CHE HAGGAI NGHA KANG

UBA24PP134

ITCS

EXERCISE 1

Cloud computing has transformed how software is developed and how organizations run. Of all the characteristics that define cloud computing, two stand out as the main characteristics related to software development: On-Demand Self-Service and Resource Pooling. They remove traditional barriers of hardware ownership and long maintaining cycles, enabling speed, scalability, and new economics of scale. examples from Netflix, and Airbnb illustrate how these characteristics influence both application design and corporate strategy.

On-Demand Self-Service This means that a developer or team can spin up servers, databases, or GPU clusters in seconds through a web console or API, without filing tickets or waiting for approval. That single feature has fundamentally changed how modern applications are designed.

Before the cloud, deploying a new feature could take weeks, given that hardware purchases and data-center rack space were required. Today, teams practice continuous deployment: code is pushed multiple times a day, and the infrastructure adjusts in real time. On Netflix, engineers deploy thousands of additional instances within minutes to handle the the options in viewer choices that gives them quick access to videos from anywhere. Without on-demand self-service, accessing Netflix from Cameroon will deliver slow services and even more costly.

From a business-model perspective, on-demand self-service turns capital expenditure (CAPEX) into operational expenditure (OPEX) and shifts risk away from the company. Airbnb's early infrastructure team was small yet they supported millions of listings worldwide. When demand became high during, engineers clicked a few buttons to add capacity; no board approval, no purchase orders. This agility let Airbnb compete with hotel chains that spend billions on physical properties. The same capability allows startups to achieve "unfair" speed: a two-person team can launch a global service that once required a large team and budget.

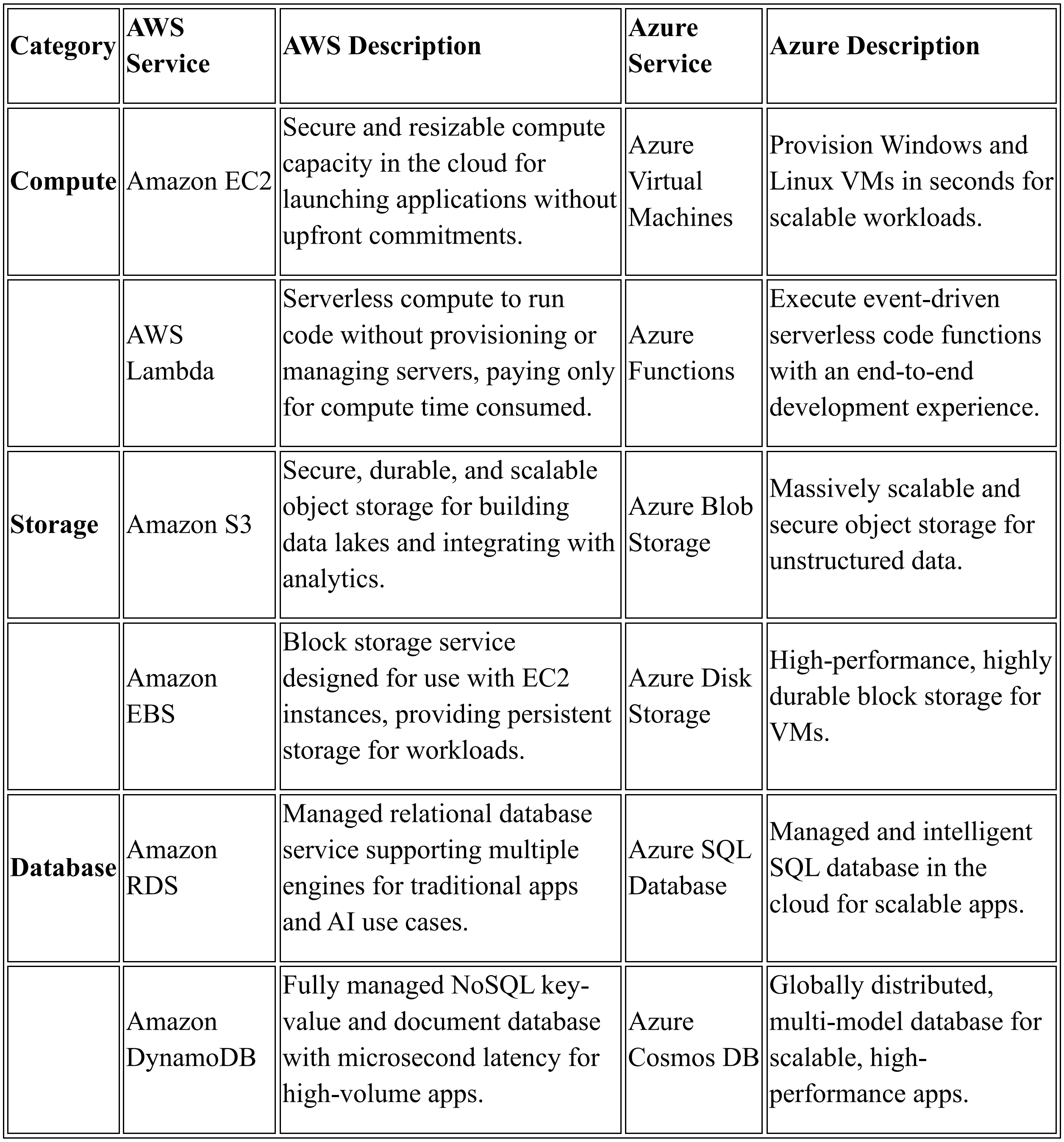
Also, Resource Pooling affects modern software development, Huge shared data centers are operated by the cloud provider and dynamically allocated virtual resources. The user never knows or cares what physical server their workload is running on. This pooling is what makes the economics work.

Pooling allows for true elasticity in application design. Spotify handles 70 million concurrent streams but pays only for the resourses it uses. The backend consists of thousands of containers powered by Google. When an artist drops a new album, in minutes, listenership can get resources from the shared pool in their Region faster and reliable

At the business-model level, resource pooling creates a pay-as-you-go utility that aligns cost directly with revenue. For example, Capital One shut eight data centers since moving to AWS, turning a $500 million a year data-center budget into a variable cloud bill that scales with transaction volume. When online transactions exploded in the 2020 pandemic, Capital One's infrastructure spend rose rather than requiring emergency capex. Today, the bank offers new digital products including an auto navigator tool that would have been expensive under the old fixed cost model.

In Conclusion, On-Demand Self-Service removes human efforts from provisioning, while Resource Pooling removes high economic from scale. Together they have turned cloud computing into the default platform for modern applications and the default operating model for competitive businesses.

