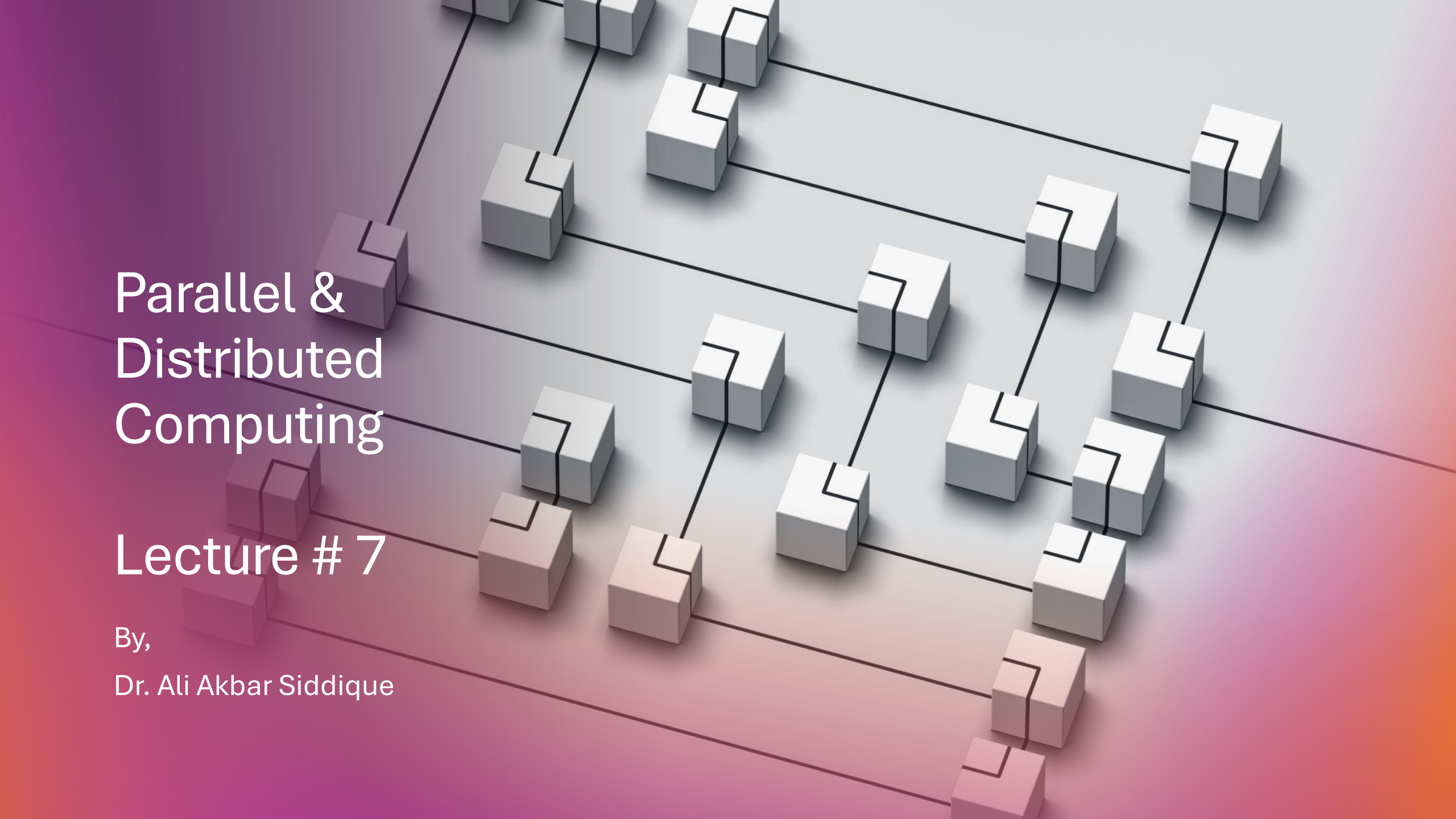


Parallel & Distributed Computing

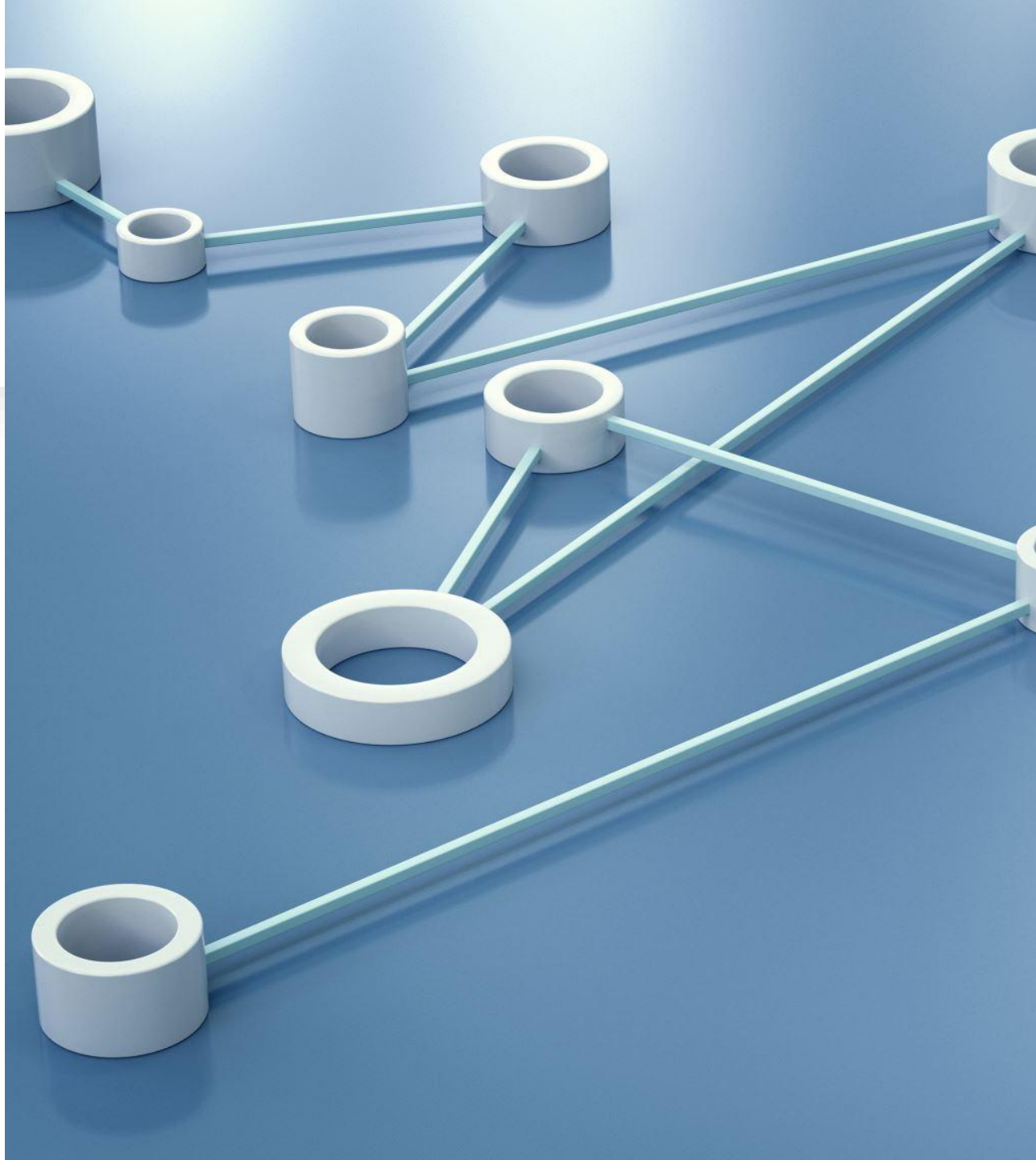
Lecture # 7

