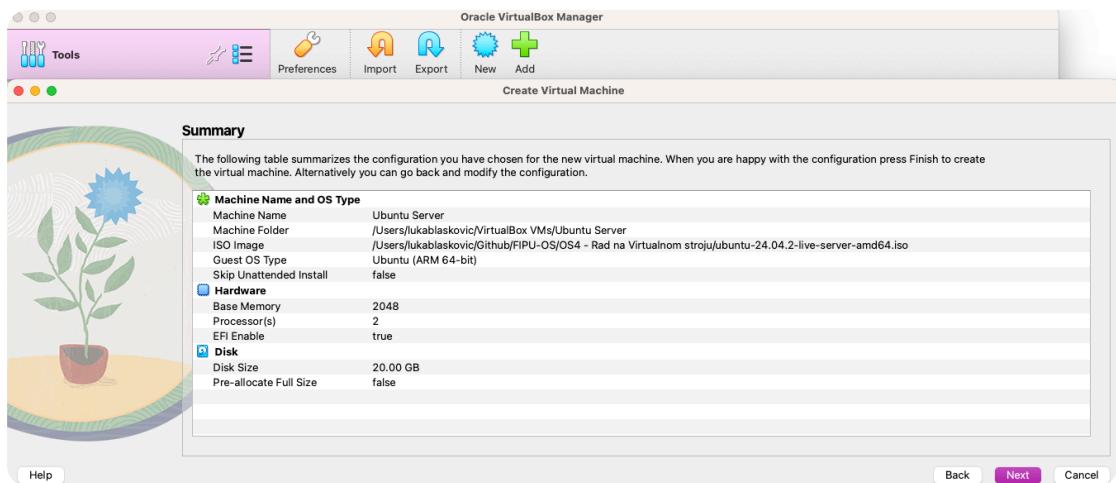
VirtualBox: Odabir radne memorije i CPU jezgri za novi virtualni stroj

Potrebno je još odrediti veličinu virtualnog diska. Prisjetite da virtualizacijom mi ustvari "rentamo" resurse našeg računala (domaćina), pa tako i prostor na disku. U ovom slučaju, odabrat ćemo opciju `"Create a Virtual Hard Disk Now"` i dodijeliti `20 GB` prostora.

Hipervizor će stvoriti virtualni disk koji će se ponašati gotovo identično kao fizički disk. Na njemu ćemo instalirati operacijski sustav Ubuntu Server i druge potrebne .aplikacije.

Napomena: Kao što dodjeljujete memoriju sa stvarnog diska, isto ju tako možete i osloboditi kada vam virtualni stroj više nije potreban – time će se prostor na vašem disku ponovno osloboditi.

Prije same izrade, provjerite još jedanput sve postavke virtualnog stroja. Ako ste sve dobro postavili, kliknite na "Next" gumb.



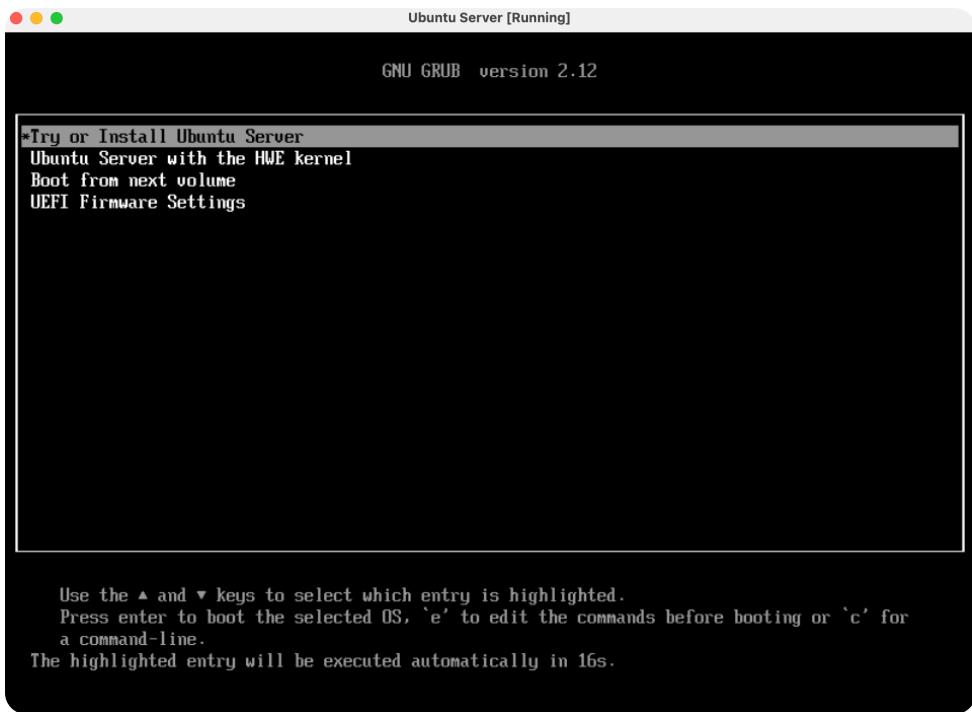
VirtualBox: Pregled postavki virtualnog stroja

To je to! Virtualni stroj možemo pokrenuti klikom na gumb "Start" 😎

2.4 Instalacija Ubuntu Servera

Jednom kada pokrenete virtualni stroj, morate proći kroz postupak instalacije operacijskog sustava. Slijedite sljedeće korake:

1. Odaberite `Try or Install Ubuntu Server` opciju
2. Odaberite jezik koji želite koristiti tijekom instalacije

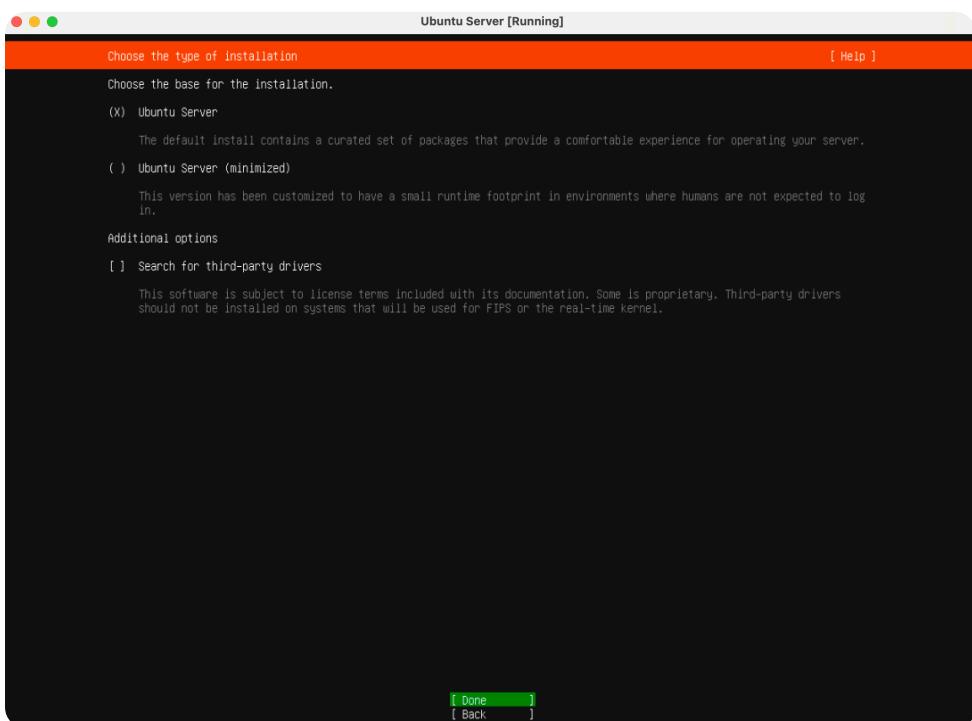


Ubuntu Server (VM): Odabir: "Try or Install Ubuntu Server"

Sada ćete morati odabrati sučelje tipkovnice. Ovisno o vašoj tipkovnici, odaberite odgovarajuće opcije ili bolje, odaberite `Identify keyboard` opciju kako bi sustav automatski prepoznao vašu tipkovnicu unosom nekoliko znakova.

Dalje, odaberite standardnu opciju `Ubuntu Server`

Mrežno sučelje ostavite zadano i odaberite `Done`, preskočite opciju `Configure proxy`.



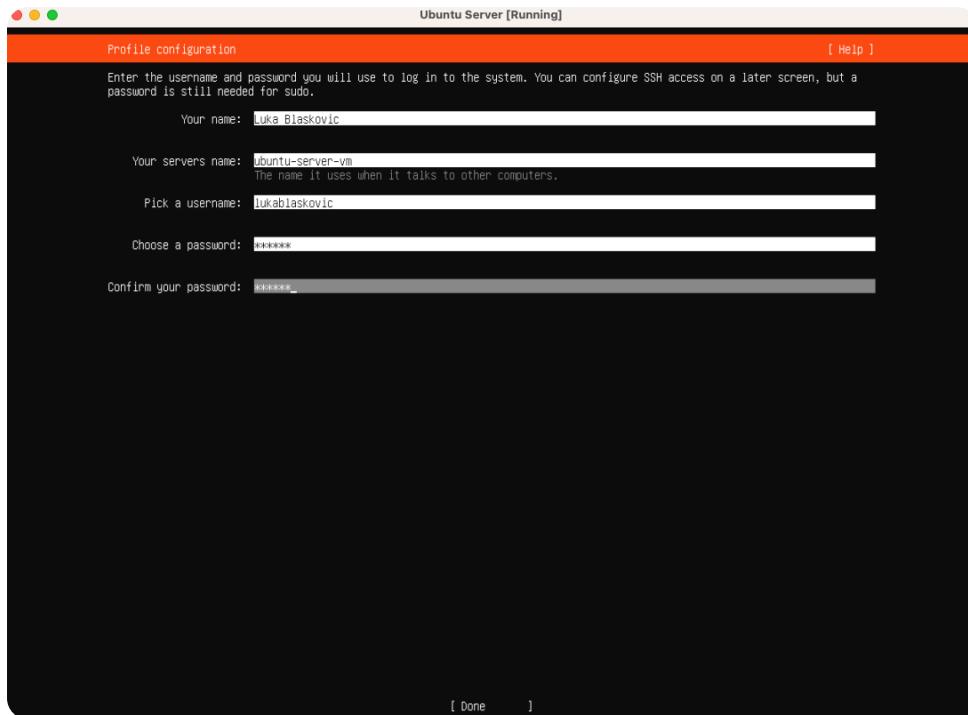
Ubuntu Server (VM): Odabir zadanih mrežnih postavki, bez proxya

Sljedeći korak je konfiguracija diska. Odaberite opciju `use an entire disk`. Ova opcija će izbrisati sve podatke na virtualnom disku koji smo ranije stvorili kroz VirtualBox hipervizor (svakako je već prazan). Odaberite `Done` pa `Continue` kako bi nastavili s instalacijom.

