Availability

...

What is Availability

- Availability: the percentage of time that the system is functioning correctly
- “High availability” is what we strive for.
- How important is availability?
  - Depends upon system!
  - Backup generators to ensure continuous power supply in hospitals: VERY
  - Random blog: NOT VERY

What is Availability

- Availability often measured in percentage of uptime, in a chart like this

- SLA (Service Level Agreements) will promise “nines” of availability

Availability %	Downtime per year	Downtime per month	Downtime per week
90% (1 nine)	36.5 days	72 hours	16.8 hours
99% (2 nines)	3.65 days	7.20 hours	1.68 hours
99.5%	1.83 days	3.60 hours	50.4 minutes
99.9% (3 nines)	8.76 hours	43.8 minutes	10.1 minutes
99.95%	4.38 hours	21.56 minutes	5.04 minutes
99.99% (4 nines)	52.56 minutes	4.32 minutes	1.01 minutes
99.999% (5 nines)	5.26 minutes	25.9 seconds	6.05 seconds
99.9999% (6 nines)	31.5 seconds	2.59 seconds	0.605 seconds
99.99999% (7 nines)	3.15 seconds	0.259 seconds	0.0605 seconds

Reasons for System Failures

- Software crashes
  - OOM
  - bugs in code
  - Corruption of files on disk
- Hardware failures
  - Machine failures
  - Network outages
- Human Error
  - Configuration and deployment mistakes
  - Example: Google took down much of Japan's internet via config error

Achieving High Availability - Fault Tolerance

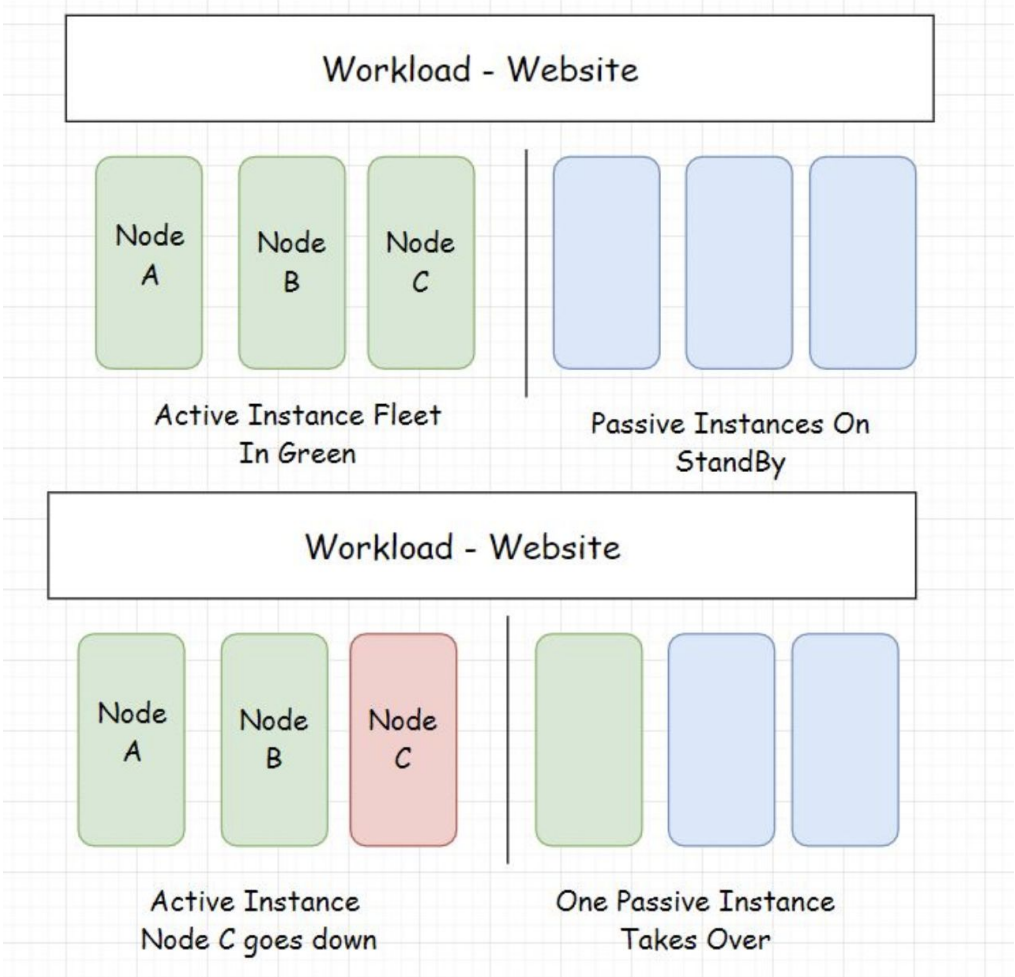
- Fault Tolerance: system's ability to continue operating properly in the event of the failure of one or more of its components
- In a fault-tolerant system:
  - Several instances/nodes running a service
    - a few go offline and bounce back without issue
  - In case of these internal failures, the system can work at a reduced level without going down entirely.
- In the case of backend node failures, a few services of the app, such as image upload, post likes, etc., may stop working.
  - However, the application as a whole will still be up.
  - Also known as fail soft.

Redundancy

- Redundancy: the intentional duplication of critical components or functions of a system with the goal of increasing reliability of the system
- Cost-Availability Balance
    - Higher availability usually comes with higher compute costs

Redundancy

- One approach:
  - Duplicating the server instances
  - Keeping duplicates on standby to take over in case any of the active server instances go down.
- Issues with this?



