

---

# Table of Contents

Foreword.....	xiii
---------------	------

Preface.....	xv
--------------	----

---

## Part I. System Design Basics

<b>1. System Design Trade-offs and Guidelines.....</b>	<b>5</b>
System Design Concepts	6
Communication	6
Consistency	8
Availability	13
Reliability	19
Scalability	20
Maintainability	21
Fault Tolerance	22
Fallacies of Distributed Computing	24
System Design Trade-offs	26
Time Versus Space	27
Latency Versus Throughput	27
Performance Versus Scalability	28
Consistency Versus Availability	28
System Design Guidelines	31
Guideline of Isolation: Build It Modularly	31
Guideline of Simplicity: Keep It Simple, Silly	32
Guideline of Performance: Metrics Don't Lie	32
Guideline of Trade-offs: There Is No Such Thing as a Free Lunch	33

Guideline of Use Cases: It Always Depends	33
Conclusion	34
<b>2. Storage Types and Relational Stores.....</b>	<b>35</b>
Data Storage Format	36
File-Based Storage	36
Block-Based Storage	37
Object-Based Storage	38
Relational Databases	39
Relational Database Concepts	42
Relational Database Management System Architecture	47
Optimizing Relational Databases	50
Scaling Relational Databases	54
Open Source Relational Database Systems	63
Conclusion	65
<b>3. Nonrelational Stores.....</b>	<b>67</b>
Nonrelational Database Concepts	68
Schema Flexibility	68
Data Models	68
Scalability	69
High Availability and Fault Tolerance	69
BASE	69
Key-Value Databases	71
Data Model	71
Data Access and Retrieval Operations	73
Scaling Key-Value Stores	73
Availability in Key-Value Stores	76
Advantages, Trade-offs, and Considerations	77
Dynamo: Key-Value Database	78
Document Databases	79
Data Model	79
Availability in Document Stores	80
Advantages, Trade-offs, and Considerations	81
MongoDB: Open Source Document Database	82
Columnar Databases	83
Data Model	83
Consistency Levels	84
Columnar Store Architecture	86
Advantages, Trade-offs, and Considerations	88
Apache Cassandra: Open Source Columnar Database	88
Graph Databases	90

Data Model	90
Data Access and Retrieval	90
Advantages, Trade-offs, and Considerations	91
Neo4j: Open Source Graph Database	91
Conclusion	93
<b>4. Caching Policies and Strategies.....</b>	<b>95</b>
Benefits of Caching	96
Cache Eviction Policies	97
Belady's Algorithm	97
Queue-Based Policies	97
Recency-Based Policies	98
Frequency-Based Policies	98
Allowlist Policy	99
Cache Invalidation	100
Caching Strategies	102
Read-Intensive Strategies	102
Write-Intensive Strategies	104
Cache Deployment	105
In-Process Caching	105
Interprocess Caching	106
Remote Caching	106
Choosing a Cache Deployment Approach	106
Caching Mechanisms	107
Content Delivery Networks	108
Push CDNs	109
Pull CDNs	109
Open Source Caching Solutions	111
Memcached	111
Redis	113
Conclusion	118
<b>5. Load Balancing Approaches and Techniques.....</b>	<b>119</b>
Networking Components	120
Benefits of Load Balancing	122
LB Deployment and Placement Strategies	123
Global Server Load Balancing	123
Local Load Balancing	123
Load Balancing Algorithms	125
Static Load Balancing Algorithms	125
Dynamic Load Balancing Algorithms	126
Session Persistence in LBs	127