EXERCISE 2



Category	AWS Service	AWS Description	Azure Service	Azure Description
Compute	Amazon EC2	Secure and resizable compute capacity in the cloud for launching applications without upfront commitments.	Azure Virtual Machines	Provision Windows and Linux VMs in seconds for scalable workloads.
	AWS Lambda	Serverless compute to run code without provisioning or managing servers, paying only for compute time consumed.	Azure Functions	Execute event-driven serverless code functions with an end-to-end development experience.
Storage	Amazon S3	Secure, durable, and scalable object storage for building data lakes and integrating with analytics.	Azure Blob Storage	Massively scalable and secure object storage for unstructured data.
	Amazon EBS	Block storage service designed for use with EC2 instances, providing persistent storage for workloads.	Azure Disk Storage	High-performance, highly durable block storage for VMs.
Database	Amazon RDS	Managed relational database service supporting multiple engines for traditional apps and AI use cases.	Azure SQL Database	Managed and intelligent SQL database in the cloud for scalable apps.
	Amazon DynamoDB	Fully managed NoSQL key-value and document database with microsecond latency for high-volume apps.	Azure Cosmos DB	Globally distributed, multi-model database for scalable, high-performance apps.

EXERCISE 3

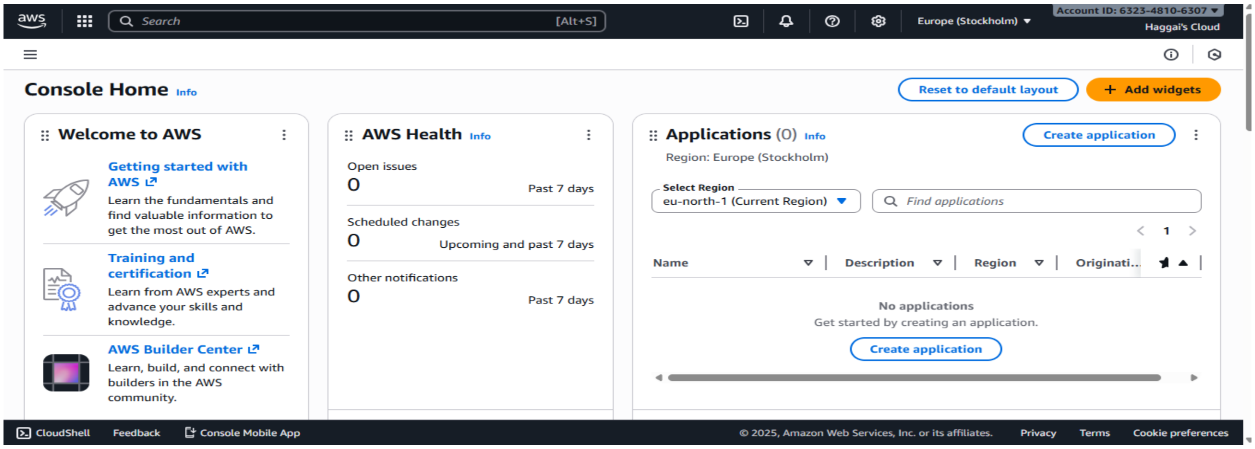
1. Creating a Free-Tier AWS Account

Steps Executed:

1. Visited aws.amazon.com → “Create an AWS Account”
2. Used email: haggairameni@gmail.com
3. Selected **Free Tier** plan
4. Added virtual card (no charge) – $1 hold refunded
5. Enabled MFA via authenticator app

Observation:

- Sign-up took **7 minutes**
- AWS automatically detected timezone (WAT) and suggested **af-south-1**
- Free tier banner visible throughout



Console Home Info

Welcome to AWS

Getting started with AWS
Learn the fundamentals and find valuable information to get the most out of AWS.

Training and certification
Learn from AWS experts and advance your skills and knowledge.

AWS Builder Center
Learn, build, and connect with builders in the AWS community.

AWS Health Info

Open issues 0 Past 7 days

Scheduled changes 0 Upcoming and past 7 days

Other notifications 0 Past 7 days

Applications (0) Info

Region: Europe (Stockholm)

Select Region: eu-north-1 (Current Region)

Find applications

Name	Description	Region	Originati…

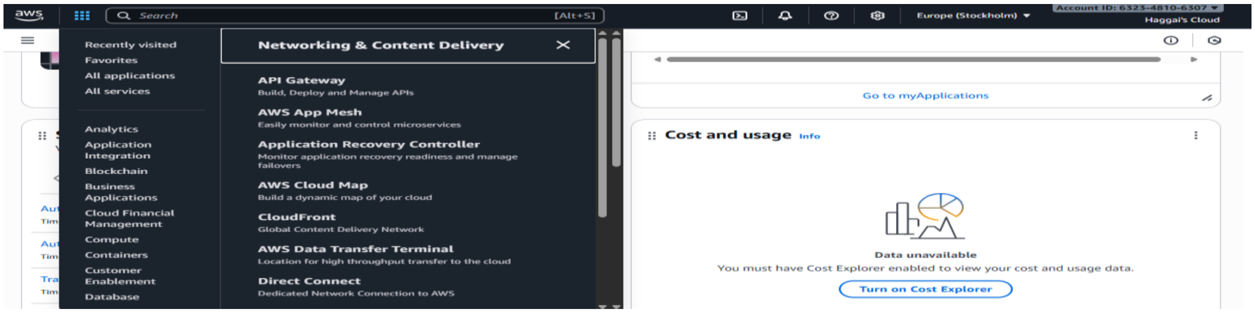
No applications
Get started by creating an application.

© 2025, Amazon Web Services, Inc. or its affiliates.

B. Exploring the AWS Management Console

2.1 Compute Section

Service Explored: Amazon EC2 **Observation:**



Networking & Content Delivery

API Gateway
Build, Deploy and Manage APIs

AWS App Mesh
Easily monitor and control microservices

Application Recovery Controller
Monitor application recovery readiness and manage failovers

AWS Cloud Map
Build a dynamic map of your cloud

CloudFront
Global Content Delivery Network

AWS Data Transfer Terminal
Location for high throughput transfer to the cloud

Direct Connect
Dedicated Network Connection to AWS

Cost and usage Info

Data unavailable
You must have Cost Explorer enabled to view your cost and usage data.

2.2 Storage Section

Service Explored: Amazon S3 **Observation:**

- Free tier: **5 GB** standard storage, 20,000 GET, 2,000 PUT requests
- Created bucket: cm-user-task-bucket-2025

2.3 Networking Section

Service Explored: Amazon VPC **Observation:**

- Default VPC auto-created per region
- Subnets in 3 Availability Zones (eu-north-1)

2.4 Security Section

Service Explored: AWS IAM **Observation:**

- Root user has full access – **avoid daily use**
- Created IAM user: task-user-cm with console access

3. Dashboard Overview (Account & Service Summary)

- **Billing:** $0.00 (Free tier)
- **Region:** eu-north-1
- **Trusted Advisor:** 0 errors
- **Recently Visited:** EC2, S3, VPC, IAM

Key Observations

Section	Free Tier Benefit	Ease of Use (1–5)	Notes
Compute	750 hrs t4g.micro	5/5	ARM instances cheaper
Storage	5 GB S3	5/5	Drag-and-drop upload
Networking	Default VPC + IGW	4/5	Subnets auto-spread
Security	IAM + MFA	4/5	Policy simulator useful

Total Time Spent: 35 minutes **Cost Incurred:** $0.00 (within free tier)