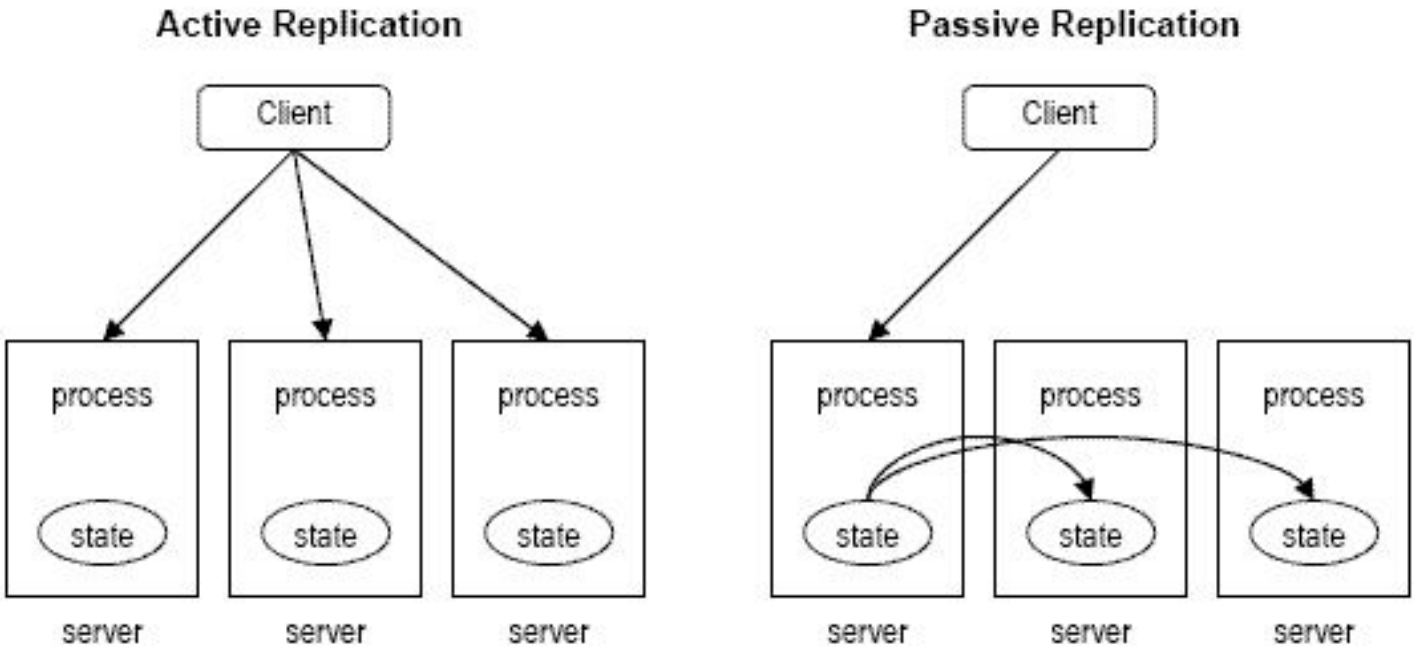
Redundancy

- Better Approach:
  - All servers share load
  - Load redistributed when one or more go down

Replication

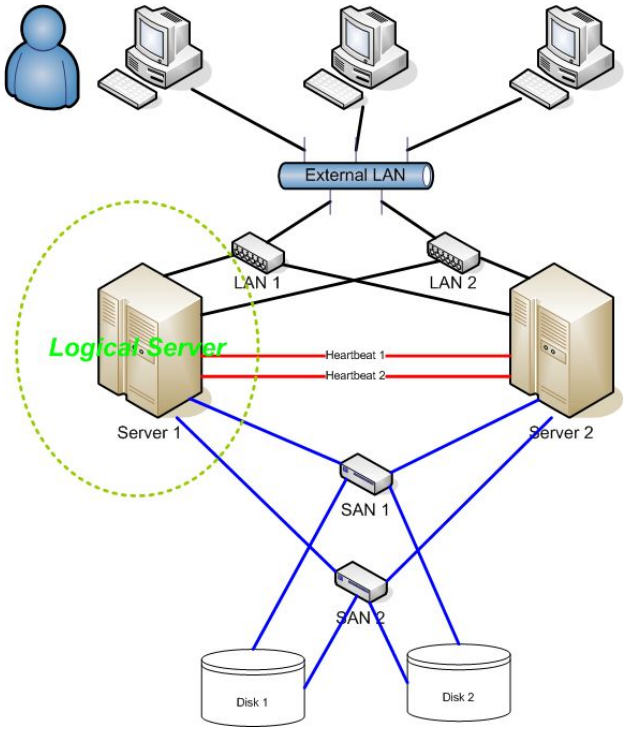
- Replication of data:
  - Duplicating in various places (DBs, machines, data centers) so that if one fails, you have extras
- Google data replication for eleven 9's of durability
  - That means that even with one billion objects, you would likely go a hundred years without losing a single one.
- Geographical distribution of workload

Replication



High Availability Clustering

- High availability clusters are groups of hosts (physical machines)
  - act as a single system and provide continuous availability.
- all hosts in the cluster must have access to the same shared storage.
- In any case of failure, a virtual machine (VM) on one host can failover to another host, without any downtime.
- The number of nodes in a high availability cluster can vary between two to dozens of nodes



Planning your pipeline is important

- Do you need low latency
- Do you need high throughput
- What size do you need to scale to
- What are your cost constraints
- Where are your users located
- Do you need high availability

In Class Work

- Get into groups of 3-6, with which you will build your Final Project
  - Also available on canva
- Choose between your design patterns submissions, and make one version that is the best
  - You will actually be building this!
  - Can be a frankenstein or combination of several of them
  - This is an exercise in collaboration and communication!
- Show me when done (if you finish. But it is OK if you need more time)