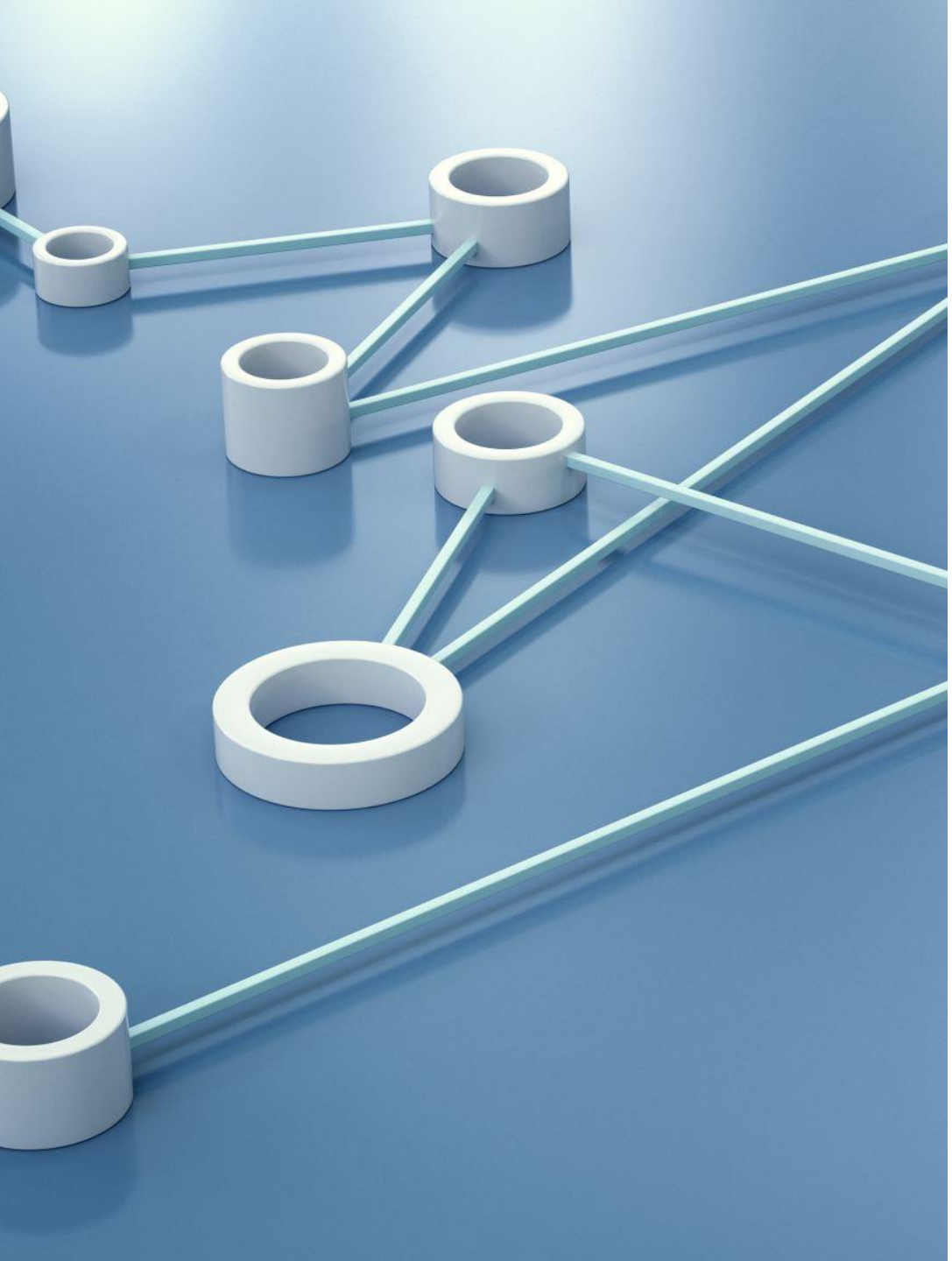
By,

Dr. Ali Akbar Siddique

What is Cloud Computing and Distributed Frameworks?