Posljednji korak je konfiguracija korisničkog profila. Unesite:

- vaše ime i prezime,
- naziv virtualnog stroja (npr. "ubuntu-server"),
- korisničko ime,
- lozinku.

Ove podaci nisu povezani s domaćinom, već su isključivo vezani za Ubuntu Server na VM-u.



Ubuntu Server (VM): Konfiguracija korisničkog profila

Pitati će vas još želite li dodatne opcije: [Ubuntu Pro](#), [OpenSSH server](#) i [Featured server snaps](#) - sve možete odbiti.

To je to! Instalacija će započeti i potrajati nekoliko sekundi do par minuta, ovisno o brzini računala.

```
Ubuntu Server [Running]
[ Help ]
Updating system
configuring partition: partition-2
configuring lvm_vggroup: lvm_vggroup-0
configuring lvm_partition: lvm_partition-0
configuring format: format-2
configuring mount: mount-2
configuring mount: mount-1
configuring mount: mount-0
executing curtin install extract step
curtin command: install
writing install sources to disk
running 'curlin extract'
curlin command: extract
    acquiring and extracting image from cp://tmp/tmpndarosax/mount
configuring keyboard
curlin command: in-target
executing curtin install curthooks step
curlin command: install
configuring installed system
running 'curlin curthooks'
curlin command: curthooks
configuring apt: configuring apt
Installing missing packages
Installing packages on target system: ['efibootmgr', 'grub-efi-arm64', 'grub-efi-arm64-signed', 'shim-signed']
configuring icsi service
configuring raid (mdadm) service
configuring NMW over TCP
installing kernel
setting up swap
apply networking config
writing etc/fstab
configuring multipath
updating packages on target system
configuring pollinate user-agent on target
updating initramfs configuration
configuring target system bootloader
installing grub to target devices
copying metadata from /cdrom
final system configuration
calculating extra packages to install
configuring cloud-init
downloading and installing security updates
curlin command: in-target
[ View full log ]
```

Ubuntu Server (VM): Instalacija u tijeku

Nakon završetka instalacije, odaberite opciju `Reboot now`. Nakon ponovnog pokretanja, trebali biste vidjeti sljedeći ispis greške:

```
[FAILED] Failed unmounting cdrom.mount - /cdrom.
Please remove the installation medium, then press ENTER:
```

- greška se javlja jer je instalacijski disk (ISO) još uvijek prisutan (montiran) u virtualnom stroju.
- Kao i kod stvarnog diska, jednom kad instaliramo operacijski sustav, možemo ga ukloniti budući da je OS sada instaliran na fizičkom tvrdom disku.

Zatvorite VM i odaberite opciju `Settings` (ikona zupčanika ) u VirtualBoxu. U lijevom izborniku odaberite `Storage`, a zatim odaberite `Empty` sliku diska. Jednostavno ju uklonite i odaberite `OK`.

Uočite `Ubuntu Server.vdi` virtualni disk koji smo ranije stvorili - on mora ostati i na njemu je instaliran Ubuntu Server.

Sada možete ponovno pokrenuti VM.

Ako je sve u redu, trebali biste vidjeti sljedeći ispis:

- Ubuntu Server će vas pitati za korisničko ime i lozinku koju ste ranije postavili

```
Ubuntu 24.04.2 LTS ubuntu-server-vm tty1
ubuntu-server-vm login:
```

Unesite prvo `username`, zatim `password` koji ste ranije postavili prilikom instalacije.

Ako ste sve dobro napravili, trebali biste vidjeti pozdravnu poruku i osnovne informacije o sustavu:



Ubuntu Server [Running]

```
Ubuntu 24.04.2 LTS ubuntu-server-vm tty1

ubuntu-server-vm login: lukablasovic
Password:
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-59-generic aarch64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Tue May  6 20:18:41 UTC 2025

System load: 0.0          Processes: 103
Usage of /: 45.6% of 9.75GB  Users logged in: 0
Memory usage: 9%           IPv4 address for enp0s8: 192.168.68.118
Swap usage: 0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
```

Ubuntu Server (VM): Uspješna prijava u Ubuntu Server OS

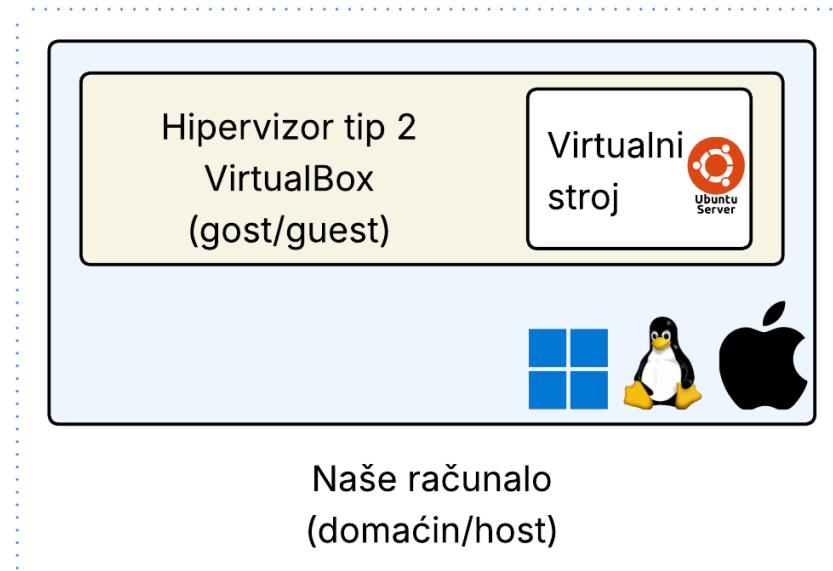
Odmah na početnoj stranici možete vidjeti nekoliko korisnih informacija:

- verziju Ubuntu Servera (24.04.2 LTS)
- naziv virtualnog stroja (ubuntu-server-vm)
- Broj aktivnih procesa (103)
- Iskorištenost radne memorije (9%)
- Iskorištenost diska (45.6% od 9.75 GB)
- Broj prijavljenih korisnika
- ...

3. Ubuntu Server

Uspješno smo instalirali Ubuntu Server unutar VirtualBoxa hipervizora na našem računalu. Sada ćemo provjeriti osnovne postavke sustava i upoznati se s nekim osnovnim naredbama.

Kratki podsjetnik: naše okruženje sada izgleda ovako:



Ilustracija hijerarhije između VirtualBox hipervizora i Ubuntu Server VM-a

Rekli smo se naredbom `echo $0` možemo uvjeriti da koristimo bash shell. Također, možemo provjeriti verziju Ubuntu Servera pomoću sljedeće naredbe:

```
→ echo $0 # aktivni shell  
→ lsb_release -a # podaci o distribuciji
```

Ispis će izgledati otprilike ovako:

```
-bash  
  
Distributor ID: Ubuntu  
Description:     Ubuntu 24.04.2 LTS  
Release:        24.04  
Codename:       noble
```

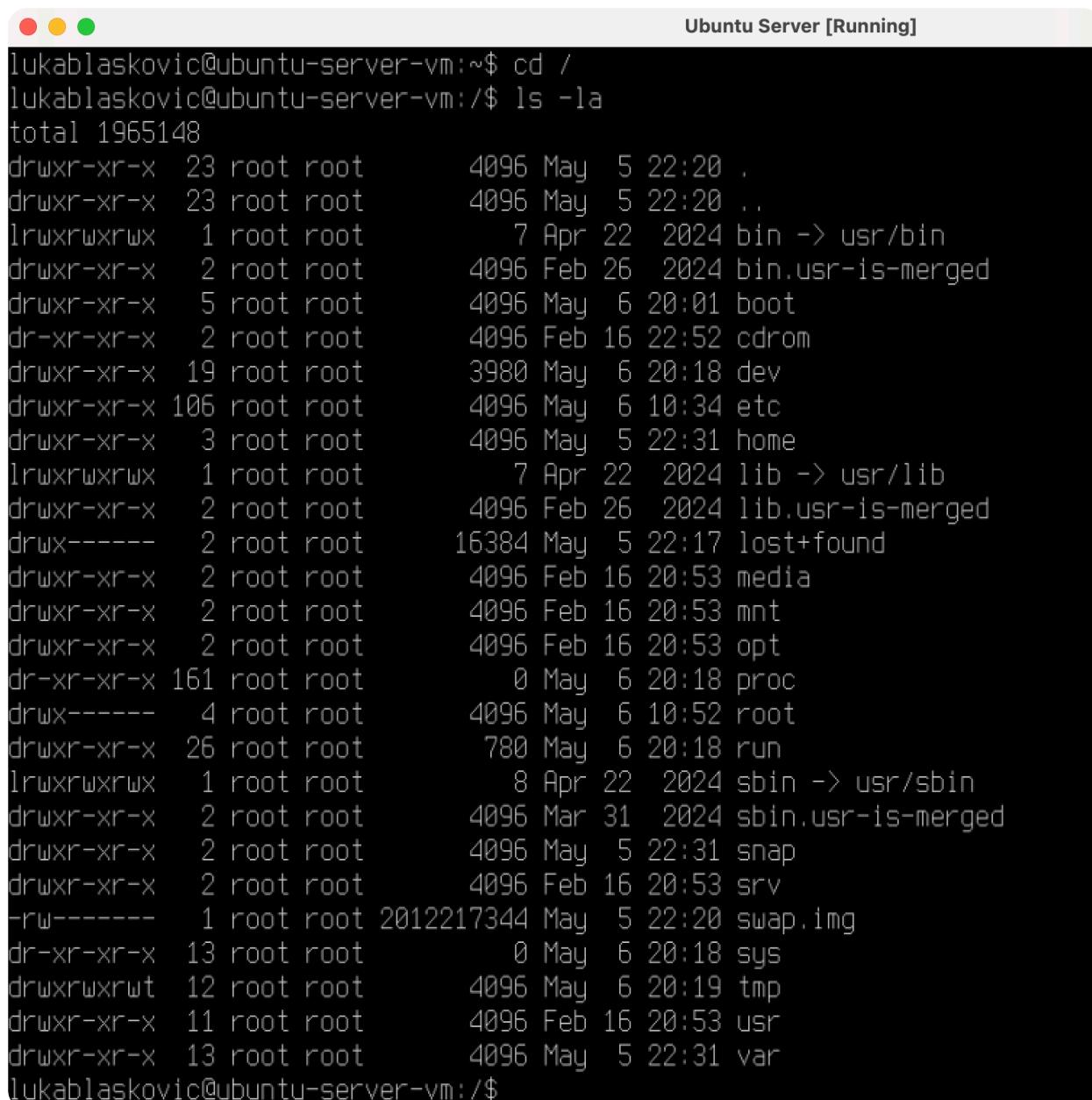
Provjerite trenutni radni direktorij - trebali biste se nalaziti u home direktoriju:

```
→ pwd # /home/<username>
```

Vidjet ćete da je home direktorij prazan. To je zato što smo tek instalirali sustav i nismo još ništa radili.

