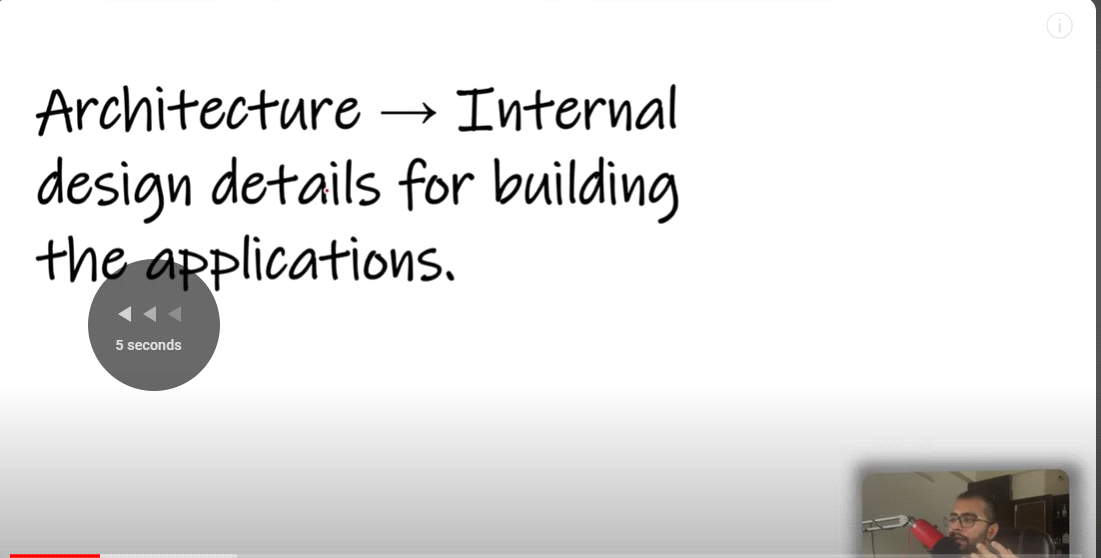
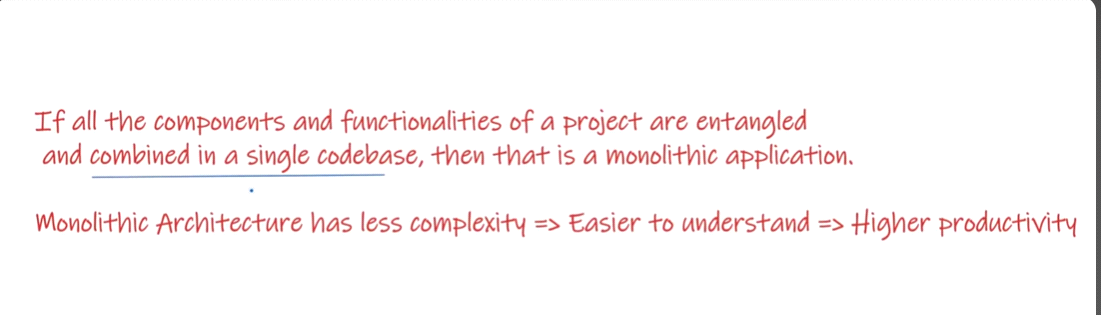
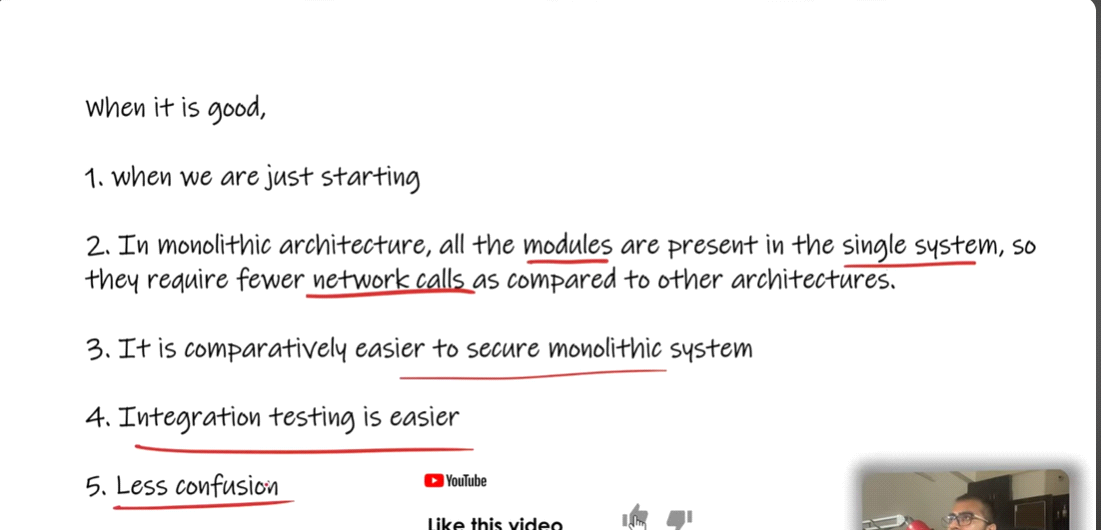