Stateful Load Balancers	127
Stateless Load Balancers	128
Types of Load Balancers	129
LB Types Based on Functionality	129
LB Types Based on Configuration	132
Nginx: Open Source Load Balancer	133
Conclusion	135
<b>6. Communication Networks and Protocols.....</b>	<b>137</b>
Communication Models and Protocols	137
OSI Model	138
TCP/IP Model	139
Communication Types	150
Pull Mechanism: HTTP Polling	151
Push Mechanism: WebSockets	151
Push Mechanism: Server-Sent Events	153
Common Communication Protocol Standards	154
Remote Procedure Call	154
REST	155
GraphQL	157
Web Real-Time Communication	159
Conclusion	160
<b>7. Containerization, Orchestration, and Deployments.....</b>	<b>163</b>
Evolution of Application Deployment	164
Containerization	166
Docker	166
Container Orchestration	174
Container Deployment Strategies	178
CI/CD Pipeline with Gitflow and Automated Deployment Strategies	180
Gitflow Workflow for Branch Management	180
Continuous Integration	181
Continuous Deployment	182
Monitoring and Incident Management	182
Conclusion	183
<b>8. Architectural Designs and Patterns.....</b>	<b>185</b>
Change Data Capture	186
Publisher-Subscriber Architecture	188
Message Brokers	188
Message Queues	189
Choreography and Orchestration	190

Choreography	190
Orchestration	191
Deciding Between Choreography and Orchestration	193
Big Data Architecture	193
Lambda Architecture	193
Kappa Architecture	194
Data Lake Architecture	195
Solution Architecture	196
Monoliths	196
N-tier Architectures	197
Microservices	198
Event-Driven Architecture	199
EDA Concepts and Implementations	199
Paradigms of Event-Driven Implementations	200
Common Cloud Architecture Patterns	202
Event-Based Patterns: CQRS and Saga	202
Failure-Tolerant Patterns: Circuit Breaker, Retry with Backoff, and Rate Limiter	203
Domain-Based Patterns: Domain-Driven Design and Decompose by Subdomains	203
API Routing Strategies and Patterns	204
Other Cloud Architecture Patterns	205
Open Source Distributed Systems Architecture	206
HDFS	206
Apache Kafka: Distributed Message Queue	211
Comparing HDFS and Kafka	215
Conclusion	215

---

## Part II. Diving Deep into AWS Services

<b>9. AWS Network Services</b>	<b>221</b>
Getting Started with AWS	222
AWS Regions	223
AWS Availability Zones	223
AWS Local Zones	224
AWS Edge Locations	224
Introduction to AWS Networking Services	225
Amazon VPC	225
Subnets	231
Internet Connectivity	233
Route Tables	233

Security Groups	234
Network Access Control Lists	236
Amazon VPC-to-Internet Connectivity	237
Connectivity Between Amazon VPCs	239
Hybrid Connectivity	243
Amazon Route 53	246
AWS Elastic Load Balancer	248
Amazon API Gateway	250
Amazon CloudFront	251
Conclusion	253
<b>10. AWS Storage Services.....</b>	<b>255</b>
Cloud Storage on AWS	256
Amazon Elastic Block Store	256
Amazon Elastic File System	257
Amazon Simple Storage Service	259
AWS Databases	262
Amazon RDS	263
Amazon DynamoDB	265
Amazon DocumentDB	268
Amazon Neptune	270
Amazon ElastiCache	272
Amazon OpenSearch	275
Amazon Timestream	276
Amazon Keyspaces	276
Conclusion	277
<b>11. AWS Compute Services.....</b>	<b>279</b>
Amazon Elastic Compute Cloud	279
Amazon Machine Image	281
Instance Type	282
Autoscaling	284
AWS Lambda	286
Containerization Services	289
Amazon Elastic Container Service	290
Amazon Elastic Kubernetes Service	293
Conclusion	295
<b>12. AWS Messaging, Orchestration, Monitoring, and Access Management Services....</b>	<b>297</b>
Amazon Managed Streaming for Apache Kafka	297
Amazon Kinesis	299
Amazon Kinesis Data Streams	299

Amazon Kinesis Data Analytics	302
Amazon Kinesis Data Firehose	303
Amazon Kinesis Video Streams	304
Amazon Simple Queue Service	305
Amazon Simple Notification Service	307
Workflow Orchestration	309
AWS Step Functions	310
Amazon Managed Workflow for Apache Airflow	313
Amazon CloudWatch	315
Application Logs	315
Metrics and Alarms	316
AWS Identity and Access Management	320
Amazon Cognito	323
AWS AppSync	326
Conclusion	328
<b>13. Big Data, Analytics, and Machine Learning Services. ....</b>	<b>329</b>
AWS Big Data and Analytics	329
Amazon Elastic MapReduce	330
AWS Glue	334
Amazon Athena	338
Amazon QuickSight	339
Amazon Redshift	340
Machine Learning on AWS	344
Amazon SageMaker	344
AWS ML Application Services	346
AWS ML Infrastructure	348
Conclusion	349

