Week 1 - Lesson 3 - Scalability : vertical vs horizontal

1 What does "scaling" really mean?

Scaling = handling more load without breaking the system

Load can be :

- more users
- more requests
- more data

2 Vertical scaling (Scale Up)

What it is

- Make one machine stronger
- more CPU, RAM, Disk

Example

- Upgrade from 8 GB RAM → 64 GB RAM

Pros

- Simple
- No code changes

Cons (very important)

- Hardware limits
- Very expensive
- Single point of failure

If this machine dies → system dies

3 Horizontal Scaling (Scale Out)

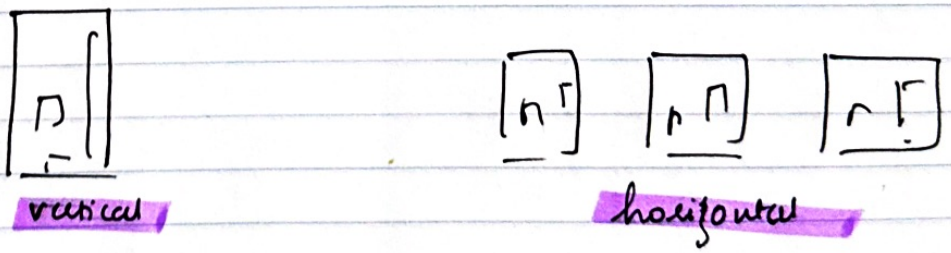
What it is
- Add more machines
- Distribute the load

Example
- 1 server → 10 servers → 100 servers

Pros
- Practically unlimited scale
- Fault tolerance
- Cheaper

Cons
- Coordination complexity
- Data consistency issues
- Needs load balancing

4 Visual comparison



5 Why big systems prefer horizontal scaling

Let's say:
- One server handles 1000 req/sec

Then:
- 10 servers → 10,000 req / sec
- One server fails → system still works

This is how:
- Google
- Netflix
- Instagram

scale

6 The hidden problem of horizontal scaling

Here's where system design becomes interesting

Problem 1: State

If user logs in on Server A:
- what if next request goes to Server B?

→ State must move out of servers

Problem 2: Data

If 10 servers write to same DB:
- DB becomes bottleneck

→ DB must have also scale

7 Important rule

Servers must be stateless to scale horizontally.

We'll revisit this many times

8 Mini Design Exercise

Task 3

You have:
- 5 application servers
- 1 database

Traffic increase 10x

Answer

1. What breaks first?
2. Why can't you just add more app servers!
3. What does this tell you about.