- Cloud Computing: Cloud computing is a model that provides on-demand access to computing resources (like servers, storage, databases, networking, software, and analytics) over the internet. It allows users to leverage powerful resources without owning the physical infrastructure.
- Distributed Frameworks: Distributed frameworks are software architectures that allow multiple computers (nodes) to work together on a single task. These frameworks are designed to enable parallel processing, efficient resource utilization, and fault tolerance.

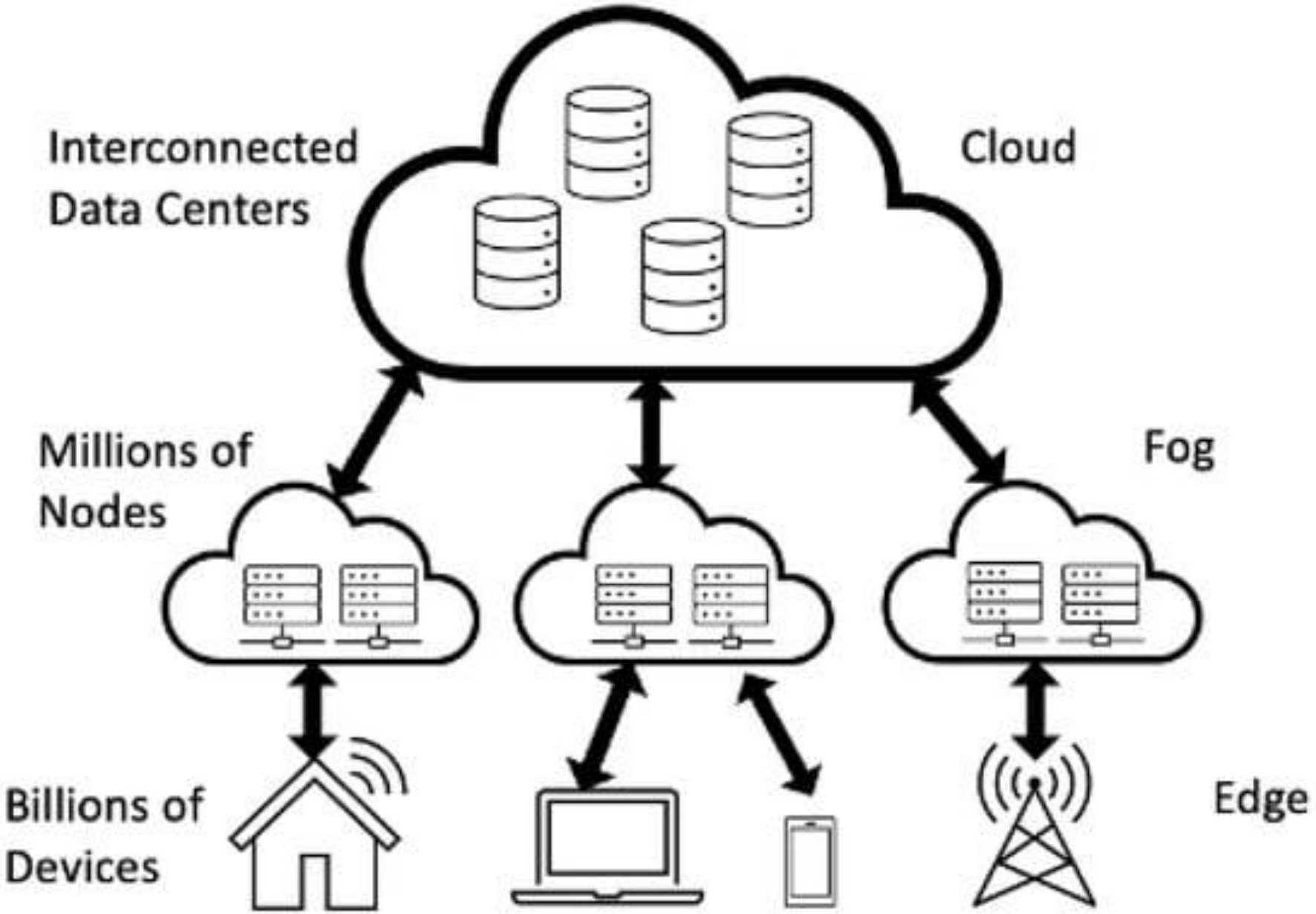




Why Study Cloud Computing and Distributed Frameworks?

- Cloud computing and distributed frameworks are the backbone of modern technology, powering everything from social media and streaming services to data analytics and artificial intelligence.
- Understanding these technologies is crucial for anyone interested in high-performance computing, big data processing, and scalable software development.

Cloud Computing



Key Characteristics of Cloud Computing

1. **On-Demand Self-Service:**
   - Users can provision computing resources (like virtual machines, storage, and applications) without requiring human intervention.
   - Example: Users can deploy a virtual server on Amazon Web Services (AWS) with a few clicks.

2. **Broad Network Access:**
   - Cloud services are accessible over the internet through various devices, including laptops, smartphones, and tablets.
   - Example: Accessing Google Drive files from any device.

