DAY 01 23-12-2024

* Networking & Data Centers.

Networking refers to interconnecting multiple computer systems together.

Networking is crucial for all the IT operations to run smoothly and effectively.

* Cloud

It is a group of data centers distributed over the globe and connected to a common web Server.

* IT Infrastructure fundamentals

1. Data Center

- Data Center consists of multiple Servers, Storage devices, routers, interfaces and many more Such components.

- The data centers which are located in the premises of the company are known as on-premises data centers.

- On-premises data centers are mostly dedicated to the company and organization and support various IT operations.

- Hence these datacenters are referred to as IT Infrastructure.

- The IT infrastructure is mandatory for any company or organizations to run their IT ~~Operations.~~

- Big companies with deep ~~pockets~~ invest huge capital investments and create on-premises data centers for themselves.

- Small IT companies, Startups, developers, & freelancers may find it Challenging to create their own IT infrastructure, or data cent

- So they purchase or lease hosting Servers/ resources from hosting companies or IT infrastructure companies.

- Either on-premises data centers or hosting companies have limitations with respect to scaling up of resources.

★ Limitations of on-premises data centers - hosting companies

i. Scaling is limited.

ii. huge fixed capital - Initial Investment cost is way too high.

iii. The billing is always more than or equal to the utilization.

iv. The data centers can experience downtim and this might impact the resources hosted (applications hosted).

★ Advantages of cloud

i. Scaling is flexible - Cloud offers scaling fro

ii. nowhere to anywhere. (eg) from few GB to ~~hig~~ 100 TB, PB, EB

[1TB - 1024GB, 1PB - 1024TB, 1EB - 1024

ii. No Initial Investment is required. - you can create an AWS cloud account free of cost.

iii. The cloud billing is as per the utilization of the resources. It is a monthly generated bill.

iv. flexible access to the cloud - An user can access cloud through a web based console

★ **Requirements for establishing a data Center**

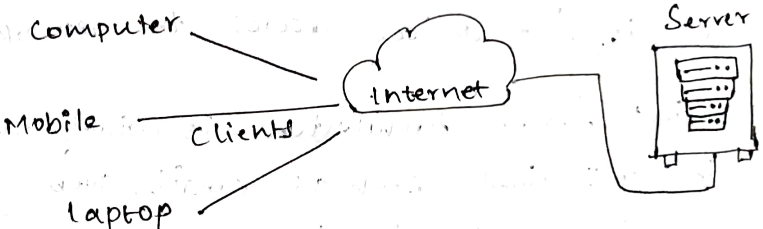
1. Place - Fixed capital - huge capital
2. Physical Infrastructure - Buildings, Ac's
3. Resources - H/w, N/w components
4. People - Hire & Technisians
5. S/w installed - O/s & Apps.

★ **why customers move to cloud**

i. Increase in Agility (↑)
ii. Increase accelerate time to market (↑)
iii. Increase Innovation (↑)
iv. Scale Seamlessly (↑)
v. Decrease optimize costs. (↓)
vi. Minimize Security vulnerabilities (↓)
vii. Reduce management Complexity (↓)
viii. reduced complexity and risk (↓)

★ **Fundamentals of Networking**

1. → **Client - Server Architecture**



• Client Server Architecture is a network model that allows communication and data exchange between different applications over a single or multiple Servers.

- The architecture can be dassified into two Parts

i. client - Client is an application that requests Services from the Server such as data retrieval, Storage, calculations, and other functions.

ii. Server - It is an application that processes client request, Sends responses or it performs specific actions. The Client & the Server may reside on the same machine or different devices accross the network.

iii. Client Server Architecture is widely used in applications such as email, web browsing, online banking, and e-commerce.

★ Components of Client - Server Architecture

1. Client
2. Server
3. Network
4. Protocol
5. Middleware
6. Application logic

- client Server architecture depende on 3 main components that need to work together for its function. These components are

1. Client

A client device or Software that requests Services from a Server. Clients are consuming facing and often include web browsers, mobile applications, or desktop applications that people can interact with. They communicate with the Server to retrieve data, make transactions, or perform other tasks by delegating that responsibility to the Server.

2. **Server** : A Server is a computing or program that offers services or solutions to client over a network. Servers handle processing of Client requests, which include tasks like file storage, database access, and application hosting, along with backend activities like computations, data management, & business logic, Significantly reducing what clients need to handle.

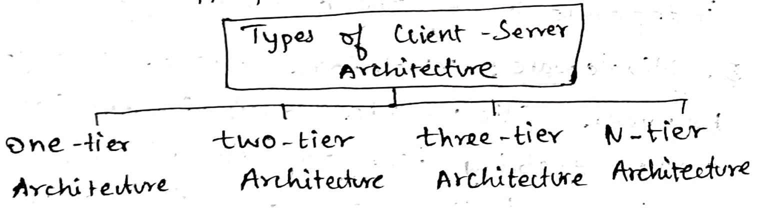
3. **Network** : This Serves as the channel through which Clients and Servers are connected for data transfer between them. Networks range from local area networks (LAN) within a single building to wide area network (WAN) and the internet, which can span Countries. It acts as the intermediary, facilitating the interchange of requests and responses between the clients & Servers, which influence the Speed and reliability of these interactions.

