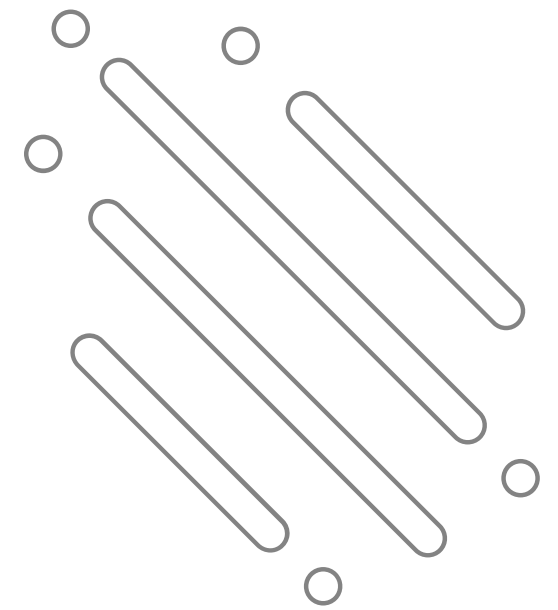
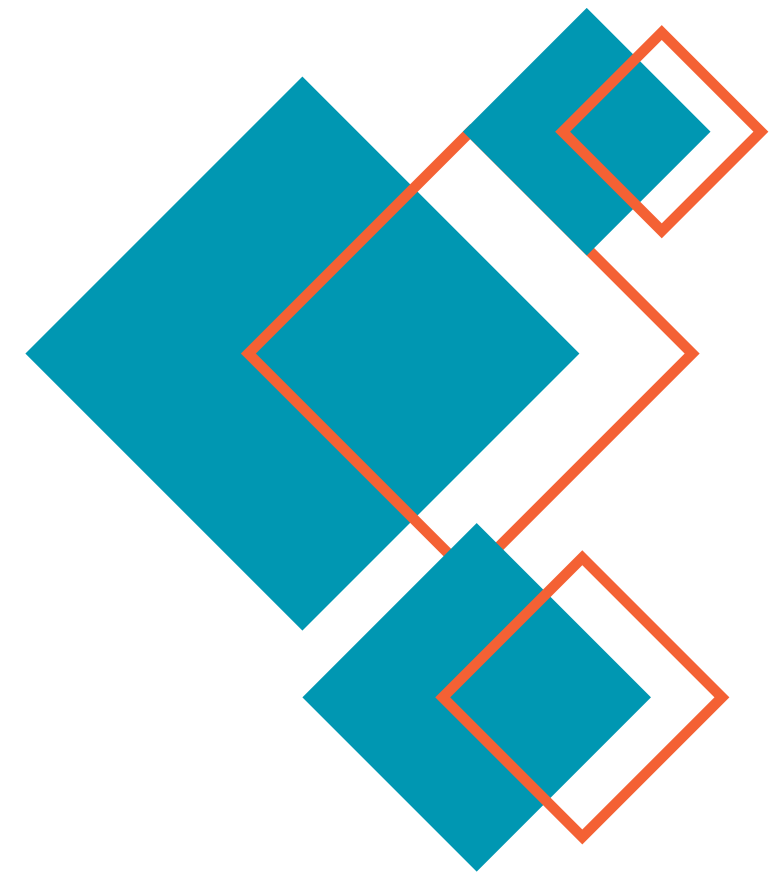




SYSTEM DESIGN



Scalability (স্কেলেবিলিটি)



Scalability (স্কেলেবিলিটি) কী?

Scalability (স্কেলেবিলিটি) system design-এর একটি অত্যন্ত গুরুত্বপূর্ণ বিষয়। Scalability system design-এর প্রাণ ও বলা যেতে পারে। ভবিষ্যতের আপনার সিস্টেম এর গ্রোথ ও ইউজার এক্সপেরিয়েন্স ভালো রাখতে হলে scalable architecture ডিজাইন করা অত্যন্ত দরকারি।

তাহলে Scalability বলতে কী বোঝায়?

Scalability হলো একটি সিস্টেমের লোড বাড়ার সাথে মানিয়ে নিয়ে টিকে থাকার সিস্টেম।

একটি সিস্টেম যখন কম ব্যবহারকারীর জন্য কাজ করে, তখন তার রিসোর্স কম লাগে।

কিন্তু যখন মিলিয়ন ব্যবহারকারী একসাথে ইউজ করে, তখন সেই সিস্টেমকে বড় করতে হয় যাতে পারফরমেন্স খারাপ না হয়।

SYSTEM DESIGN-এ SCALABILITY আমাদের কেন প্রয়োজন?