Prebacite se u korijenski direktorij i provjerite sadržaj:

```
→ cd /  
→ ls -la
```



Ubuntu Server [Running]

```
lukablavkovic@ubuntu-server-vm:~$ cd /  
lukablavkovic@ubuntu-server-vm:/$ ls -la  
total 1965148  
drwxr-xr-x 23 root root 4096 May  5 22:20 .  
drwxr-xr-x 23 root root 4096 May  5 22:20 ..  
lrwxrwxrwx  1 root root  7 Apr 22 2024 bin -> usr/bin  
drwxr-xr-x  2 root root 4096 Feb 26 2024 bin usr-is-merged  
drwxr-xr-x  5 root root 4096 May  6 20:01 boot  
dr-xr-xr-x  2 root root 4096 Feb 16 22:52 cdrom  
drwxr-xr-x 19 root root 3980 May  6 20:18 dev  
drwxr-xr-x 106 root root 4096 May  6 10:34 etc  
drwxr-xr-x  3 root root 4096 May  5 22:31 home  
lrwxrwxrwx  1 root root  7 Apr 22 2024 lib -> usr/lib  
drwxr-xr-x  2 root root 4096 Feb 26 2024 lib usr-is-merged  
drwx----- 2 root root 16384 May  5 22:17 lost+found  
drwxr-xr-x  2 root root 4096 Feb 16 20:53 media  
drwxr-xr-x  2 root root 4096 Feb 16 20:53 mnt  
drwxr-xr-x  2 root root 4096 Feb 16 20:53 opt  
dr-xr-xr-x 161 root root  0 May  6 20:18 proc  
drwx----- 4 root root 4096 May  6 10:52 root  
drwxr-xr-x 26 root root 780 May  6 20:18 run  
lrwxrwxrwx  1 root root  8 Apr 22 2024 sbin -> usr/sbin  
drwxr-xr-x  2 root root 4096 Mar 31 2024 sbin usr-is-merged  
drwxr-xr-x  2 root root 4096 May  5 22:31 snap  
drwxr-xr-x  2 root root 4096 Feb 16 20:53 srv  
-rw-------  1 root root 2012217344 May  5 22:20 swap.img  
dr-xr-xr-x 13 root root  0 May  6 20:18 sys  
drwxrwxrwt 12 root root 4096 May  6 20:19 tmp  
drwxr-xr-x 11 root root 4096 Feb 16 20:53 usr  
drwxr-xr-x 13 root root 4096 May  5 22:31 var  
lukablavkovic@ubuntu-server-vm:/$
```

 Ubuntu Server (VM): Detaljni prikaz sadržaja korijenskog direktorija (`ls -la`)

3.1 APT (Advanced Package Tool)

Advanced Package Tool (`apt`) je besplatni alat za upravljanje paketima (eng. *package manager*) koji se koristi u Debian i Ubuntu distribucijama Linuxa. Ovaj alat omogućava korisnicima lakše upravljanje softverskim paketima, uključujući **instalaciju**, **nadogradnju** i **uklanjanje** paketa kroz jednostavne CLI naredbe.

Unesite naredbu `apt` i pritisnite `Enter`. Trebali biste dobiti sljedeći ispis:

```
lukablašković@ubuntu-server-vm:~$ apt
apt 2.7.14 (arm64)
Usage: apt [options] command

apt is a commandline package manager and provides commands for
searching and managing as well as querying information about packages.
It provides the same functionality as the specialized APT tools,
like apt-get and apt-cache, but enables options more suitable for
interactive use by default.

Most used commands:
list - list packages based on package names
search - search in package descriptions
show - show package details
install - install packages
reinstall - reinstall packages
remove - remove packages
autoremove - automatically remove all unused packages
update - update list of available packages
upgrade - upgrade the system by installing/upgrading packages
full-upgrade - upgrade the system by removing/installing/upgrading packages
edit-sources - edit the source information file
satisfy - satisfy dependency strings

See apt(8) for more information about the available commands.
Configuration options and syntax is detailed in apt.conf(5).
Information about how to configure sources can be found in sources.list(5).
Package and version choices can be expressed via apt_preferences(5).
Security details are available in apt-secure(8).

This APT has Super Cow Powers.

lukablašković@ubuntu-server-vm:~$
```

 Ubuntu Server (VM): Prikaz metapodataka i osnovnih naredbi `apt` alata

U prikazu ćete vidjeti sljedeće informacije:

- verziju alata
- kratki opis alata
- popis najčešće korištenih naredbi i njihove kratke opise

Naredba `apt list` će krenuti listati sve dostupne pakete koje je moguće instalirati. Ovo može potrajati par sekundi, a kako biste zaustavili listanje, pritisnite `CTRL + C` tipke.

Primijetit će se da ova naredba izlistava na stotine paketa - od kud svi ti paketi? Kako je moguće da imamo toliko paketa na našem sustavu ako smo tek instalirali sustav koji ima 2-3 GB? Više informacija u nastavku.

Paketi su osnovne jedinice softvera u Linuxu, a svaki sadrži:

- implementaciju softvera
- njegove zavisnosti
- metapodatke koji definiraju kako se taj paket koristi i instalira.

Paketi su obično komprimirani i dolaze u različitim formatima, ovisno o distribuciji Linuxa.

Paketi koji se izlistavaju nisu instalirani na Ubuntu Serveru, već su dostupni za instalaciju. Informacije o dostupnim paketima `apt` preuzima iz **repositorija** (eng. *repository*) - **centraliziranog skladišta softverskih paketa koji su dostupni za preuzimanje i instalaciju**.

Do sad ste se već susreli s repozitorijima, i to `git` repozitorijima koji se mogu klonirati i preuzeti na lokalno računalo, ali i pohraniti na udaljeni poslužitelj (npr. GitHub, GitLab, Bitbucket i sl.). U ovom slučaju, **repositorij je centralizirano mjesto gdje se nalaze svi dostupni paketi za instalaciju**.

`apt` preuzima pakete s online repozitorija prema potrebi. Informacije o dostupnim paketima se pohranjuju u `/etc` konfiguracijski direktorij u koji se pohranjuju razno-razni konfiguracijski podaci i datoteke, uključujući i bash skripte. Konkretno, informacije o repozitorijima se nalaze u `/etc/apt/sources.list.d` direktoriju.

Ispišite sadržaj direktorija:

```
→ ls -la /etc/apt/sources.list.d
```

Vidjet ćete datoteku `ubuntu.sources` koja sadrži informacije o repozitorijima. Pročitat ćemo sadržaj datoteke pomoću `cat` naredbe:

```
→ cd /etc/apt/sources.list.d  
→ cat ubuntu.sources
```

```
lukablaskovic@ubuntu-server-vm:/etc/apt/sources.list.d$ cat ubuntu.sources
Types: deb
URIs: http://ports.ubuntu.com/ubuntu-ports/
Suites: noble noble-updates noble-backports
Components: main restricted universe multiverse
Signed-By: /usr/share/keyrings/ubuntu-archive-keyring.gpg

Types: deb
URIs: http://ports.ubuntu.com/ubuntu-ports/
Suites: noble-security
Components: main restricted universe multiverse
Signed-By: /usr/share/keyrings/ubuntu-archive-keyring.gpg
lukablaskovic@ubuntu-server-vm:/etc/apt/sources.list.d$
```

Ubuntu Server (VM): Ispis sadržaja `ubuntu.sources` datoteke

Uočite dva zapisa u sljedećem formatu:

```
Types: deb
URIs: http://ports.ubuntu.com/ubuntu-ports
Suites: noble-security
Components: main restricted universe multiverse
Signed-By: /usr/share/keyring/ubuntu-archive-keyring.gpg
```

- `.deb` označava da se radi o binarnom paketu (eng. *binary package*). Općenito govoreći, binarni paketi su već kompajlirani i spremni za instalaciju. Konkretno, `.deb` označava da se radi o Debian paketu (distribucija Linuxa na kojoj se temelji i sam Ubuntu). Ove pakete moguće je instalirati direktno, ili preuzimanjem i instalacijom putem `apt` alata.
- `URIs` označava URL repozitorija s kojeg se paketi preuzimaju

Još nam je zanimljiv ključ `signed-By` koji označava putanju do datoteke s javnim ključevima koji se koriste za potpisivanje paketa. Ovaj ključ se koristi za **provjeru autentičnosti paketa** i kao osiguranje da su paketi preuzeti s pouzdanog izvora, a ne modificirani od trećih strana. Javni ključevi se pohranjuju u `/usr/share/keyrings` direktoriju.

```
→ ls /usr/share/keyrings # ispisuje sve javne ključeve (gpg format) pohranjene na sustavu
```

Poveznica s repozitorijem biblioteka razlikuje se ovisno o verziji Ubuntu Servera i arhitekturi procesora. Primjerice, kod 64-bitnih x86 arhitektura, poveznica će biti `http://archive.ubuntu.com/ubuntu/`, dok će kod ARM64 arhitektura u pravilu biti `http://ports.ubuntu.com/ubuntu-ports`.

Ako otvorite poveznicu u web pregledniku, **vidjet ćete datotečni sustav pohranjen na web stranici koji sadrži sve dostupne pakete za instalaciju**. Svaki od tih paketa je pohranjen u nekom od poddirektorija.

Index of /ubuntu-ports

Name	Last modified	Size
Parent Directory		-
dists/	2025-04-22 14:39	-
indices/	2025-05-06 08:35	-
ls-LR.gz	2025-05-06 08:02	31M
pool/	2010-02-27 06:30	-
project/	2024-11-24 21:31	-
ubuntu-ports/	2025-05-06 08:12	-

Apache/2.4.29 (Ubuntu) Server at ports.ubuntu.com Port 443



Datotečni sustav javnog repozitorija - moguće ga je otvoriti na webu

Vratimo se na `apt` alat. Da bismo preuzele najnovije informacije o dostupnim paketima, **ali bez da ih instaliramo**, koristimo naredbu `update`:

```
→ apt update
```

Da bismo pozvali naredbu, moramo biti *root* korisnik ili imati **administratorske privilegije**. Trenutno nismo *root* korisnik, već običan korisnik. U tom slučaju, dodajemo `sudo` ispred naredbe - što označava da ćemo izvršiti naredbu kao super korisnik. `sudo` je skraćenica za "superuser do" i **omogućava običnim korisnicima da izvrše određene naredbe koje zahtijevaju administratorske privilegije**.