Exercise 4

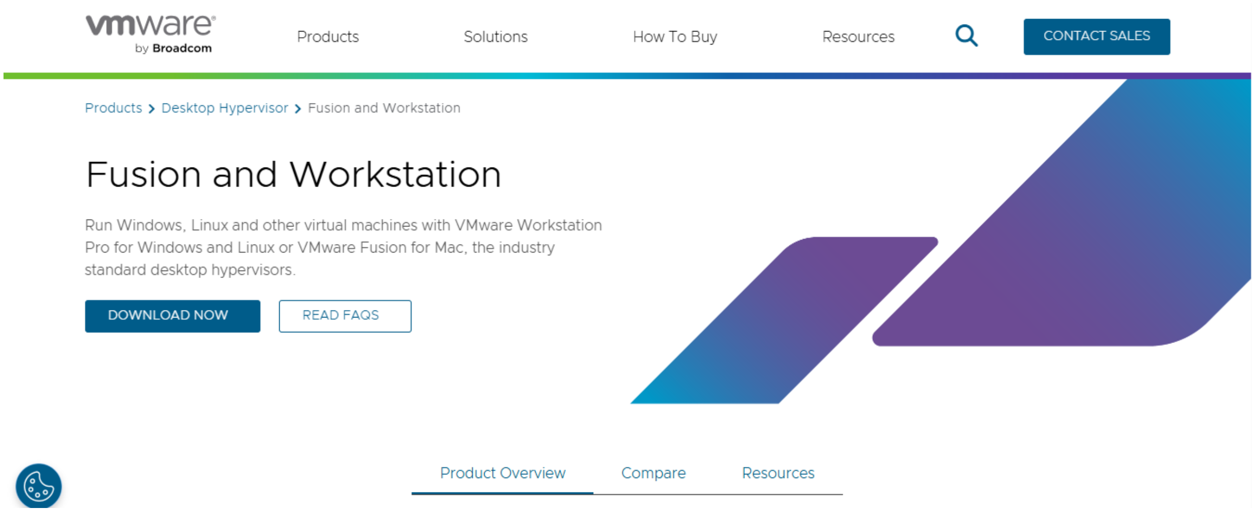
using VMware workstation player and KALI LINUX

1. Install VMware Workstation Player

- ➤ Download VMware Workstation 17 Player from:

https://www.vmware.com/products/workstation-player.html

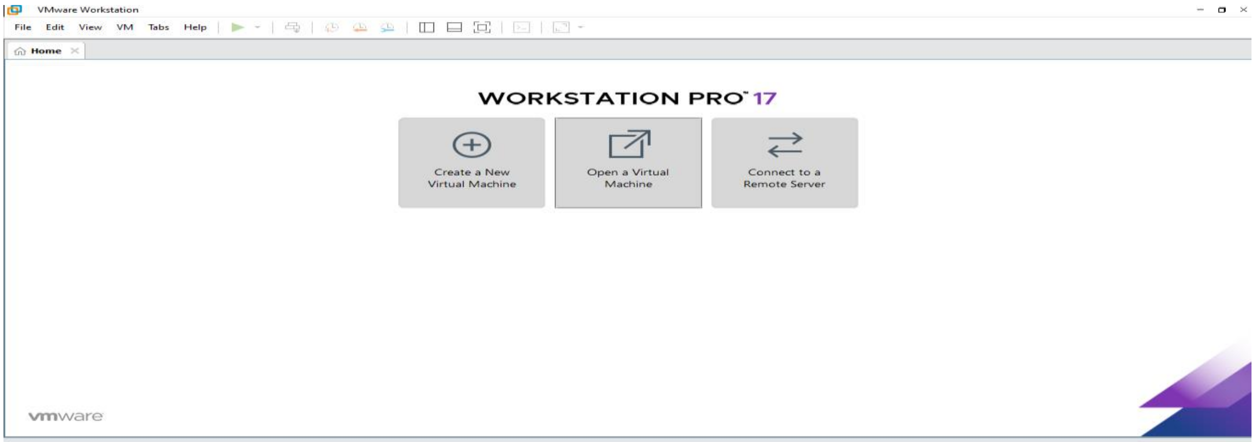
→ Choose Workstation 17 Player for Windows (or Linux host if applicable).



Fusion and Workstation

Run Windows, Linux and other virtual machines with VMware Workstation Pro for Windows and Linux or VMware Fusion for Mac, the industry standard desktop hypervisors.

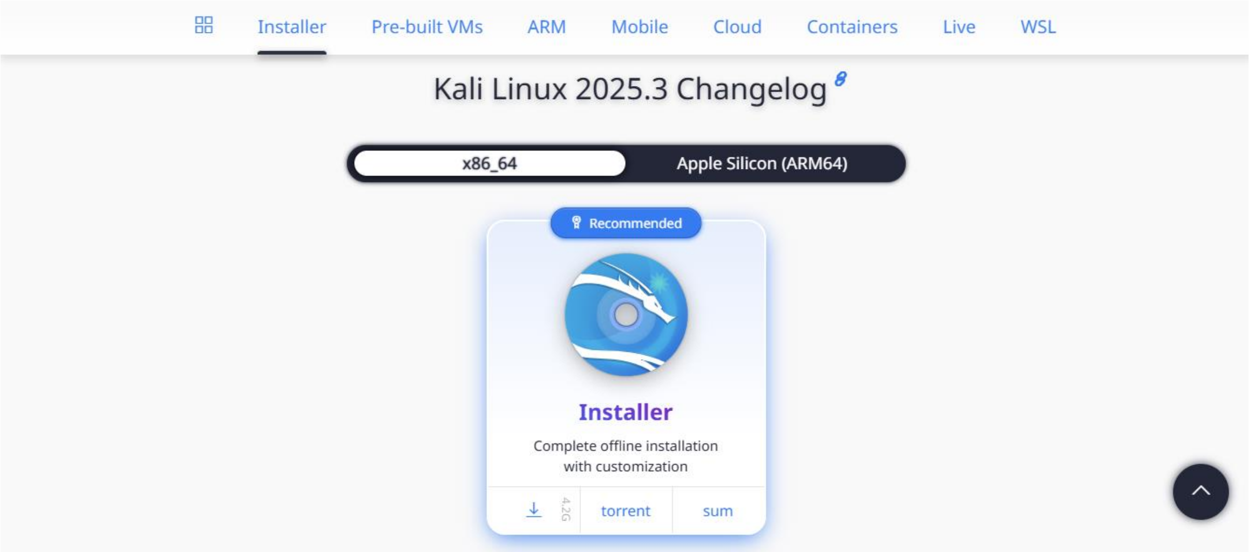
- ➤ Run the installer → Accept license → Typical install → Finish.
- ➤ .Launch VMware Workstation Player.



