

**On-Premises–Definition**  
 On-premises (often written “on-prem”) refers to IT infrastructure—such as servers, storage, and software—that is installed and operated within an organization’s own physical location, rather than being hosted in the cloud. The organization owns, maintains, and manages all the hardware and software in-house.

Example:

In an on-premises setup, the company’s servers are physically located in its own office or data centre, and the IT team is responsible for their maintenance.

**Cloud Computing–Definition**

A style of computing in which scalable and elastic IT-enabled capabilities are delivered as a service using Internet technologies.

--Gartner

Example:

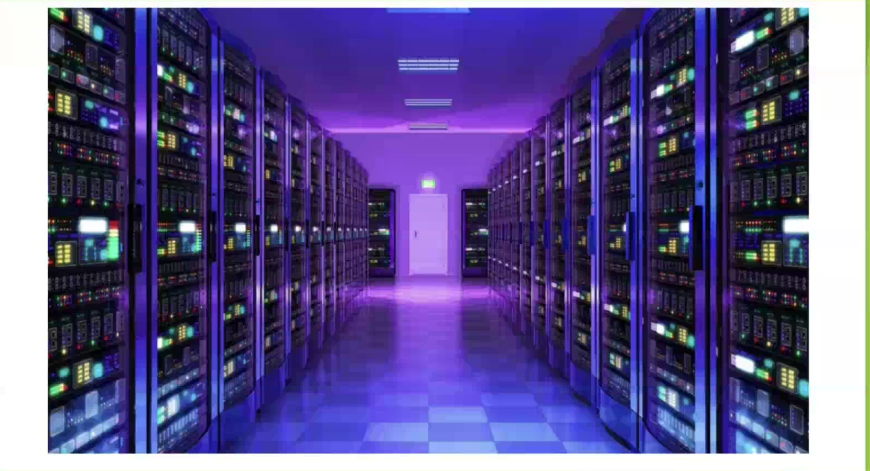
Storing photos in **Google Drive** instead of your computer’s hard drive is cloud computing.  
You can access them anytime, anywhere, without worrying about storage space or server maintenance.

**Just like Google offers Google Drive, Apple offers iCloud, and Microsoft offers OneDrive, on-premises solutions are hosted locally rather than in the cloud.**

**Or**

**Google has Google Drive, Apple has Dropbox, and Microsoft has OneDrive — but in an on-premises setup, storage stays inside the company’s own servers.**

****

****

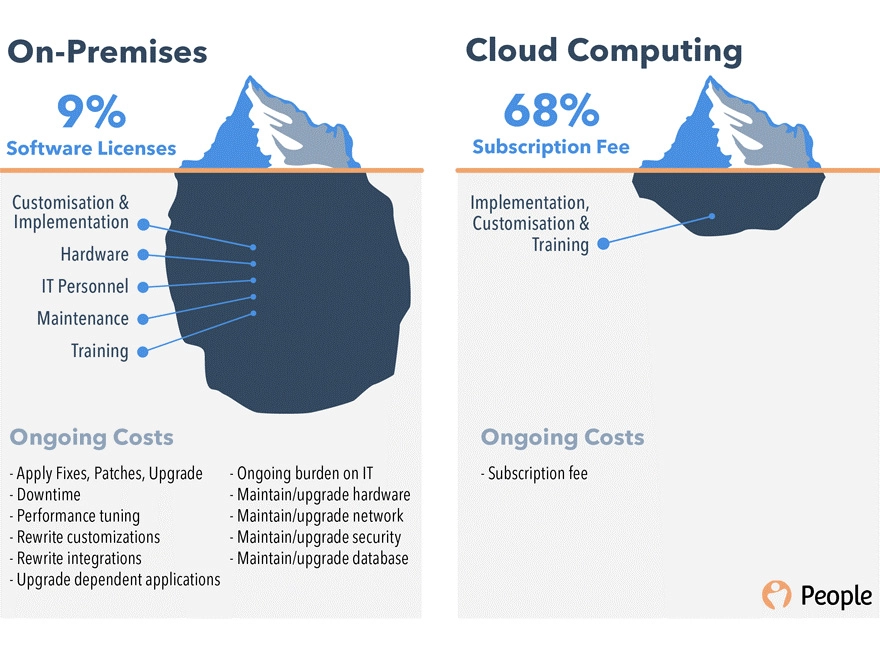
* **This image shows a server room or data centre, which is an example of an on-premises setup.**

In on-premises computing, the organization owns and manages these physical servers in its own facility. They are responsible for:

* Purchasing hardware
* Maintaining equipment
* Ensuring power, cooling, and security
* Upgrading when capacity needs grow

In contrast, cloud computing stores and manages data in third-party providers’ data centres (like AWS, Azure, or Google Cloud), so you don’t have to own or maintain such server rooms yourself.

