# **Guía de Servicios Relevantes en AWS**

## **1. Computación**

**EC2 (Elastic Compute Cloud)**

* Servidores virtuales bajo demanda.
* Úsalo cuando necesitas control total de SO, red, almacenamiento.
* Ejemplo: montar un servidor de aplicaciones o un backend clásico.

**ECS (Elastic Container Service) / EKS (Elastic Kubernetes Service)**

* Orquestación de contenedores.
* ECS es el servicio nativo de AWS, más simple.
* EKS es Kubernetes gestionado, más flexible y estándar.
* Ejemplo: microservicios con Docker.

**Lambda**

* Funciones serverless que se ejecutan bajo demanda.
* Pago por uso (ejecuciones y tiempo).
* Ejemplo: procesar una imagen cuando se sube a S3.

**Comparación rápida**

* EC2: control absoluto, pero más gestión.
* ECS/EKS: para contenedores y microservicios.
* Lambda: para tareas event-driven, sin servidores.

## **2. Almacenamiento**

**S3 (Simple Storage Service)**

* Objetos: ficheros de cualquier tamaño
* Muy duradero, barato, escalable.
* Ejemplo: guardar audios, logs, datasets, backups.

**EFS (Elastic File System)**

* Sistema de archivos compartido, accesible desde varias instancias.
* Ejemplo: apps que requieren un filesystem común.

**Glacier**

* Archivos fríos y de bajo coste (backups largos).

**Comparación rápida**

* S3: datos inmutables y masivos.
* EBS: almacenamiento de bloques para servidores.
* EFS: compartido y accesible como NAS.
* Glacier: archivo barato y a largo plazo.

## **3. Bases de datos**

**Aurora (MySQL/Postgres compatible)**

* Base de datos relacional gestionada, muy rápida y escalable.
* Ejemplo: catálogo de productos, usuarios, pagos.

**RDS (Relational Database Service)**

* Varias engines (MySQL, Postgres, Oracle, SQL Server).
* Aurora es una versión más optimizada de RDS.

**DynamoDB**

* NoSQL clave-valor/documental, latencia milisegundos.
* Ejemplo: sesiones, carritos de compra, IoT.

**DocumentDB**

* Compatible con MongoDB, orientado a documentos JSON.
* Ejemplo: playlists, configuraciones flexibles.

**ElastiCache (Redis/Memcached)**

* Cache en memoria para lecturas rápidas.
* Ejemplo: top charts, sesiones activas.

**OpenSearch Service (antes Elasticsearch)**

* Motor de búsqueda y analítica de logs.
* Ejemplo: buscador de canciones, dashboards de logs.

**Comparación rápida**

* Aurora/RDS: relacional, integridad, joins.
* DynamoDB/DocumentDB: no relacional, flexible, escalable.
* ElastiCache: lecturas ultrarrápidas.
* OpenSearch: búsqueda de texto libre y log analytics.

## **4. Redes y entrega de contenido**

**VPC (Virtual Private Cloud)**

* Red privada para tus recursos AWS.
* Control de subnets, firewalls, seguridad.

**API Gateway**

* Puerta de entrada a APIs serverless o backend.
* Ejemplo: exponer microservicios a clientes móviles.

## **5. Analítica y Big Data**

**Kinesis (o MSK: Managed Streaming for Kafka)**

* Streaming en tiempo real.
* Ejemplo: ingesta de clics, logs, eventos de reproducción.

**Glue**

* ETL serverless para preparar datos.
* Ejemplo: transformar logs en datasets limpios en S3.

**Athena**

* SQL serverless sobre S3.
* Ejemplo: consultar logs o datasets sin cargar a base de datos.

**EMR (Elastic MapReduce)**

* Cluster gestionado de Hadoop/Spark.
* Ejemplo: procesamiento batch de grandes volúmenes.

**Redshift**

* Data Warehouse relacional optimizado para analítica.
* Ejemplo: informes financieros, reporting BI.

**QuickSight**

* BI para dashboards interactivos.
* Ejemplo: visualización de KPIs de negocio.

**Comparación rápida**

* Kinesis/MSK: tiempo real.
* Glue/EMR: batch y ETL pesados.
* Athena: SQL sobre data lake.
* Redshift: Data Warehouse clásico.
* QuickSight: visualización BI.

## **6. Machine Learning e IA**

**SageMaker**

* Entrenamiento y despliegue de modelos de ML.
* Ejemplo: recomendaciones personalizadas, clasificación de imágenes.

**Comprehend:** NLP gestionado (análisis de texto, sentimientos).

**Rekognition:** Análisis de imágenes y vídeo (detección de rostros, objetos).

**Lex:** Chatbots conversacionales.

**7. Monitoreo, seguridad y gestión**

**CloudWatch**

* Métricas, logs, alarmas.
* Ejemplo: monitorizar uso de CPU o errores de Lambda.

**Secrets Manager**

* Gestión de permisos y roles.

**Organizations**

* Multi-cuenta y facturación consolidada.

## **8. Orquestación y automatización**

**Step Functions**

* Orquesta servicios de AWS con lógica de estados.
* Ejemplo: pipeline de aprobación de documentos.

**Airflow (Managed Workflows for Apache Airflow - MWAA)**

* Orquestación de flujos complejos, ETLs multi-servicio.

**CloudFormation / CDK / Terraform**

* Infraestructura como código.
* Ejemplo: desplegar toda tu arquitectura en YAML/TypeScript.

## **9. Resumen mental**

* **Aurora/RDS** → relacional, integridad.
* **DynamoDB/DocumentDB** → NoSQL, flexibilidad.
* **S3** → objetos, barato y escalable.
* **Glue/EMR** → batch pesado.
* **Kinesis/MSK** → streaming real-time.
* **SageMaker** → ML end-to-end.
* **QuickSight** → dashboards de negocio.
* **Airflow/Step Functions** → orquestación.