

TABLE OF CONTENTS

PART I: CONCEPTS & TECHNOLOGIES	21
1 Introduction to Cloud Computing	23
1.1 Introduction	24
1.1.1 Definition of Cloud Computing	24
1.2 Characteristics of Cloud Computing	25
1.3 Cloud Models	27
1.3.1 Service Models	27
1.3.2 Deployment Models	27
1.4 Cloud Computing Concepts & Technologies	31
1.4.1 Virtualization	31
1.4.2 Containerization	33
1.4.3 Load Balancing	34
1.4.4 Scalability & Elasticity	37
1.4.5 Deployment	39
1.4.6 Replication	40
1.4.7 Monitoring	42
1.4.8 Software Defined Networking	43
1.4.9 Network Function Virtualization	46
1.4.10 Identity and Access Management	48
1.4.11 Service Level Agreements	50
1.4.12 Billing	50
1.5 Cloud Computing Reference Model	51
1.6 Open Source Private Cloud Software	53
1.6.1 CloudStack	53
1.6.2 Eucalyptus	53
1.6.3 OpenStack	54
1.7 Design Considerations for Cloud Applications	55
1.7.1 Scalability	55
1.7.2 Reliability & Availability	56

1.7.3	Security	57
1.7.4	Maintenance & Upgradation	57
1.7.5	Performance	57
1.8	Reference Architectures for Cloud Applications	58
2	Fundamentals of Microservices	63
2.1	Definition of a Microservice	64
2.2	Characteristics of a Microservice	64
2.3	Microservices Architecture	67
2.3.1	Components of Microservices Architectures	69
2.4	Monolithic versus Microservice Architecture	71
2.4.1	Monolithic Application Architecture	71
2.4.2	Drawbacks of Monolithic Applications	72
2.4.3	Benefits of Microservices	73
2.4.4	Challenges of Microservices	76
2.5	Decomposing a Monolith into Microservices	79
2.5.1	Decomposition Strategies	79
2.5.2	Steps for Decomposing a Monolithic Application into Microservices	82
2.6	Case Study: Transitioning a Monolithic App to Microservices	85
3	Microservice Architecture Patterns	89
3.1	Introduction	90
3.2	Decomposition Patterns	90
3.2.1	Decompose by Business Capability	90
3.2.2	Decompose by Subdomain	93
3.2.3	Decompose by Transaction	95
3.2.4	Strangler Pattern	97
3.3	Composition & Integration Patterns	99
3.3.1	Aggregator Pattern	100
3.3.2	API Gateway Pattern	101
3.3.3	Proxy Pattern	102
3.3.4	Client-Side UI Composition	103
3.3.5	Chained Microservices	104
3.3.6	Branched Microservices	105
3.4	Database Patterns	106
3.4.1	Database per Service	106
3.4.2	Shared Database per Service	107
3.4.3	Command Query Responsibility Segregation (CQRS)	109
3.4.4	Event Sourcing Pattern	110
3.4.5	Saga Pattern	112

3.5	Observability Patterns	113
3.5.1	Log Aggregation Pattern	114
3.5.2	Distributed Tracing Pattern	115
3.5.3	Performance Metrics Pattern	117
3.5.4	Health Check Pattern	119
3.6	Deployment & Operational Patterns	120
3.6.1	Blue-Green Deployment Pattern	120
3.6.2	Service Registry Pattern	122
3.6.3	External Configuration Store Pattern	124
3.6.4	Circuit Breaker Pattern	126
3.7	Communication Patterns	127
3.7.1	Request-Response	128
3.7.2	Publish-Subscribe	128
3.7.3	Exclusive Pair	130
3.7.4	Remote Procedure Call	131
3.7.5	Push-Pull	132
4	Design Methodologies for Microservices Applications	135
4.1	Introduction	136
4.2	REST Web Services	137
4.2.1	Using REST Principles for Microservices	138
4.2.2	Building RESTful Web Services	139
4.2.3	Benefits of RESTful Web Services for Microservices	140
4.3	Cloud Component Model (CCM)	140
4.3.1	Using CCM Principles for Microservices	142
4.3.2	Building Microservices with CCM	144
4.3.3	Benefits of using CCM for Microservices	145
4.4	Serverless Architecture	146
4.4.1	Using Serverless Principles for Microservices	147
4.4.2	Building Serverless Microservices Applications	149
4.4.3	Benefits of using Serverless Architecture for Microservices	149
4.5	Event-Driven Architecture	150
4.5.1	Using Event-Driven Architecture Principles for Microservices	150
4.5.2	Building Microservices with Event-Driven Architecture	151
4.5.3	Benefits of using Event-Driven Architecture for Microservices	152
4.6	Model View Controller (MVC)	153
4.6.1	Using MVC Principles for Microservices	153
4.6.2	Building Microservices with MVC	154
4.6.3	Benefits of using MVC for Microservices	154

