Prathamesh Arvind Jadhav

Unit-4

1.Compare & contrast between static & Dynamic provisioning, cost & performance metrix?

Static vs Dynamic Provisioning: A Comparative Study

Provisioning refers to allocating resources (like compute, storage, or network) in a system or cloud environment. Static and dynamic provisioning are two approaches with different implications for cost, performance, and flexibility.

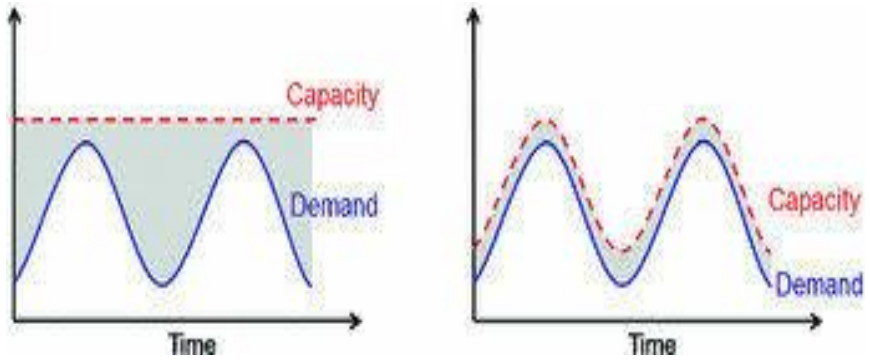
---

1. Definition

Aspect	Static Provisioning	Dynamic Provisioning
Meaning	Resources are allocated manually and remain fixed regardless of usage.	Resources are allocated automatically based on real-time demand.
Example	Allocating a fixed 8GB RAM to a server even if it only uses 2GB.	Automatically scaling a server from 2GB to 8GB RAM when needed.

---

2. Diagram:



---

3. Comparison: Cost Metrics

Criteria	Static Provisioning	Dynamic Provisioning
Cost Efficiency	Often **inefficient**—pay for unused resources.	**Highly efficient**—only pay for what is used.
Billing	Based on **maximum capacity provisioned**.	Based on **actual usage** or consumption.

Criteria	Static Provisioning	Dynamic Provisioning
Wastage	High, if resource usage is less than provisioned.	Low, as resources scale with demand.

---

4. Comparison: Performance Metrics

Criteria	Static Provisioning	Dynamic Provisioning
Performance Under Load	May perform poorly under unexpected load unless overprovisioned.	Performs well under load due to **auto-scaling**.
Responsiveness	Slower to react to usage spikes; manual intervention needed.	Fast and responsive to changes in demand.
Resource Utilization	Low or inefficient in most cases.	High and optimal utilization.

---

5. Use Cases

Static Provisioning	Dynamic Provisioning
Legacy systems, low variability workloads	Cloud-native apps, web servers, AI/ML jobs

---

6. Advantages and Disadvantages

Feature	Static	Dynamic
Simplicity	✔ Easy to configure	□ Complex setup (automation required)
Flexibility	□ Rigid	✔ Highly adaptable
Cost Saving	□ Not cost-effective	✔ Cost-efficient
Performance	□ Wastage or bottlenecks	✔ Balanced performance

2.Explain about hybrid cloud & resource interaction?

Hybrid Cloud and Resource Interaction

---

1. Introduction to Hybrid Cloud

A **Hybrid Cloud** is a computing environment that combines:

- **Private Cloud** (on-premises or dedicated resources)

- **Public Cloud** (services offered by providers like AWS, Azure, GCP)
- **Orchestration** and integration between them

It allows **data and applications** to move seamlessly between the two environments, offering greater flexibility and optimization of resources.

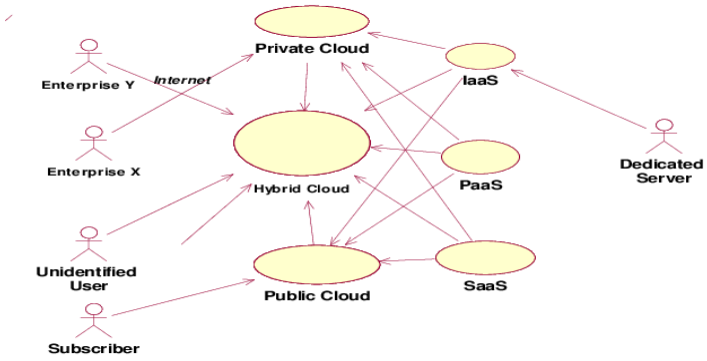
---

2. Key Features of Hybrid Cloud

Feature	Description
Flexibility	Run workloads in public or private cloud as needed
Scalability	Use public cloud for dynamic scaling
Security	Keep sensitive data on private infrastructure
Cost Efficiency	Balance between cost-saving (public cloud) and control (private cloud)