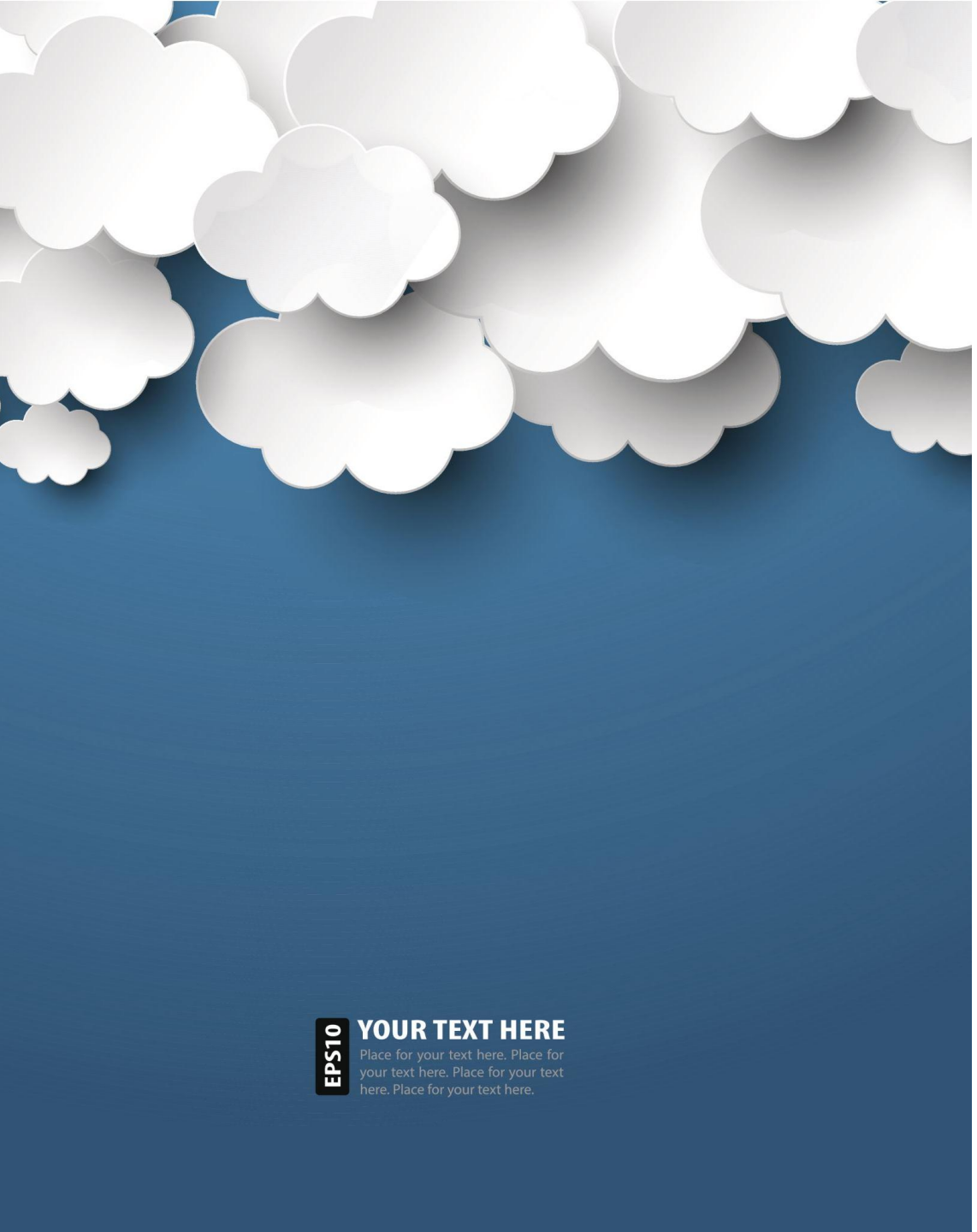
3. **Resource Pooling:**
   - Cloud providers use multi-tenant models to serve multiple customers using the same physical resources.
   - Example: Microsoft Azure uses a shared pool of servers to provide computing power to multiple users.

4. **Rapid Elasticity:**
   - Resources can be quickly scaled up or down depending on user demand.
   - Example: Netflix scales its server capacity during peak hours to support more users.

5. **Measured Service:**
   - Resource usage is monitored and billed on a pay-as-you-go or subscription basis.
   - Example: Users are billed for the amount of cloud storage or computing power they consume.

How Cloud Computing Works:

- **User Interaction**: Users access cloud services through a web browser or a dedicated application.
- **Service Delivery**: Cloud providers manage and maintain the underlying infrastructure.
- **Resource Allocation**: Virtual resources are allocated based on user requirements.



Types of Cloud Environments:

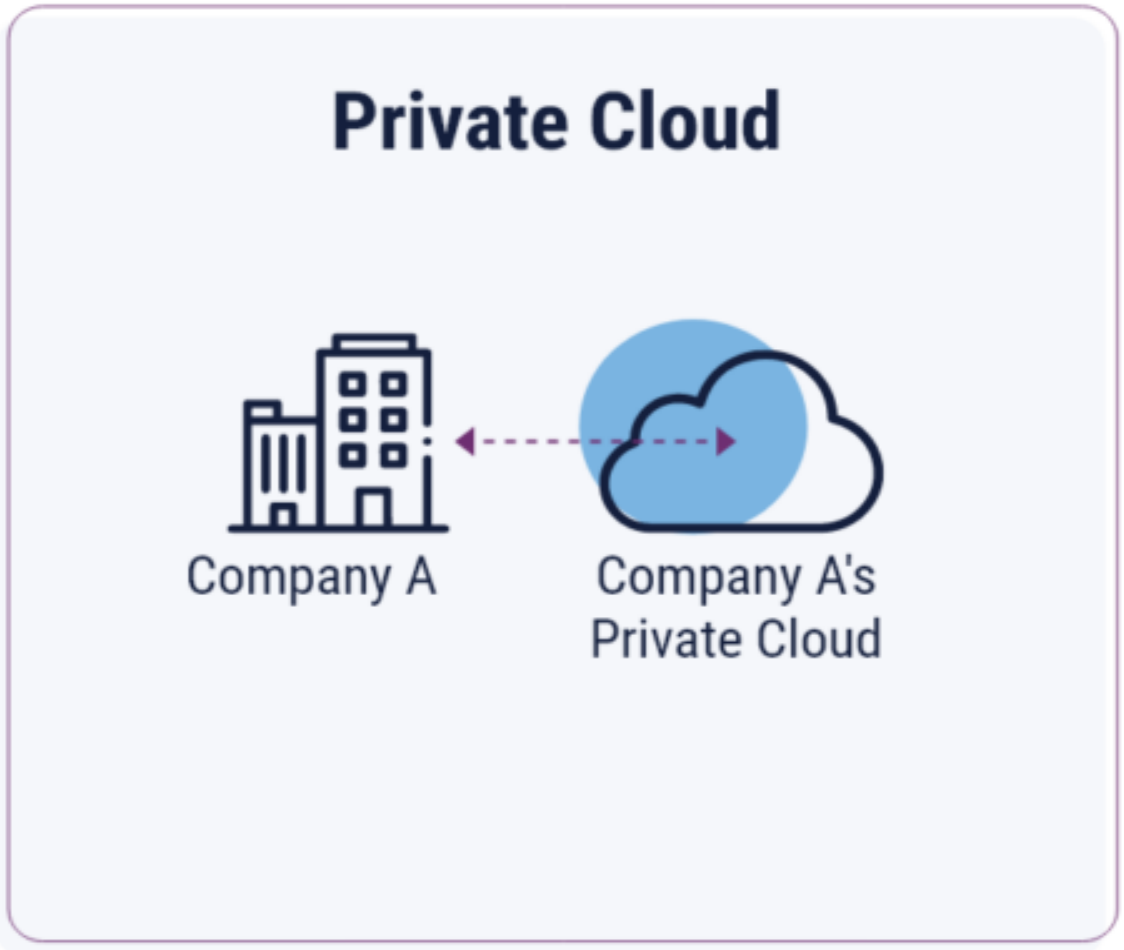
1. **Public Cloud:** Services are delivered over the internet and are accessible to anyone.
   - Example: Google Cloud Platform (GCP), Amazon Web Services (AWS).