2. Download Kali Linux ISO

- ➢ Go to https://www.kali.org/get-kali/

→ Installer Images → 64-Bit → kali-linux-2025.3-installer-amd64.iso



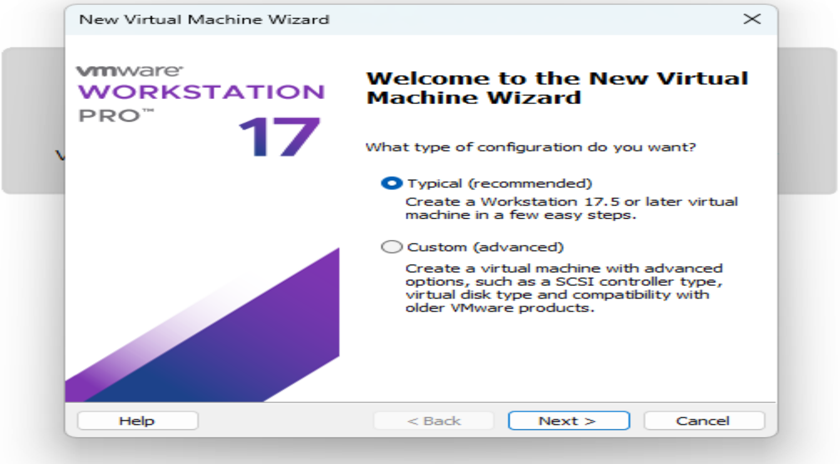
Kali Linux 2025.3 Changelog

Installer

Complete offline installation with customization

3. Create the Virtual Machine (Configuration & Resource Allocation)

- ➢ .In VMware Player → “Create a New Virtual Machine”.
- ➢ Choose “Installer disc image file (iso)” → Browse to the Kali ISO → Next.



Welcome to the New Virtual Machine Wizard

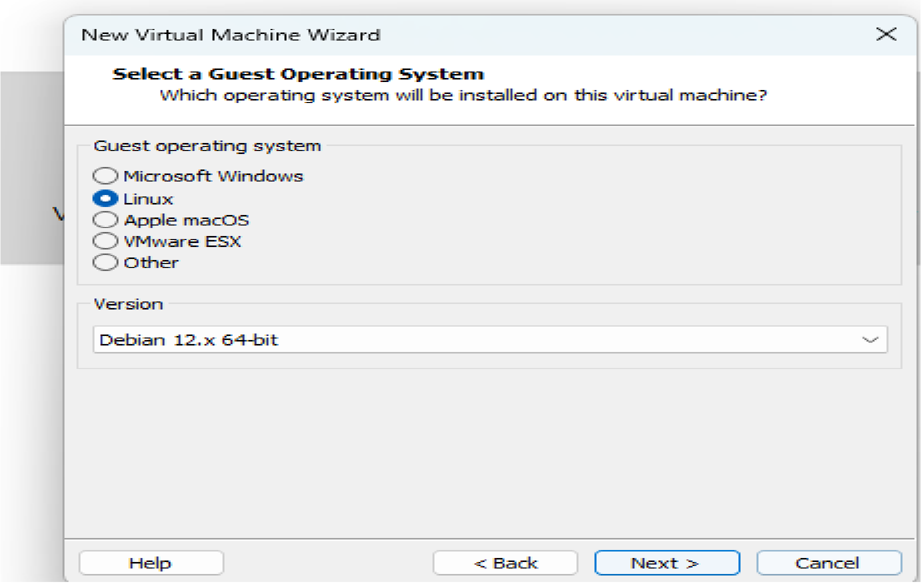
What type of configuration do you want?

● Typical (recommended)
Create a Workstation 17.5 or later virtual machine in a few easy steps.

○ Custom (advanced)
Create a virtual machine with advanced options, such as a SCSI controller type, virtual disk type and compatibility with older VMware products.

- ➢ Guest operating system

Select Linux → Version: Debian 12.x 64-bit (best match)



New Virtual Machine Wizard

Select a Guest Operating System

Which operating system will be installed on this virtual machine?

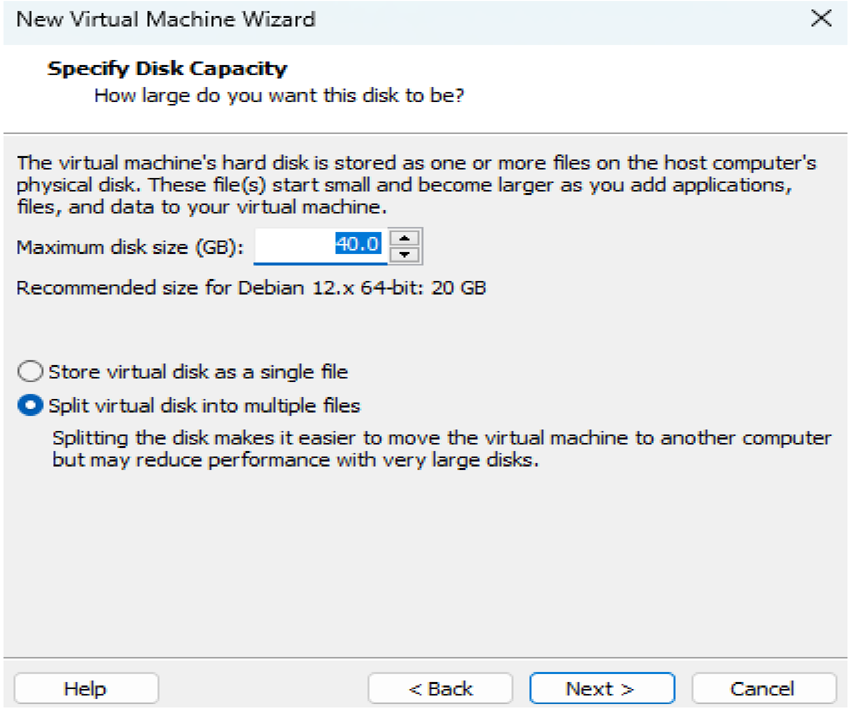
Guest operating system

○ Microsoft Windows
● Linux
○ Apple macOS
○ VMware ESX
○ Other

Version

Debian 12.x 64-bit

- ➢ Disk size:
  - Maximum disk size: 40 GB (recommended for Kali).
  - Select “Store virtual disk as a single file”.



New Virtual Machine Wizard

Specify Disk Capacity

How large do you want this disk to be?

The virtual machine's hard disk is stored as one or more files on the host computer's physical disk. These file(s) start small and become larger as you add applications, files, and data to your virtual machine.

