**AWS Summary**

|  |  |
| --- | --- |
| **Name** | **Description** |
| 1. **Compute** | |
| **EC2** | Elastic compute - virtual machines |
| **Lambda** | Serverless compute service - runs your code on AWS |
| **ECR** | Elastic container registry to manage container images |
| **ECS** | EC2 container [orchestration] service - run, scale Docker apps on EC2 and Fargate |
| **Fargate** | Serverless compute engine for containers, works w/ECS and EKS (A. Elastic Kubernetes Service) – auto-allocates compute resources |
| **Elastic Beanstack** | Auto-deploys web apps |
| 1. **DB** | |
| **DynamoDB** | NoSQL key-value and doc store, scalable, low-latency |
| **ElastiCache** | In-memory data store (Redis and Memcached) |
| **Aurora** | Commercial RDB (transactional, for backend) |
| **RDS (Relational Database Service)** | Manages RDBs in cloud: Aurora, PostgreSQL, MySQL, Microsoft SQL, MariaDB |
| 1. **Analytics** | |
| **Redshift** | Analytical RDB, standard SQL queries for viz & BI |
| **Athena** | Serverless, analyzes large dsets in S3 using standard SQL |
| **AWS Glue** | Serverless data integration from multiple sources |
| **Kinesis** | Analyze real-time data streams instantly |
| 1. **Storage** | |
| **S3** | Simple storage service (generic) |
| **Glacier** | Data archiving + long-term backups (retrieval 5min. – 10hr) |
| **EBS** | Elastic block storage for EC2 instances |
| **EFS** | Elastic file system - scalable distributed network file system |
| 1. **Networking** | |
| **Elastic Load Balancing** | Distributes app traffic 2 multiple servers, containers, Lambda f(x) |
| **Route 53** | Scalable, low-latency **DNS Service** (IP lookup, GeoDNS, etc.) |
| **CloudFront** | Content delivery network (**CDN**) |
| **API Gateway** | Manage RESTful and WebSocket APIs |
| **VPC** | Virtual private cloud - logically isolates **virtual networks** |
| **Other** |  |
| **Integration Services** | *AmazonMQ* (broker for ActiveMQ and RabbitMQ), *Simple Queue Service, Simple Email Service, AWS Secrets Manager* |
| **Management & Governance** | *CloudFormation* (describe/manage resources with code), *CloudWatch* (monitoring) |

**Most Used AWS Services (Full)**

Here we describe only a few out of ~200 AWS services which are scalable, reliable, easy-to-use, and secure platform as a service (PaaS) and infrastructure as a service (IaaS) solutions:

1. **Computation Services**

**Amazon EC2 (Elastic Compute Cloud)**

Virtual servers / machines, varying CPU, RAM, etc.

**Amazon ECR (Elastic Container Registry)**

Storage, management, and deployment of container images.

**Amazon ECS (EC2 Container Service)**

Container orchestration service - run, scale, secure Docker apps on Amazon EC2 and Fargate.

**AWS Fargate**

Serverless compute engine for containers, works with ECS and EKS (Amazon Elastic Kubernetes Service) - automatically allocates the right amount of isolated compute resources for each container; isolation also improves its security.

**AWS Lambda**

Serverless compute service - runs your code on AWS.

**AWS Elastic Beanstack**

Auto-deployment of web apps (provisioning, load balancing, auto-scaling, app health monitoring).

1. **Databases**

**Amazon DynamoDB**

NoSQL key-value and doc store, scalable, low-latency.

**Amazon ElastiCache**

Redis and Memcached high-throughput low-latency in-memory data stores.

**Amazon Aurora**

Commercial RDB (transactional, for backend), compatible, but much faster than MySQL and PostgreSQL.

**Amazon RDS (Relational Database Service)**

Manages relational DBs in the cloud (hardware, setup, patching, and backups); supports various DB engines: Aurora, PostgreSQL, MySQL, Microsoft SQL, MariaDB.

1. **Analytics**

**Amazon Kinesis**

Analyze real-time data streams w/low-latency at any scale - collect, buffer, process streaming data instantly (no waiting for hours).

**Amazon Redshift**

Cost-effective data warehouse for standard SQL queries for viz & BI.

**Amazon Athena**

Serverless, analyzes large dsets in S3 using standard SQL, fast, easy, doesn’t require complex ETL to prepare data, pay per query.

**AWS Glue**

Serverless data integration from multiple sources, data preparation and placement in DB, warehouses, lake for further analysis.

1. **Data Storage**

**Amazon S3 (Simple Storage Service)**

Generic object storage - durability, high scalability, availability, security, and performance. Amazon Athena - analyze data in S3 with SQL queries.

**Amazon S3 Glacier**

Data archiving + long-term backups at extremely low-cost. Expedited retrievals – takes 1-5 minutes, standard - 3-5 hours, bulk - 5-12 hours.

**Amazon EBS (Elastic Block Storage)**

Block storage for EC2 instances (various throughput and latency suitable for your needs, scales to petabytes).

**Amazon EFS (Elastic File System)**

Fully managed scalable elastic NFS (distributed network file system) - grows and shrinks automatically, provides massively parallel shared access to thousands of EC2 instances w/high throughput and IOPS + consistent latency.

1. **Networking and Content Delivery Services**

**Amazon Route 53**

Highly available, scalable fault-tolerant, low-latency DNS Service (IP lookup, GeoDNS, Geoproximity, Latency Based Routing).

**Amazon CloudFront**

Fast, secure, programmable content delivery network (CDN).

**Amazon API Gateway**

Create, publish, monitor, and secure RESTful and WebSocket APIs.

**AWS Elastic Load Balancing**

Distributes incoming app traffic across multiple servers, containers, Lambda f(x) – makes apps highly available, reliable, fault-tolerant.

**Amazon VPC (Virtual Private Cloud)**

Logically isolates virtual networks inside AWS.

1. **Integration Services**

**Amazon MQ**

Broker service to run ActiveMQ and RabbitMQ on AWS infrastructure.

**Amazon SQS (Simple Queue Service)**

Send, store, and receive messages between microservices and serverless apps.

**Amazon SES (Simple Email Service)**

Email service that supports mass emails, marketing / transactional emails at scale.

**AWS Secrets Manager**

Protects secrets for various APIs and resources (to avoid hardcode them), controlled with fine-grained permissions.

1. **Management & Governance**

**AWS CloudFormation**

Describe and manage (add, update, etc.) desired resources and their dependencies with a code template (without managing them individually).

**AWS CloudWatch**

Monitor AWS resources and services of your apps for optimization, abnormal behavior detection, alarms, app health, and troubleshooting with logs and metrics.