**Monolithic Architecture**

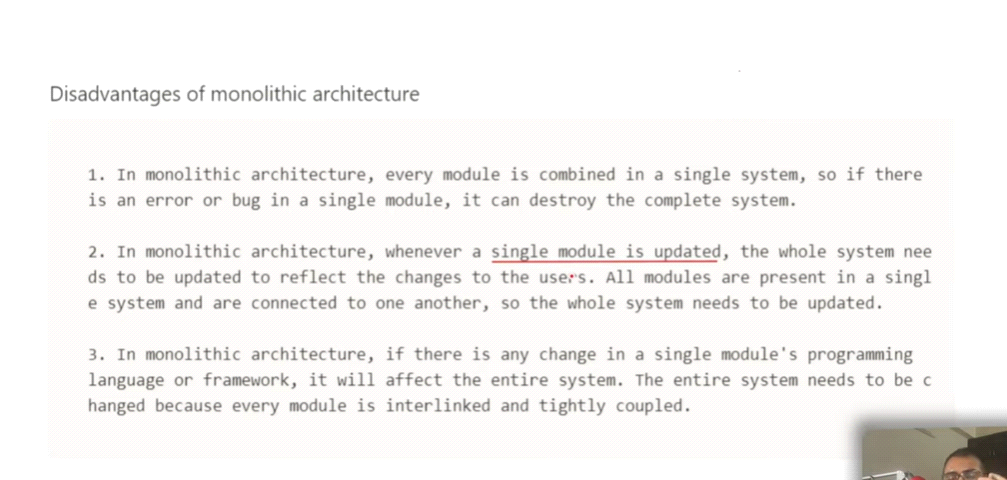




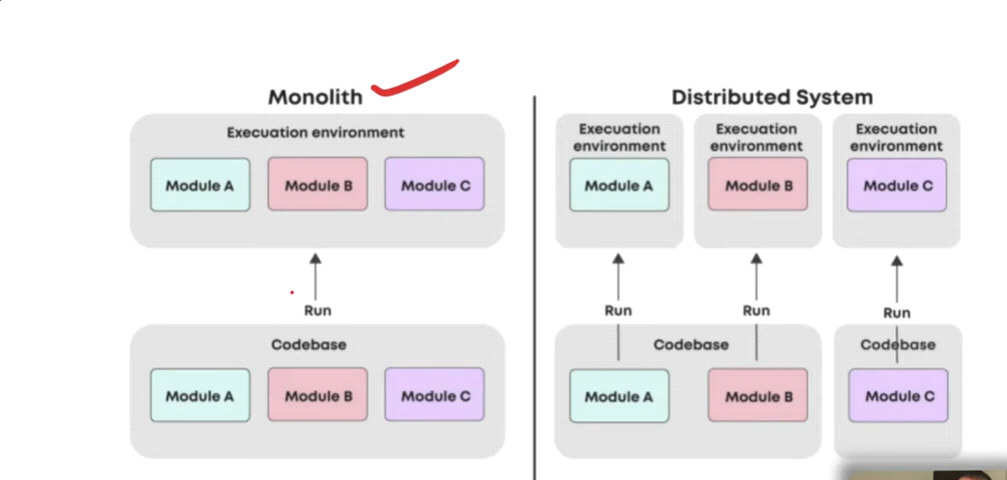
Monolithic Architecture is also know as centralized system.