Maximum disk size (GB): 40.0

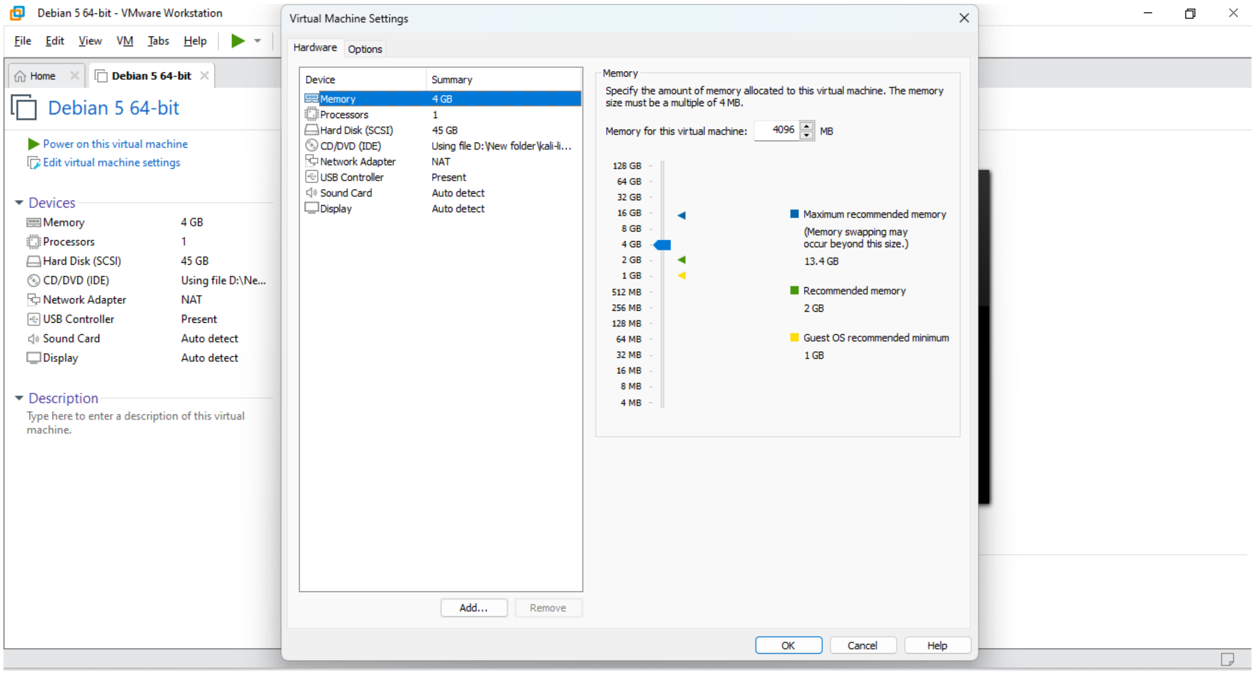
Recommended size for Debian 12.x 64-bit: 20 GB

○ Store virtual disk as a single file
● Split virtual disk into multiple files

Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

- Customize Hardware (critical step):

Resource	**Allocation**	**Reason**
Memory	4 GB (4096 MB)	Kali GUI needs ≥2 GB; 4 GB smooth
Processors	2 cores	Enables parallel installs
Network Adapter	NAT (default)	Internet for packages
USB Controller	Present (optional)	Not needed for text install
Display	Uncheck “Accelerate 3D graphics”	Text-mode install
CD/DVD	Already connected to ISO	Keep



Devices	
Memory	4 GB
Processors	1
Hard Disk (SCSI)	45 GB
CD/DVD (IDE)	Using file D:\Ne...
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Display	Auto detect

Type here to enter a description of this virtual machine.

Device	Summary
Memory	4 GB
Processors	1
Hard Disk (SCSI)	45 GB
CD/DVD (IDE)	Using file D:\New folder\kali-li...
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Display	Auto detect

Memory

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

Memory for this virtual machine: 4096 MB

🟦 Maximum recommended memory
(Memory swapping may occur beyond this size.)
13.4 GB

🟩 Recommended memory
2 GB

🟨 Guest OS recommended minimum
1 GB

- Click Finish. VM is created.

4. Boot into Text-Mode Installer (Terminal-Only)

- Select the VM → “Play virtual machine”.

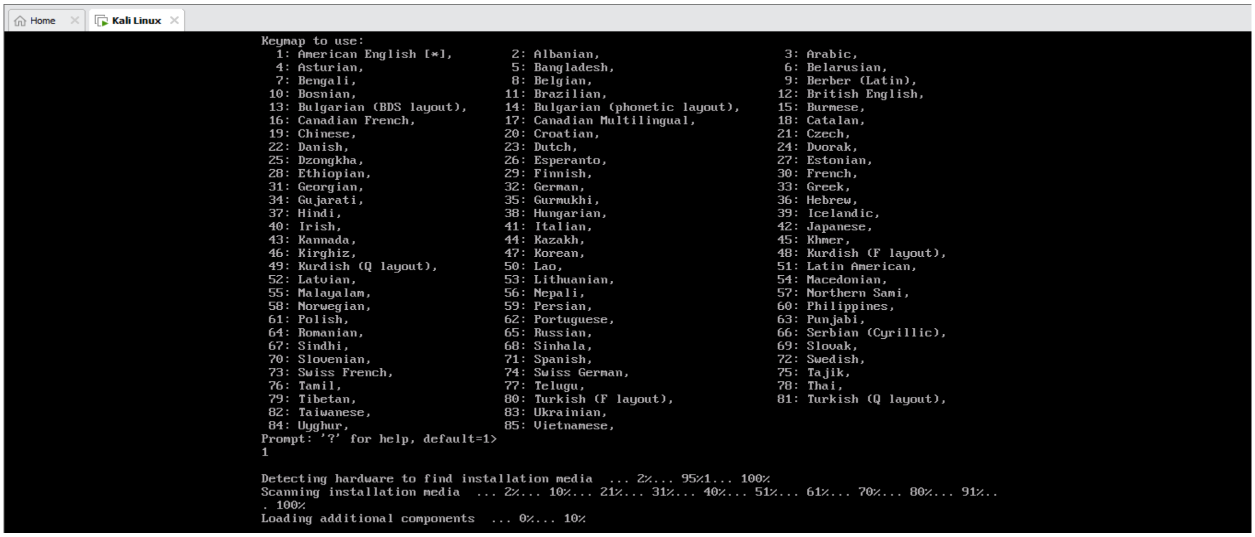
- Choose “Text-Mode Install or GUI install” (going with text mode install).

→ This starts Debian installer in pure terminal mode.

5. Kali Linux Terminal Installation (Step-by-Step)

> All following steps occur inside the VM console (black screen with white text).

- Language, Location, Keyboard
  - Language: English → Continue
  - Location: Cameroon
  - Keyboard: American English