Ova naredba će zahtijevati unos lozinke koju ste postavili prilikom instalacije sustava. Međutim, prema zadanim postavkama `sudo` će zapamtiti vašu lozinku najčešće na 15 minuta, tako da nećete morati unositi lozinku svaki put kada koristite `sudo`.

```
→ sudo apt update
```

```
lukablaskovic@ubuntu-server-vm:/etc/apt/sources.list.d$ sudo apt update
Hit:1 http://ports.ubuntu.com/ubuntu-ports noble InRelease
Hit:2 http://ports.ubuntu.com/ubuntu-ports noble-updates InRelease
Hit:3 http://ports.ubuntu.com/ubuntu-ports noble-backports InRelease
Hit:4 http://ports.ubuntu.com/ubuntu-ports noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
64 packages can be upgraded. Run 'apt list --upgradable' to see them.
lukablaskovic@ubuntu-server-vm:/etc/apt/sources.list.d$
```

 Ubuntu Server (VM): Rezultat `apt update` naredbe - uočite da postoji 64 paketa koje možemo ažurirati

Ova naredba ustvari **ažurira lokalno pohranjenu listu dostupnih paketa i njihovih verzija** i preporučuje se pokrenuti prije instalacije ili ažuriranja paketa. Na taj način, `apt` zna koji su paketi dostupni za instalaciju, odnosno koje verzije paketa su dostupne.

Uočite u ispisu na slici iznad da postoji 64 paketa koji su dostupni za nadogradnju. To su paketi koji su već instalirani na vašem sustavu, ali postoje novije verzije na **udaljenom repozitoriju**.

Nakon izvršavanja naredbe `apt update`, možemo izlistati pakete dostupne za nadogradnju pomoću zastavice `--upgradable` naredbe `apt list`:

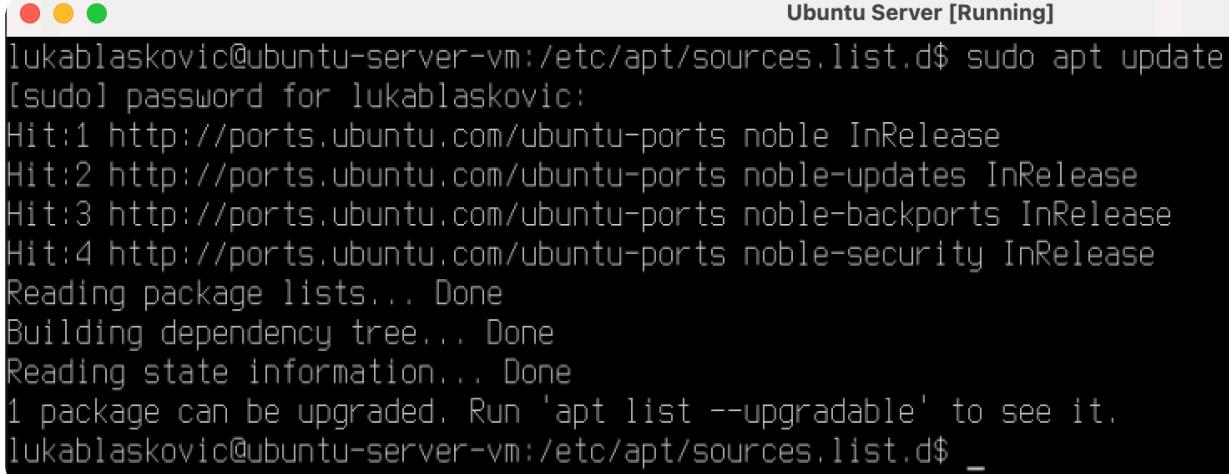
```
→ apt list --upgradable
```

Naredba `apt upgrade` će **nadograditi sve pronađene pakete** na najnoviju verziju - prema informacijama koje smo preuzeли u prethodnoj naredbi.

```
→ sudo apt upgrade
```

Ovo može potrajati nekoliko sekundi do nekoliko minuta, ovisno o tome koliko paketa imate instalirano i koliko je nadogradnji dostupno te o brzini vaše internetske veze. Neki paketi se potencijalno ne mogu nadograditi zbog razno-raznih zavisnosti i potrebno ih je ručno pregledati.

Nakon što je nadogradnja završena, možemo ponovo pozvati naredbu `apt update` kako bismo provjerili ima li novih nadogradnji.



```
lukablaskovic@ubuntu-server-vm:/etc/apt/sources.list.d$ sudo apt update
[sudo] password for lukablaskovic:
Hit:1 http://ports.ubuntu.com/ubuntu-ports noble InRelease
Hit:2 http://ports.ubuntu.com/ubuntu-ports noble-updates InRelease
Hit:3 http://ports.ubuntu.com/ubuntu-ports noble-backports InRelease
Hit:4 http://ports.ubuntu.com/ubuntu-ports noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
lukablaskovic@ubuntu-server-vm:/etc/apt/sources.list.d$ _
```

 Ubuntu Server (VM): Nakon `apt upgrade` ponovo pozivamo naredbu `apt update` kako bismo provjerili koliko je paketa ažurirano

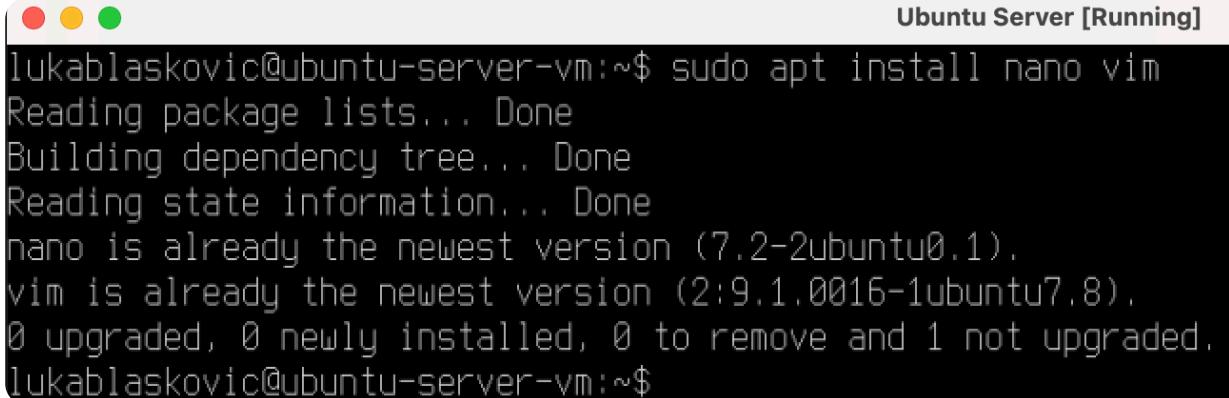
Kako bismo instalirali novi paket koristimo naredbu `apt install`. Primjerice, želimo instalirati `nano` i `vim` CLI uređivače teksta.

```
→ sudo apt install nano
→ sudo apt install vim
```

Možemo instalirati više paketa u isto vrijeme:

```
→ sudo apt install nano vim
```

Vidimo da su oba uređivača već instalirana na našem sustavu i najnovije verzije.



```
lukablaskovic@ubuntu-server-vm:~$ sudo apt install nano vim
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nano is already the newest version (7.2-2ubuntu0.1).
vim is already the newest version (2:9.1.0016-1ubuntu7.8).
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
lukablaskovic@ubuntu-server-vm:~$
```

 Ubuntu Server (VM): Instalacija `nano` i `vim` CLI uređivača

Što ako izađe nova verzija nekog paketa?

Prvo je potrebno ponovo pokrenuti `apt update` kako bismo preuzele najnovije informacije o dostupnim paketima. Nakon toga možemo ponovo pokrenuti `apt upgrade` kako bismo nadogradili sve pakete na najnoviju verziju ili pojedini paket koristeći `apt install` naredbu i zastavicu `--only-upgrade`.

```
→ sudo apt update # dohvati najnovije informacije o dostupnim paketima
→ sudo apt upgrade # nadogradi sve pakete na najnoviju verziju
```

ili pojedinačno (nadogradnja samo jednog paketa):

```
→ sudo apt update # dohvati najnovije informacije o dostupnim paketima  
→ sudo apt install --only-upgrade nano # nadogradi samo nano paket
```

Zastavicom `--installed` kod naredbe `apt list` možemo **izlistati sve instalirane pakete na sustavu**:

```
→ apt list --installed
```

Paket možemo ukloniti pomoću `apt remove` naredbe:

```
→ sudo apt remove nano
```

Ova naredba će ukloniti paket, ali ostaviti sve njegove zavisnosti. Ako želite ukloniti i zavisnosti (druge pakete koji su instalirani zajedno s ovim paketom), koristite `autoremove` naredbu:

```
→ sudo apt autoremove nano # briše nano i sve njegove zavisnosti
```

Ako želite ukloniti i konfiguracijske datoteke paketa, koristite `apt purge` naredbu:

```
→ sudo apt purge nano
```

Ova naredba će ukloniti paket i sve njegove zavisnosti, uključujući i konfiguracijske datoteke.

U nastavku se nalazi tablica najčešće korištenih `apt` naredbi:

Naredba	Sintaksa	Primjer	Objašnjenje
Update	<code>sudo apt update</code>	<code>sudo apt update</code>	Osvježava lokalnu listu dostupnih paketa iz udaljenih repozitorija
Upgrade	<code>sudo apt upgrade</code>	<code>sudo apt upgrade</code>	Nadograđuje sve instalirane pakete na najnovije verzije, ukoliko je moguće.
Full Upgrade	<code>sudo apt full-upgrade</code>	<code>sudo apt full-upgrade</code>	Nadograđuje sve instalirane pakete, uklanjajući stari ako je potrebno.
Install	<code>sudo apt install <package></code>	<code>sudo apt install curl</code>	Instalira navedeni paket iz repozitorija.
Remove	<code>sudo apt remove <package></code>	<code>sudo apt remove curl</code>	Uklanja instalirani paket, ali zadržava konfiguracijske datoteke.
Purge	<code>sudo apt purge <package></code>	<code>sudo apt purge curl</code>	Uklanja instalirani paket zajedno s njegovim konfiguracijama.
Autoremove	<code>sudo apt autoremove</code>	<code>sudo apt autoremove</code>	Uklanja nepotrebne pakete koji su instalirani kao zavisnosti drugih paketa i nisu više potrebni.
Search	<code>apt search <package></code>	<code>apt search nginx</code>	Pretražuje paket u repozitorijima prema danom nazivu paketa.
Show Package Info	<code>apt show <package></code>	<code>apt show git</code>	Ispisuje detaljne informacije o paketu.
List (Installed)	<code>apt list --installed</code>	<code>apt list -installed</code>	Prikazuje popis svih dostupnih paketa, ili samo onih instaliranih ako se pozove zastavicom <code>--installed</code> .
Clean Cache	<code>sudo apt clean</code>	<code>sudo apt clean</code>	Briše sve preuzete instalacijske datoteke iz lokalne predmemorije.

3.2 SSH (Secure Shell)

SSH (eng. *Secure Shell*) je kriptografski mrežni protokol koji omogućava sigurno povezivanje i upravljanje udaljenim računalima (ili virtualnim strojevima) putem nesigurnih mreža. Protokol je razvijen kao zamjena za početne "remote access" protokole poput Telnet-a i rlogin-a, koji su slali osjetljive podatke (poput korisničkih imena i lozinki) u običnom tekstu, što ih čini ranjivima na napade.