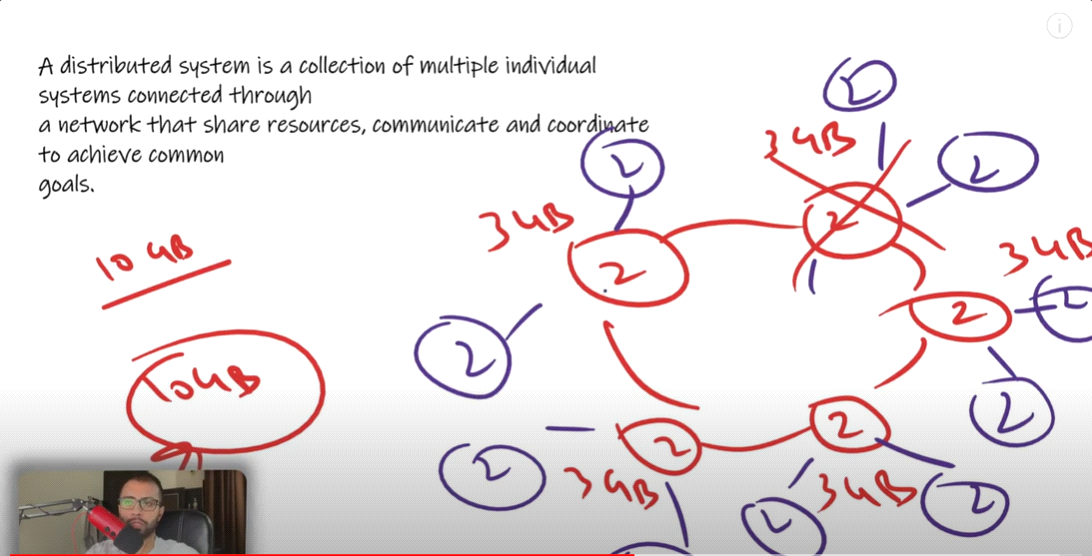


* Single point failure
* Not Scalable



**Microservices architecture**

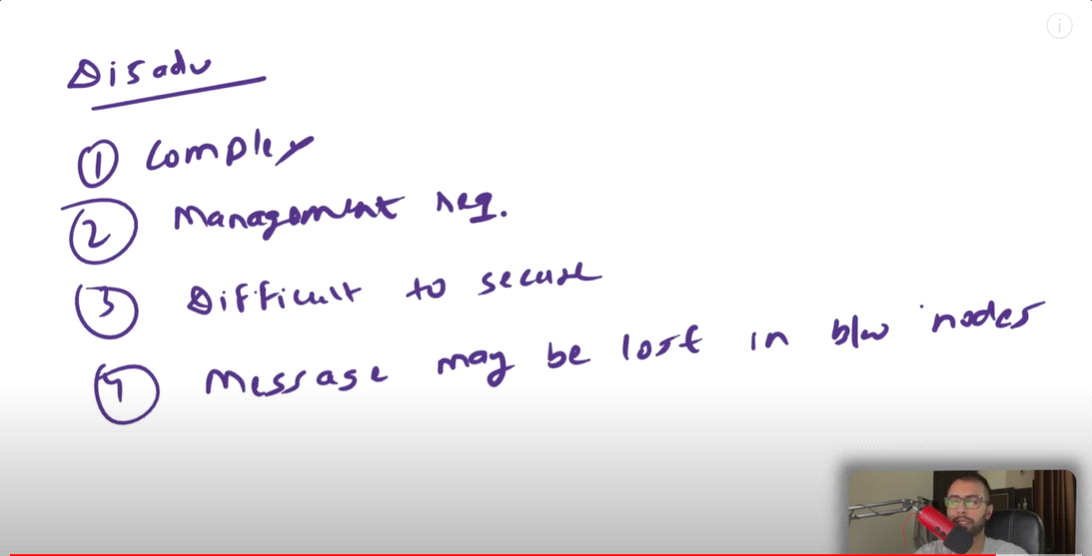




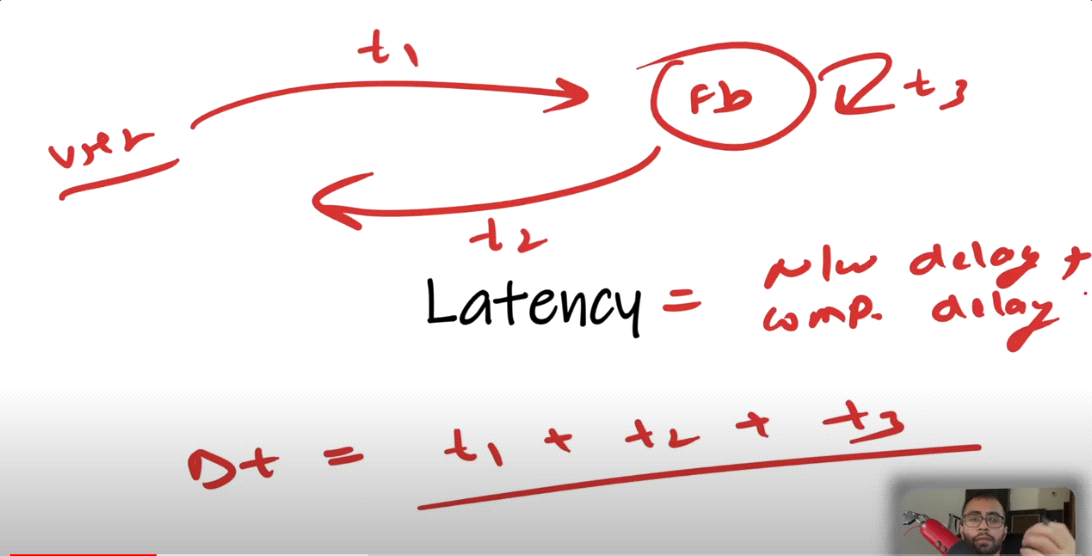
**Advantages:-**

* **Scalable :- system can be scalable in horizontally so if there one job was done by one machine then may be 2 machines distributes the job in half-half.**
* **SPOF:- Achieve the solution of single point of failure.**
* **Low latency**

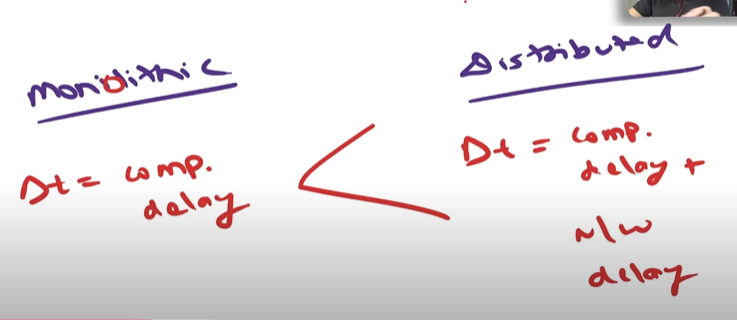
**Disadvantages:-**



**Latency**



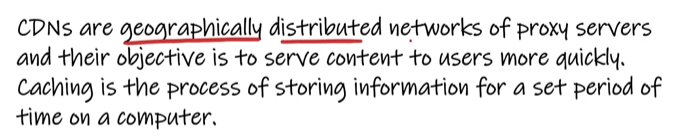
**Latency = Network Delay + Computational Delay**