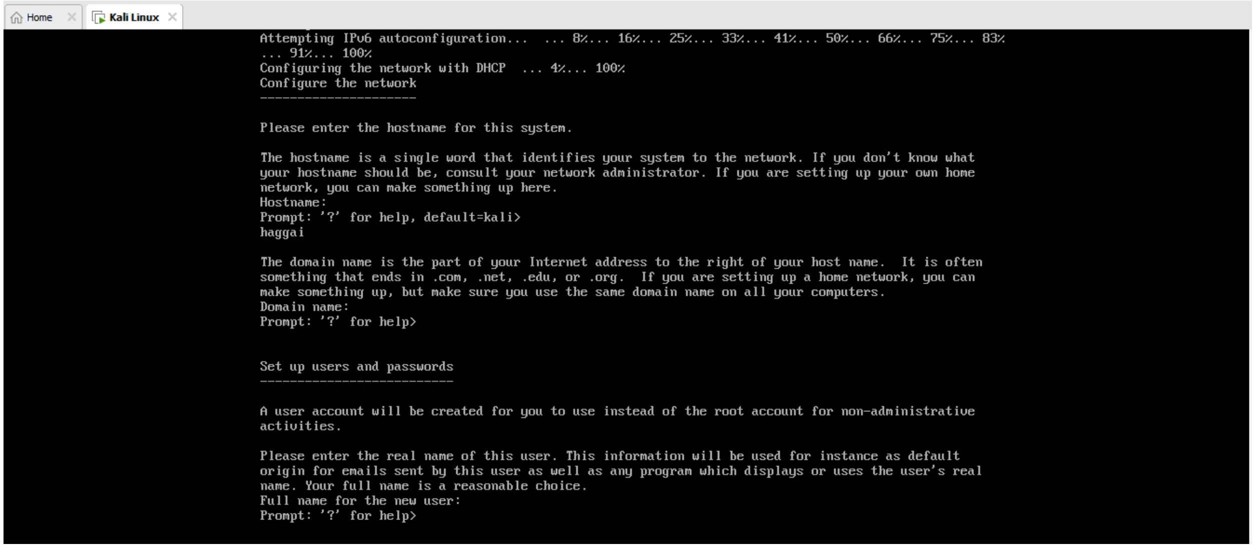


```
Keymap to use:
  1: American English [*],        2: Albanian,                      3: Arabic,
  4: Asturian,                    5: Bangladesh,                    6: Belarusian,
  7: Bengali,                     8: Belgian,                       9: Berber (Latin),
 10: Bosnian,                    11: Brazilian,                    12: British English,
 13: Bulgarian (BDS layout),     14: Bulgarian (phonetic layout),  15: Burmese,
 16: Canadian French,            17: Canadian Multilingual,        18: Catalan,
 19: Chinese,                    20: Croatian,                     21: Czech,
 22: Danish,                     23: Dutch,                        24: Dvorak,
 25: Dzongkha,                   26: Esperanto,                    27: Estonian,
 28: Ethiopian,                  29: Finnish,                      30: French,
 31: Georgian,                   32: German,                       33: Greek,
 34: Gujarati,                   35: Gurmukhi,                     36: Hebrew,
 37: Hindi,                      38: Hungarian,                    39: Icelandic,
 40: Irish,                      41: Italian,                      42: Japanese,
 43: Kannada,                    44: Kazakh,                       45: Khmer,
 46: Kirghiz,                    47: Korean,                       48: Kurdish (F layout),
 49: Kurdish (Q layout),         50: Lao,                          51: Latin American,
 52: Latvian,                    53: Lithuanian,                   54: Macedonian,
 55: Malayalam,                  56: Nepali,                       57: Northern Sami,
 58: Norwegian,                  59: Persian,                      60: Philippines,
 61: Polish,                     62: Portuguese,                   63: Punjabi,
 64: Romanian,                   65: Russian,                      66: Serbian (Cyrillic),
 67: Sindhi,                     68: Sinhala,                      69: Slovak,
 70: Slovenian,                  71: Spanish,                      72: Swedish,
 73: Swiss French,               74: Swiss German,                 75: Tajik,
 76: Tamil,                      77: Telugu,                       78: Thai,
 79: Tibetan,                    80: Turkish (F layout),           81: Turkish (Q layout),
 82: Taiwanese,                  83: Ukrainian,
 84: Uyghur,                     85: Vietnamese,
Prompt: '?' for help, default=1>
1

Detecting hardware to find installation media  ... 2%... 95%1... 100%
Scanning installation media  ... 2%... 10%... 21%... 31%... 40%... 51%... 61%... 70%... 80%... 91%..
. 100%
Loading additional components  ... 0%... 10%
```

- 5.2 Network Configuration
  - DHCP auto-config → Continue
  - Hostname: `kali-exercise`
  - Domain: leave blank

- 5.3 Root Password

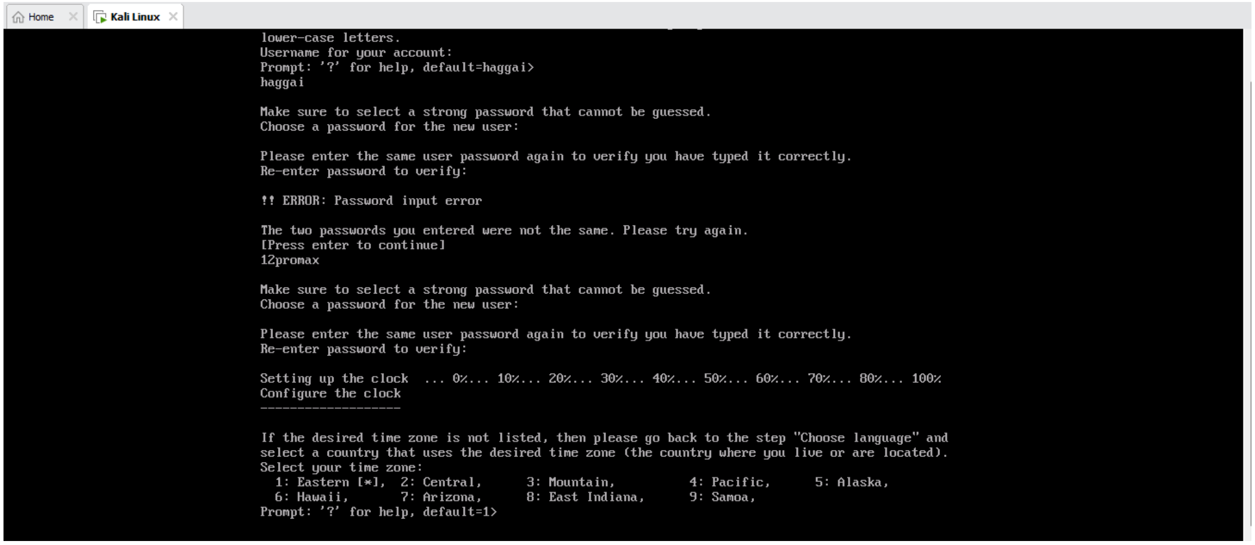
  - Set root password: `kaliroot2025` (or strong password)



```
Attempting IPv6 autoconfiguration...  ... 8%... 16%... 25%... 33%... 41%... 50%... 66%... 75%... 83%
... 91%... 100%
Configuring the network with DHCP  ... 4%... 100%
Configure the network
---------------------

Please enter the hostname for this system.

The hostname is a single word that identifies your system to the network. If you don't know what
your hostname should be, consult your network administrator. If you are setting up your own home
network, you can make something up here.
Hostname:
Prompt: '?' for help, default=kali>
haggai

The domain name is the part of your Internet address to the right of your host name.  It is often
something that ends in .com, .net, .edu, or .org.  If you are setting up a home network, you can
make something up, but make sure you use the same domain name on all your computers.
Domain name:
Prompt: '?' for help>


Set up users and passwords
--------------------------

A user account will be created for you to use instead of the root account for non-administrative
activities.

Please enter the real name of this user. This information will be used for instance as default
origin for emails sent by this user as well as any program which displays or uses the user's real
name. Your full name is a reasonable choice.
Full name for the new user:
Prompt: '?' for help>
```

