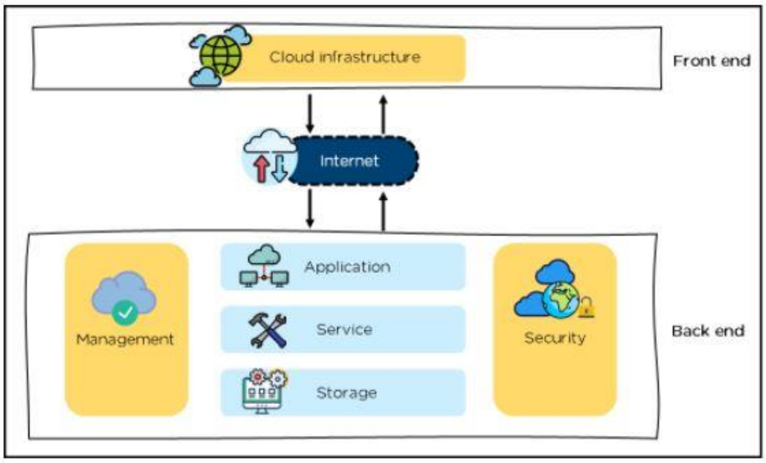
2. Cloud Computing Architecture

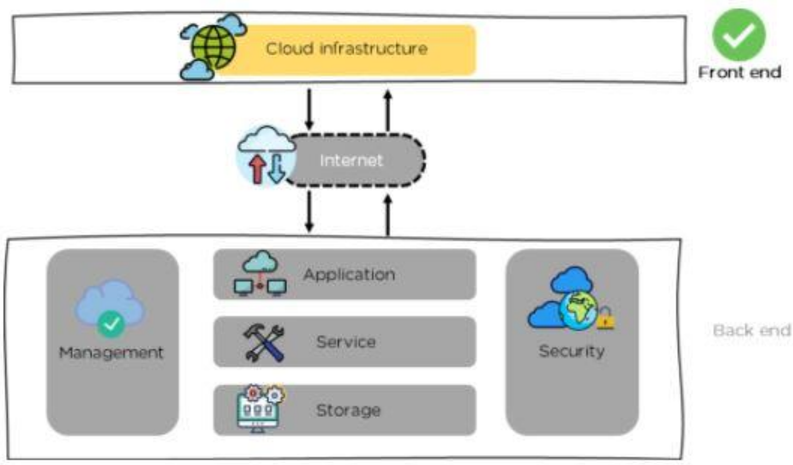
Architecture of cloud computing is the combination of both SOA (Service Oriented Architecture) and EDA (Event Driven Architecture). Client infrastructure, application, service, runtime cloud, storage, infrastructure, management and security all these are the components of cloud computing architecture.

Cloud Computing Architecture is divided into two parts, i.e., front-end and back-end. Front-end and back-end communicate via a network or internet. A diagrammatic representation of cloud computing architecture is shown below:



Front-End

- It provides applications and the interfaces that are required for the cloud-based service.
- It consists of client's side applications, which are web browsers such as Google Chrome and Internet Explorer.
- Cloud infrastructure is the only component of the front-end. Let's understand it in detail.

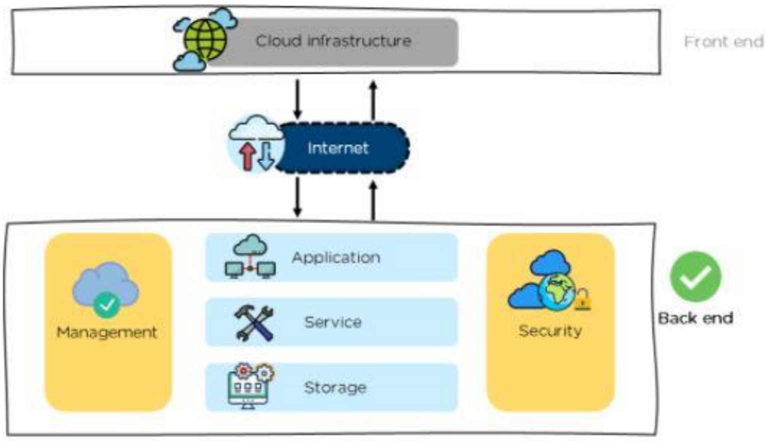


- Cloud infrastructure consists of hardware and software components such as data storage, server, virtualization software, etc.

