Dharma Devi-April 21st HW

1.What is ETL in data engineering, and why is it important?

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Description** | **Examples / Tasks** | **Why It's Important** |
| **Extract** | Pull data from various source systems. | - Read from databases - Call APIs - Load files (CSV, JSON, etc.) | Brings together scattered data from multiple sources. |
| **Transform** | Clean, standardize, and reshape the extracted data. | - Remove duplicates - Convert formats - Join data - Apply business logic | Ensures data is accurate, consistent, and useful. |
| **Load** | Store the transformed data into a target system for use. | - Write to a data warehouse (e.g., Snowflake) - Load into databases | Makes data available for analytics, reporting, and decision-making. |

2. What are the key differences between cloud storage and traditional on-premise storage?

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Cloud Storage** | **On-Premise Storage** |
| **Location** | Data is stored on remote servers managed by a cloud provider. | Data is stored locally on physical servers at your own facility. |
| **Scalability** | Highly scalable—add storage on demand with little effort. | Limited by physical hardware; scaling requires buying equipment. |
| **Cost Structure** | Pay-as-you-go (operational expense). | Upfront capital expense (hardware, maintenance, etc.). |
| **Maintenance** | Managed by the provider (automatic updates, backups, etc.). | Managed in-house—requires IT staff for updates and support. |
| **Accessibility** | Accessible from anywhere via the internet. | Limited to local network unless additional access is configured. |
| **Security** | Shared responsibility—cloud providers offer strong security tools. | Full responsibility for physical and network security. |
| **Disaster Recovery** | Built-in redundancy and geo-replication options. | Requires custom setup; backup systems must be managed internally. |
| **Deployment Speed** | Fast—can provision storage in minutes. | Slower—requires hardware setup, configuration, etc. |
| **Compliance Control** | May have limitations depending on provider’s certifications. | Full control—can meet niche or local compliance needs more easily. |
| **Use Case** | **Cloud Storage** | **On-Premise Storage** |
| **Startup or Small Business** | ✅ Cost-effective with no large upfront investment. | ❌ High setup cost, not ideal for limited budgets. |
| **Rapidly Growing Company** | ✅ Scales easily with demand—just increase cloud resources. | ❌ Scaling requires new hardware purchases and setup time. |
| **Remote or Global Workforce** | ✅ Accessible from anywhere with internet access. | ❌ May need complex VPNs or remote access solutions. |
| **Data Backup & Disaster Recovery** | ✅ Built-in backup, geo-redundancy, and quick recovery. | ❌ Must build and manage disaster recovery systems manually. |
| **High Security/Compliance Requirements** | ❌ May face data residency or compliance challenges. | ✅ Full control over compliance and security measures. |
| **Latency-Critical Applications** | ❌ May suffer from network latency in some scenarios. | ✅ Data stays close to the source, reducing latency. |
| **Machine Learning & Big Data Workloads** | ✅ Integration with cloud compute (e.g., AWS, GCP, Azure) is easy. | ❌ Needs massive infrastructure and ongoing tuning. |
| **Highly Regulated Industries** | ❌ Must ensure provider meets all required certifications. | ✅ Better suited for granular control and audits. |
| **Short-Term Projects or Prototypes** | ✅ Quick to spin up and decommission as needed. | ❌ Overhead not justified for short-term needs. |

3.What is Google Cloud Storage (GCS), and what are some common use cases?

|  |  |
| --- | --- |
| **Aspect** | **Google Cloud Storage (GCS)** |
| **What It Is** | A highly scalable, durable, and secure object storage service provided by Google Cloud. |
| **Storage Type** | Object storage for unstructured data (files, images, videos, backups, etc.). |
| **Storage Classes** | - **Standard** (frequent access) - **Nearline** (low-cost, monthly access) - **Coldline** (for data rarely accessed) - **Archive** (long-term storage, very infrequent access) |
| **Key Features** | - Global distribution - Data encryption at rest - Integrated with other Google Cloud services (e.g., BigQuery, AI tools) - Easy-to-use APIs and management interfaces |
| **Security** | - IAM roles & policies - Encryption at rest & in transit - Compliance certifications (e.g., GDPR, HIPAA) |
| **Integration** | - Google Cloud services (e.g., Compute Engine, Kubernetes Engine) - Third-party tools - Data analytics (e.g., BigQuery, Dataflow) |
| **Cost Model** | Pay-as-you-go model based on storage volume and access frequency (e.g., Standard, Nearline, etc.). |