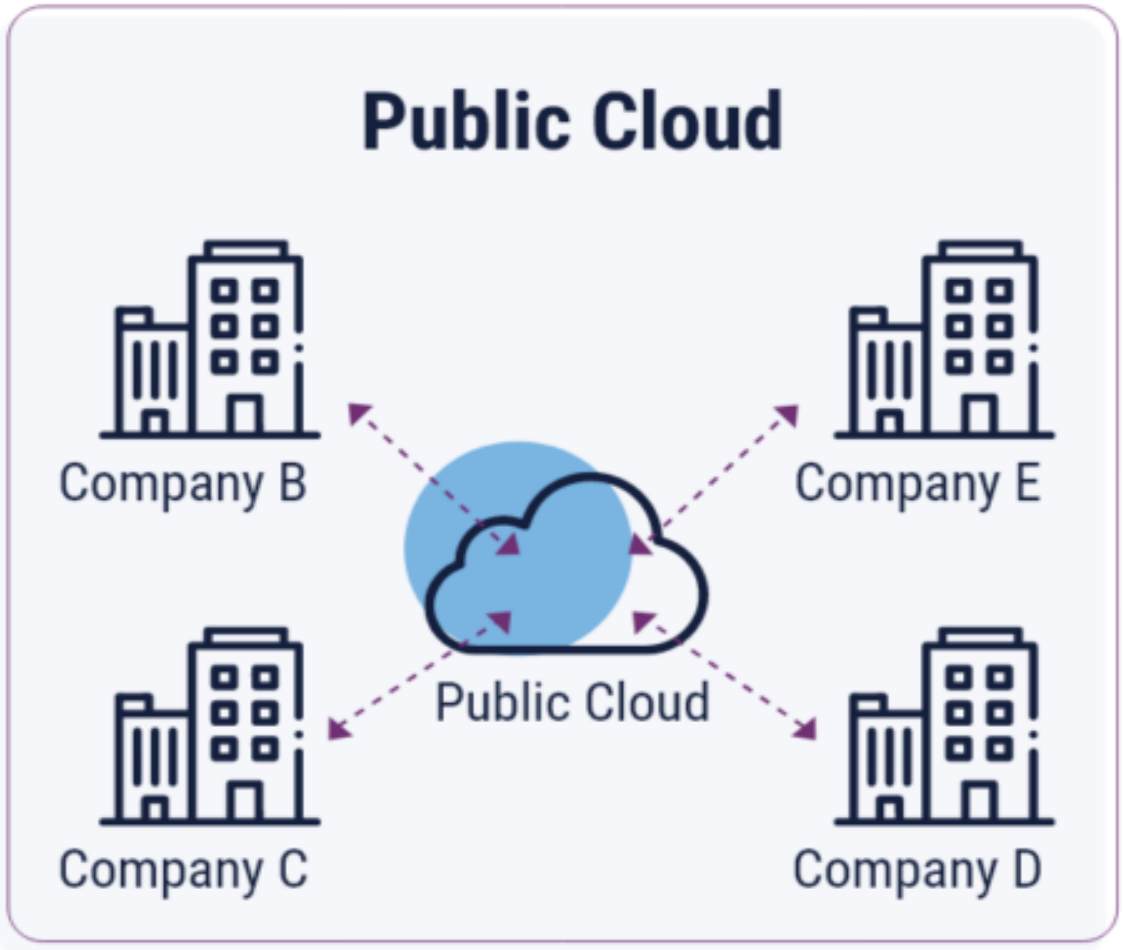
2. **Private Cloud:** Cloud infrastructure is dedicated to a single organization, providing more control and security.
   - Example: An organization’s internal cloud data center.

3. **Hybrid Cloud:** Combines public and private cloud environments, allowing data and applications to be shared between them.
   - Example: A company using both AWS for public cloud storage and a private cloud for sensitive data.

4. **Multi-Cloud:** The use of multiple cloud providers (like AWS, Azure, and Google Cloud) to avoid vendor lock-in.