**On-Premises VS Cloud Computing:**

****

**Cloud Computing – Scalability and Elasticity:**

**EXAMPLE – TaxSmile**

1. Scalability

* Definition: The ability of a system to handle an increased workload by adding resources.
* Example case:
  + On-Premises (Feb): TaxSmile tax submission portal starts with 2 servers.
  + Load & Stress Testing (Apr): Need for 3 more servers identified. (Due to traffic on peak months)
  + Scaling in on-premises requires: (Challenges of on-perm)
    1. Approvals from stakeholders (Directors, COO, CTO).
    2. Placing orders with a vendor (Germany).
    3. Retrofitting servers by network engineers (24/7, 3 teams + 1 team).
    4. Additional costs (room rent, power supply).
  + This process is time-consuming and involves capital expenses (CapEx).

2. Elasticity

* Definition: The ability to automatically increase or decrease resources based on demand.
* Example case:
  + Cloud Solution (Feb): TaxSmile on Azure with 2 servers.
  + When needed, resources can be increased or decreased instantly through a portal.
    - Example: Add a text box, increase from 2 servers to more, save changes — done in 1 minute.
  + Billing is based on actual usage (operational expenses – OpEx).

3. On-Premises vs Cloud Approach

|  |  |  |
| --- | --- | --- |
| **Aspect** | **On-Premises** | **Cloud (Azure)** |
| Initial Setup | Physical servers | Virtual servers |
| Scaling Time | Weeks/Months (ordering, shipping, setup) | Minutes |
| Costs | High upfront CapEx | Pay-as-you-go OpEx |
| Resource Adjustment | Manual & slow | Automated & instant |
| Downtime Risk | Higher (hardware constraints) | Lower (redundancy) |

4. Seasonal Demand Example

* Apr–Aug: Traffic decreases after tax season.
* On-premises: Extra servers remain idle (waste of resources).
* Cloud: Can scale down to save costs.

5. Benefits of Cloud Scalability & Elasticity

* Faster deployment of additional resources.
* Reduced operational complexity.
* Cost efficiency (scale down during low demand).
* Global access to vendor services.

**Computing Service Models:**

On-Premises + IaaS + PaaS + SaaS together cover the whole range of computing service models, from fully self-managed to fully provider-managed.

**1. IaaS – Infrastructure as a Service**

**Meaning:**

* Think of it as *renting raw computer resources* over the internet.
* You get **servers, storage, and networking** on demand — but you manage the operating system, applications, and data yourself.

**What you manage:**

* OS, applications, runtime, and data.
* The provider manages physical servers, networking, and virtualization.

**Example scenario:**

* You want to build a custom website and choose everything yourself (OS, programming language, DB, etc.).
* Instead of buying servers, you rent them from a provider.

**Examples:**

* Amazon Web Services (AWS) EC2
* Microsoft Azure Virtual Machines
* Google Compute Engine

**Analogy:**

* Like renting an empty house — you bring your own furniture, paint, and appliances.

**2. PaaS – Platform as a Service**

**Meaning:**

* You get a *ready-made platform* with OS, runtime, and tools to develop, run, and manage applications.
* You don’t worry about infrastructure or OS setup — you just focus on your code.

**What you manage:**

* Only your applications and data.
* Provider manages OS, middleware, runtime, and infrastructure.

**Example scenario:**

* You want to develop an app but don’t want to handle OS updates, server setup, or database management.
* PaaS gives you a workspace to code and deploy quickly.

**Examples:**

* Google App Engine
* Microsoft Azure App Service
* Heroku

**Analogy:**

* Like renting a fully furnished apartment — you just bring your clothes and start living.

**3. SaaS – Software as a Service**

**Meaning:**

* You get *ready-to-use software* that runs on the provider’s servers.
* No installation, no updates, no infrastructure worries — you just use it via browser or app.

**What you manage:**

* Nothing technical — just your data and how you use the software.

**Example scenario:**

* You want an email service or CRM without worrying about servers, storage, or security.

**Examples:**

* Gmail, Google Drive
* Microsoft Office 365
* Dropbox
* Salesforce

**Analogy:**

* Like booking a hotel room — everything is ready; you just check in and use it.

**IAAS AND PAAS:**

**To deploy:**

1. Install Python
2. Install any other extensions
3. Install IIS or Tomcat or web sphere (servers)
4. Set Env variables
5. Take your website and deploy to Tomcat – make it public

Until this it is Iaas…,

Whereas, in paas, Give the website to Azure (Providing platform), the above 5 steps are taken care of azure which reduces the headache.

