

 ashishps1 /
awesome-system-design-resources




 Code  Issues 2  Pull requests  Actions  Projects  Security  Insights



Learn System Design concepts and prepare for interviews using free resources.




blog.algomaster.io



 GPL-3.0 license


 22.1k stars  5.3k forks  314 watching  Branches  Activity


 Tags


 Public repository

 2 Branches  0 Tags  

 Go to file 

Go to file 

Add file 

Code 



ashishps1 Update README.md

01d4d20 · 2 days ago 

 diagrams

update interview template

last year

 implementations

add consistent hashing implementati...

2 months ago

 LICENSE

Add LICENSE

2 years ago


 README.md

Update README.md

2 days ago



 README

 GPL-3.0 license



This repository contains free resources to learn System Design concepts and prepare for interviews.

 Subscribe to my [AlgoMaster Newsletter](#) and get a **FREE System Design Interview Handbook** in your inbox.

 If you are new to System Design, start here: [System Design was HARD until I Learned these 30 Concepts](#)

System Design Key Concepts

- [Scalability](#)

- [Availability](#)
- [CAP Theorem](#)
- [ACID Transactions](#)
- [Consistent Hashing](#)
- [Rate Limiting](#)
- [SPOF](#)
- [Fault Tolerance](#)
- [Consensus Algorithms](#)
- [Gossip Protocol](#)
- [Service Discovery](#)
- [API Design](#)
- [Disaster Recovery](#)
- [Distributed Tracing](#)

System Design Building Blocks

- [APIs](#)
- [Content Delivery Network \(CDN\)](#)
- [Proxy vs Reverse Proxy](#)
- [Domain Name System \(DNS\)](#)
- [Caching](#)
- [Caching Strategies](#)
- [Distributed Caching](#)
- [API Gateway](#)
- [Load Balancing](#)
- [Databases Types](#)
- [SQL vs NoSQL](#)
- [Database Indexes](#)
- [Consistency Patterns](#)
- [HeartBeats](#)
- [Circuit Breaker](#)
- [Idempotency](#)
- [Database Scaling](#)
- [Data Replication](#)
- [Data Redundancy](#)
- [Database Sharding](#)
- [Database Architectures](#)
- [Failover](#)
- [Bloom Filters](#)
- [Message Queues](#)
- [WebSockets](#)
- [Checksums](#)

- [Microservices Guidelines](#)
- [Distributed Locking](#)



System Design Tradeoffs

- [Top 15 Tradeoffs](#)
- [Vertical vs Horizontal Scaling](#)
- [Concurrency vs Parallelism](#)
- [Long Polling vs WebSockets](#)
- [Batch vs Stream Processing](#)
- [Stateful vs Stateless Design](#)
- [Strong vs Eventual Consistency](#)
- [Read-Through vs Write-Through Cache](#)
- [Push vs Pull Architecture](#)
- [REST vs RPC](#)
- [Synchronous vs. asynchronous communications](#)
- [Latency vs Throughput](#)

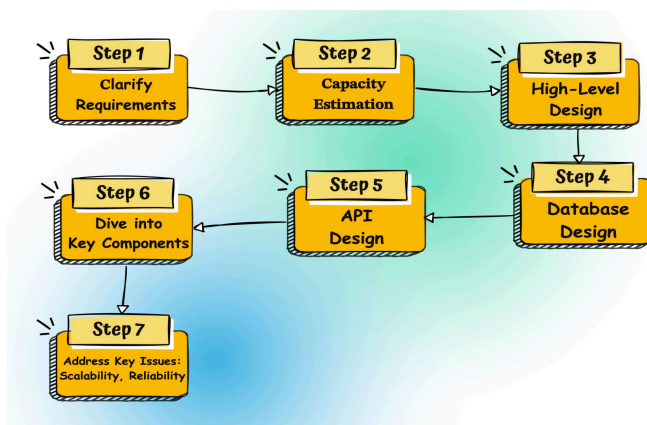


System Design Architectural Patterns

- [Client-Server Architecture](#)
- [Microservices Architecture](#)
- [Serverless Architecture](#)
- [Event-Driven Architecture](#)
- [Peer-to-Peer \(P2P\) Architecture](#)



