A single point of failure(SPOF) in computing is a critical point in the system whose failure can take down the entire system. A lot of resources and time is spent on removing single points of failure in an architecture/design.

Single points of failure often pop up when setting up coordinators and proxies. These services help distribute load and discover services as they come and leave the system. Because of the critical centralized tasks of these services, they are more prone to being SPOFs.

One way to mitigate the problem is to use multiple instances of every component in the service. The graph of dependencies then becomes more flexible, allowing the system to resiliently switch to another service instead of failing requests.

Another approach is to have backups that allow a quick switch over on failure. The backups are useful in components dealing with data, like databases.

Allocating more resources, distributing the system and replication are some ways of mitigating the problem of SPOF. Hence designs include horizontal scaling capabilities and partitioning.

It is important to note that the CAP theorem does not allow removing SPOFs if perfect consistency is required.

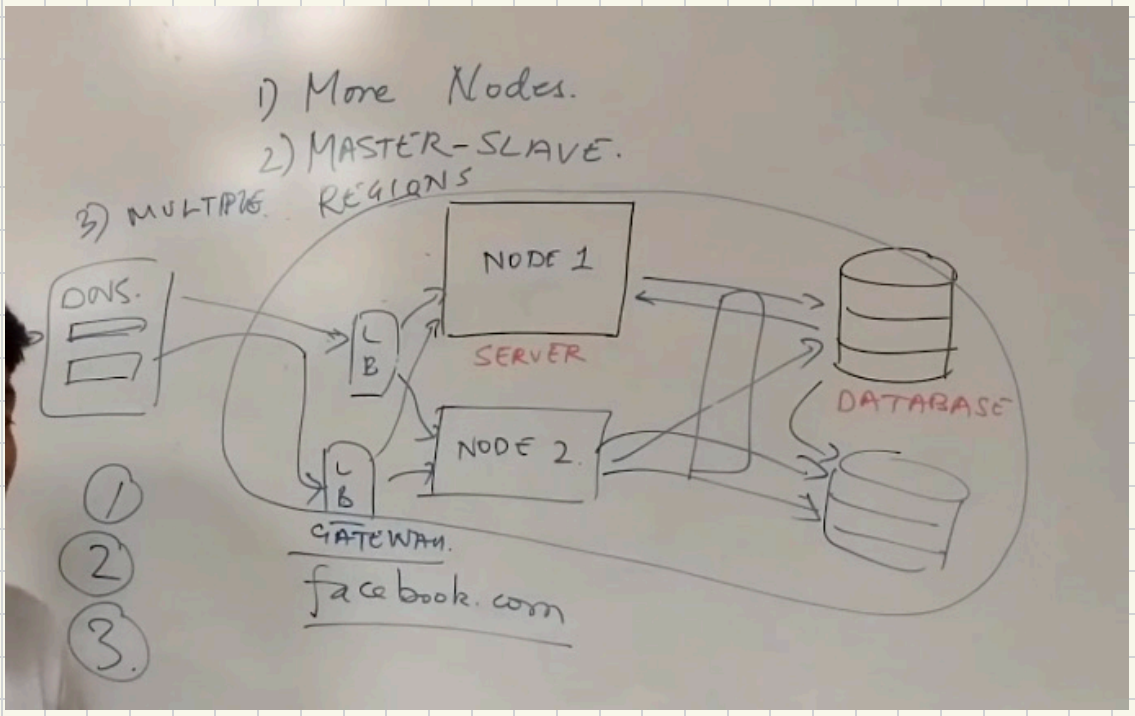
CHAOS MONKEY, NETFLIX
https://docs.oracle.com/cd/E19693-01/819-0992/fjdch/index.html

https://stackoverflow.com/questions/7943211/web-app-high-availability-how-to-prevent-a-single-point-of-failure

https://ieeexplore.ieee.org/document/6214760/

http://www.spkaa.com/blog/how-to-deal-with-single-points-of-failure-software/

https://en.wikipedia.org/wiki/Single_point_of_failure


1) More Nodes.
2) MASTER-SLAVE.
3) MULTIPLE REGIONS
DNS.
LB
NODE 1
SERVER
DATABASE
NODE 2.
LB
GATEWAY.
facebook.com
1
2
3.