**Reduce Latency**

**-> Caching**

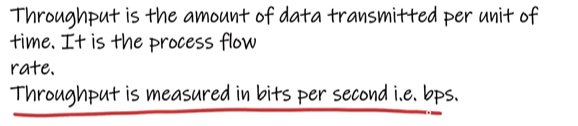
**-> CDN (Content Delivery Network)**

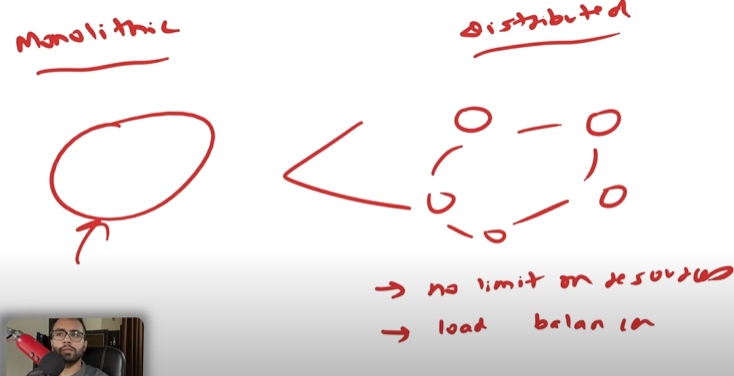


**-> Upgrade System**

**Throughput**

**The volume of work or infromation flowing through a system.**





**Causes of Low Throughtput**

**-> Latency**

**-> protocol overhead**

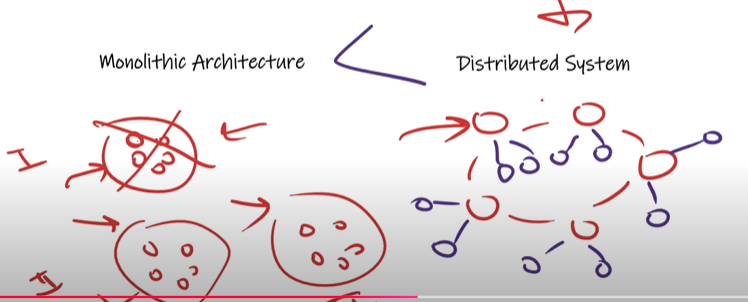
**-> Congestion  
  
Improving Throughput**

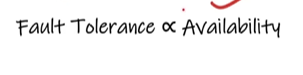
**-> CDN  
-> Caching**

**-> DS**

**-> Load Balacer  
-> Improve Resources**

**Availability**



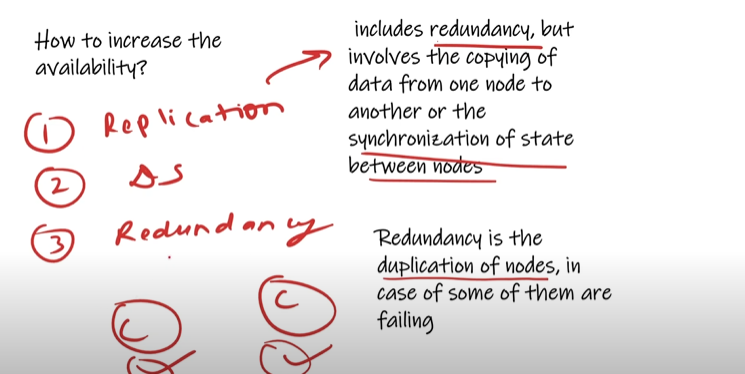


**How to increase the abailability**

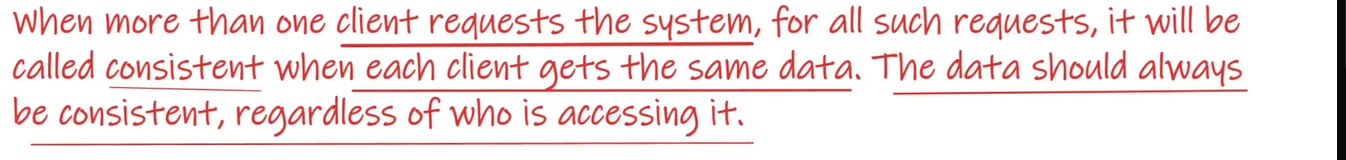
**->Replication**

**-> DS**

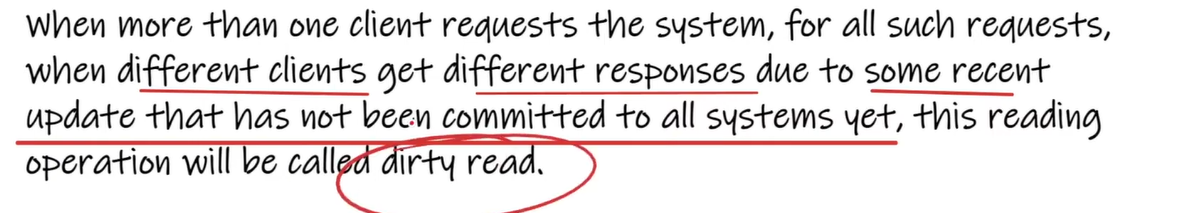
**--> Redundancy**

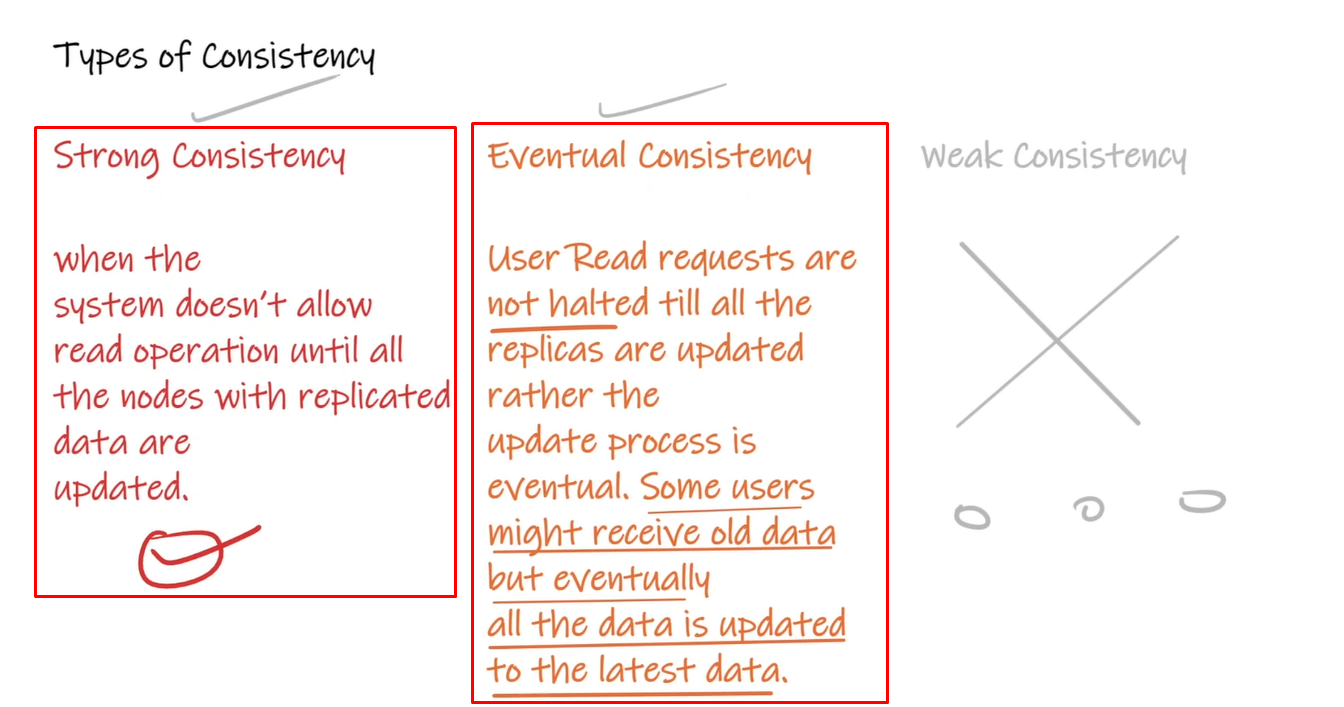
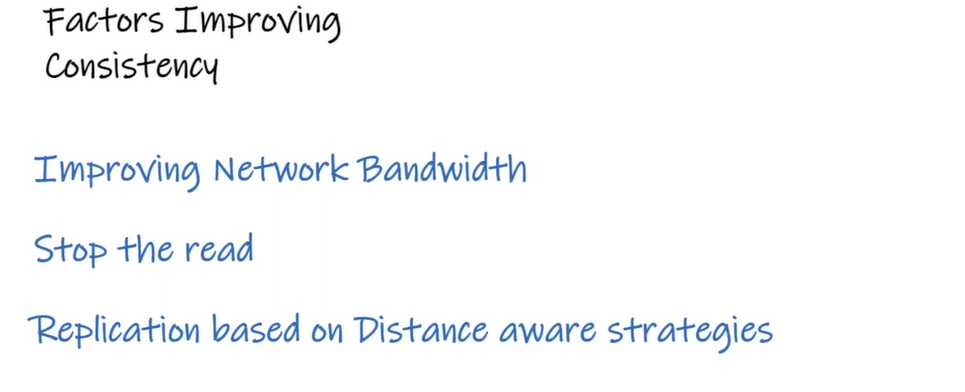


