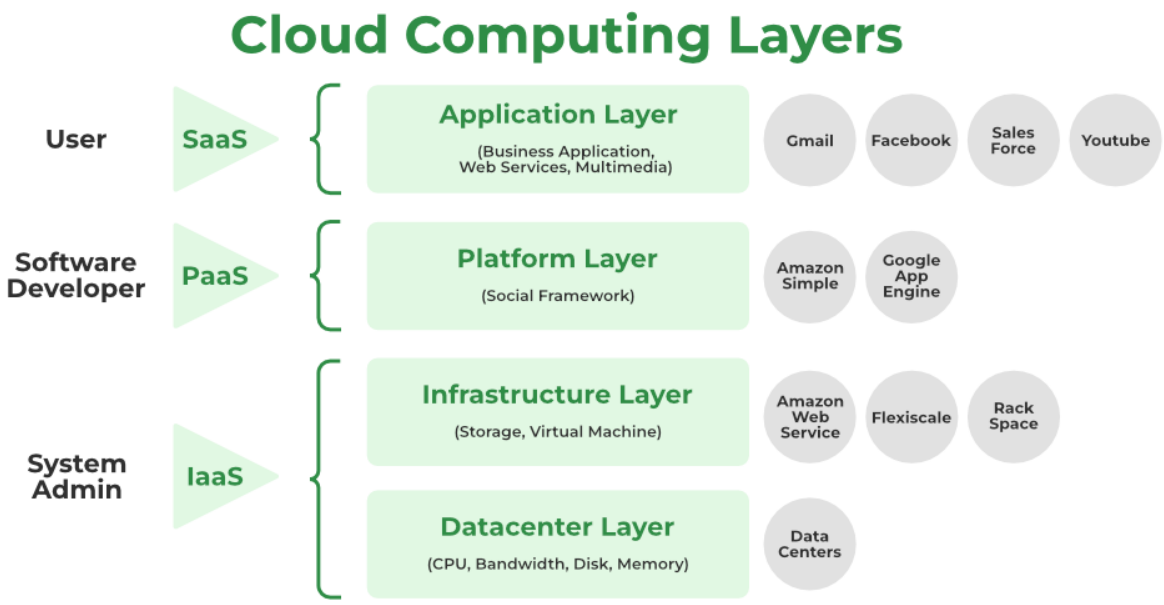
3 - Layer Architecture of Cloud Computing



Application Layer

1. The application layer, which is at the top of the stack, is where the actual cloud apps are located. Cloud applications, as opposed to traditional applications, can take advantage of the **automatic-scaling** functionality to gain greater performance, availability, and lower operational costs.
2. This layer consists of different Cloud Services which are used by cloud users. Users can access these applications according to their needs. Applications are divided into **Execution layers** and **Application layers**.
3. In order for an application to transfer data, the application layer determines whether communication partners are available. Whether enough cloud resources are accessible for the required communication is decided at the application layer. Applications must cooperate in order to communicate, and an application layer is in charge of this.
4. The application layer, in particular, is responsible for processing IP traffic handling protocols like Telnet and FTP. Other examples of application layer systems include web browsers, SNMP protocols, HTTP protocols, or HTTPS, which is HTTP's successor protocol.

Platform Layer

1. The operating system and application software make up this layer.
2. Users should be able to rely on the platform to provide them with **Scalability, Dependability, and Security Protection** which gives users a space to create their

apps, test operational processes, and keep track of execution outcomes and performance. SaaS application implementation's application layer foundation.

3. The objective of this layer is to deploy applications directly on virtual machines.
4. Operating systems and application frameworks make up the platform layer, which is built on top of the infrastructure layer. The platform layer's goal is to lessen the difficulty of deploying programmers directly into VM containers.
5. By way of illustration, Google App Engine functions at the platform layer to provide API support for implementing storage, databases, and business logic of ordinary web apps.

Infrastructure Layer

1. It is a layer of virtualization where physical resources are divided into a collection of virtual resources using virtualization technologies like Xen, KVM, and VMware.
2. **This layer serves as the Central Hub of the Cloud Environment**, where resources are constantly added utilizing a variety of virtualization techniques.
3. A base upon which to create the platform layer. constructed using the virtualized network, storage, and computing resources. Give users the flexibility they want.
4. Automated resource provisioning is made possible by virtualization**,** which also improves infrastructure management.
5. The infrastructure layer sometimes referred to as the virtualization layer, partitions the physical resources using virtualization technologies like **Xen, KVM, Hyper-V, and VMware** to create a pool of compute and storage resources.
6. The infrastructure layer is crucial to cloud computing since virtualization technologies are the only ones that can provide many vital capabilities, like dynamic resource assignment.

Datacenter Layer

- In a cloud environment, this layer is responsible for **Managing Physical Resources** such as servers, switches, routers, power supplies, and cooling systems.
- Providing end users with services requires all resources to be available and managed in data centers.
- Physical servers connect through high-speed devices such as routers and switches to the data center.
- In software application designs, the division of business logic from the persistent data it manipulates is well-established. This is due to the fact that the same data cannot be incorporated into a single application because it can be used in numerous ways to support numerous use cases. The requirement for this data to become a service has arisen with the introduction of microservices.
- A single database used by many microservices creates a very close coupling. As a result, it is hard to deploy new or emerging services separately if such services need database modifications that may have an impact on other services. A data layer containing many databases, each serving a single microservice or perhaps a few closely related microservices, is needed to break complex service interdependencies.