- It also provides a Graphical User Interface to the end-users to perform respective tasks.

Back-End

- It is responsible for monitoring all the programs that run the application on the front-end
- It has a large number of data storage systems and servers. The back-end is an important and huge part of the whole cloud computing architecture, as shown below:



Benefits of Cloud Computing Architecture

The cloud computing architecture is designed in such a way that:

- It solves latency issues and improves data processing requirements

- It reduces IT operating costs and gives good accessibility to access data and digital tools

- It helps businesses to easily scale up and scale down their cloud resources

- It has a flexibility feature which gives businesses a competitive advantage

- It results in better disaster recovery and provides high security

- It automatically updates its services

- It encourages remote working and promotes team collaboration

Cloud Computing Architecture Components

Some of the important components of Cloud Computing architecture that we will be looking into are as follows:

- Hypervisor
- Management Software
- Deployment Software
- Network
- Cloud Server
- Cloud Storage

2.1 EXPLORING THE CLOUD COMPUTING STACK

The cloud creates a system where resources can be pooled and partitioned as needed. Cloud architecture can couple software running on virtualized hardware in multiple locations to provide an on-demand service to user-facing hardware and software. It is this unique combination of abstraction and metered service that separates the architectural requirements of cloud computing systems from the general description given for an n-tiered Internet application. Many descriptions of cloud computing describe it in terms of two architectural layers: A client as a front end The “cloud” as a backend.

This is a very simplistic description because each of these two components is composed of several component layers, complementary functionalities, and a mixture of standard and proprietary protocols. Cloud computing may be differentiated from older models by describing an encapsulated information technology service that is often controlled through an Application Programming Interface (API), thus modifying the services that are delivered over the network. A cloud can be created within an organization's own infrastructure or outsourced to another datacentre. While resources in a cloud can be real physical resources, more often they are virtualized resources because virtualized resources are easier to modify and optimize. A compute cloud requires virtualized storage to support the staging and storage of data. From a user's perspective, it is important that the resources appear to be infinitely scalable, that the service be measurable, and that the pricing be metered.

2.2 Understanding Cloud Architecture

Many descriptions of cloud computing describe it in terms of two architectural layers:

- A client as a front end
- The “cloud” as a backend

To understand the Cloud Architecture and the type of applications that can run on Cloud, an understanding on the below will be useful:

- Composability
- Infrastructure
- Platforms
- Virtual Appliances

- Communication Protocols

Composability

A cloud-based application has the property of being built from a collection of components. This feature is known as composability.

A composable component must be:

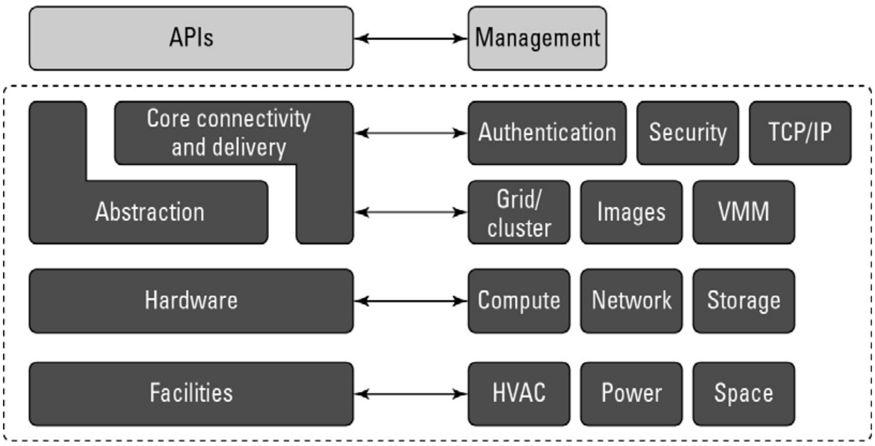
Modular: It is self-contained and independent unit that is cooperative reusable and replaceable.