**Consistency**

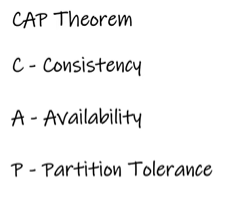


**Usually Monolithik System is More consistant than Microlitjik beause in monolithic all structure and database available on one server**



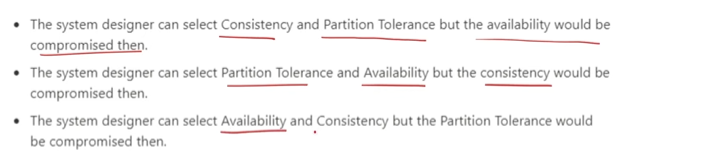
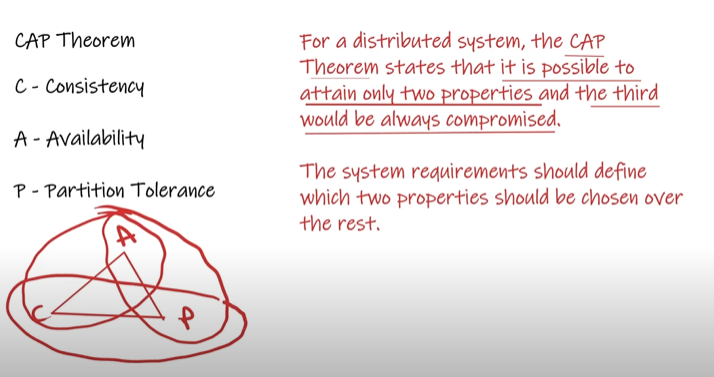


**CAP Theorem**

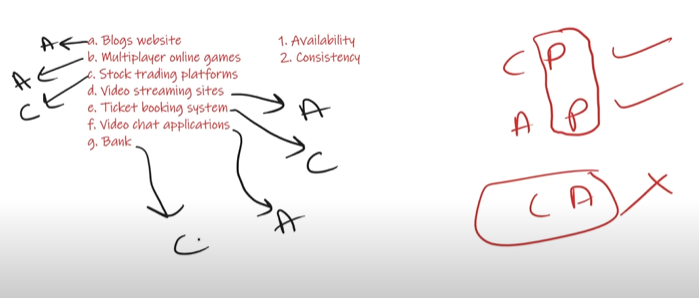
 **Consistency - > Synchronized**

**Availability -> High fault Tolerance**

**Partition Tolerance -> made up with two words Partition ( In Distributed System or microlithic architecture application is divided in components and deployed on different server but when we any server goes down it is called Network Partitining) Tolerance ( It basically means we have to make less chances of failur)**







**Scalability**

1. **Vartical Scaling : In one machine increase only hardware specification.**

***Pros :***

**Easy implementation**

**Less power in mentaince**

**Management is easy**

***Cons:***

**SPOF**

**Limit**

**price**

1. **Horizontal Scaling: We increase By same machine acc. To need**

**Pros:**

**Reduces SPOF**

**No limit**

**Price is low comparing with vertical scaling**

**Cons:**

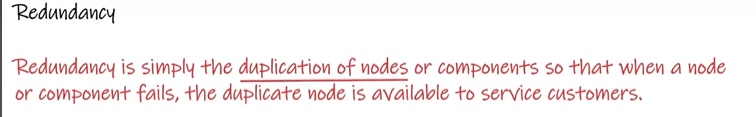
**Power Consumption is more**

**Implementation is tough**

**Management is difficult**

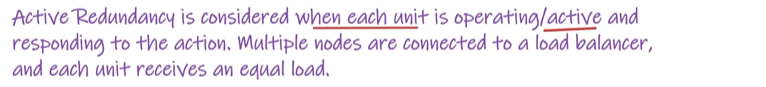
**Difference between Redundancy and Replication**