---

## Part III. System Design Use Cases

<b>14. Designing a URL Shortener Service. ....</b>	<b>355</b>
System Requirements	355
Functional and Nonfunctional Requirements	356
System Scale	357
Storage Space	358
Starting with the Design	360
URL Shortening Algorithm	360
System APIs	365
System Considerations	367
Database Selection	368

Custom Domain Support	369
Launching the System on AWS	371
Day Zero Architecture	371
Scaling to Millions and Beyond	376
Day N Architecture	379
Conclusion	383
<b>15. Designing a Web Crawler and Search Engine.....</b>	<b>387</b>
System Requirements	388
Functional and Nonfunctional Requirements	388
System Scale	389
Starting with the Design	391
Designing the Web Crawler	393
Designing the Search Engine	398
Launching the System on AWS	404
Day 0 Architecture	405
Scaling to Millions and Beyond	407
Day N Architecture	409
Conclusion	414
<b>16. Designing a Social Network and Newsfeed System.....</b>	<b>415</b>
System Requirements	415
Functional and Nonfunctional Requirements	416
System Scale	417
Starting with the Design	417
Handling New Posts	418
Managing User Connections	424
Search Service	425
Launching the System on AWS	427
Day 0 Architecture	427
Scaling to Millions and Beyond	428
Day N Architecture	436
Conclusion	440
<b>17. Designing an Online Game Leaderboard.....</b>	<b>441</b>
System Requirements	442
Functional and Nonfunctional Requirements	442
System Scale	444
Starting with the Design	446
Concepts and Principles	446
A Rough System Design	448
Launching the System on AWS	451



Day 0 Architecture	451
Scaling to Millions and Beyond	454
Day N Architecture	460
Conclusion	463
<b>18. Designing a Hotel Reservation System.....</b>	<b>465</b>
System Requirements	465
Functional and Nonfunctional Requirements	466
System Scale	467
Starting with the Design	469
Property Onboarding Architecture	469
Property Search Architecture	471
Property Booking Architecture	478
Property Reviews Architecture	488
Launching the System on AWS	490
Day 0 Architecture	490
Scaling to Millions and Beyond	493
Day N Architecture	497
Conclusion	498
<b>19. Designing a Chat Application.....</b>	<b>501</b>
System Requirements	501
Functional and Nonfunctional Requirements	502
System Scale	503
Starting with the Design	503
Messaging Architecture	504
WhatsApp Architecture with Erlang	511
Launching the System on AWS	516
Day 0 Architecture	516
Scaling to Millions and Beyond	517
Day N Architecture	520
Conclusion	522
<b>20. Designing a Video-Processing Pipeline for a Streaming Service.....</b>	<b>525</b>
System Requirements	525
Functional and Nonfunctional Requirements	526
System Scale	527
Starting with the Design	528
Video Encoding	529
Video-Quality Validation	531
Content Indexing	531
Content Distribution	532

Launching the System on AWS	534
Day 0 Architecture	534
Scaling to Millions and Beyond	536
Day N Architecture	540
Conclusion	542
<b>21. Designing an Online Stock-Trading Platform.....</b>	<b>543</b>
System Requirements	544
Functional and Nonfunctional Requirements	544
System Scale	545
Starting with the Design	545
Designing a Stock Tick System	546
Designing the Order Management System	550
Designing Ultra-Low-Latency Systems	553
Building the P&L Dashboard	555
Launching the System on AWS	557
Day 0 Architecture	557
Scaling to Millions and Beyond	559
Day N Architecture	562
Conclusion	566
<b>Index.....</b>	<b>569</b>