

AWS Services Mind Map – Detailed Guide

Compute

• EC2 (Elastic Compute Cloud)

What: Virtual servers in the cloud.

Why: Run applications, test environments.

When to Use: When you need scalable compute power.

Features: Scalable, Pay-as-you-go, Customizable

Pros: Flexible, Global reach

Cons: Management overhead, Cost if idle

• Lambda

What: Serverless compute service.

Why: Run code without managing servers.

When to Use: When you need event-driven apps.

Features: No server mgmt, Auto scaling

Pros: Low cost for sporadic tasks

Cons: Not for heavy compute, Cold starts

• Elastic Beanstalk

What: PaaS for app deployment.

Why: Quickly deploy/manage apps.

When to Use: When you want managed infra for apps.

Features: Auto scaling, Easy deploy

Pros: Fast setup

Cons: Less control, Limited customizations

• ECS/EKS

What: Containers on AWS.

Why: Run Docker/K8s workloads.

When to Use: When using microservices.

Features: Managed orchestration

Pros: Scalable

Cons: Learning curve

Storage

• S3 (Simple Storage Service)

What: Object storage in the cloud.

Why: Store/retrieve any amount of data.

When to Use: When storing backups, media, logs.

Features: Durable, Scalable, Secure

Pros: Cheap for storage

Cons: Latency vs block storage

• EBS (Elastic Block Store)

What: Block storage for EC2.

Why: Attach persistent disks to EC2.

When to Use: When running DBs or OS storage.

Features: Persistent, Fast

Pros: High performance

Cons: Tied to AZ, Costly at scale

• Glacier

What: Archival storage service.

Why: Low-cost long-term storage.
When to Use: When storing old, infrequently used data.
Features: Very cheap, Durable
Pros: Low cost
Cons: Slow retrieval

• EFS (Elastic File System)

What: Managed NFS for AWS.
Why: Shared file storage for EC2.
When to Use: When multiple instances need same files.
Features: Scalable, Managed
Pros: Easy to use
Cons: Costly for large data

Database

• RDS (Relational Database Service)

What: Managed SQL DB.
Why: Run MySQL, PostgreSQL, etc.
When to Use: When you need relational DB.
Features: Managed backups, Scaling
Pros: Less admin work
Cons: Less control, Cost

• DynamoDB

What: Managed NoSQL DB.
Why: Key-value and doc storage.
When to Use: When you need high-speed NoSQL.
Features: Low latency, Scalable
Pros: Fast
Cons: Expensive for high throughput

• Redshift

What: Data warehouse.
Why: Analytics on large data.
When to Use: When running BI reports.
Features: Columnar storage, Fast queries
Pros: Good for analytics
Cons: Costly for small workloads

• ElastiCache

What: Managed cache service.
Why: Redis/Memcached.
When to Use: When improving app speed.
Features: Low latency, Managed
Pros: Boosts performance
Cons: Cost if idle

Networking/CDN

• VPC (Virtual Private Cloud)

What: Isolated AWS network.
Why: Control networking setup.
When to Use: When you need private cloud infra.
Features: Custom IP ranges, Security
Pros: Control
Cons: Complex setup

• CloudFront

What: Content Delivery Network.

Why: Distribute content globally.

When to Use: When you want low-latency delivery.

Features: Global edge locs, Secure

Pros: Fast

Cons: Cost with heavy use

• Route 53

What: DNS & domain mgmt.

Why: Route traffic globally.

When to Use: When managing domains.

Features: Scalable, Health checks

Pros: Reliable

Cons: Cost for many queries

Security

• IAM (Identity & Access Management)

What: Manage AWS users/roles.

Why: Control who can do what.

When to Use: Always for security control.

Features: Granular perms, Secure

Pros: Essential

Cons: Complex policies

• KMS (Key Management Service)

What: Manage encryption keys.

Why: Encrypt data easily.

When to Use: When storing sensitive data.

Features: Secure, Integrated

Pros: Strong security

Cons: Cost per request

• Shield

What: DDoS protection.

Why: Protect apps from attacks.

When to Use: When hosting public apps.

Features: Always on, Integrated

Pros: Automatic

Cons: Only for network attacks

Monitoring

• CloudWatch

What: Monitor AWS resources.

Why: Track metrics/logs.

When to Use: When you need visibility.

Features: Metrics, Alerts

Pros: Detailed

Cons: Cost for high logs

• CloudTrail

What: Track API calls.

Why: Audit AWS actions.

When to Use: When auditing changes.

Features: Logging, Security

Pros: Accountability

Cons: Log storage cost