**Redundancy**

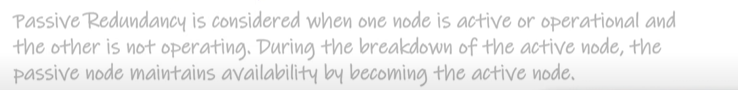


**Types Of Redundancy**

**1.**



**2.**



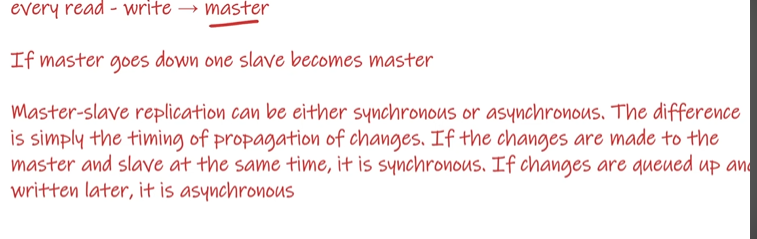
**Replication**



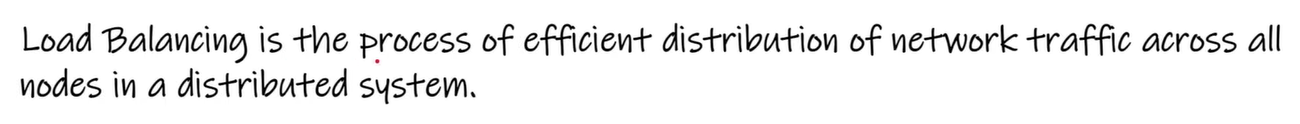
**At same time data should be same on all the severs**

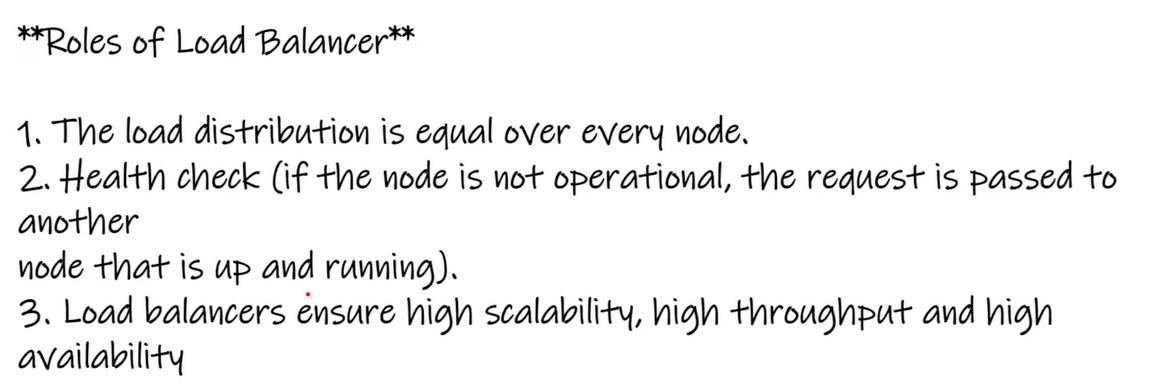
**Basically Replication use in database**

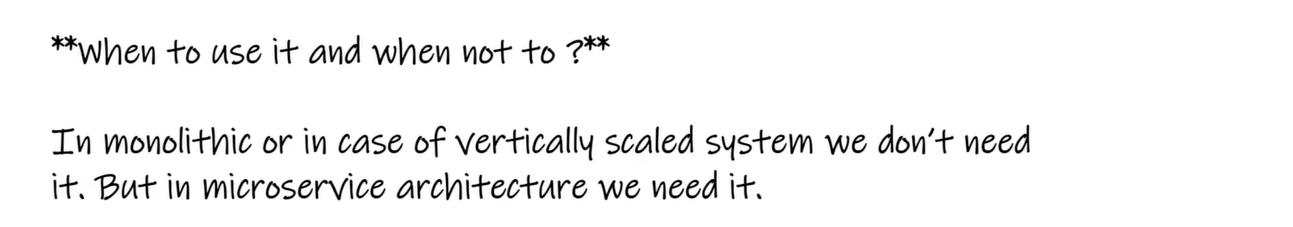
1. **Active replication is a computing technique that processes the same request at every replica. In an active replication environment, nodes can replicate data to other nodes in a cluster. This allows all nodes to handle read and write operations simultaneously, unlike traditional master-slave configurations.**
2. **Passive replication**