Glavne funkcionalnosti SSH protokola:

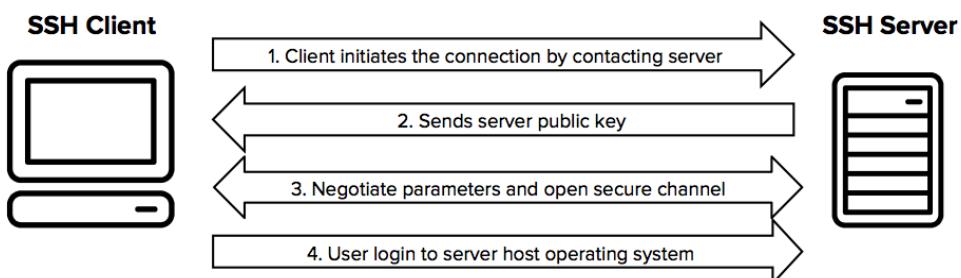
- **Enkripcija:** Svi podaci koji se šalju između klijenta i poslužitelja su **enkriptirani**
- **Autentifikacija:** SSH koristi različite **metode autentifikacije**, uključujući lozinke i javne ključeve
- **Integritet:** SSH **osigurava da podaci nisu izmijenjeni** tijekom prijenosa

- **Prijenos podataka:** SSH omogućava **siguran prijenos datoteka** između klijenta i poslužitelja putem protokola SCP (Secure Copy Protocol) i SFTP (SSH File Transfer Protocol)

SSH je široko korišten u administraciji sustava, a danas postoji mnogo različitih SSH klijenata i poslužitelja dostupnih za različite operacijske sustave. Najpoznatiji SSH klijent je `openssh`, koji je uključen u većinu Linux distribucija, uključujući i Ubuntu Server.

Instalirat ćemo `openssh` poslužitelj na našem Ubuntu Serveru kako bismo omogućili udaljeni pristup sustavu putem SSH protokola. Konkretnije, udaljeni pristup ćemo izvršavati preko računala koje je domaćin VirtualBox hipervizora. Međutim, na ovaj način možemo pristupiti i bilo kojem drugom računalu na udaljenoj lokaciji, pod uvjetom da imamo pristup mreži i potrebne autentifikacijske podatke.

Prema tome, naše računalo domaćina će biti **SSH klijent** (eng. *SSH client*), a naš virtualni stroj će biti **SSH poslužitelj** (eng. *SSH server*).



Ilustracija SSH protokola - <https://www.ssh.com/academy/ssh>

3.2.1 Instalacija OpenSSH poslužitelja



OpenSSH je besplatan alat za udaljeno povezivanje putem SSH protokola

Unutar Ubuntu Servera ćemo instalirati `OpenSSH` poslužitelj pomoću `apt` alata:

```

→ sudo apt update
→ sudo apt install openssh-server

```

Alat `systemctl`

Nakon instalacije, virtualni stroj će automatski restartirati određene servise i pokrenuti OpenSSH poslužitelj. Da bismo provjerili je li SSH poslužitelj pokrenut, možemo koristiti paket `systemctl`.

Općenito, alat `systemctl` se koristi za upravljanje sustavnim servisima i procesima na Linux sustavima. Ovaj alat omogućava korisnicima da pokreću, zaustavljaju, restartaju i provjeravaju status raznih servisa koji se izvode na sustavu.

Argumentom `status` možemo dobiti status određenog servisa. U ovom slučaju, provjeravamo status SSH poslužitelja:

```
→ sudo systemctl status ssh
```



```
lukablasovic@ubuntu-server-vm:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
  Active: inactive (dead)
TriggeredBy: • ssh.socket
    Docs: man:sshd(8)
          man:sshd_config(5)
lukablasovic@ubuntu-server-vm:~$ _
```



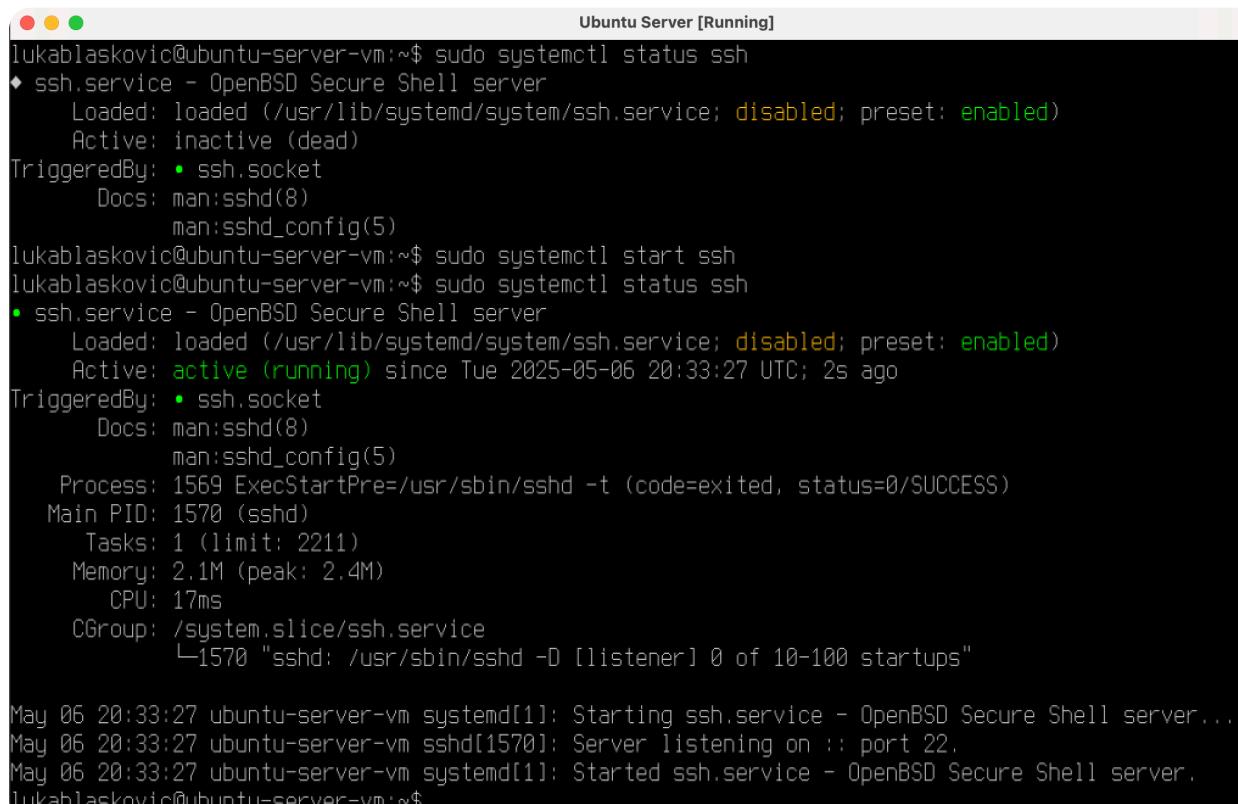
Ako je instalacija uspješno prošla, trebali biste vidjeti "OpenBSD Secure Shell server", međutim **vjerojatno nije pokrenut**, uočite: `Active: inactive (dead)`.

Da bismo ga pokrenuli, koristimo `systemctl` naredbu i `start` naredbu:

```
→ sudo systemctl start ssh
```

Nakon toga možemo ponovo provjeriti status:

```
→ sudo systemctl status ssh
```



```
lukablasovic@ubuntu-server-vm:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
  Active: active (running) since Tue 2025-05-06 20:33:27 UTC; 2s ago
TriggeredBy: • ssh.socket
    Docs: man:sshd(8)
          man:sshd_config(5)
Process: 1569 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
Main PID: 1570 (sshd)
  Tasks: 1 (limit: 2211)
  Memory: 2.1M (peak: 2.4M)
    CPU: 17ms
   CGroup: /system.slice/ssh.service
           └─1570 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

May 06 20:33:27 ubuntu-server-vm systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
May 06 20:33:27 ubuntu-server-vm sshd[1570]: Server listening on :: port 22.
May 06 20:33:27 ubuntu-server-vm systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
lukablasovic@ubuntu-server-vm:~$ _
```



U Linux OS-u postoji [princip/pristup koji kaže da je "sve datoteka"](#). To znači da se gotovo sve – uključujući obične dokumente, direktorije, uređaje poput tvrdog diska, pa čak i mrežne priključke ili servise – tretira kao datoteka. Navedeno omogućuje jednostavnije i ujednačeno rukovanje svim tim resursima.

Kada pišemo bash skripte, vidjeli smo da često trebamo provjeriti što neka "datoteka" zapravo predstavlja. U tu svrhu smo koristili zastavice:

- f** provjerava je li nešto regularna datoteka (npr. tekstualni dokument, slika, program).
- d** provjerava je li nešto direktorij.

Alat **lsof**

lsof (eng. *list open files*) je alat koji se koristi za prikaz svih otvorenih datoteka i procesa koji ih koriste. Ovaj alat je posebno koristan za dijagnostiku problema s datotekama i procesima, kao i za praćenje aktivnosti sustava. Općenito govoreći, na Linux sustavima, datoteke mogu biti:

- regularne datoteke (eng. *regular files*)
- direktoriji (eng. *directories*)
- mrežni priključak (eng. *network sockets*)
- uređaj (eng. *devices*)
- biblioteka (eng. *libraries*)
- i drugi resursi...

Proces u Linuxu je instanca programa koji se izvršava u radnoj memoriji. Svaki proces ima svoj jedinstveni identifikator (**PID**) i može imati **više otvorenih datoteka za vrijeme svog izvršavanja**.