---

3. Architecture Diagram



---

4. Resource Interaction in Hybrid Cloud

Resource interaction refers to how computing resources across **private** and **public** clouds **interact and exchange data/workloads**. This includes:

A. Data Integration

- Data from private cloud databases is shared with public cloud analytics tools.
- E.g., logs from internal systems sent to public cloud for AI processing.

B. Application Layer Interaction

- Front-end of the app hosted on a public cloud (for speed).
- Backend (with sensitive data) on a private cloud.

C. Load Balancing

- Load balancers distribute traffic between public and private cloud.
- Helps in disaster recovery and performance optimization.

D. APIs and Middleware

- APIs enable communication between services in both environments.
- Middleware ensures compatibility and synchronization.

E. Orchestration Tools

- Tools like Kubernetes, Ansible, or Terraform manage deployment and scaling across hybrid setups.

---

5. Benefits of Hybrid Cloud

Benefit	Explanation
Control	Sensitive data remains in-house.
Agility	Public cloud handles variable workloads.
Cost Efficiency	Avoid overpaying for unused private infrastructure.
Disaster Recovery	Redundant systems in public cloud provide high availability.

---

6. Challenges in Hybrid Cloud

Challenge	Explanation
Integration Complexity	Difficult to unify different environments.
Security	Managing security policies across clouds.
Latency	Data transfer between clouds may introduce delays.
Compliance	Ensuring compliance across multiple platforms.

---

7. Use Cases

- **Healthcare:** Patient data in private cloud, analytics in public cloud.
- **Finance:** Customer data in-house, risk modeling in public cloud.
- **Education:** Student records in private; e-learning tools in public.

3.Solve the case which is the best suitable among cloud type, service type, platform type for video streaming platform?

Best Cloud Type, Service Type, and Platform Type for a Video Streaming Platform

Designing a video streaming platform (like Netflix, YouTube, or Hotstar) requires choosing the **most efficient cloud model**, **service model**, and **platform type** to ensure high availability, scalability, cost-efficiency, and performance.

---

1. Cloud Type: Public Cloud (Best Suitable)

Criteria	Justification
Scalability	Public cloud providers (like AWS, Azure, GCP) offer global auto-scaling to handle millions of users.
Availability	High uptime and redundancy using multiple data centers across regions.
Cost-Effective	Pay-as-you-go model fits fluctuating demand (e.g., viral videos or live events).
CDN Support	Public clouds offer built-in **Content Delivery Networks (CDN)** for faster video delivery globally.
Examples	Netflix uses AWS Public Cloud.

 Why not Private or Hybrid?

- Private cloud is too expensive and not scalable enough.
- Hybrid cloud is complex and better suited for data-sensitive enterprises like healthcare.

---

2. Service Type: PaaS + SaaS Combo

Service Model	Reason
PaaS (Platform as a Service)	Used for application development, video encoding, streaming logic, API management (e.g., using AWS Elastic Beanstalk, Azure App Services).
SaaS (Software as a Service)	Used for ready-made services like analytics, AI recommendations, or video player SDKs.
Benefits	Reduces infrastructure management, supports fast development and deployment.

 Why not IaaS?
IaaS requires managing VMs, storage, networks — more effort, less agility compared to PaaS/SaaS.

---

3. Platform Type: Cross-Platform (Best Fit)

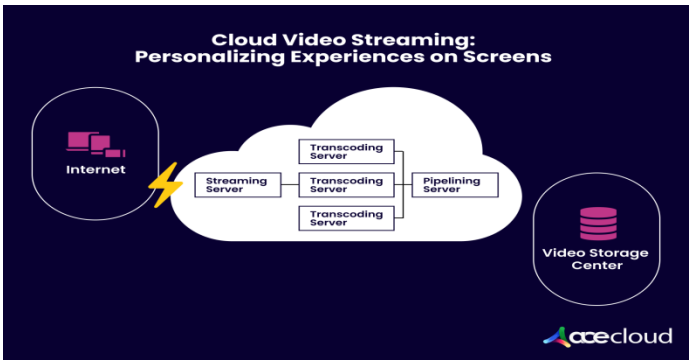
Platform	Justification
Cross-Platform (Web + Mobile + Smart TV)	Ensures users can access the platform on **Android, iOS, browsers, Smart TVs, etc.**
Technologies	React Native, Flutter, or Progressive Web Apps (PWAs) for frontend. Backend hosted on scalable cloud infrastructure.
User Reach	Maximizes user base and improves UX across devices.

