Phase 1

⭐ *Capacity (Agility, Elasticity, Scalability)*

- **Capacity** = How much your system can handle.
- **Agility** = How quickly you can adjust resources.
- **Elasticity** = Ability to **grow or shrink automatically** based on demand.
- **Scalability** = Ability to **handle more load** without breaking.

Example:
Imagine a website:

- On weekdays, 50 users visit → one small server is enough.
- On weekends, 5000 users visit → you add more servers automatically (elasticity).

⭐ *Availability*

- **Availability** = Your service stays **up and running**, even if something goes wrong.
- Think of it like a **backup plan** for failure.

Example:

- You have a website hosted in one data center. If that data center goes down, your site goes offline → low availability.

- If you have two data centers and traffic switches to the second when the first fails → high availability.

⭐ *Blast Radius*

- **Blast radius** = How much damage happens if something fails.
- Smaller blast radius → less impact when failure happens.

Example:

- One giant server crashes → your entire app goes down → huge blast radius.
- Many small servers → if one crashes, only part of your app is affected → small blast radius.

⭐ *Disaster Recovery*

- **Disaster recovery** = How quickly you can **restore your service after a big failure**.

Example:

- Your data center floods → you switch to another region and your website is back online.

⭐ *Vertical Scaling vs Horizontal Scaling*

Vertical Scaling (Scale Up)

- Add **more power to one server** (CPU, memory).
- **Problem:**
    - Often requires **downtime** to upgrade.
    - Not all apps can handle huge servers.
- **Example:**
    - Upgrade a small server from 2 CPUs → 16 CPUs → your server must restart.
    - If it crashes during upgrade → downtime.

Horizontal Scaling (Scale Out)

- Add **more servers** instead of making one bigger.
- **Better for cloud** → handles failure and traffic smoothly.
- **Example:**
    - Your website has 2 small servers → traffic increases → add 3 more servers automatically.
    - Traffic decreases → remove 1 server.
    - You **never go below 2 servers** to ensure availability.

Rule of thumb in cloud: Horizontal scaling is safer, more flexible, and keeps your service always available.

✅ Simple Analogy

- **Vertical Scaling =** One big pizza → hard to eat, and if it burns, you lose all.

- **Horizontal Scaling** = Many small pizzas → easy to share, if one burns, others are still fine.

⭐ *Capital Expenditure (CapEx)*

- **What it is:** Buying physical resources upfront, which you own and depreciate over time.
- **Example in the real world:** Buying servers, storage devices, or networking equipment for your company's on-premises data center.
- **Simple analogy:** Like buying a car—you pay upfront, and it's yours for years.

In cloud context (Phase 1 / Azure labs):

- On-premises equivalent: If you wanted to practice Azure security but instead bought your own physical servers, firewalls, and networking gear to test labs.
- You spend **a lot upfront**, but you **own the hardware**.

⭐ *Operational Expenditure (OpEx)*

- **What it is:** Paying only for what you use, usually subscription or consumption-based. No big upfront cost.
- **Example in the real world:** Using a streaming service like Netflix—you pay monthly only for what you watch.
- **Simple analogy:** Like renting a car—you pay only when you drive it.

In cloud context (Phase 1 / Azure labs):

- Using Azure free-tier or pay-as-you-go services for VMs, Storage, Key Vault, and AKS in your labs.
- You **pay based on usage**, scale up or down, and don't need to buy servers or networking devices upfront.
- Ideal for learning: You can practice security labs **without spending thousands on hardware**.

Key difference (super simple)

CapEx	OpEx
Buy and own (servers, devices)	Rent or pay-as-you-go (Azure services)
High upfront cost	Low/no upfront cost
Fixed capacity (limited by hardware)	Flexible capacity (scale up/down anytime)
Depreciates over time	Costs treated as ongoing expense

⭐ *What does "responsibility" mean in cloud computing?*

In cloud computing, **responsibility** means:

👉 *Which tasks you (the customer) must manage*
vs.
👉 *Which tasks the cloud provider (Azure, AWS, GCP) manages*

This is called the **shared responsibility model**.

Simple explanation with examples

1. Example: Using a Virtual Machine (VM)

Your responsibility:

- Install updates
- Secure passwords
- Install antivirus
- Configure firewall rules
- Manage who can log in

Cloud provider's responsibility:

- Physical servers
- Data center buildings
- Power, cooling, hardware

✔ If your VM gets hacked because you didn't update it → **your responsibility**
✔ If a physical server in Azure breaks → **Azure's responsibility**

---

2. Example: Using cloud storage

Your responsibility:

- Decide who can access the files
- Encrypt sensitive data if needed
- Set access policies correctly

Cloud provider's responsibility:

- Keep storage hardware running
- Ensure data is replicated/redundant
- Fix hardware failures

✔ If you accidentally share a storage container with "Public access = ON" → **your responsibility**
✔ If a disk in the data center physically breaks → **cloud provider's responsibility**

---

3. Example: Using cloud networking

Your responsibility:

- Create secure network rules
- Close unused ports
- Create strong access policies

Cloud provider's responsibility:

- Maintain routers, switches, cables in the data center
- Ensure global network connectivity

✔ If you leave port 22 open to the internet → **your responsibility**
✔ If an internet cable inside the Azure data center fails → **Azure handles it**

---

4. Example: Using databases in the cloud

If you use a fully managed database (like Azure SQL):

Your responsibility:

- Protect your data
- Control who can access the database
- Configure firewalls

Cloud provider's responsibility:

- Patch the database engine
- Maintain servers
- Handle automatic backups (depending on service)

✔ If someone steals your data because permissions were too open → **your responsibility**

Super simple summary

Responsibility in cloud computing = what YOU must secure, configure, and manage vs. what the provider handles.

Cloud providers handle the **physical stuff**.
You handle the **things you run inside the cloud**.

⭐ *What you manage in IaaS(Infrastructure-as-a-Service)*

When you use **IaaS**, you are renting the *infrastructure* from Azure — like servers, storage, and networking — but **you are still responsible for what happens inside your virtual machine and inside your cloud environment**.

Think of it like renting an empty apartment:

- Azure = landlord (manages the building)

- You = tenant (manage everything inside the apartment)

Here are the responsibilities in **simple words + examples**:

✅ What YOU manage (your responsibility)

1. Operating System

You decide:

- Windows or Linux
- Update it
- Patch vulnerabilities

Example:
If your Linux VM has a security update available, *you must install it*.

---

2. Applications

You control:

- What applications you install
- How they run
- Their security settings

Example:
If you install NGINX on your VM, *you configure it, secure it, and update it*.

---

3. Security settings

You must set:

- Firewalls inside the VM
- Anti-malware
- Correct permissions

Example:
If someone can SSH into your VM because your password is weak, **that's your responsibility**, not Azure's.

---

4. Network rules

You configure:

- NSGs (Network Security Groups)
- Subnets
- Routing rules

Example:
If port 3389 (RDP) is accidentally left open to the world, **you fix it**.

---

5. Identity and Access

You handle:

- Who can access your VM
- Role assignments
- Permissions

Example:
If you give admin rights to someone who shouldn't have them, **that's on you**, not Azure.

---

6. Monitoring

You must set up:

- Logs
- Alerts
- Health checks

Example:
If you want alerts when someone logs into your VM at 3 AM, **you must configure that** through Monitor or Sentinel.

❌ What Azure manages (NOT your responsibility)

Azure takes care of:

- Physical servers
- Data center buildings
- Power & cooling
- Internet connections in data centers
- Disk failures (hardware)
- Racks, cables, network devices

Example:
If a physical server in Azure breaks, **Azure replaces it**, not you.