ব্যবহারকারী বাড়লে যেত সমস্যা না হয়:

1. ধরুন একটি অ্যাপ আজকে ১০০ জন ব্যবহার করছে, কালকে ১ লাখ হয়ে গেলো তখন হটাৎ কি হবে
2. স্কেলেবল ডিজাইন না থাকলে সার্ভার ক্র্যাশ করবে।
3. অনেক বেশি ট্র্যাফিক আসলে পারফরমেন্স ঠিক রাখতে স্কেল করতে হয়।

Scalability সাধারণত দুই ধরনের হয়ে থাকে :

Vertical Scaling (Scale-Up) - একটি সার্ভারের র‍্যাম, CPU, SSD বাড়িয়ে শক্তিশালী করা

Horizontal Scaling (Scale-Out) - একাধিক সার্ভার/নোড যোগ করে কাজ ভাগ করে নেওয়া

VERTICAL SCALING

আপনার যেই সার্ভার আছে, সেই একই সার্ভারে hardware upgrade করে তার performance বাড়ানোই হলো Vertical Scaling।

উদাহরণ:

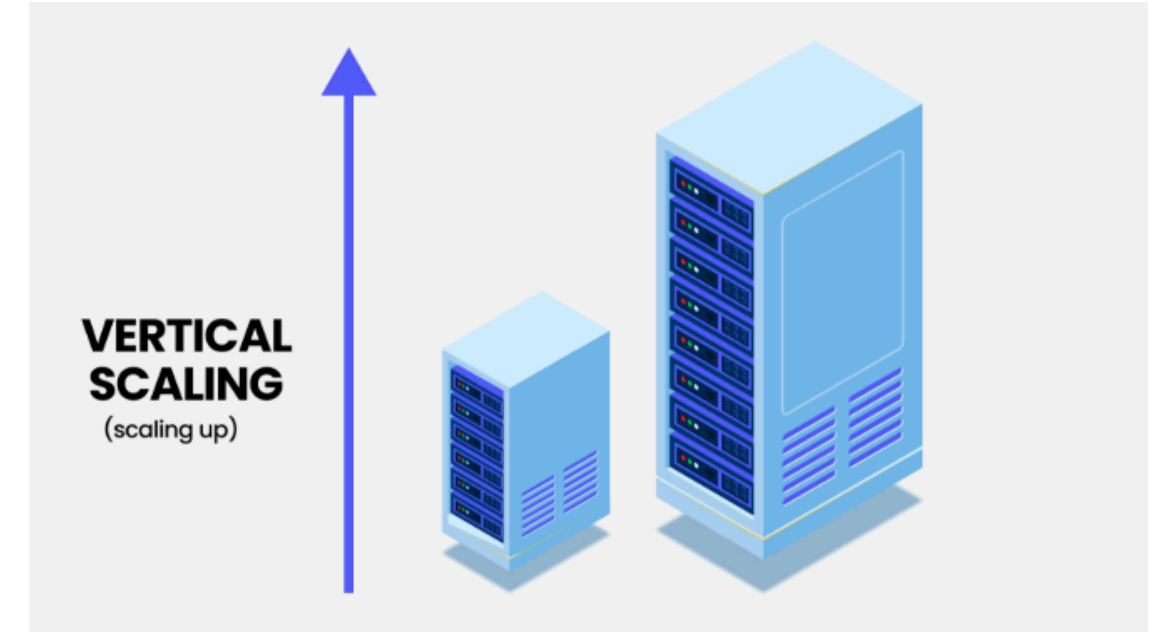
ধরুন আপনার Laravel অ্যাপ এখন একটি 2 GB RAM এবং 2 Core CPU বিশিষ্ট সার্ভারে চলছে।

কিন্তু অ্যাপে ইউজার বেড়ে যাচ্ছে এবং সার্ভার চাপ নিতে পারছে না।
তখন আপনি নিচের মতো করবেন:

2 GB RAM → 8 GB RAM

2 vCPU → 4 vCPU

মানে: সার্ভার টা বদলালেন না, কিন্তু সেটার স্পেস বাড়ালেন।



VERTICAL SCALING এর সুবিধা গুলো কি কি এবং অসুবিধা গুলো কি কি ?

সুবিধা:

ইমপ্লিমেন্ট করা সহজ, শুধু হার্ডওয়্যার আপগ্রেড করবেন।
কমপ্লেক্সিটি কম

অসুবিধা:

সীমিত স্কেল (এক সময় হার্ডওয়্যারের সীমা পেরিয়ে যায়) কস্ট বেশি হতে পারে
একটাই ফেইল হলে সব কিছু বন্ধ

Horizontal Scaling কী?