➢ 5.4 User Account

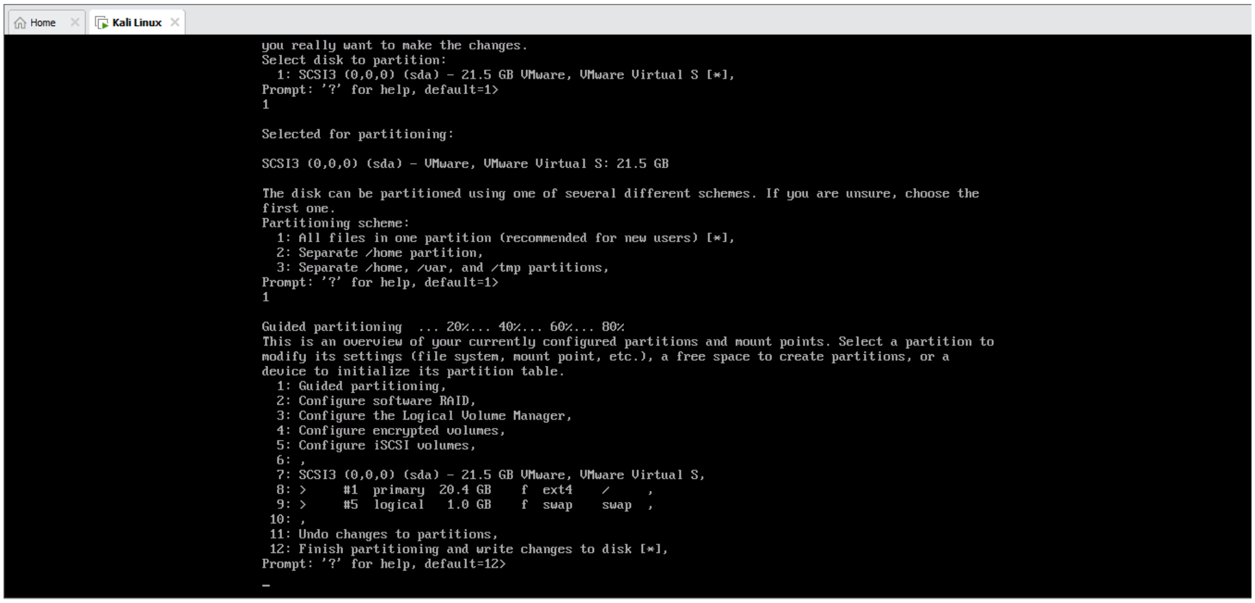
- Full name: `Haggai`
- Username: `haggai`
- Password: `******`



```
lower-case letters.
Username for your account:
Prompt: '?' for help, default=haggai>
haggai

Make sure to select a strong password that cannot be guessed.
Choose a password for the new user:

Please enter the same user password again to verify you have typed it correctly.
Re-enter password to verify:

!! ERROR: Password input error

The two passwords you entered were not the same. Please try again.
[Press enter to continue]
12promax

Make sure to select a strong password that cannot be guessed.
Choose a password for the new user:

Please enter the same user password again to verify you have typed it correctly.
Re-enter password to verify:

Setting up the clock  ... 0%... 10%... 20%... 30%... 40%... 50%... 60%... 70%... 80%... 100%
Configure the clock
-------------------

If the desired time zone is not listed, then please go back to the step "Choose language" and
select a country that uses the desired time zone (the country where you live or are located).
Select your time zone:
  1: Eastern [*],  2: Central,      3: Mountain,         4: Pacific,      5: Alaska,
  6: Hawaii,       7: Arizona,      8: East Indiana,     9: Samoa,
Prompt: '?' for help, default=1>
```

➢ Disk Partitioning

- Choose “Guided – use entire disk” → Continue
- Select the 40 GB virtual disk → Continue
- Scheme: “All files in one partition” → Continue
- “Finish partitioning and write changes” → Yes



```
you really want to make the changes.
Select disk to partition:
  1: SCSI3 (0,0,0) (sda) - 21.5 GB VMware, VMware Virtual S [*],
Prompt: '?' for help, default=1>
1

Selected for partitioning:

SCSI3 (0,0,0) (sda) - VMware, VMware Virtual S: 21.5 GB

The disk can be partitioned using one of several different schemes. If you are unsure, choose the
first one.
Partitioning scheme:
  1: All files in one partition (recommended for new users) [*],
  2: Separate /home partition,
  3: Separate /home, /var, and /tmp partitions,
Prompt: '?' for help, default=1>
1

Guided partitioning  ... 20%... 40%... 60%... 80%
This is an overview of your currently configured partitions and mount points. Select a partition to
modify its settings (file system, mount point, etc.), a free space to create partitions, or a
device to initialize its partition table.
  1: Guided partitioning,
  2: Configure software RAID,
  3: Configure the Logical Volume Manager,
  4: Configure encrypted volumes,
  5: Configure iSCSI volumes,
  6: ,
  7: SCSI3 (0,0,0) (sda) - 21.5 GB VMware, VMware Virtual S,
  8: >     #1  primary  20.4 GB    f  ext4    /     ,
  9: >     #5  logical   1.0 GB    f  swap    swap  ,
 10: ,
 11: Undo changes to partitions,
 12: Finish partitioning and write changes to disk [*],
Prompt: '?' for help, default=12>

_
```

➢ Package Manager & Mirror

- Use a network mirror: **Yes**
- Country: Choose closest (e.g., United States)
- Mirror: `http.kali.org`

➢ Software Selection

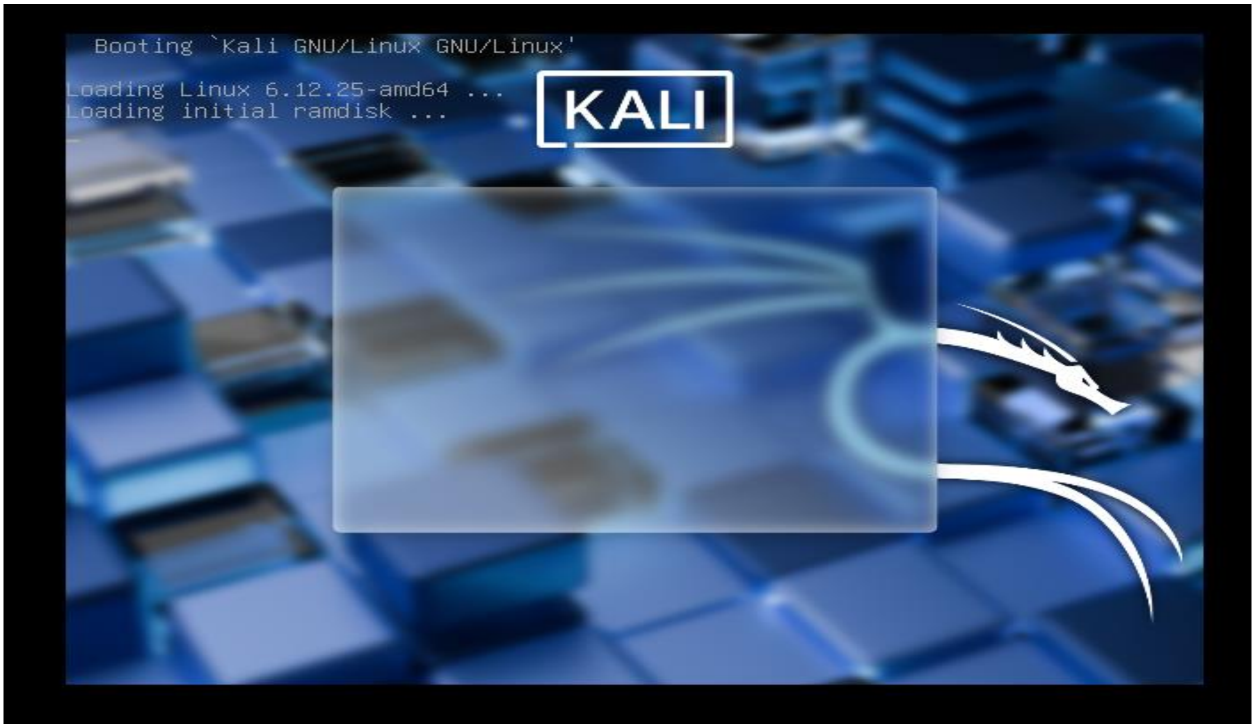
- Kali Linux desktop environment
- standard system utilities
- SSH server (optional but useful)