Stateless: A transaction is executed without regard to other transactions or requests.

It isn't an absolute requirement that transactions be stateless, some cloud computing applications provide managed states through brokers, transaction monitors and service buses. In rare cases full transactional systems are deployed in the clouds, but these systems are harder to architect in a distributed architecture.

Infrastructure

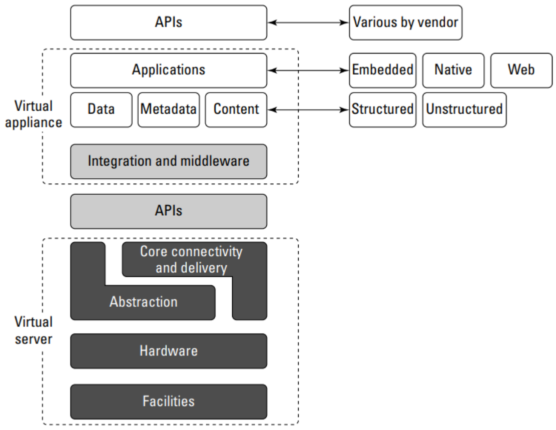
Cloud Infrastructure depends on the virtual machine technology. A particular physical machine can run multiple virtual machines. The low-level software that runs these virtual machines is called as Virtual Machine monitor (VMM) or Hypervisor.



Platforms

A cloud platform provides the hardware and software needed to build web apps or services that are custom built to use the capabilities of that platform. Platforms represent the full software stack except the presentation layer.

Depending upon the PAAS vendor you may find developer tools for team collaboration, testing tools instrumentation for measuring program performance and attributes, versioning, database and webservice integration and storage tools.



some major cloud platforms are:

- Salesforce.com's Force.com platform
- Windows Azure platform
- Google Apps and Google App engine

2.3 Virtual Appliances

A virtual appliance is a pre-configured virtual machine image, ready to run on hypervisor virtual appliances are a subset of the broader class of software appliances.

A virtual appliance is not a complete virtual machine platform, but rather a software image containing a software stack designed to run on a virtual computer, a hypervisor is merely a platform for running an operating system environment and does not provide application software itself. For example, applications such as a web server or database server that can run on a virtual machine image are referred to as virtual appliances.

A virtual appliance is a platform instance; therefore, virtual appliances occupy the middle of a cloud computing stack.

IAAS systems such as Amazon's elastic compute cloud uses virtual appliances. Amazon machine images are a collection of virtual appliances that can be installed on their Xen hypervisor servers. AMI includes a variety of operating systems both proprietary and open source, a set of enterprise applications such as Oracle BPM, SQL server and even complete application stacks such as LAMP (Linux Apache, MySQL and PHP).

Similarly, VMware's virtual appliance marketplace sells virtual appliances that run on VMware's hypervisor in cloud computing applications.

2.4 Communication Protocols

The popular communication protocols used in the cloud computing world are SOAP, XML-RPC, CORBA, APP and REST.

APP is Atom publishing protocol. Atom is a syndication format that allows HTTP protocols to create and update information.

CORBA is Common object requesting broker architecture. It is an early effort at a standard to enable object interactions within a network. It uses an interface definition language to define the face these objects present and an object request broker (ORB) to negotiate the exchange of information about objects.

REST is representational state transfer. It assigns a global identifier to a resource so there is an uniform method of accessing information sources. That identifier is a URI expressed in HTTP form. This is the most widely used protocol in cloud apps.