PART II: IMPLEMENTATION ON AWS 157

5	Overview of Amazon Web Services	159
5.1	Introduction	160
5.2	Compute Services	161
5.3	Amazon EC2	161
5.3.1	EC2 Instance Types	162
5.3.2	Amazon Machine Image	162
5.3.3	Security Groups	162
5.3.4	Tenancy Options	163
5.3.5	Pricing Options	163
5.3.6	Placement Groups	163
5.4	AWS EC2 Auto Scaling	169
5.4.1	Launch Configuration	169
5.4.2	Auto Scaling Group	169
5.4.3	Scaling Policy	169
5.5	AWS Elastic Load Balancing	173
5.5.1	Internet-Facing or Internal Load Balancers	173
5.5.2	Listeners	173
5.5.3	Health Checks	173
5.5.4	Sticky Sessions	173
5.5.5	Connection Draining	174
5.5.6	Using an Application Load Balancer	174
5.6	Storage Services	178
5.7	Amazon Simple Storage Service (S3)	178
5.7.1	Buckets	179
5.7.2	Objects	182
5.7.3	Managing Access with Bucket Policies and ACLs	184
5.7.4	Cross-Origin Resource Sharing (CORS)	186
5.7.5	Versioning	187
5.7.6	Encryption	187
5.7.7	Static Website Hosting	187
5.7.8	Transfer Acceleration	188
5.7.9	Durability and Availability	188
5.7.10	Consistency	188
5.7.11	Storage Classes	188
5.7.12	Object Lifecycle Management	189
5.7.13	Cross-Region Replication	189
5.8	Amazon Elastic File System (EFS)	190
5.8.1	EFS Performance Modes	193
5.8.2	EFS Throughput Modes	193
5.8.3	Encryption	193

5.9	Amazon Elastic Block Store (EBS)	194
5.9.1	Volumes	194
5.9.2	Snapshots	196
5.9.3	Encryption	196
5.10	Relational Databases	197
5.10.1	OLTP vs OLAP	198
5.11	MySQL	199
5.12	Amazon Relational Data Store	201
5.12.1	Setting up an RDS Database Instance	203
5.13	Amazon ElastiCache	210
5.14	Non-Relational Databases	212
5.15	Consistency, Availability & Partition Tolerance (CAP)	214
5.16	Key-Value Databases	215
5.16.1	Amazon DynamoDB	216
5.17	Document Databases	220
5.17.1	MongoDB	221
5.17.2	Amazon DocumentDB	224
5.18	Graph Databases	225
5.18.1	Amazon Neptune	225
5.19	Serverless Computing	230
5.20	AWS Lambda	230
5.20.1	Triggers	231
5.20.2	Lambda Function Example	233
5.21	Identity & Access Management	239
5.22	AWS Identity and Access Management (IAM)	240
5.22.1	Users	241
5.22.2	Groups	244
5.22.3	Policies	245
5.22.4	Roles	245
5.23	Key Management	247
5.24	AWS Key Management Service (KMS)	248
5.25	Amazon Route 53	249
5.26	Amazon CloudFront	251
5.27	Amazon SNS	252
6	Developing Microservices with Flask on EC2	255
6.1	Introduction	256

6.2	Key Concepts	256
6.2.1	Flask	256
6.2.2	CRUD functionality	257
6.2.3	SQLAlchemy and Object-Relational Mapping (ORM)	257
6.2.4	Database Migrations	258
6.2.5	Authentication and Authorization	258
6.2.6	Routing	258
6.2.7	Sessions Management	259
6.2.8	HTML Templating with Jinja	259
6.2.9	Forms and Validation	259
6.2.10	Responsive Design	259
6.2.11	File Uploads	260
6.3	Monolithic Implementation of E-Commerce application	260
6.3.1	Application Architecture	260
6.3.2	Data Modeling	261
6.3.3	Application Flow and UI	262
6.3.4	Flask Application Implementation	266
6.3.5	Flask Templates	278
6.4	Microservices Implementation of E-Commerce application	284
6.4.1	Microservices Implementation	285
6.5	Deploying Microservices Application on Amazon EC2	307
6.5.1	Nginx Reverse Proxy	308
6.5.2	Web Server Gateway Interface (WSGI)	309
6.5.3	Gunicorn	309
6.5.4	Supervisor	309
6.5.5	SSL Certificate	310
6.5.6	EC2 Security Groups	310
6.5.7	Setting Up and Configuring EC2 Instances	311
6.6	Configuring a Domain with Route 53	315
7	Persisting Microservices Data with S3, RDS and DynamoDB	317
7.1	Introduction	318
7.2	Storing Files on Amazon S3	320
7.3	Serving Files with CloudFront CDN	325
7.4	Persisting Data with RDS	329
7.5	Microservices App Using EC2, S3, CloudFront & RDS	334
7.6	Persisting Data with DynamoDB	340
7.7	Microservices App Using EC2, S3, CloudFront & DynamoDB	343