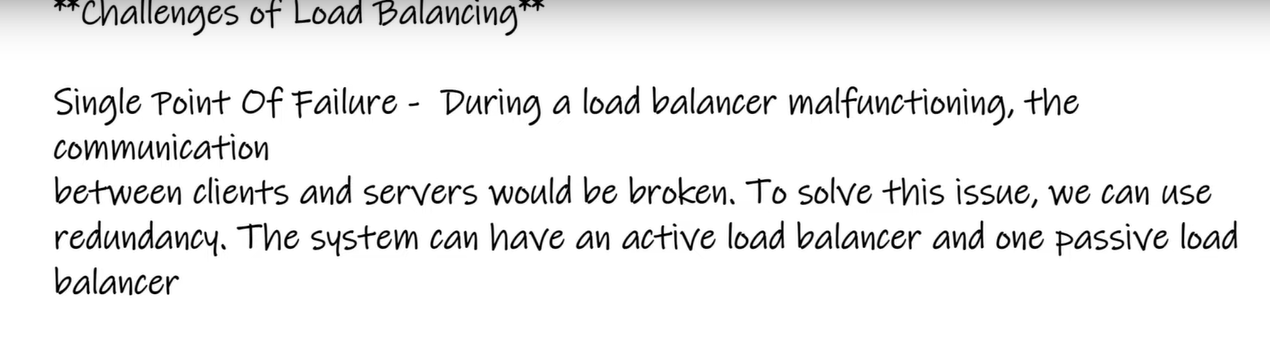


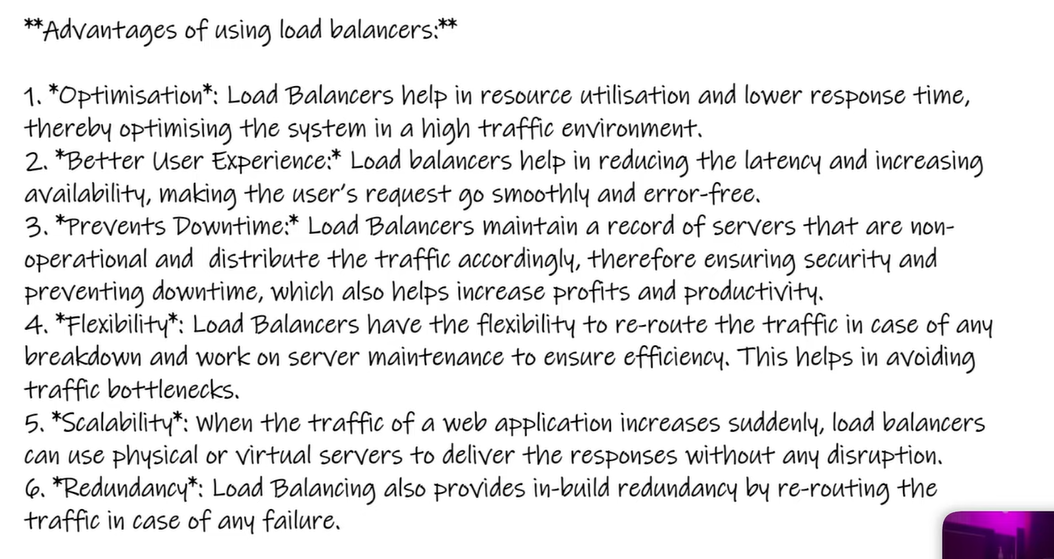
**load balancer**

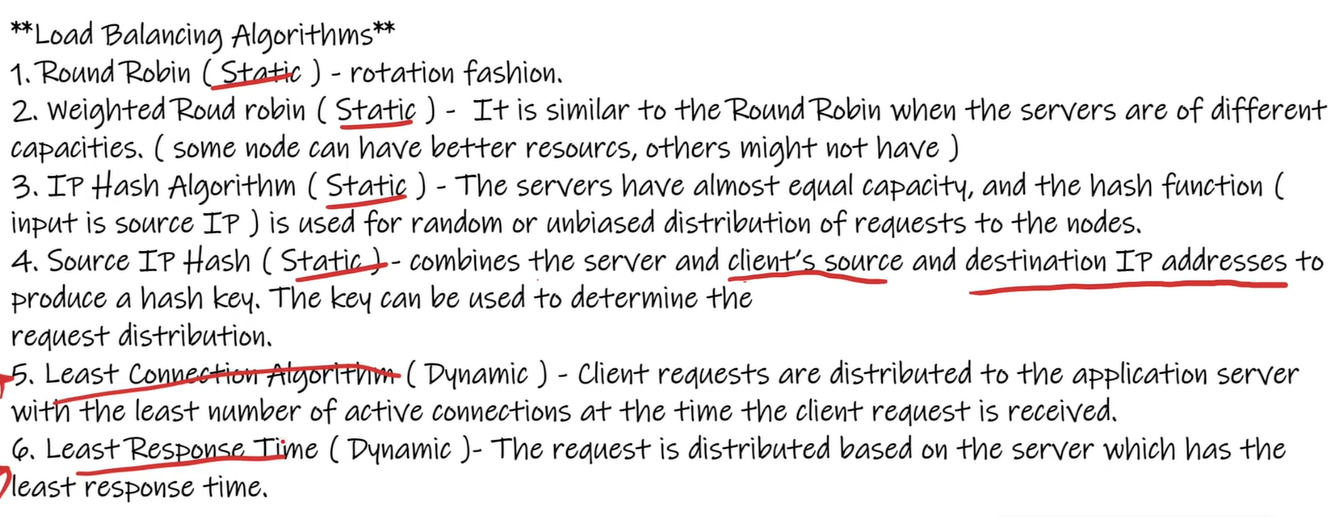










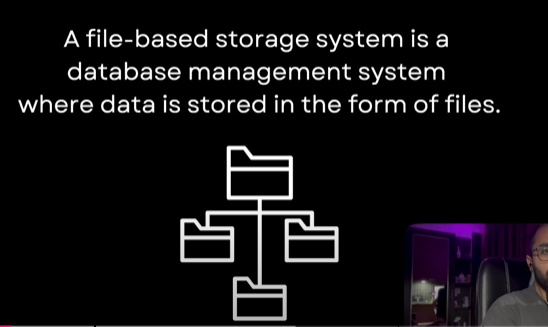


**Caching**

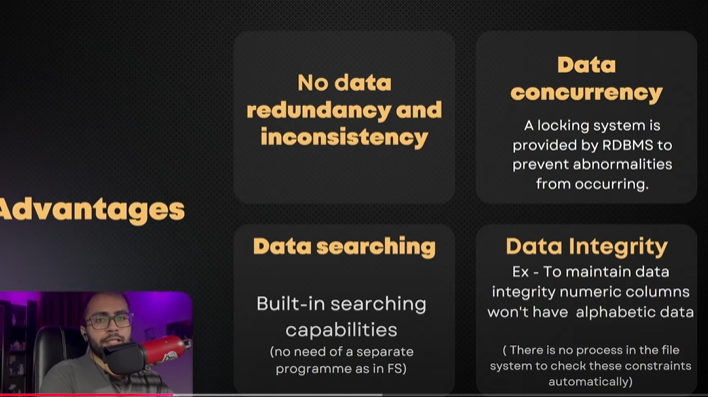
**Caching is a technique that stores data in a cache, a hardware or software component, to make it available faster when requested again. Caching can improve a system's performance and scalability by making data more accessible**