2.5 Connecting to cloud

Clients can connect to a cloud service in a number of different ways. These are the two most common means:

- A Web browser
- A proprietary application

These applications can be running on a server, a PC, a mobile device, or a cell phone. There are three basic methods for securely connecting over a connection:

- Use a secure protocol to transfer data such as SSL (HTTPS), FTPS, or IPsec, or connect using a secure shell such as SSH to connect a client to the cloud.
- Create a virtual connection using a virtual private network (VPN), or with a remote data transfer protocol such as Microsoft RDP or Citrix ICA, where the data is protected by a tunnelling mechanism.
- Encrypt the data so that even if the data is intercepted or sniffed, the data will not be meaningful.

2.6 The Jolicloud Netbook OS

What Does Joli OS Mean?

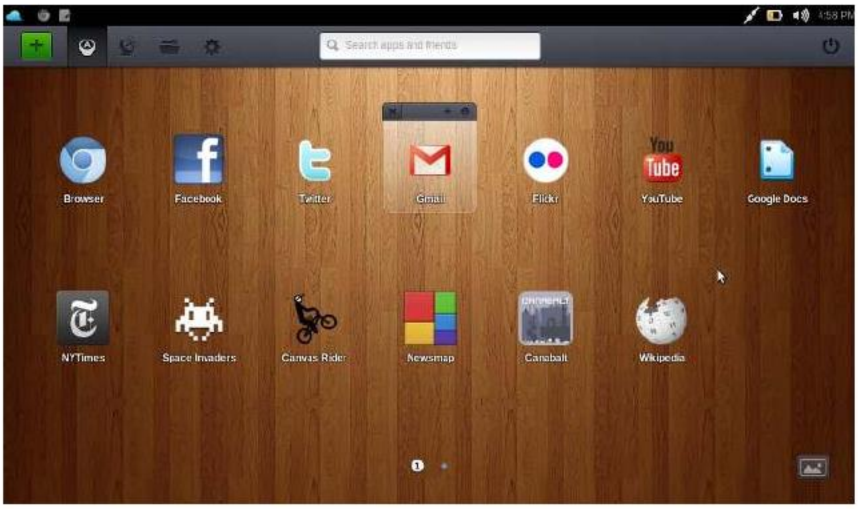
Joli OS, developed by Jolicloud, provides file sharing and access to Web applications (apps) and desktops from the cloud. Based on the Ubuntu Linux kernel, Joli OS was designed to give netbook and low-end processors the ability to utilize Web app and basic computing services without hardware upgrades.

- Joli OS is installed as a thin client on a host desktop and provisions a variety of Web apps from the cloud, including standard Web browsers, Gmail, Dropbox, Google Docs and Flickr.

- Joli OS hosts a number of apps that may be accessed and easily added to the cloud desktop via the default launcher. Joli OS also provides social bookmarking capabilities for user sharing of popular apps and services.

Jolicloud concentrates on building a social platform with automatic software updates and installs. The application launcher is built in HTML 5 and comes preinstalled with Gmail, Skype, Twitter, Firefox, and other applications.

Any HTML 5 browser can be used to work with the Jolicloud interface. Jolicloud maintains a library or App Directory of over 700 applications as part of an app store. When you click to select an application, the company both installs and updates the application going forward, just as the iPhone manages applications on that device.



2.7 Chromium OS

Chromium OS is designed with cloud technology at its core, emphasizing lightweight computing, security, and seamless integration with web-based applications and services. Here are the key features of Chromium OS in relation to cloud technology:

1. Cloud-Centric Architecture

- **Web-Based Applications**: Chromium OS primarily relies on web applications accessed through the Chrome browser. This cloud-centric approach minimizes the need for local storage and processing power, leveraging the cloud for computing tasks.
- **Data Synchronization**: Users' data, settings, and preferences are synchronized with their Google Account, allowing seamless access to personalized content across devices connected to the internet.