How to Answer a System Design Interview Problem



[Read the Full Article](#)



System Design Interview Problems

Easy

- [Design URL Shortener like TinyURL](#)
- [Design Text Storage Service like Pastebin](#)
- [Design Leaderboard](#)
- [Design Content Delivery Network \(CDN\)](#)
- [Design Parking Garage](#)
- [Design Vending Machine](#)
- [Design Distributed Key-Value Store](#)
- [Design Distributed Cache](#)
- [Design Authentication System](#)
- [Design Unified Payments Interface \(UPI\)](#)

Medium

- [Design WhatsApp](#)
- [Design Spotify](#)
- [Design Distributed Job Scheduler](#)
- [Design a Scalable Notification Service](#)
- [Design Instagram](#)
- [Design Tinder](#)
- [Design Facebook](#)
- [Design Twitter](#)
- [Design Reddit](#)
- [Design Netflix](#)
- [Design Youtube](#)
- [Design Google Search](#)
- [Design E-commerce Store like Amazon](#)
- [Design TikTok](#)
- [Design Shopify](#)
- [Design Airbnb](#)
- [Design Autocomplete for Search Engines](#)
- [Design Rate Limiter](#)
- [Design Distributed Message Queue like Kafka](#)
- [Design Flight Booking System](#)
- [Design Online Code Editor](#)
- [Design Stock Exchange System](#)
- [Design an Analytics Platform \(Metrics & Logging\)](#)
- [Design Payment System](#)
- [Design a Digital Wallet](#)

Hard

- [Design Location Based Service like Yelp](#)
- [Design Uber](#)

- [Design Food Delivery App like Doordash](#)
- [Design Google Docs](#)
- [Design Google Maps](#)
- [Design Zoom](#)
- [Design Distributed Counter](#)
- [Design File Sharing System like Dropbox](#)
- [Design Ticket Booking System like BookMyShow](#)
- [Design Distributed Web Crawler](#)
- [Design Code Deployment System](#)
- [Design Distributed Cloud Storage like S3](#)
- [Design Distributed Locking Service](#)
- [Design Slack](#)
- [Design Live Comments](#)



Books

- [Designing Data-Intensive Applications](#)



Newsletters

- [AlgoMaster Newsletter](#)



YouTube Channels

- [Tech Dummies Narendra L](#)
- [Gaurav Sen](#)
- [codeKarle](#)
- [ByteByteGo](#)
- [System Design Interview](#)
- [sudoCODE](#)
- [Success in Tech](#)



Must-Read Engineering Articles

- [How Discord stores trillions of messages](#)
- [Building In-Video Search at Netflix](#)
- [How Canva scaled Media uploads from Zero to 50 Million per Day](#)
- [How Airbnb avoids double payments in a Distributed Payments System](#)
- [Stripe's payments APIs - The first 10 years](#)
- [Real time messaging at Slack](#)



Must-Read Distributed Systems Papers

- [Paxos: The Part-Time Parliament](#)
- [MapReduce: Simplified Data Processing on Large Clusters](#)
- [The Google File System](#)
- [Dynamo: Amazon's Highly Available Key-value Store](#)
- [Kafka: a Distributed Messaging System for Log Processing](#)
- [Spanner: Google's Globally-Distributed Database](#)
- [Bigtable: A Distributed Storage System for Structured Data](#)
- [ZooKeeper: Wait-free coordination for Internet-scale systems](#)
- [The Log-Structured Merge-Tree \(LSM-Tree\)](#)
- [The Chubby lock service for loosely-coupled distributed systems](#)

Releases

No releases published

Packages

No packages published

Contributors 2



ashishps1 Ashish Pratap Singh



systemdesign42 Neo Kim

Languages

● Java 54.8% ● Python 45.2%