 Why not Native Only or Single Platform?
Limiting to native or single platform cuts off a large user base and increases development cost.

---

4. Architecture Diagram

Prathamesh Arvind Jadhav



4.List all cloud resource allocation model, Explain in details?

Cloud Resource Allocation Models

---

1. Introduction

Cloud resource allocation is the process of assigning available computing, storage, and networking resources to cloud applications or services in an optimal way.

Efficient resource allocation ensures:

- Maximum utilization
- Better performance
- Lower operational cost
- SLA (Service Level Agreement) compliance

---

2. Types of Cloud Resource Allocation Models

There are **four major cloud resource allocation models**:

---

1. Static Resource Allocation

Feature	Description
Definition	Resources are pre-allocated at the time of deployment and do not change during runtime.
Usage	Suitable for applications with **predictable workloads**.

Feature	Description
Pros	Simple to implement, low overhead.
Cons	Poor resource utilization if demand fluctuates.

Example:
A video encoding service with fixed CPU/GPU usage regardless of traffic spikes.

---

2. Dynamic Resource Allocation

Feature	Description
Definition	Resources are allocated **on-demand** based on real-time usage and workload variation.
Usage	Suitable for **web apps, media streaming, or e-commerce platforms** with unpredictable loads.
Pros	Cost-effective, high scalability, responsive to changes.
Cons	Complex to manage, risk of SLA violations if scaling is delayed.

Example:
Autoscaling in AWS EC2 or Azure VM scale sets.

---

3. Market-Based Resource Allocation

Feature	Description
Definition	Resources are allocated based on **bidding, auctions, or pricing strategies**.
Usage	Used in **spot pricing or reserved instances** models.
Pros	Helps providers optimize revenue; allows users to get cheaper resources.
Cons	Resources may be reclaimed; not suitable for critical applications.

Example:
AWS Spot Instances where users bid for unused EC2 capacity.

---

4. Priority-Based Resource Allocation

Feature	Description
Definition	Resources are assigned based on **user/application priority levels**.

Feature	Description
Usage	Suitable for **multi-tenant systems or enterprise clouds** with tiered access.
Pros	Ensures VIP users or critical apps get resources first.
Cons	May starve lower-priority tasks.

Example:
Gold/Silver/Bronze tier resource quotas in private cloud environments.

---

3. Comparison Table

Model	Allocation	Use Case	Pros	Cons
Static	Fixed at start	Batch jobs	Simple	Wasteful
Dynamic	On demand	Web apps	Flexible	Complex
Market-Based	Bid-based	Cost-saving tasks	Cheap	Unstable
Priority-Based	Based on levels	Tiered services	Fair allocation	Starvation risk

---

5.Explain about intercloud management to share trade resource between cloud service providers & cloud users?

Intercloud Management: Sharing and Trading Resources Between Cloud Providers & Users

---

1. Introduction

As cloud computing grows rapidly, **no single cloud provider can meet all customer demands**. To overcome limitations like resource shortages, vendor lock-in, or geographical constraints, **Intercloud Computing** (also called Cloud Federation) is used.

Intercloud Management enables:

- Collaboration between **multiple cloud service providers (CSPs)**
- Seamless **resource sharing and trading**
- **Load balancing**, **cost optimization**, and **high availability**

---

2. What is Intercloud?

Prathamesh Arvind Jadhav

The **Intercloud** is a **network of interconnected clouds** (public, private, or hybrid) that cooperate to offer scalable, on-demand services beyond a single cloud's limits.

It is similar to how the **Internet** connects multiple networks — the **Intercloud** connects multiple clouds.

---

3. Intercloud Management Functions

Function	Description
Resource Discovery	Identify available virtual machines, storage, or services across clouds.
Resource Allocation	Assign resources to users from the best-fit provider.
SLA Negotiation	Ensure service-level agreements (latency, uptime) are maintained across providers.
Billing & Metering	Track cross-provider usage for accurate cost sharing.
Interoperability	Use common APIs and standards (e.g., OpenStack, OCCI) for compatibility.