2. Security and Privacy

- **Sandboxing**: Applications are sandboxed, meaning each application runs in its own secure environment, isolated from other processes on the device. This enhances security by preventing malware and unauthorized access.
- **Verified Boot**: Chromium OS uses a verified boot process that ensures the integrity of the operating system before it loads. It verifies the integrity of firmware, kernel, and system partitions, protecting against tampering and ensuring a secure boot environment.

3. Fast Boot and Performance

- **Quick Boot Times**: Chromium OS is optimized for fast boot times, allowing users to start their devices quickly and efficiently.
- **Efficient Resource Management**: It efficiently manages system resources, ensuring smooth performance even on low-powered devices like netbooks and Chromebooks.

4. Integration with Google Services

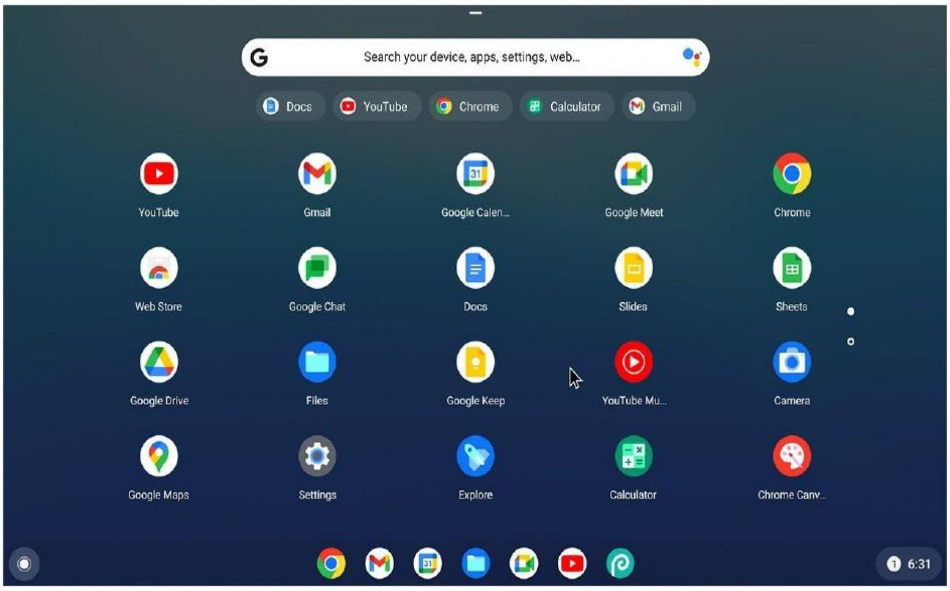
- **Google Account Integration**: Users log in with their Google Account, enabling access to Google services such as Gmail, Google Drive, Google Calendar, and Google Docs. This integration simplifies data management and collaboration.
- **Chrome Web Store**: Users can install web apps and extensions from the Chrome Web Store, expanding functionality and customization options directly within the browser interface.

5. Ecosystem and Community

- **Open-Source Foundation**: Chromium OS is developed as an open-source project, encouraging community contributions and innovation. It allows developers to customize and enhance the operating system for specific needs and use cases.
- **Chromebooks**: Chromium OS is prominently featured on Chromebook devices, which are popular in education, enterprise, and consumer markets due to their affordability, ease of management, and integration with Google services.

6. Limitations and Considerations

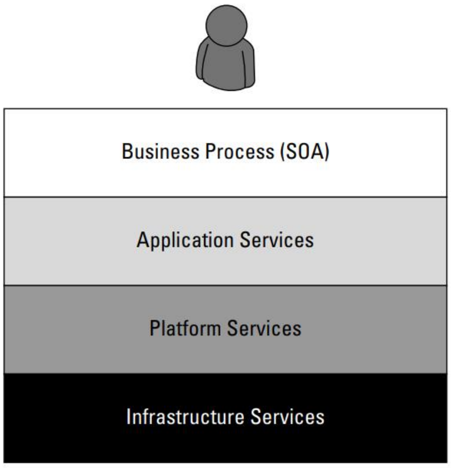
- **Offline Capability**: While Chromium OS is optimized for cloud-based usage, its functionality may be limited in offline environments where internet connectivity is unreliable or unavailable.
- **Dependency on Google Ecosystem**: The integration with Google services may limit user choices and preferences for alternative cloud providers or applications, potentially leading to vendor lock-in.