--------------------------------------------------------------------------------------------------------------

|  |  |
| --- | --- |
| **Use Case** | **Description** |
| **Backup and Archiving** | Storing large volumes of backup data or long-term archives (e.g., log files, old records). |
| **Big Data Analytics** | Storing large datasets that are analyzed using tools like BigQuery or Dataflow. |
| **Media & Content Storage** | Hosting images, videos, and other media for applications, websites, or streaming services. |
| **Data Sharing & Collaboration** | Sharing large datasets across teams or external partners with secure access controls. |
| **Disaster Recovery** | Ensuring data availability and quick recovery in case of system failures. |
| **Machine Learning & AI Data Storage** | Storing training data, models, and other assets required for ML and AI applications. |
| **IoT Data Storage** | Storing time-series data generated by IoT devices for later analysis. |
| **Website Hosting** | Hosting static websites (e.g., HTML, CSS, JavaScript files) directly from GCS. |
| **Data Migration** | Moving data between on-premise systems and the cloud or between cloud regions. |

4.What is the role of BigQuery in Google Cloud, and how is it different from a traditional database?

BigQuery plays a central role in **Google Cloud Platform (GCP)** as a **fully-managed, serverless, cloud-based data warehouse** that is designed for **fast SQL-based analytics on large datasets**. It's purpose-built to handle **petabyte-scale data** with ease, and is widely used for **business intelligence (BI), data science, and real-time analytics**.

### **1. Serverless Architecture**

* **BigQuery**: No infrastructure to manage — Google handles all provisioning, scaling, and maintenance.
* **Traditional DB**: Requires manual provisioning and scaling (e.g., hardware, storage, compute).

### **⚡ 2. Performance at Scale**

* **BigQuery**: Optimized for **very large datasets**, processing queries across thousands of nodes in parallel.
* **Traditional DB**: Performance can degrade with large datasets unless heavily optimized or scaled out.

### **💸 3. Pay-per-Query Pricing**

* **BigQuery**: Charges based on the **amount of data processed** per query or through a flat-rate model.
* **Traditional DB**: Typically involves **license fees**, hardware costs, or always-on cloud instances.

### **🧠 4. SQL-Based Analytics + ML Integration**

* **BigQuery**: Supports **standard SQL**, with built-in machine learning via **BigQuery ML**, geospatial analysis, and support for federated queries (across GCS, Cloud SQL, etc.).
* **Traditional DB**: Generally offers SQL querying but may not have built-in ML capabilities or require separate tools.

### **🔄 5. Real-Time Analytics**

* **BigQuery**: Can **stream data in real-time** for immediate querying (e.g., from IoT devices or event streams).
* **Traditional DB**: Often not optimized for real-time ingestion and analytics at this scale.

### **🌍 6. Integration with Google Cloud Ecosystem**

* Seamless integration with **Dataflow, Dataproc, Looker, Cloud Storage, Pub/Sub**, and more

1. What are the main types of cloud service models (IaaS, PaaS, SaaS), and which Google Cloud services fall under each?

## **Cloud Service Models**

### **1. IaaS (Infrastructure as a Service)**

You manage: applications, data, runtime, middleware  
 Cloud handles: virtualization, servers, storage, networking

* Think of it like renting hardware.
* You set up and control the environment (OS, runtime, apps).

#### **🔧 Google Cloud IaaS Services:**

* **Compute Engine** – VMs you fully control
* **Cloud Storage** – object storage, raw files
* **Persistent Disks** – block storage for VMs
* **VPC (Virtual Private Cloud)** – networking
* **Cloud Load Balancing** – traffic distribution
* **Cloud VPN / Interconnect** – hybrid networking

### **2. PaaS (Platform as a Service)**

You manage: applications, data  
 Cloud handles: runtime, middleware, OS, infrastructure

* You focus on code and data; the platform handles the rest.
* Best for developers who don’t want to manage infrastructure.

#### **🔧 Google Cloud PaaS Services:**

* **App Engine** – deploy code, GCP handles everything else
* **Cloud Functions** – event-driven serverless functions
* **Cloud Run** – containerized apps, serverless
* **Cloud SQL / Firestore / BigQuery** – managed databases
* **Dataflow** – stream & batch data pipelines (managed Apache Beam)
* **AI Platform** – deploy ML models without managing infrastructure

### **3. SaaS (Software as a Service)**

You manage: nothing (just use the software)  
 Cloud handles: everything

* End-user apps delivered over the internet.
* You just log in and use them.

#### **🔧 Google Cloud SaaS Offerings (often via Google Workspace):**

* **Gmail** – cloud email
* **Google Drive** – cloud file storage
* **Google Docs / Sheets / Slides** – collaboration tools
* **Google Meet** – video conferencing**Looker / Looker Studio** – data analytics and visualization tools (also a bit of PaaS)