Primjer: želimo ispisati sve otvorene datoteke

```
→ sudo lsof # ispisuje sve otvorene datoteke
```

Ako želimo provjeriti samo datoteke asocijirane s **mrežnim protokolima**, možemo koristiti **-i** opciju:

```
→ sudo lsof -i # ispisuje sve otvorene datoteke asocijirane s mrežnim protokolima
```

Možemo i navesti port koji nas zanima, primjerice, **SSH obično koristi port 22**:

```
→ sudo lsof -i :22 # koristimo :<port>
```

```

Ubuntu Server [Running]
lukablaslavic@ubuntu-server-vm:~$ sudo lsof -i
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
systemd 1 root 93u IPv6 9127 0t0 TCP *:ssh (LISTEN)
systemd-n 529 systemd-network 21u IPv4 7655 0t0 UDP ubuntu-server-vm:bootpc
systemd-r 549 systemd-resolve 14u IPv4 8968 0t0 UDP _localdnsstub:domain
systemd-r 549 systemd-resolve 15u IPv4 8969 0t0 TCP _localdnsstub:domain (LISTEN)
systemd-r 549 systemd-resolve 16u IPv4 8970 0t0 UDP _localdnsproxy:domain
systemd-r 549 systemd-resolve 17u IPv4 8971 0t0 TCP _localdnsproxy:domain (LISTEN)
sshd 1014 root 3u IPv6 9127 0t0 TCP *:ssh (LISTEN)
lukablaslavic@ubuntu-server-vm:~$ sudo lsof -i :22
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
systemd 1 root 93u IPv6 9127 0t0 TCP *:ssh (LISTEN)
sshd 1014 root 3u IPv6 9127 0t0 TCP *:ssh (LISTEN)
lukablaslavic@ubuntu-server-vm:~$ -

```

 Ubuntu Server (VM): Ispis svih otvorenih datoteka i procesa koji koriste port 22 (SSH)

3.2.2. Povezivanje na SSH poslužitelj

Sada kada imamo pokrenut SSH poslužitelj, možemo se **povezati na njega putem SSH klijenta** (naše računalo domaćin VM-a). Da bismo to učinili, trebamo znati IP adresu našeg virtualnog stroja.

IP adresu možemo dobiti pomoću više naredbi, ali najčešće korištena je `ip` naredba:

Alat `ip`

```

→ ip a
# ili
→ ip address

```

Zapis naredbe je dosta kompleksan i dugačak, a u pravilu se sastoji od 2 dijela:

1. **Loopback interface** (`lo`) - sučelje predstavlja virtualni mrežni adapter koji se koristi za komunikaciju unutar samog virtualnog stroja, od tuda naziv *loopback*. Možemo iščitati IP adresu `127.0.0.1`. Ovu adresu nećemo koristiti za komunikaciju s našim VM-om
2. **Network interface** (`enp0s8`) - ovo sučelje predstavlja fizički adapter koji se koristi za komunikaciju s vanjskim svijetom, zamislimo ga poput mrežnog kabla koji je spojen na naš VM. Ovdje ćemo pronaći:
 - **MAC adresu**: (npr. `08:00:27:e0:02:dc`) koja predstavlja jedinstveni identifikator mrežnog adaptera (hardvera)
 - **IP adresu**: (`10.0.0.15`) koja predstavlja adresu našeg VM-a unutar mreže - ona nam omogućuje komunikaciju s drugim uređajima, kao i pristup internetu (ako je omogućen)
 - **Subnet masku**: (`/24`) koja određuje veličinu mreže (u našem slušaju ukupno 254 uređaja može biti spojeno na istoj mreži budući da je mrežni dio maske fiksan za prvih 24 bita)
 - **Broadcast adresu** (`10.0.0.255`) kojom se šalju poruke svim uređajima na mreži

O mrežama više na kolegiju [Mrežni sustavi](#).

```
Ubuntu Server [Running]
lukablasovic@ubuntu-server-vm:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:e0:02:dc brd ff:ff:ff:ff:ff:ff
        inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s8
            valid_lft 85766sec preferred_lft 85766sec
        inet6 fd00::a00:27ff:fee0:2dc/64 scope global dynamic mngtmpaddr noprefixroute
            valid_lft 86119sec preferred_lft 14119sec
        inet6 fe80::a00:27ff:fee0:2dc/64 scope link
            valid_lft forever preferred_lft forever
lukablasovic@ubuntu-server-vm:~$ _
```

 Ubuntu Server (VM): Ispis IP adrese virtualnog stroja naredbom ip address

Samo IP adresu mrežnog sučelja možemo dobiti i pomoću naredbe `hostname`:

```
→ hostname -I
```

Do sad smo podesili SSH poslužitelj i znamo njegovu IP adresu. SSH klijent je već instaliran na vašem računalu domaćinu, a to je OpenSSH klijent (dodano za Windows 10 i 11). Drugi poznati SSH klijenti su PuTTY i MobaXterm.

Kako bi se povezali na SSH poslužitelj, jednostavno otvorimo bilo koji terminal i unosimo `ssh` naredbu te naziv korisnika na VM-u i njegovu IP adresu.

Sintaksa:

```
ssh <username>@<ip_address>
```

Primjer:

```
→ ssh username@10.0.2.15
```

Ipak, ako se pokušamo spojiti na ovaj način dobit ćemo grešku!

```
ssh: connect to host 10.0.2.15 port 22: Operation timed out
```

Prema zadanim postavkama, VirtualBox koristi **NAT mrežni adapter** za svoje virtualne strojeve. NAT (*eng. Network Address Translation*) je tehnologija koja omogućuje da više uređaja dijeli jednu javnu IP adresu. U ovom slučaju, VirtualBox koristi ugrađeni virtualni usmjerivač koji upravlja prometom između vašeg računala i virtualnog stroja.

To znači da:

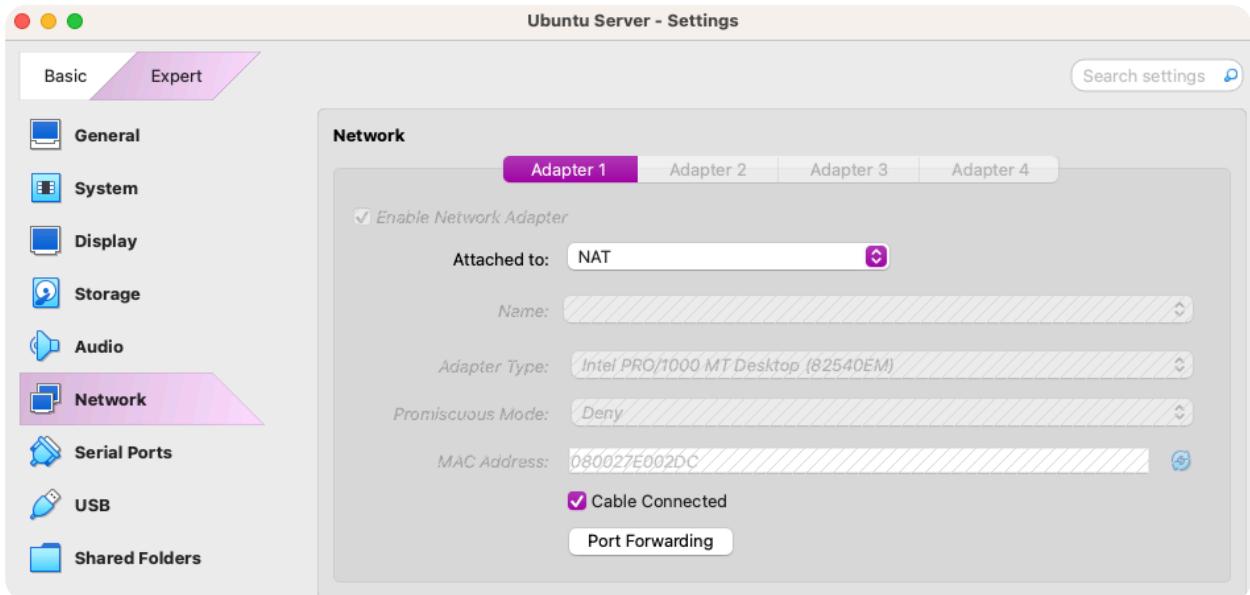
- Virtualni stroj nema izravno dostupnu IP adresu unutar lokalne mreže.
- Domaćin ne može inicirati SSH vezu prema virtualnom stroju, jer je promet iz vanjske mreže prema "NAT-iranom" VM-u blokiran.

Zamislite da je vaš VM „iza“ usmjerivača – baš kao što su vaša računala i pametni uređaji kod kuće iza kućnog routera. U tom slučaju, ne možete pristupiti VM-u izvan VirtualBoxa bez dodatne konfiguracije.

Ovaj problem možemo riješiti na 2 načina:

Rješenje 1: Odrađujemo *port forwarding* da bismo mogli pristupiti SSH poslužitelju na VM-u. Isto omogućuje da se promet s **određenog porta na domaćinu preusmjeri na određeni port na VM-u.**

Otvorite VirtualBox i otvorite postavke VM-a. Na vrhu ćete vidjeti odabir `Basic`, `Expert`. Odaberite `Expert` kako bi vam se otvorila dodatna opcija `Port Forwarding`. Odaberite `Network → Port Forwarding`.



VirtualBox: Settings → Expert → Port Forwarding

Alat `ss`

Rekli smo da je SSH poslužitelj na VM-u pokrenut na portu `22`. Isto možemo provjeriti pomoću alata `ss` (socket statistics) koji prikazuje sve otvorene portove i aktivne veze na VM-u:

```
→ ss -tul
```

Kombinacija zastavica:

- `t` - TCP ([Transmission Control Protocol](#)) socket
- `u` - UDP ([User Datagram Protocol](#)) socket
- `l` - listening socket (svi portovi koji su otvoreni i čekaju na vezu)

Trebali bismo vidjeti TCP zapis `*:ssh` koji označava da je ssh dostupan na svim mrežnim sučeljima VM-a (odnosno sluša na svim IP adresama virtualnog stroja). Dodajemo zastavicu `-n` kako bismo dobili **numeričku reprezentaciju portova i IP adresa.**

```
→ ss -tuln
```

Vidimo da se port `:ssh` zamijenio brojem `22`, a port `:http` brojem `80` itd. Isto potvrđuje da je SSH poslužitelj pokrenut i da sluša na portu `22`.

```

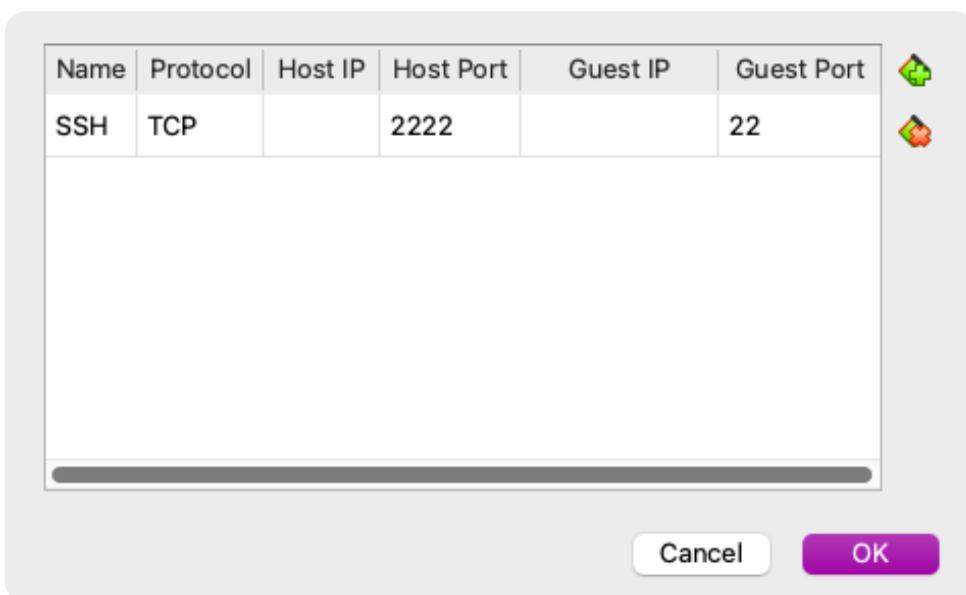
Ubuntu Server [Running]
lukablasovic@ubuntu-server-vm:~$ ss -tul
Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port Process
udp UNCONN 0 0 127.0.0.54:domain 0.0.0.0:*
udp UNCONN 0 0 127.0.0.53%lo:domain 0.0.0.0:*
udp UNCONN 0 0 10.0.2.15%enp0s8:bootpc 0.0.0.0:*
tcp LISTEN 0 4096 127.0.0.54:domain 0.0.0.0:*
tcp LISTEN 0 4096 127.0.0.53%lo:domain 0.0.0.0:*
tcp LISTEN 0 4096 *:ssh 0.0.0.0:*
lukablasovic@ubuntu-server-vm:~$ ss -tuln
Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port Process
udp UNCONN 0 0 127.0.0.54:53 0.0.0.0:*
udp UNCONN 0 0 127.0.0.53%lo:53 0.0.0.0:*
udp UNCONN 0 0 10.0.2.15%enp0s8:68 0.0.0.0:*
tcp LISTEN 0 4096 127.0.0.54:53 0.0.0.0:*
tcp LISTEN 0 4096 127.0.0.53%lo:53 0.0.0.0:*
tcp LISTEN 0 4096 *:22 0.0.0.0:*
lukablasovic@ubuntu-server-vm:~$ 