2.8 Defining Infrastructure as a Service (IaaS)

Infrastructure as a Service (IaaS) creates what may be determined to be a utility computing model, something that you can tap into and draw from as you need it without significant limits on the scalability of your deployment. You pay only for what you need when you need it. IaaS may be seen to be an incredibly disruptive technology, one that can help turn a small business into a large business nearly overnight. This is a most exciting prospect; one that is fuelling a number of IaaS startups during one of the most difficult recessions of recent memory.

Infrastructure as a Service (IaaS) is a cloud computing service model in which hardware is virtualized in the cloud. In this particular model, the service vendor owns the equipment: servers, storage, network infrastructure, and so forth. The developer creates virtual hardware on which to develop applications and services. Essentially, an IaaS vendor has created a hardware utility service where the user provisions virtual resources as required.

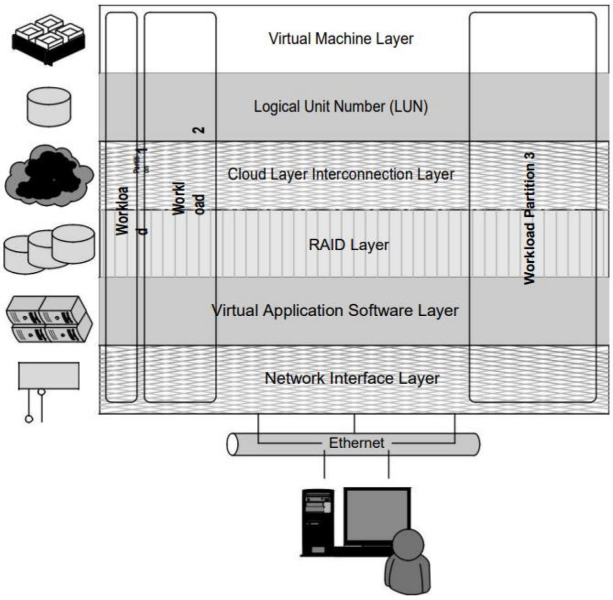


The developer interacts with the IaaS model to create virtual private servers, virtual private storage, virtual private networks, and so on, and then populates these virtual systems with the applications and services it needs to complete its solution. In IaaS, the virtualized resources are mapped to real systems. When the client interacts with an IaaS service and requests resources from the virtual systems, those requests are redirected to the real servers that do the actual work

2.9 IAAS WORKLOADS

The fundamental unit of virtualized client in an IaaS deployment is called a workload. A workload simulates the ability of a certain type of real or physical server to do an amount of work. The work done can be measured by the number of Transactions Per Minute (TPM) or a similar metric against a certain type of system. In addition to throughput, a workload has certain other attributes such as Disk I/Os measured in Input/Output Per Second IOPS, the amount of RAM consumed under load in MB, network throughput and latency, and so forth. In a hosted application environment, a cli-ent's application runs on a dedicated server inside a server rack or perhaps as a standalone server in a room full of servers. In cloud computing, a provisioned server called an instance is reserved by a customer, and the necessary amount of computing resources needed to achieve that type of physical server is allocated to the client's needs.

A client would reserve a machine equivalent required to run each of these workloads. The IaaS infrastructure runs these server instances in the data center that the service offers, drawing from a pool of virtualized machines, RAID storage, and network interface capacity. These three layers are expressions of physical systems that are partitioned as logical units. LUNs, the cloud interconnect layer, and the virtual application software layer are logical constructs. LUNs are logical storage containers, the cloud interconnect layer is a virtual network layer that is assigned IP addresses from the IaaS network pool, and the virtual application software layer contains software that runs on the physical VM instance(s) that have been partitioned from physical assets on the IaaS' private cloud.