একটি বড় সার্ভারে resource বাড়ানোর পরিবর্তে (Vertical Scaling), আপনি আরও নতুন সার্ভার যোগ করেন – যাতে সবার মাঝে লোড ভাগ হয় এবং সার্ভার চাপ নিতে পারে।

ধরুন Laravel অ্যাপ ১টি সার্ভারে চলছে। ট্রাফিক বাড়ছে।
তখন আপনি এমন করবেন:

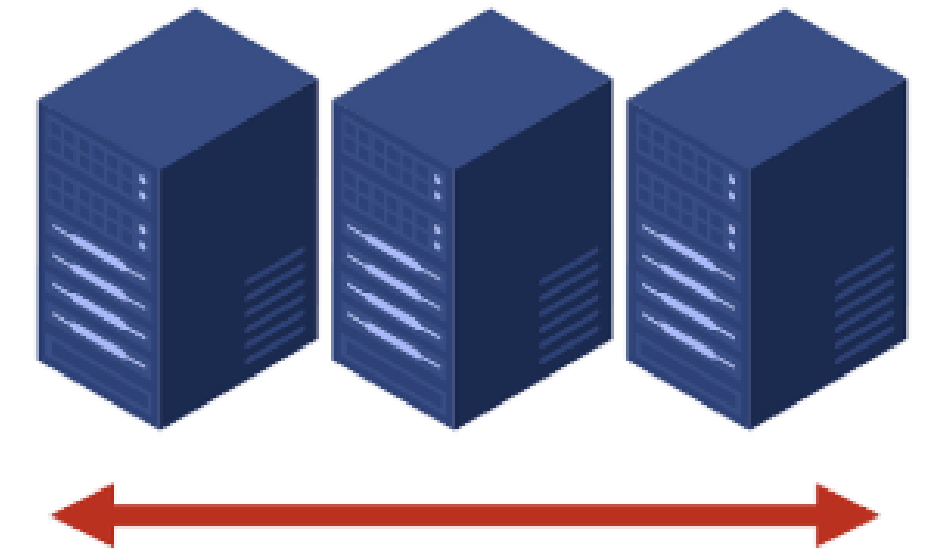
1 Server → 3 Servers

এখন ইউজারদের মধ্যে লোড ভাগ হয়ে যাবে:

- Server 1: কিছু request
- Server 2: কিছু request
- Server 3: কিছু request

Horizontal Scaling

Add more resources like virtual machines to your system to spread out the workload across them.



কিভাবে কাজ করে?

1. আপনার একাধিক সার্ভার থাকবে (Load Balanced)
2. Front এ থাকবে একটি Load Balancer
3. Load Balancer request গুলো ভাগ করে দেবে backend সার্ভারগুলোর মধ্যে

Optional:

- Database replication / clustering দরকার হতে পারে
- Session কে Redis/Database এ রাখতে হয়, যেন সব সার্ভার access করতে পারে

Horizontal Scaling এর সুবিধা:

1. High Availability: একাধিক সার্ভার, তাই একটিতে সমস্যা হলে অন্যটি চালু থাকে
2. No hardware limit: প্রয়োজনমতো সার্ভার যোগ করতে পারবেন
3. Auto-scaling support: Cloud platform গুলোতে অটো স্কেল করা যায়
4. Zero Downtime Deployment সম্ভব

অসুবিধা:

1. ইমপ্লিমেন্ট জটিল তুলনামূলক ভাবে।
2. Load balancing, distributed systems, consistency সমস্যা তৈরি হতে পারে

SCALABILITY DESIGN প্যাটার্ন করার টেকনিক শিখিয়ে দেই :

Load Balancing - মানে রিকোয়েস্ট গুলো বিভিন্ন সার্ভারে ভাগ করে দেওয়া।

Caching - বারবার রিকোয়েস্ট করা ডেটা মেমোরি তে রেখে দ্রুত নিয়ে আশা।

Database Sharding - আপনার বড় ডেটাবেসকে ছোট ছোট অংশে ভাগ করে আলাদা সার্ভারে রাখা।


Replication - ডেটার একাধিক কপি রাখলে ডাটা রিড করা দ্রুত হয়।

Asynchronous Processing - Heavy task গুলোকে queue তে রেখে পরে প্রসেস করা।

CDN (Content Delivery Network) স্ট্যাটিক ফাইলগুলো ইউজারের কাছাকাছি সার্ভারে রাখা।

VERTICAL VS HORIZONTAL SCALING:

বিষয়	Vertical Scaling	Horizontal Scaling
কৌশল	1 সার্ভারে resource বাড়ানো	নতুন সার্ভার যোগ করা
Setup সহজ?	হ্যাঁ	না (Load balancer, config লাগে)
Limit আছে?	আছে (RAM, CPU-এর limit)	নেই (scalable)
Cost	কম	বেশি (বেশি সার্ভার মানে বেশি খরচ)
Reliability	কম (single point of failure)	বেশি (redundant system)



শেষ কথা:

সুতরাং এই ছিল আজকের ছোট আলোচনা আমরা Scalability খুব ভালো করে জানলাম এগুলো আমাদের কি? কেত দরকার উপকারিতা গুলো কি কি। পরবর্তীতে আমরা Basic Networking নিয়ে আলোচনা করবো সাথেই থাকুন