---

4. Resource Sharing and Trading

Role	How Resources are Shared/Traded
Cloud Providers (CSPs)	Offer unused resources to peer providers or users via federated systems.
Cloud Users	Access compute/storage/network from external clouds if their primary CSP lacks resources.
Intercloud Brokers	Mediate between CSPs and users, helping match demand to supply and managing SLA.

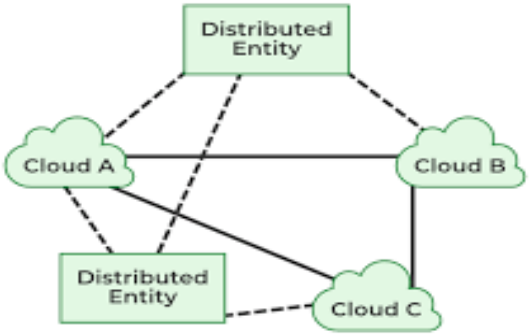
---

5. Benefits of Intercloud Management

Benefit	Description
Scalability	If one cloud is overloaded, it offloads to a partner cloud.
Cost Efficiency	Users can choose cheaper or more efficient clouds.
Reliability	Ensures fault-tolerance via resource redundancy.
Avoid Vendor Lock-in	Freedom to switch or combine multiple providers.
Better Resource Utilization	Idle resources in one cloud can be used by another.

---

6. Diagram: Intercloud Resource Management



7. Real-Life Example

- **Netflix** primarily uses AWS but can shift resources to **Azure or Google Cloud** during peak traffic or outages.
- **Science Grid or Research Projects** use federated clouds from different universities or regions to run simulations or share compute power.

---

8. Challenges in Intercloud Management

Challenge	Description
Security & Trust	Sharing resources between clouds requires strict authentication.
Interoperability	Diverse APIs and platforms can cause integration issues.
SLA Conflicts	Different clouds may define service guarantees differently.

Prathamesh Arvind Jadhav

6.Solve the case using dynamic provisioning techniques for online app & cloud resources during peakoff peak usage time?

Case Study: Dynamic Provisioning for Online App During Peak/Off-Peak Usage

---

1. Introduction

Dynamic provisioning is a cloud technique where computing resources (CPU, memory, storage, etc.) are **allocated and deallocated automatically** based on **real-time workload** or **user demand**.

It helps in:

- Handling **traffic spikes**
- Saving **operational cost**
- Ensuring **scalability and performance**

An **online application** (e.g., video streaming, food delivery, e-commerce, or social media) faces **fluctuating traffic** — high during peak hours and low during off-peak.

---

2. Problem Statement

How to handle **varying load** on an online application:

- During **peak hours** (e.g., 7–10 PM), demand surges.
- During **off-peak** (e.g., 2–5 AM), demand drops.
- Static provisioning leads to:
    - Over-provisioning → **Resource wastage**.
    - Under-provisioning → **Performance issues**.

---

3. Solution Using Dynamic Provisioning

Step 1: Enable Auto-Scaling

- Use services like **AWS Auto Scaling**, **Azure VM Scale Sets**, or **Google Cloud Instance Groups**.
- Automatically **increase/decrease** number of servers based on:
    - CPU usage

- Network traffic
- Number of active sessions

Step 2: Configure Load Balancer

- Distribute traffic among **available instances** using **Elastic Load Balancer (ELB)** or **Cloud Load Balancing**.
- Ensure no single instance is overwhelmed.

Step 3: Use Pay-as-you-Go Billing

- Cloud charges only for **actual resource usage** (not idle time).
- Reduces **cost during off-peak hours**.

