HLD 4th question

There are several architectural improvements we can implement to enhance **high availability**, **reliability**, **and scalability** across the platform:

- **Horizontal Scaling:** Each microservice—including the load balancer, CDN, and API layers—can be horizontally scaled to handle increased traffic and ensure zero downtime.
- **Database Sharding & Indexing:** For better performance and distribution, we can shard the database based on user ID or asset type. Additionally, efficient indexing will significantly speed up query retrievals for user portfolios.
- **Optimized Read/Write Separation:** A standard practice is to route all **write operations** to the primary database while handling **read operations** through multiple replicas. This improves throughput and ensures consistency.
- **Stateless Microservices:** By designing services to be stateless, they can be replicated easily and scaled independently without session affinity.
- **Caching Layers:** Frequently accessed data such as portfolio snapshots or asset metadata can be cached using Redis/Memcached to reduce database hits and improve response times.

These strategies together ensure the system can reliably serve real-time portfolios for hundreds or thousands of users while maintaining performance even under heavy load.