Consider a transactional eCommerce system, for which a typical stack contains the following components:

- Web server
- Application server
- File server
- Database
- Transaction engine

This e-Commerce system has several different workloads that are operating: queries against the database, processing of business logic, and serving up clients' Web pages.

2.10 Pods, aggregation, and silos

1. **Pods**:

   - **Definition**: In cloud-native terminology, a "pod" typically refers to a group of one or more containers (e.g., Docker containers) that are deployed together on the same host and share the same network and storage resources.

- **Usage**: Pods are fundamental units of deployment in container orchestration platforms like Kubernetes. They allow for efficient resource utilization and management of applications by grouping related containers together.

2. **Aggregation**:

   - **Definition**: Aggregation in cloud computing often refers to the process of combining or consolidating data, resources, or services from multiple sources into a unified view or operation.
   - **Usage**: For example, in cloud data analytics, aggregation might involve combining data from different sources (like databases, logs, or IoT devices) to derive insights or perform computations. In networking, aggregation could refer to consolidating multiple network links into a larger, more efficient link.

3. **Silos**:

   - **Definition**: In cloud computing, silos refer to isolated or separate environments where data, applications, or resources are kept distinct from each other, often leading to inefficiencies or barriers to collaboration.
   - **Usage**: Breaking down silos is a common goal in cloud integration and digital transformation efforts. It involves integrating different systems or departments within an organization to enable better communication, sharing of resources, and alignment of goals.

Relationship in Cloud Computing:

- **Pods** are used to encapsulate and manage containers efficiently within cloud environments, facilitating scalability and deployment.
- **Aggregation** involves combining resources or data to optimize operations and enhance efficiency in cloud services.
- **Silos** represent barriers that cloud technologies aim to overcome by integrating systems and promoting collaboration across different parts of an organization.

2.11 Defining Platform as a Service (PaaS)

Platform as a Service (PaaS) is a cloud computing service model that provides a platform allowing customers to develop, run, and manage applications without the complexity of building and maintaining the underlying infrastructure typically associated with developing and launching an application.

Here are the key characteristics and components of PaaS:

1. **Application Development Tools**: PaaS provides developers with a suite of tools and services to develop, test, and deploy applications. These tools often include IDEs (Integrated Development Environments), code repositories, version control, and debugging capabilities.
2. **Middleware**: PaaS platforms include middleware services such as databases, messaging queues, caching, and application servers that enable developers to build

applications without worrying about the underlying hardware or software infrastructure.

3. **Deployment and Management**: PaaS automates and simplifies the deployment process, providing tools for deploying applications to the cloud environment with scalability and reliability. It also includes management features like monitoring, logging, and auto-scaling to ensure optimal performance.
4. **Scalability and Availability**: PaaS platforms typically offer built-in scalability features, allowing applications to automatically scale resources up or down based on demand. They also provide high availability through redundancy and failover mechanisms to minimize downtime.
5. **Multi-Tenancy**: PaaS services are designed to support multiple users or organizations (tenants) on the same platform securely, ensuring isolation of resources and data between different users.
6. **Integration and APIs**: PaaS platforms often include APIs and integration capabilities that allow developers to easily connect their applications with other services, both within the platform and externally.
7. **Billing and Metering**: PaaS providers typically offer a pay-as-you-go pricing model, where customers are billed based on their usage of resources such as compute instances, storage, and data transfer.

PaaS is particularly beneficial for developers and businesses looking to accelerate application development, reduce infrastructure management overhead, and leverage the scalability and flexibility of cloud computing without having to build and maintain their own underlying infrastructure stack. It abstracts away the complexities of infrastructure management, allowing developers to focus more on writing code and delivering value to their users.