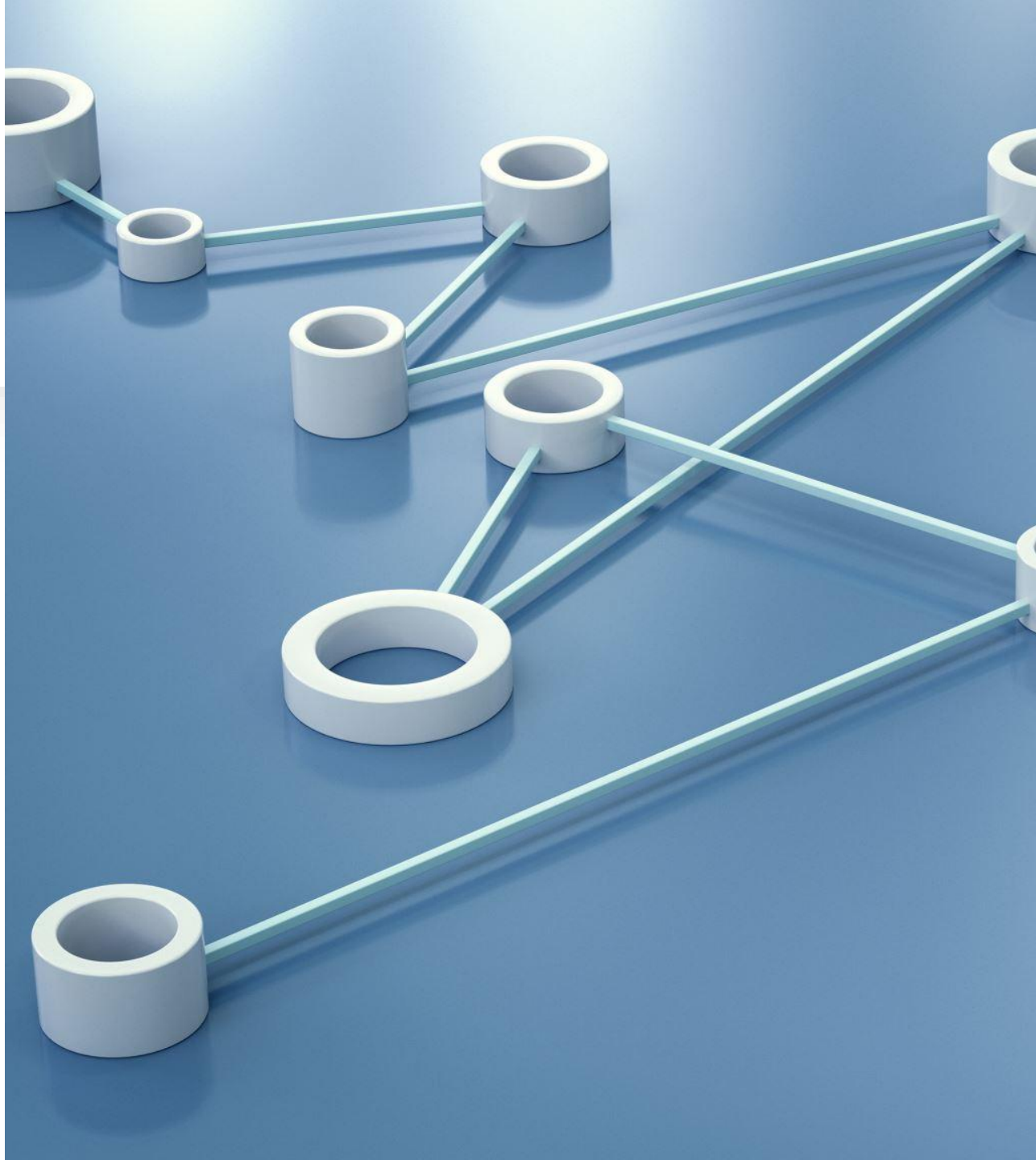
Private Vs Public Cloud





Cloud Service Models

- Cloud computing services are categorized into three main models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). These models define the level of control and management provided to users.



Infrastructure as a Service (IaaS):

1. **Definition:** Provides virtualized computing resources over the internet.

2. **What it Offers:** Virtual machines, storage, networks, and operating systems.

3. **User Responsibility:** Manages OS, applications, middleware, and runtime.

4. **Examples:**
   - Amazon Web Services (AWS) EC2 (Elastic Compute Cloud).
   - Google Compute Engine (GCE).
   - Microsoft Azure Virtual Machines.

5. **Use Cases:**
   - Hosting websites and applications.
   - Storage and backup solutions.
   - Development and testing environments.

Infrastructure as a Service (IaaS):