4. **Protocol** : Protocols are rules that define how data is exchanged between Clients and Servers, ensuring communication is orderly, secure and understandable. Common protocols include HTTP or HTTPS for web Services, FTP for file transfers, and SMTP for email. They help bridge communication between different systems, independent of their technology stack.

5. **Middleware** : Middlewara acts as a bridge between Client-side and Server-Side code, enabling them to communicate. It performs tasks such as authentication, load balancing, data translation and message queing, Simplifying interactions

within the Client-Server model by enhancing transaction speed, Scalability and integration.

6. Application logic: Application logic is the code and processes that determine how a server responds to client requests, involving business rules, big data Processing, and workflows. on the Server side, It ensures the server correctly interrupts client requests, performs necessary Calculations or data manipulations and delivers appropriate responses.



There are different types of Client-Server architecture, depending on how many tiers or layers are involved in the communication process. Some of the types are

★ One-tier Architecture:

- A self-contained application on a single platform. In one-tier architecture, the client, server and database are all on the same machine. The client handles user-interaction and business logic, the Server provides Services like data storage and processing, and the database manages data. While simple and popular for Small apps, this architecture is rarely used in production because it doesn't meet most system requirements.

★ Two - tier architecture :

- This basic Client-Server Architecture involves direct communication between the client and Server without an intermediate layer. The Client manages the User interface (UI) and business logic, while the Server handles data storage and processing. (An example): is a web browser requesting pages from a web Server, which responds with HTML files. It's easy to implement but has drawbacks like low scalability, high network traffic, and Security risks.

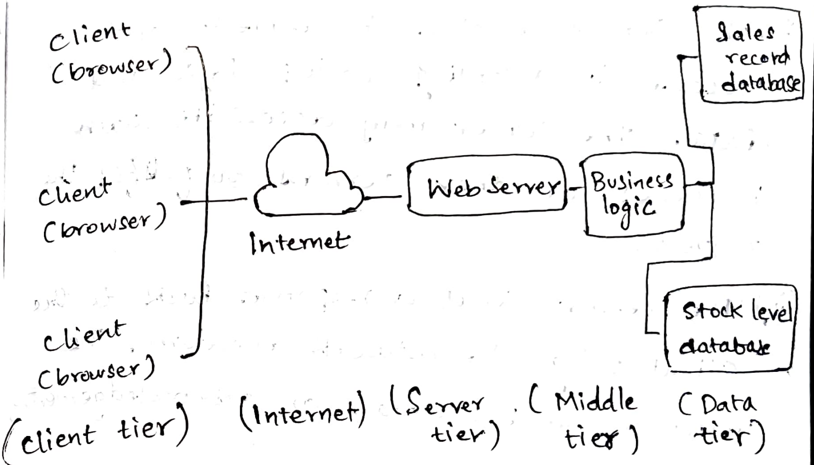
★ Three - tier Architecture :

- A more complex Client-Server architecture Setup with an intermediate layer (usually middleware or an application Server) that handles business logic, acting as a bridge between the Client and Server. The Client deals with the user interface (UI), while the server manages data Storage.

- (An example): is an Online banking System where the client is a web Server, the middleware checks transactions, and the Server Stores account data. This architecture improves Scalability, performance, and Security but increases complexity and cost.

★ N-tier Architecture :

- A more flexible client-Server architecture with more than 3 tiers, allowing greater Scalability, flexibility, and modularity.
- Each tier can be distributed across different machines or networks and updated independently.
- (An Example): is an E-commerce System with a web browser displaying the product catalog, a web server handling HTTP requests, an application Server processing business logic, and a database storing product information.
- while Suitable for complex systems, it requires more resources and management.



N-tier Architecture

★ How does Client-Server Architecture work?

The basic steps of how Client-Server architecture works are:

i) The client sends a request to the server using the network medium. The request can be a query, a command or a message.

ii) The Server receives the request and processes it according to its logic and data. The server may access its own resources or other servers to fulfil the request.

iii) The Server Sends a response back to the Client using the network medium. The response can be data, an acknowledgement, or an error message.

iv) The client receives the response and displays it to the user or performs forther actions based on it.

★ Some Examples of Client-Server Architecture

i. Email Servers: It has evolved into the Primary communication method for business due to its speed and convenience. Various Server components work together to deliver email between users across different mail servers.

ii. File Servers: When Saving documents on services like Google Docs or Microsoft office, you're interacting with file Servers. These Servers Store data Centrally and allow multiple Clients to access it.

iii. Web Servers: These high-powered Servers host websites, which web clients access through DNS or an IP address.

Here's a Simplified process:

a) A user enters a URL in the browser.

b) The browser requests the IP address from the Domain Name System (DNS).

c) The DNS server provides the IP address to the browser.

d) The browser sends an HTTPS or HTTP request to the Web Server.

e) The Server sends back the requested files.

f) The user retrieves the files, and the process Continves as needed.