➢ Install GRUB Bootloader

- Install to /dev/sda → Yes

➢ Installation Complete

- “Continue” → VM reboots into terminal login prompt.



```
  Booting `Kali GNU/Linux GNU/Linux'
Loading Linux 6.12.25-amd64 ...
Loading initial ramdisk ...
_
```

7. Brief Reflection: How Virtualization Underpins Cloud Computing

Virtualization is the foundational layer of cloud computing. By abstracting physical hardware into virtual machines (VMs), a single physical server can host dozens of isolated guest OSes (multi-tenancy). In this exercise, VMware Workstation Player emulated CPU, RAM, disk, and network for Kali Linux exactly how hypervisors like VMware ESXi, KVM, or Hyper-V operate in data centers.

Cloud providers (AWS EC2, Azure VMs, GCP Compute Engine) run Type-1 hypervisors on bare metal, dynamically allocating vCPUs, memory, and storage from resource pools. Features demonstrated resource customization, snapshot capability (not shown but available), network NAT mirror cloud elasticity, isolation, and self-service provisioning.

Without virtualization, cloud scale, cost-efficiency, and rapid deployment would be impossible. This lab illustrates the core principle: one physical host → many logical servers.

Exercise 5

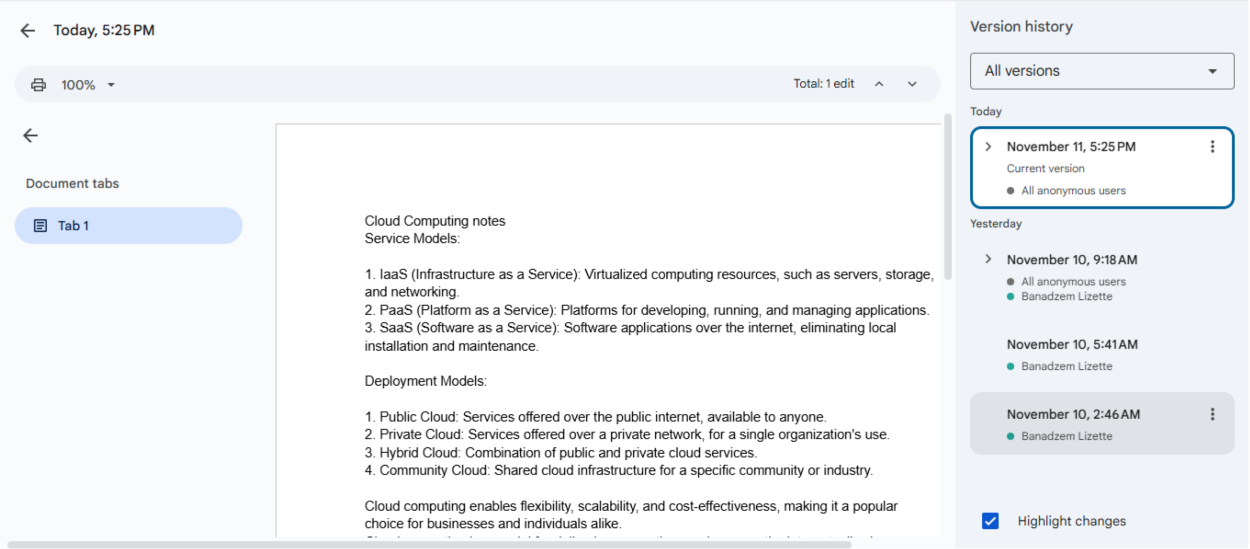
For this exercise, I interviewed Timo, a freelance web developer who has five years of experience building SaaS applications for small startups. Most of the time, Timo leverages AWS services-Lambda for serverless functions, S3 for storage, and EC2 for compute instances-sometimes augmented with Cloudflare for CDN and edge caching. He connects all of these to Vercel. He shared that the biggest benefits have been rapid scalability during traffic spikes, such as auto-scaling during product launches without manual server provisioning, and cost savings on idle resources compared to maintaining physical hardware. However, challenges include vendor lock-in, where migrating data between providers feels risky and time-consuming, unexpected billing surprises due to over-provisioning or data transfer fees, and cold start latencies in serverless setups frustrating user experience during low-traffic periods. These insights make it clear how such cloud tools accelerate development but require thoughtful planning to avoid pitfalls like bloated costs or debugging complexities.

The experiences Timo has directly include all the core characteristics of cloud computing discussed in class: on-demand self-service with AWS Lambda, where he can spin functions up instantly via the console without involving IT support; broad network access, where he has visually managed deployments from any browser around the world; resource pooling, where AWS shares the underlying infrastructure between clients, allowing him cost-effectively to store their data on S3 for multiple clients dynamically; rapid elasticity through EC2, quickly scaling instances up or down based on demand, which Alex credits with handling viral growth without downtime. However, measured services are modeled by per-usage billing, which introduces its challenges-rampant hidden costs due to unoptimized queries or egress fees, for instance, decorators highlight the importance of vigilant monitoring to avoid budget overruns.

This interview, in general, reinforced how cloud computing democratizes powerful infrastructure for developers such as Timo, who foster innovation in web applications but require a balance of enthusiasm with caution. It echoes class discussions on trade-offs: characteristics such as elasticity drive efficiency, but real world frictions termed lock in remind us that clouds are tools, not panaceas. Finally, going forward, I would recommend incorporating cost-optimization tools early in projects for maximum benefit.

EXERCISE 6 OPTIONAL

https://docs.google.com/document/d/1q43-XeF1cP_cmnCsxC26-HLSuINuSeFtVCJQIByd5e0/edit?usp=drivesdk



Cloud Computing notes
Service Models:

1. IaaS (Infrastructure as a Service): Virtualized computing resources, such as servers, storage, and networking.
2. PaaS (Platform as a Service): Platforms for developing, running, and managing applications.
3. SaaS (Software as a Service): Software applications over the internet, eliminating local installation and maintenance.

Deployment Models:

1. Public Cloud: Services offered over the public internet, available to anyone.
2. Private Cloud: Services offered over a private network, for a single organization's use.
3. Hybrid Cloud: Combination of public and private cloud services.
4. Community Cloud: Shared cloud infrastructure for a specific community or industry.

Cloud computing enables flexibility, scalability, and cost-effectiveness, making it a popular choice for businesses and individuals alike.