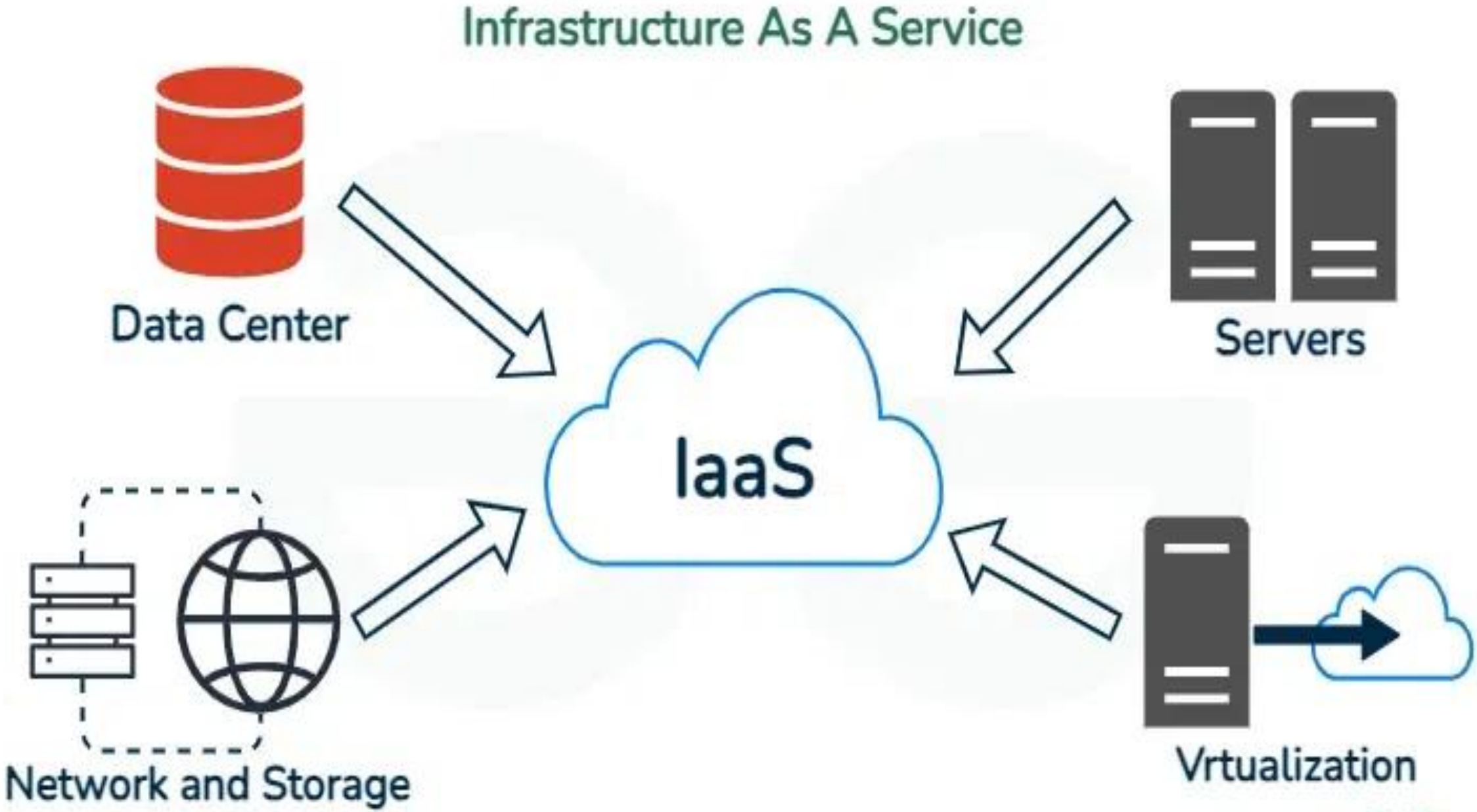
- Infrastructure as a Service (IaaS) is a cloud computing service model that gives virtualized computing resources over the web, with IaaS, associations can get to and manage versatile infrastructure assets like virtual machines, storage, and networking administration parts without the need to put resources into or keep up with actual equipment.
- IaaS allows business to outsource their whole IT infrastructure to a cloud service provider, empowering them to arrange, deploy, and manage computing resources on-demand, this adaptability allows organizations to increase their infrastructure or down in view of fluctuating interest, pay just for the resources they consume, and keep away from the expenses and intricacies related with customary on-premises infrastructure.

How does IaaS Architecture Work ?

- **On-Demand Access:** With IaaS, users can get to processing resources on-demand, allowing them to rapidly arrange and deploy infrastructure components depending on the situation. This disposes of the requirement for a forthright interest in equipment and empowers quick scaling to meet changing workload demands.
- **Self-Service Provisioning:** IaaS platforms offer self-support interfaces, for example, online interfaces or APIs, that empower users to freely arrangement and manage systems resources. This self-service model engages users to control their infrastructure deployments without depending on IT administrators.
- **Scalability:** IaaS platforms regularly offer level adaptability, allowing users to scale resources up or down based on demand, this adaptability ensures that associations can deal with changes in responsibility without encountering margin time or execution corruption.
- **Pay-Per-Use Billing:** IaaS providers normally utilize a pay-per-use billing model, where users are charged on their actual use of computing resources, this utilization based estimating model offers cost effectiveness, as associations just compensation for the resources they consume, as opposed to putting resources into excess limit.

What are the Types of Infrastructure As a Service Resources?

- **Virtual Machines (VMs):** Virtual machines are virtual instances of computing conditions that emulate the usefulness of physical servers. Users can arrangement VMs with specific configurations, including central processor, memory, storage, and operating systems, to run applications and administrations.
- **Networking:** IaaS platforms give organizing parts that empower clients to associate their virtualized infrastructure to the internet and establish communication between various resources, this includes virtual networks, subnets, firewalls, load balancers, and VPN gateways for managing network traffic and ensuring availability.
- **Load Balancers:** Load balancers convey incoming network traffic across numerous virtual machines or instances to advance execution, unwavering quality, and accessibility, they help uniformly distribute workloads and prevent overloading of individual resources, ensuring a smooth and steady user experience.
- **Databases:** A few IaaS suppliers offer managed database benefits that empower users to send and manage database in the cloud. These services incorporate relational databases like MySQL, PostgreSQL, and SQL Server, as well as NoSQL databases like MongoDB, Cassandra, and Redis.
- **Containers:** IaaS platforms may likewise offer help for containerized conditions, allowing users to deploy and manage containerized applications utilizing tools like Docker and Kubernetes, container services give a lightweight and versatile way to deal with application deployment and management, empowering quick turn of events and deployment of cloud-native applications.


Infrastructure As A Service
Data Center
Servers
IaaS
Network and Storage
Vrtualization

Platform as a Service (PaaS)

•**Definition:** Provides a complete development and deployment environment in the cloud.

•**What it Offers:** A platform for developers to build, run, and manage applications without managing the underlying infrastructure.

•**User Responsibility:** Manages the application and its data.

•**Examples:**

- Google App Engine.
- Microsoft Azure App Services.
- Heroku.

•**Use Cases:**

- Web application development.
- API development and hosting.
- Automated software deployment.