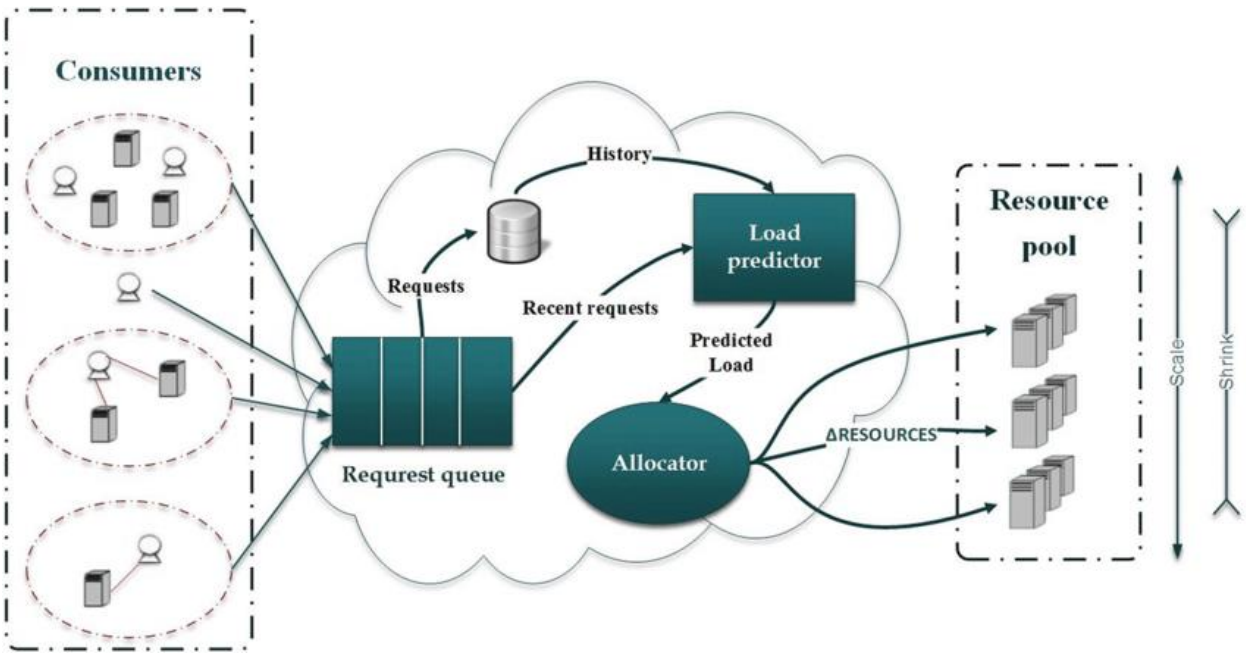
Step 4: Monitor and Predict Workload

- Use cloud monitoring tools (AWS CloudWatch, GCP Operations) to track patterns.
- **Schedule provisioning** based on historical trends (e.g., pre-scale at 6:45 PM).

Step 5: Use Containers or Serverless

- **Containerization (Docker + Kubernetes)** allows quick scaling.
- **Serverless (AWS Lambda, Azure Functions)**: No need to provision servers at all — ideal for sporadic tasks.

4. Diagram: Dynamic Provisioning During Peak vs. Off-Peak



5. Benefits

Benefit	Description
□ Performance	Ensures smooth app operation under all loads
□ Cost-Efficient	Pay only for what is used
□ Scalable	Grows/shrinks with demand
□ Reliable	Avoids server crashes during peak usage

---

6. Real-Life Example

- **Swiggy/Zomato** food delivery apps:
  - Face high traffic during lunch/dinner time.
  - Auto-scale backend services to meet demand.
- **Netflix/YouTube**:
  - Use autoscaling to stream high-quality content without buffering.

7. Explain security standards and its challenges?

Security Standards and Its Challenges in Cloud Computing

---

1. Introduction

With the growth of **cloud computing**, organizations store and manage large amounts of sensitive data online. To ensure **confidentiality, integrity, and availability** of this data, **security standards** are implemented.

These **security standards** are **guidelines, best practices, and protocols** designed to protect cloud systems from threats like data breaches, cyberattacks, and unauthorized access.

---

2. Key Security Standards in Cloud Computing

Standard	Description
ISO/IEC 27001	Provides a framework for Information Security Management Systems (ISMS). Ensures risk assessment, asset control, and access management.
ISO/IEC 27017	Cloud-specific security controls – addresses responsibilities of both providers and users.
ISO/IEC 27018	Focuses on protection of Personally Identifiable Information (PII) in public clouds.

Standard	Description
NIST SP 800-53	A set of security controls developed by the U.S. National Institute of Standards and Technology. Ensures compliance with federal security policies.
SOC 1, SOC 2, SOC 3	Service Organization Controls. SOC 2 is popular for demonstrating cloud security to customers.
CSA (Cloud Security Alliance) CCM	Cloud Controls Matrix – a cybersecurity control framework for cloud environments.
GDPR (General Data Protection Regulation)	Applies to data privacy for individuals in the EU. Mandatory for companies handling European users' data.