8	Containerizing Microservices with Docker, ECR and ECS . .	365
8.1	Introduction	366
8.2	Virtualization vs Containerization	367
8.3	Docker Overview	369
8.3.1	Docker Architecture	369
8.3.2	Docker Storage and Volumes	376
8.3.3	Dockerfile	376
8.3.4	Docker Compose	378
8.4	Containerizing Microservices Application with Docker	380
8.5	Deploying Containerized Microservices Application on AWS	394
8.5.1	Amazon Elastic Container Registry (ECR)	395
8.5.2	Amazon Elastic Container Service (ECS)	398
8.5.3	Amazon Application Load Balancer (ALB)	404
8.5.4	Creating ECS Services	410
9	Implementing High-Performance Microservices with gRPC	415
9.1	Introduction	416
9.2	gRPC Overview	417
9.3	Microservices Application using gRPC	418
9.3.1	Define Proto Files	419
9.3.2	Generate stubs	424
9.3.3	Update frontend microservice	425
9.3.4	Modify backend microservices	437
10	Serverless Microservices with Lambda and API Gateway . .	451
10.1	Introduction	452
10.2	Serverless Microservices Application	455
10.2.1	Cognito User Pool	455
10.2.2	DynamoDB Tables	459
10.2.3	IAM Role	462
10.2.4	Lambda Functions	462
10.2.5	API Gateway	483
10.2.6	Static Website Frontend	486
11	Microservices Observability and Monitoring	505
11.1	Introduction	506
11.2	Logging	507
11.3	Metrics	509
11.4	Distributed Tracing with OpenTelemetry	510

11.5	OpenTelemetry and Jaeger	512
11.6	Prometheus and Grafana	521
11.7	AWS X-Ray and CloudWatch	531
11.8	Best Practices for Observability and Monitoring	539
12	Microservices Security	543
12.1	Introduction	544
12.2	Authentication and Authorization	545
12.2.1	Centralized Identity and Access Management (IAM)	545
12.2.2	Token-based Authentication	545
12.2.3	Role-based Access Control (RBAC)	547
12.2.4	Service-to-Service Authentication	548
12.3	API Gateway and Perimeter Security	550
12.4	Secure Communication	552
12.4.1	Encryption in Transit	552
12.4.2	Secure Communication Channels	552
12.4.3	Auditing and Monitoring	553
12.5	Data Security	553
12.5.1	Data Classification and Sensitivity	553
12.5.2	Encryption at Rest and in Transit	554
12.5.3	Secure Storage Solutions	554
12.5.4	Data Access Controls and Restricted Access	555
12.5.5	Data Lifecycle Management and Disposal	555
12.6	Monitoring and Logging	556
12.6.1	Centralized Logging and Monitoring	556
12.6.2	Collecting Logs, Metrics, and Traces	556
12.6.3	Log Analysis and Security Event Detection	556
12.6.4	Alerting and Notifications	557
12.7	Secure Development Practices	557
12.7.1	Secure Coding Practices and Code Reviews	557
12.7.2	Security Testing	558
12.7.3	Secure Libraries, Frameworks, and Tools	558
12.7.4	Dependency Management and Patching	558
12.7.5	Secure Software Development Lifecycle (SSDLC)	559
12.7.6	Security Champions and Training	559
12.8	Secure Deployment and Automation	559
12.8.1	Secure Deployment Pipelines	560
12.8.2	Infrastructure as Code (IaC)	560
12.8.3	Secure Automation and Orchestration	561

12.9	Incident Response and Disaster Recovery	562
12.10	Containerization and Isolation	563
12.10.1	Consistent Runtime Environments	563
12.10.2	Container Isolation and Security	564
12.10.3	Patching and Updating Container Images	564
12.11	Least Privilege and Separation of Concerns	565
12.11.1	Identity and Access Management	565
12.11.2	OS-Level Security and Containerization	566
12.11.3	Secure Service-to-Service Communication	566
12.11.4	Access Controls on Shared Resources	566
12.11.5	Separation of Concerns and Domain Decomposition	566
12.11.6	Microservice Domains as Trust Boundaries	566
12.12	Network Security	567
12.12.1	Network Segmentation and Isolation	567
12.12.2	Secure Communication Protocols	567
12.12.3	Firewalls and Network Access Controls	567
12.12.4	Network Monitoring and Logging	568
12.12.5	Network Access Controls for External Connections	568
12.12.6	Network Security Automation	569
12.13	Vulnerability Management	569
12.13.1	Understanding Vulnerabilities	569
12.13.2	Vulnerability Management Process	570
12.13.3	Automation and Integration	571
12.14	Chaos Engineering for Microservices Security	571
12.14.1	Implementing Chaos Engineering in Microservices	572
12.14.2	Tools and Frameworks for Chaos Engineering	573
12.15	Security Automation	573
12.15.1	Automating Security Checks and Scans	574
12.15.2	Automated Configuration Management	574
12.15.3	Compliance and Governance Automation	575
12.15.4	Incident Response Automation	575
12.16	DevSecOps for Microservices Security	576
	Bibliography	584
	Index	589