Platform As A Service (PaaS)

- Platform as a Service (PaaS) is a cloud computing model designed for developers, offering a complete environment to build, test and deploy applications. Unlike traditional infrastructure management, PaaS takes care of things like servers, storage and networking allowing developers to focus mainly on writing code and delivering applications quickly.
- In the cloud computing ecosystem, PaaS acts as a middle layer between Infrastructure as a Service (IaaS) and Software as a Service (SaaS). While IaaS provides the fundamental infrastructure like servers and storage, and SaaS delivers ready-made applications, PaaS provides developers with the necessary tools and environment to create custom applications from scratch.

How does Platform as a Service(PaaS) work?

1. **Core Infrastructure:** PaaS is built on cloud infrastructure provided by platforms like AWS, Microsoft Azure and Google Cloud. The provider handles everything behind the scenes, including servers, storage, and networking.
    - Servers: The provider manages hardware, load balancing, and scaling for you.
    - Storage: Applications and data are stored in secure cloud data centers.
    - Networking: The provider ensures secure, fast communication between resources.
2. **Built-In Platform Service:** On top of the infrastructure, PaaS offers all the tools and services you need to develop and run applications:
    - Operating Systems: Pre-configured systems like Linux or Windows.
    - Runtime Environments: Ready-to-use environments for languages like Java, Python, Node.js, Ruby or .NET.
    - Middleware: Services like caching, authentication, and messaging for applications.
    - Development Tools: Access to code editors, debugging tools, and CI/CD pipelines to streamline coding and deployment.

How does Platform as a Service(PaaS) work?

3. **Simplified Development and Deployment:** PaaS takes care of the heavy lifting in the development process
   - Development: You can write code using built-in frameworks and tools. For example, a developer might use Node.js and connect it to a pre-configured MySQL database.
   - Testing: Applications can be tested in sandbox environments that simulate real-world conditions.
   - Deployment: PaaS automates the deployment process with CI/CD pipelines, making it easy to push updates and changes.
4. **Automatic Scalability:** One of the best features of PaaS is its ability to scale based on traffic:
   - Horizontal Scaling: Adds more application instances to handle increased demand.
   - Vertical Scaling: Boosts the resources (e.g., CPU or RAM) of an existing instance.

How does Platform as a Service(PaaS) work?

5. **Easy Integration with Databases and APIs:** PaaS makes connecting to databases and third-party services straightforward:
   - Databases: Whether it's SQL (like PostgreSQL) or NoSQL (like MongoDB), PaaS simplifies setup and management.
   - APIs: You can easily integrate external services like payment systems or analytics tools to enhance your application.
6. **Built-In Security:** Security is handled by the provider, so developers can focus on building their applications:
   - Data Encryption: Ensures that data is secure both during transfer and at rest.
   - Access Control: Role-based permissions and identity management tools are included.
   - Compliance: Many providers follow regulations like GDPR or HIPAA to meet legal and industry requirements.

IaaS vs PaaS vs SaaS

Feature	IaaS	PaaS	SaaS
Definition	Provides virtualized computing resources like servers, storage, and networking.	Offers a platform with tools and environments for application development.	Delivers ready-to-use software applications over the internet.
Control Level	High: Users manage OS, middleware, apps, and data.	Medium: Users control apps and data; the provider manages infrastructure.	Low: Users only manage software configuration and usage.
Examples	AWS EC2, Microsoft Azure VM, Google Compute Engine.	AWS Elastic Beanstalk, Google App Engine, Heroku.	Google Workspace, Salesforce, Dropbox.
Target Users	IT administrators, developers requiring full control of infrastructure.	Developers looking for a managed platform to build and deploy applications.	End-users needing ready-to-use applications without technical expertise.
Use Cases	Hosting websites, storage, disaster recovery, virtual machines.	Software development, app testing, and deployment.	Email, CRM, file sharing, and collaboration tools.
Infrastructure Access	Provides direct access to virtualized hardware.	Abstracts the infrastructure, offering tools and frameworks.	No access to underlying infrastructure.

Software as a Service (SaaS)

- **Definition:** Delivers software applications over the internet on a subscription basis.

- **What it Offers:** Complete software solutions managed by the service provider.

- **User Responsibility:** Only uses the software without worrying about updates, security, or maintenance.

- **Examples:**
  - Google Workspace (Gmail, Google Drive).
  - Microsoft 365 (Word, Excel, PowerPoint).
  - Salesforce CRM.

- **Use Cases:**
  - Email communication.
  - Customer relationship management (CRM).
  - Office productivity.

Software as a Service (SaaS)

- In today's competitive market, businesses must adopt flexible, scalable and cost-effective solutions to stay ahead. Cloud services like Google Cloud, AWS, and Microsoft Azure offer these solutions. Among them, Amazon Web Services (AWS) leads the market, offering a range of services such as Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS).
- AWS’s SaaS solutions help businesses improve operations by making them more scalable and cost-effective while reducing infrastructure and maintenance costs. SaaS provides software applications over the Internet on a subscription basis, eliminating the need for local installation and management. It is commonly used in customer relationship management (CRM), project management, and collaboration tools.

How does a SaaS work?

- Overview of SaaS Delivery Model

In a SaaS delivery model, the software is hosted on the provider's cloud infrastructure and made available to customers via the internet. Users access the application remotely through a browser, typically via a subscription-based pricing model. This setup eliminates the need for businesses to maintain their own servers, install software on each device or worry about software upgrades and patches.

- Cloud-based Access and Subscription Model

With SaaS, businesses only pay for the software they use often on a monthly or annual subscription basis. The subscription model is advantageous as it allows businesses to access cutting-edge software without upfront costs. Additionally, the software can be accessed from anywhere, enabling remote work and collaboration, which is especially important in today's dynamic business environment.

Top AWS SaaS Solutions for Modern Cloud Applications