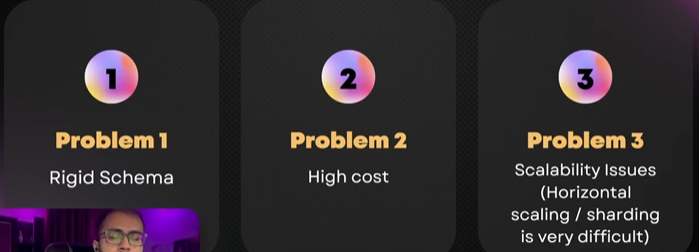
**Read Some More From Google**

**File based storage system**

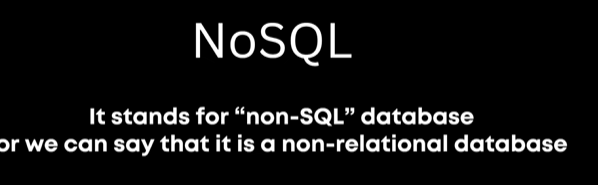


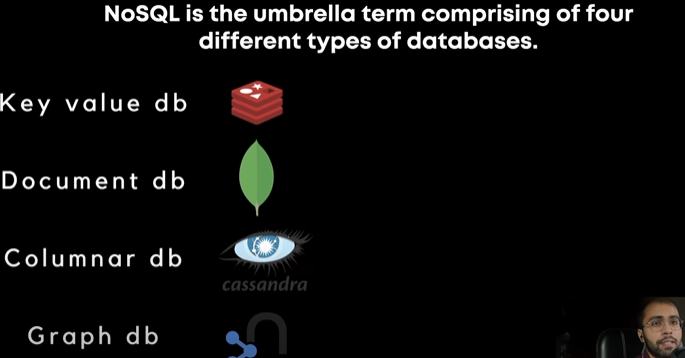
**RDBMS**

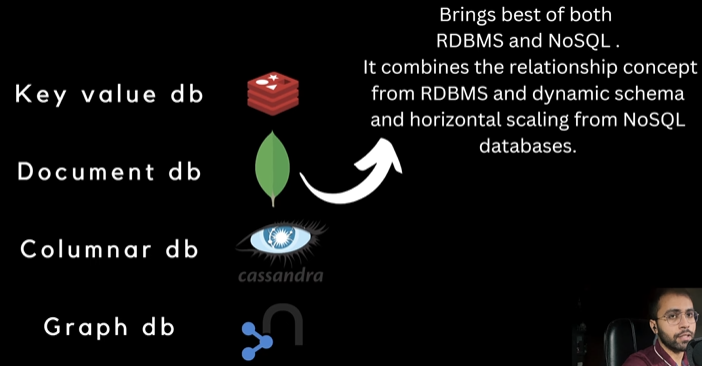


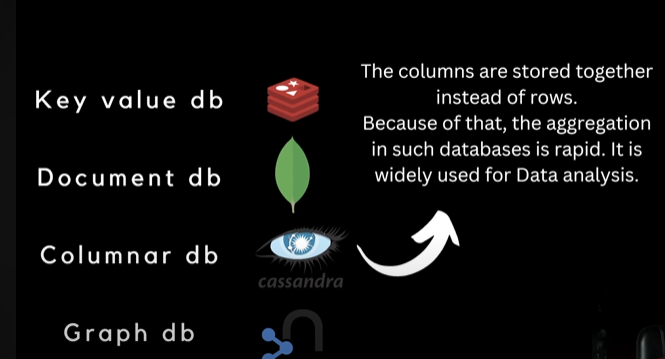


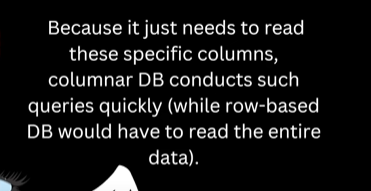
**NoSQL Databases**



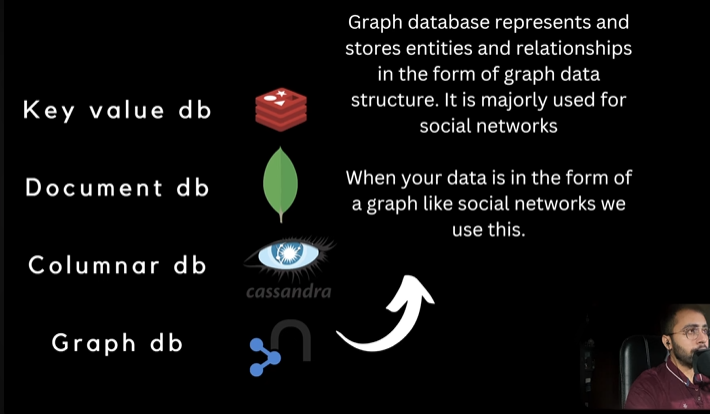








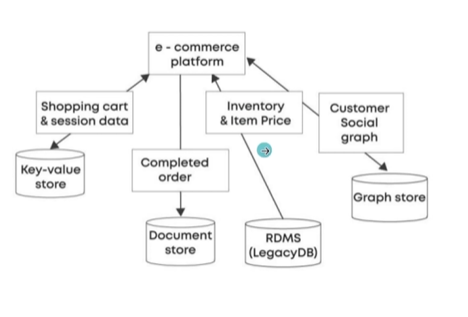




**Polyglot Persistence**

**when one data base is not enough for our application and we implement multiple database in one pplication then it is know as Polyglot Persistence.**





**What is denormalization in RDBMS**

**Normalization**

 **Denormalization**