```

Sada možemo zatvoriti VM i vratiti se u mrežne postavke VirtualBoxa. U `Port Forwarding` opcijama dodajte novi unos:

- `Name`: SSH (proizvoljno)
- `Protocol`: TCP
- `Host IP`: (prazno)
- `Host Port`: 2222 (proizvoljno)
- `Guest IP`: (prazno)
- `Guest Port`: 22

`Host IP` predstavlja adresu domaćina na kojoj će se slušati promet, dok je `Guest IP` adresa VM-a na kojoj će se slušati promet. `Host Port` predstavlja port na domaćinu koji će se koristiti za preusmjeravanje prometa, dok je `Guest Port` port na VM-u na kojem će se slušati promet - u našem slučaju port 22 (SSH).



VirtualBox: Dodavanje novog unosa za port forwarding (TCP: Host 2222 -> Guest 22)

Nakon što ste dodali unos, pokrenite VM i ponovo provjerite status SSH poslužitelja.

- Ako nije pokrenut, pokrenite ga naredbom `sudo systemctl start ssh`.

Sada se možemo povezati na SSH poslužitelj putem SSH klijenta. U ovom slučaju, koristimo port `2222` na domaćinu koji je preusmjeren na port `22` na VM-u.

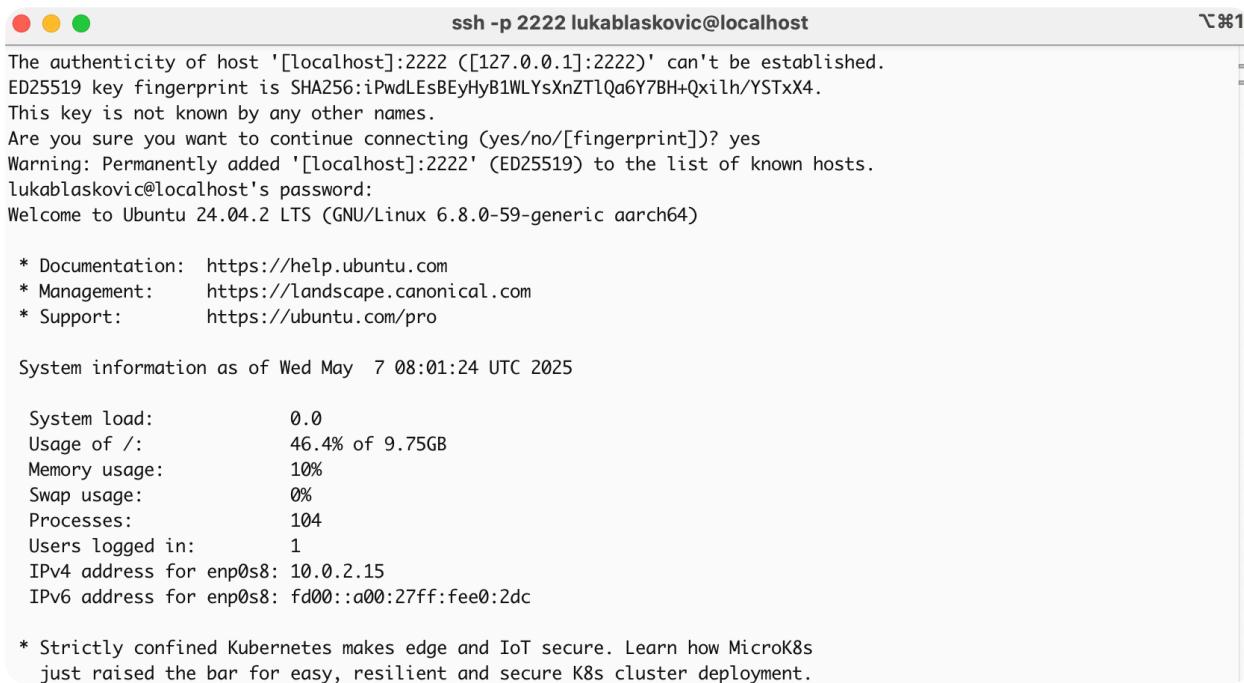
Otvorite ponovo terminal. Više nećemo koristiti IP adresu VM-a, budući da kad koristimo *port forwarding*, na neki način ustvari govorimo VirtualBoxu da **preusmjeri sav promet koji dolazi s domaćina na portu `2222` prema VM-u na portu `22`**. Budući da u unosu iznad nismo unijeli IP adresu, VirtualBox će pravilo preusmjeravanja primijeniti samo na `localhost` adresu, odnosno `127.0.0.1`.

 Hint: localhost predstavlja specijalnu IP adresu `127.0.0.1` koja se koristi za pristup mrežnim servisima koji se izvode na istom računalu koristeći loopback sučelje. [Pročitajte više...](#)

```
→ ssh username@localhost -p 2222  
# ili  
→ ssh username@127.0.0.1 -p 2222
```

- gdje je `username` korisničko ime koje ste definirali na Ubuntu Serveru

Dobit ćete upozorenje da je poslužitelj nepoznat i da se ne može provjeriti njegov identitet. Ovo je normalno, jer SSH klijent ne prepoznaje javni ključ poslužitelja - **povezujemo se prvi put**. Jednostavno unesite `yes`, a zatim lozinku koju ste postavili prilikom instalacije sustava.



```
ssh -p 2222 lukablaskovic@localhost
The authenticity of host '[localhost]:2222 ([127.0.0.1]:2222)' can't be established.
ED25519 key fingerprint is SHA256:iPwdLEsBEyHyB1WLYsXnZTlQa6Y7BH+Qxilh/YSTxX4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[localhost]:2222' (ED25519) to the list of known hosts.
lukablaskovic@localhost's password:
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-59-generic aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Wed May  7 08:01:24 UTC 2025

System load:          0.0
Usage of /:           46.4% of 9.75GB
Memory usage:         10%
Swap usage:           0%
Processes:            104
Users logged in:     1
IPv4 address for enp0s8: 10.0.2.15
IPv6 address for enp0s8: fd00::a00:27ff:fee0:2dc

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.
```

 Lokalno računalo: Uspješno povezivanje na SSH poslužitelj

Nakon uspješne autentifikacije, trebali biste biti povezani na SSH poslužitelj i vidjeti iste uvodne informacije kao i na VM-u. Sada možete upravljati VM-om putem SSH klijenta, odnosno putem vašeg terminala na lokalnom računalu.

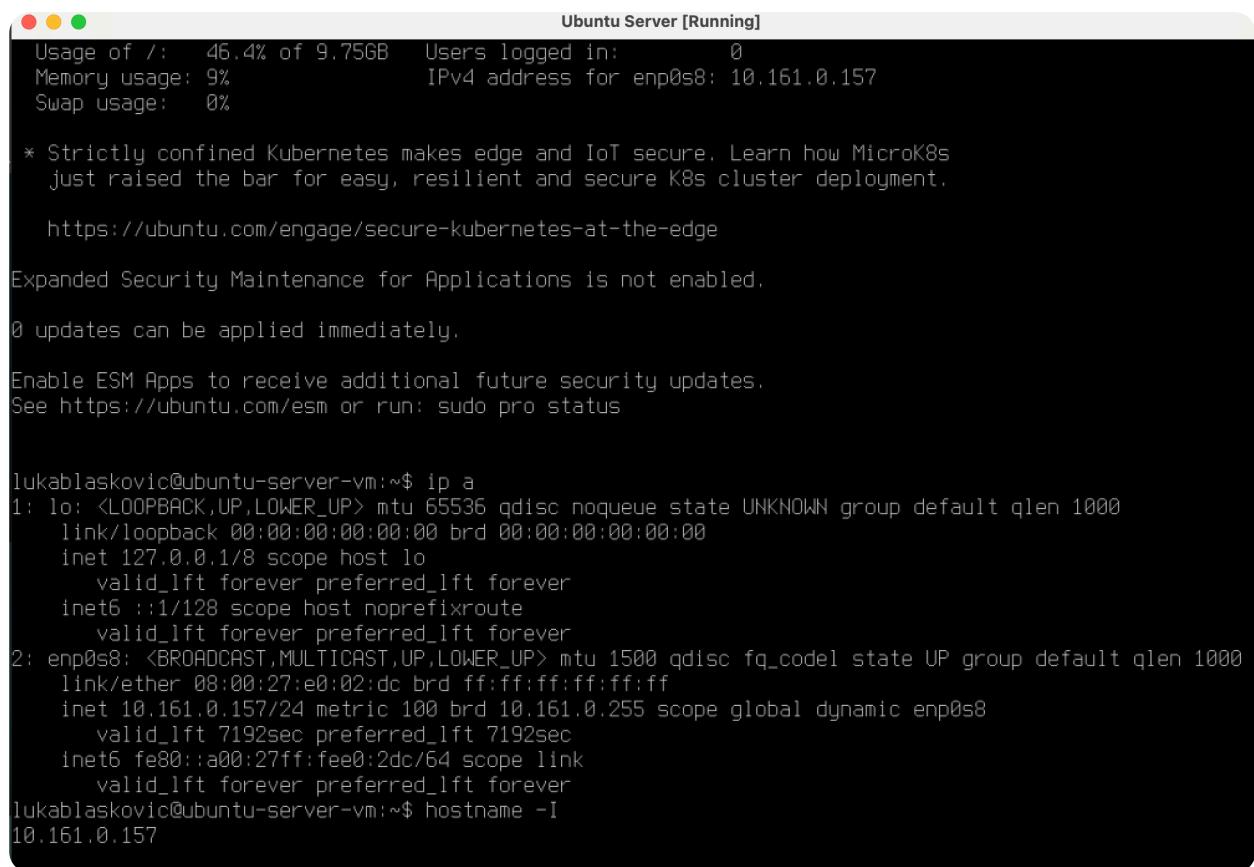
Rješenje 2: drugo rješenje je promjena mrežnog adaptera VM-a iz `NAT` u `Bridged Adapter`. Bridged adapter omogućuje VM-u da se ponaša kao fizički uređaj unutar lokalne mreže. To znači da će VM dobiti svoju vlastitu IP adresu unutar lokalne mreže, a ne "NAT-iranu" adresu. Na taj način, možete pristupiti VM-u putem njegove IP adrese **bez potrebe za port forwardingom**.

