AWS System Design Interview Cheat Sheet

Compute

EC2: Virtual machines in the cloud. Use when you need fine-grained control over compute resources. Tradeoff: More management overhead vs serverless.

ECS: Managed container orchestration (Docker). Good for microservices. Tradeoff: less flexibility vs EKS, but easier to manage.

EKS: Managed Kubernetes service. Use when you need full Kubernetes ecosystem. Tradeoff: more complex setup/management.

Lambda: Serverless compute for event-driven workloads. Great for lightweight functions. Tradeoff: cold starts, limited runtime duration, not for heavy workloads.

Auto Scaling Groups: Automatically scale EC2 instances. Good for handling variable workloads. Tradeoff: Only works with EC2, not serverless.

Storage

S3: Object storage for images, videos, backups. Virtually unlimited. Tradeoff: not low latency, not block storage.

EBS: Block storage for EC2. Low latency, like a disk. Tradeoff: tied to an EC2 instance/AZ.

EFS: Shared file system (NFS-like). Good for multiple instances. Tradeoff: more expensive, slower than EBS for single instance workloads.

Glacier: Archive storage for rarely accessed data. Cheap, but retrieval is slow (minutes to hours).

Databases

RDS: Managed relational databases (MySQL, Postgres, etc.). Good for structured data. Tradeoff: vertical scaling limits, can be costly.

Aurora: High-performance relational DB (MySQL/Postgres compatible). Good for high throughput apps. Tradeoff: higher cost, AWS-specific.

DynamoDB: Managed NoSQL key-value/document store. Great for serverless and scale. Tradeoff: query flexibility is limited compared to SQL.

ElastiCache: In-memory cache (Redis/Memcached). Use for low-latency caching. Tradeoff: volatile, not persistent storage.

Neptune: Graph database. Use for relationship-heavy data. Tradeoff: niche use case, higher complexity.

Networking & Content Delivery

VPC: Virtual Private Cloud for isolation and networking. Foundation for AWS apps. Tradeoff: requires setup and networking knowledge.

Route 53: DNS and traffic routing. Can be used for global failover. Tradeoff: DNS caching delays can affect failover speed.

CloudFront: CDN for caching and global delivery. Good for static assets and videos. Tradeoff: adds caching layer complexity.

ALB/NLB: Load balancers for HTTP (ALB) or TCP (NLB). ALB is good for web apps, NLB for high-performance networking.

API Gateway: Managed API service. Integrates well with Lambda. Tradeoff: can be costly at scale, request limits apply.

Messaging & Streaming

SQS: Simple Queue Service for async processing. Good for decoupling. Tradeoff: polling model, no push delivery.

SNS: Pub/Sub notifications. Push-based. Good for fan-out messaging. Tradeoff: simple filtering only.

Kinesis: Real-time streaming data ingestion. Tradeoff: shard management required, cost at scale.

MSK: Managed Kafka. Full Kafka features. Tradeoff: more ops overhead vs Kinesis.

Security & Identity

IAM: Identity and access management. Core AWS security. Tradeoff: complex policies can be hard to debug.

KMS: Key Management Service for encryption. Tradeoff: costs for many keys/operations.

Cognito: Authentication (user pools, federated identity). Good for apps. Tradeoff: limited customization vs custom auth.

Secrets Manager: Store and rotate secrets. Tradeoff: costlier than Parameter Store.

Monitoring & Observability

CloudWatch: Metrics, logs, alarms. Tradeoff: costs grow with log storage.

X-Ray: Distributed tracing. Good for microservices. Tradeoff: requires instrumentation.

CloudTrail: Audit of all API calls. Good for compliance. Tradeoff: generates a lot of logs.

Deployment & DevOps

CloudFormation: Infrastructure as code. Declarative. Tradeoff: verbose, slower deployments.

CDK: Code-first IaC. More developer friendly. Tradeoff: tied to supported languages.

CodePipeline/CodeBuild/CodeDeploy: CI/CD pipeline tools. Tradeoff: simpler than Jenkins but less flexible.

Analytics & Big Data

Athena: Query S3 with SQL. Serverless. Tradeoff: slower than RDS for frequent queries.

EMR: Managed Hadoop/Spark. Tradeoff: cluster management overhead.

Redshift: Data warehouse. Optimized for OLAP. Tradeoff: not for OLTP.

Glue: ETL service. Tradeoff: slower startup, less flexible than custom Spark jobs.

Other Useful Services

Step Functions: Orchestrate workflows. Good for microservice coordination. Tradeoff: step limits, cost at scale.

MediaConvert: Video transcoding. Good for media workflows. Tradeoff: costs per minute processed.

OpenSearch Service: Search & analytics (Elasticsearch). Tradeoff: can be expensive, ops overhead.