1st October

01 (services that we will learn)

Analytics.

Amazon EMR,
AWS Lake Formation,
Amazon Redshift,
Kinesis,
AWS GLUE,
Athena,
Managed Kafka

API Integration.

Amazon EventBridge,
AWS Step Functions,
SNS, SQS
Apache Airflow

Compute → AWS Batch,
Amazon Elastic Cloud (EC2) Compute,
AWS Lambda
AWS Serverless Repository

Containers → ECR → Amazon Elastic Container Registry
ECS → Amazon Elastic Container Services
EKS → Amazon Elastic Kubernetes Services.

Database.

Amazon Document DB (Mongo DB).
Amazon Dynamo DB
Amazon Keyspaces (for Cassandra).
Amazon Memory DB (Redis).
Amazon RDS (Relational Database)

(Developer tools)

AWS CLI (Command Line Interface).

(Management And Governance)

CloudFormation,
CloudWatch,
Managed Grafana,

(Networking And Content Delivery)

CloudFront, PrivateLink, Route 53, VPC (Virtual Private Network).

(Security, Identity & Compliance)

IAM,
AWS KMS → Key Management System,
AWS Secrets Manager

Storage

Amazon S3,
Amazon EFS → Elastic File System.

(02 Data Engineering Fundamentals)

(Semi Structured Data)

(JSON, XML) → Some part is structured, some part might be not.

[Schema might not be consistent, but it exists]

Data Warehouse → Storing data in structured format.
For Read Heavy Operations.

Data Lake → Raw Storage, (Structured, Semi, Unstructured)

Data Warehouse → (Schema on write).
(ETL) → (first transform then Load.) [Defined during writing].

Data Lake → Schema on Read.
(ELT) (Defined during Read data)
(first Load into system then Transform)

Data Warehouse → More expensive because of optimisations for complex queries during read.

Data Lakes → Cost effective. But cost rises when processing Large amount of data.

Data Lakehouse → (Data Lake + Data Warehouse)

Supports both
(→ Schema on write
+
→ Schema on read)

Ex [AWS Lake Formation]. (S3 + Red Shift)

02 Oct

(02 Data Engineering Fundamentals)

ETL → (Extract, Transform, Load)

Extract → Fetching data from raw sources, extracting data from APIs.

Transform ⇒ Data cleaning, enrichment, Aggregations, Formatting to have it in structured format to Load it into Data Warehouse.

(Data Formats)

CSV → (Structured) Comma Separated Values. A line represents a row and elements in the row are separated by ','.

TSV → (Structured) Same as CSV, just the separator used is a 'Tab' instead of a ','.

JSON → (semi-structured) JavaScript Object Notation.

AVRO → Binary format. (Both data & Schema).

Parquet → Columnar Storage
[Stores data in colums, useful for analytics].
Allows for efficient compression & storage.

When to use: → Use cases where reading specific columns instead of entire records is benificial.
↳ Storing data on distributed systems where I/O operations and storage need optimisation.