Zatvorite VM i otvorite postavke VM-a na VirtualBoxu. U `Network` odjeljku, promijenite `Attached to` opciju iz `NAT` u `Bridged Adapter`. To je to! Možete ostaviti naziv adaptera `en0: Wi-Fi`.

Ponovno pokrenite virtualni stroj i provjerite njegovu IP adresu pomoću `ip a` ili `hostname -I` naredbe.

```
→ ip a  
# ili  
→ hostname -I
```

Uočit ćete da se IP adresa promjenila.



Ubuntu Server [Running]

```
Usage of /: 46.4% of 9.75GB Users logged in: 0
Memory usage: 9% IPv4 address for enp0s8: 10.161.0.157
Swap usage: 0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

lukablaskovic@ubuntu-server-vm:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:e0:02:dc brd ff:ff:ff:ff:ff:ff
    inet 10.161.0.157/24 metric 100 brd 10.161.0.255 scope global dynamic enp0s8
        valid_lft 7192sec preferred_lft 7192sec
        inet6 fe80::a00:27ff:fee0:2dc/64 scope link
            valid_lft forever preferred_lft forever
lukablaskovic@ubuntu-server-vm:~$ hostname -I
10.161.0.157
```

 Ubuntu Server (VM): IP adresa nakon promjene mrežnog adaptera u Bridged Adapter

Pokrenite ponovo SSH poslužitelj. Sada se možemo direktno povezati na VM putem SSH klijenta koristeći njegovu IP adresu:

```
→ ssh username@<ip_address>
```

- gdje je `username` korisničko ime koje ste definirali na Ubuntu Serveru
- gdje je `<ip_address>` IP adresa VM-a kada je pokrenut s `Bridged` mrežnim adapterom

Kako biste izašli iz prethodne SSH sesije, jednostavno upišite `exit` ili pritisnite `Ctrl + D`.

Ponovno će nas pitati da li želimo dodati javni ključ poslužitelja u našu SSH poznatu listu. Unesite `yes` i lozinku koju ste postavili prilikom instalacije sustava.

```
ssh lukablaskovic@10.161.0.157
Last login: Wed May  7 09:55:58 on ttys004
~ ssh lukablaskovic@10.161.0.157
The authenticity of host '10.161.0.157 (10.161.0.157)' can't be established.
ED25519 key fingerprint is SHA256:iPwdLEsBEyHyB1WLysXnZTlQa6Y7BH+Qxilh/YSTxX4.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:4: [localhost]:2222
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.161.0.157' (ED25519) to the list of known hosts.
lukablaskovic@10.161.0.157's password:
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-59-generic aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

System information as of Wed May  7 08:12:53 UTC 2025

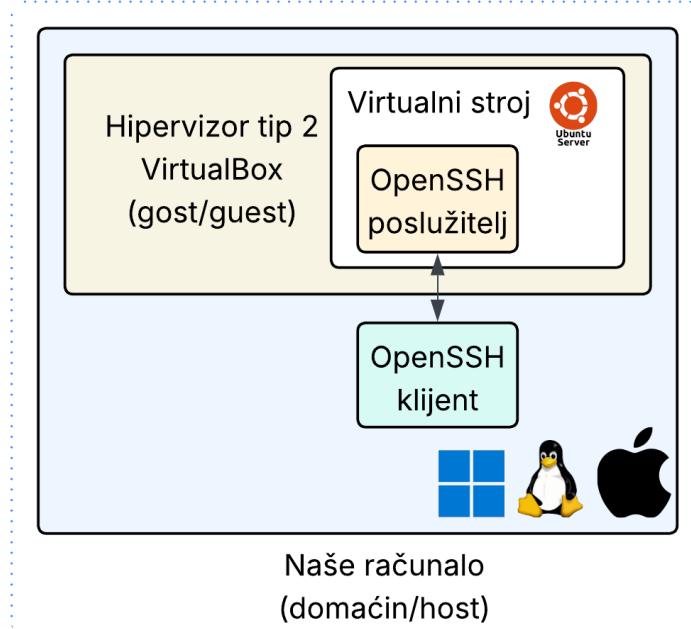
System load: 0.0          Processes:           106
Usage of /:   46.5% of 9.75GB  Users logged in:      1
Memory usage: 9%          IPv4 address for enp0s8: 10.161.0.157
Swap usage:   0%         

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.
```

 Lokalno računalo: Uspješno povezivanje na SSH poslužitelj (Bridged Adapter) - bez port forwardinga
Uspješno smo se povezali na virtualni stroj putem SSH klijenta! 

 **Hint:** Kako bismo pogledati listu javnih ključeva poznatih SSH poslužitelja, možemo otvoriti datoteku `~/.ssh/known_hosts` na našem lokalnom računalu. Ova datoteka sadrži popis svih poznatih SSH poslužitelja i njihovih javnih ključeva na koje smo se povezivali.

Okruženje koje smo sada postavili možemo prikazati sljedećom ilustracijom:



Ilustracija komunikacije između domaćina i VM-a putem SSH protokola

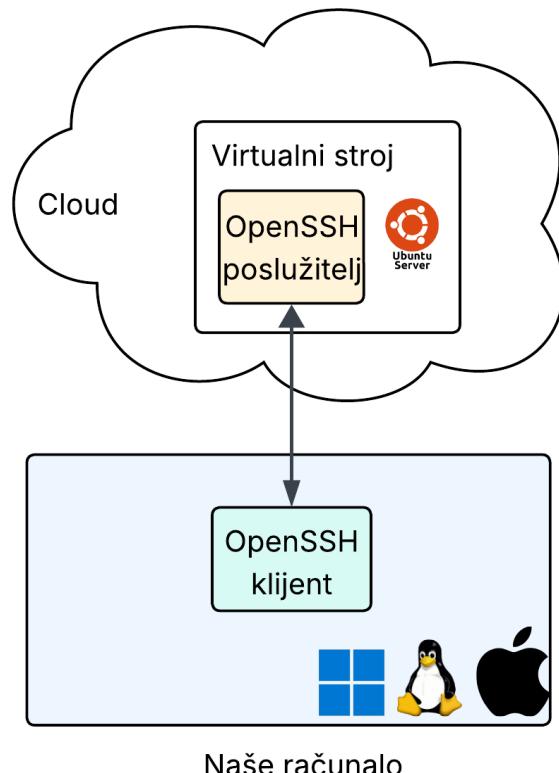
I dalje izvodimo VM putem hipervizora VirtualBox, ali sada imamo mogućnost povezivanja putem SSH klijenta. Naše računalo je domaćin VM-a, ali istovremeno i klijent u kontekstu SSH veze. Na VM smo instalirali SSH poslužitelj koji omogućuje udaljeni pristup sustavu putem SSH protokola. Naše računalo domaćin koristi SSH klijent za povezivanje na definirani SSH poslužitelj.

Sjetite se priče o "sve je datoteka". U ovom slučaju, SSH klijent i poslužitelj komuniciraju putem TCP/IP protokola, a sve se odvija unutar složenog sustava mrežnih sučelja i portova. Naša su računala povezana putem fizičkih mrežnih adaptera, dok virtualni stroj koristi virtualne mrežne adaptore za komunikaciju s vanjskim svijetom. Sve to omogućuje nam da upravljamo našim VM-om kao da je lokalno računalo, unatoč tome što se zapravo nalazi u virtualiziranom okruženju.

Na jednak način kao što smo se povezivali s VM-om koji se izvodi u VirtualBoxu, možemo se povezati i s **bilo kojim drugim udaljenim poslužiteljem koji podržava SSH protokol** i za koji imamo potrebne **autorizacijske ovlasti**. Ovo je posebno korisno za administraciju strojeva koji su u oblaku, gdje možemo upravljati svojim resursima putem SSH veze.

Također, prisjetimo se naše tvrtke s početka ove skripte. Recimo da su ipak odlučili koristiti Cloud okruženje za postavljanje svoja 3 virtualna stroja. U tom slučaju, svaki od njih će imati svoju javnu IP adresu (obzirom da je u Cloudu) i, ako je tako podešeno, svi mogu biti dostupni putem SSH veze.

U tom slučaju, našu situaciju možemo ilustrirati na sljedeći način:



Ilustracija komunikacije između našeg računala i VM-a u Cloudu putem SSH protokola

4. Zadaci za Vježbu 4

1. **Preuzmite i instalirajte** VirtualBox na svoje računalo.
2. **Preuzmite i instalirajte** Ubuntu Server (LTS verziju) te **izradite novi virtualni stroj**.
Tijekom instalacije Ubuntu Servera, za `username` unesite svoje ime i prezime u formatu: `ime.prezime`.
3. Nakon instalacije, **napravite screenshot početnog zaslona** Ubuntu Servera na kojem se jasno vidi vaše korisničko ime i naziv virtualnog stroja.

Za svaki od sljedećih zadataka napravite odgovarajući screenshot:

- Ažurirajte lokalnu listu dostupnih paketa i verzija te nadogradite sve pakete na najnovije verzije.
- Instalirajte paket `openssh-server`, pokrenite SSH poslužitelj i provjerite njegov status.
- Pronađite IP adresu virtualnog stroja i provjerite koji su mrežni portovi otvoreni.
Kako ćete provjeriti koji port koristi SSH poslužitelj?
- Povežite se na SSH poslužitelj putem SSH klijenta na dva načina:
 - korištenjem **NAT adaptera i port forwardinga**
 - korištenjem **Bridged adaptera**
- Kada uspostavite SSH vezu s poslužiteljem, **napravite screenshot terminala domaćina (SSH klijenta)**.

Putem domaćina, **izradite novu bash skriptu** unutar virtualnog stroja, u direktoriju `/home/username/` (zamijenite `username` svojim korisničkim imenom). Skripta treba napraviti **detaljan ispis svih datoteka** (uključujući skrivene) iz korijenskog direktorija VM-a (`/`).

Za ovaj zadatak također napravite screenshot i na VM-u i u terminalu domaćina.