---

3. Goals of Security Standards

- **Data Confidentiality** – Prevent unauthorized access
- **Data Integrity** – Ensure information is accurate and unaltered
- **Availability** – Ensure data is accessible when needed
- **Accountability** – Maintain logs and user actions
- **Compliance** – Meet regulatory and legal requirements

---

4. Challenges in Implementing Cloud Security Standards

Challenge	Explanation
1. Multi-Tenancy Risks	Data from different users share the same physical resources. Risk of data leakage between tenants.
2. Data Location & Jurisdiction	Data stored in other countries may face legal issues due to local regulations (e.g., GDPR, CCPA).
3. Lack of Standardization	Different CSPs (Cloud Service Providers) follow different protocols, making integration difficult.
4. Insider Threats	Employees or administrators may misuse access rights.
5. Insecure APIs	Application interfaces may be exploited by attackers if not properly secured.
6. Key Management	Ensuring secure encryption key generation, storage, and rotation is complex.
7. Continuous Compliance	Standards evolve over time; staying up-to-date is resource-intensive.
8. Shared Responsibility Confusion	Customers and providers often misunderstand their roles in security responsibilities.
9. Performance vs. Security	Stronger security may increase latency and affect system performance.

Prathamesh Arvind Jadhav

Unit-5

1.Explain about Hadoop with it's architecture?

Hadoop and Its Architecture

1. Introduction to Hadoop

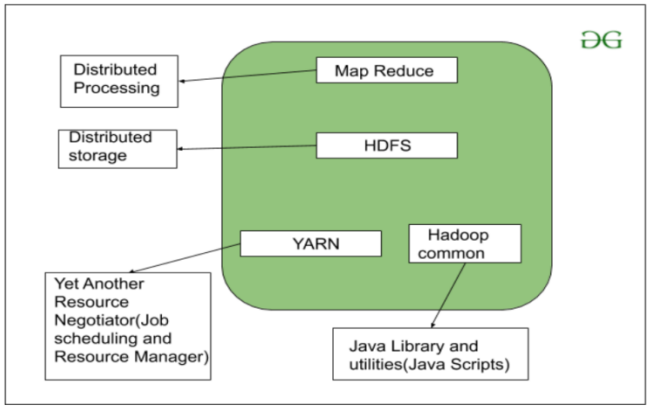
- **Hadoop** is an open-source framework designed for **processing and storing big data** in a **distributed and parallel** manner.
- It provides a **reliable**, **scalable**, and **cost-effective** way to handle vast amounts of data using **commodity hardware**.

---

2. Hadoop Architecture Overview

The architecture in the diagram consists of **four main components**:

1. **HDFS (Hadoop Distributed File System)**
2. **MapReduce**
3. **YARN (Yet Another Resource Negotiator)**
4. **Hadoop Common**



---

3. Detailed Explanation of Components

A. HDFS (Distributed Storage)

- HDFS is the **storage layer** of Hadoop.
- It stores **large files** by breaking them into **blocks** (default 128 MB) and distributing them across a **cluster** of machines (DataNodes).

- **NameNode (Master):** Manages metadata, file locations.
- **DataNode (Slave):** Stores actual data blocks.
- Ensures **fault tolerance** by **replicating blocks** (default 3 copies).

In the diagram: HDFS is shown connected to "Distributed Storage", which highlights its key role in managing large-scale data storage.

B. MapReduce (Distributed Processing)

- MapReduce is the **processing engine** in Hadoop.
- It follows a **Map → Shuffle → Reduce** model to process data in **parallel** across nodes.
- **Map task:** Processes input and produces intermediate key-value pairs.
- **Reduce task:** Aggregates and processes intermediate data to produce final results.

In the diagram: MapReduce is linked with "Distributed Processing", indicating its role in parallel computing over stored data.

C. YARN – Resource Management

- YARN stands for **Yet Another Resource Negotiator**.
- Manages **job scheduling**, **resource allocation**, and **task monitoring**.
- Core components:
    - **ResourceManager:** Allocates resources to applications.
    - **NodeManager:** Manages resources on individual nodes.
    - **ApplicationMaster:** Controls execution of a single job.

In the diagram: YARN connects with a label showing “Job scheduling and Resource Manager,” summarizing its functions.

D. Hadoop Common (Utilities & Libraries)

- Includes **Java libraries, scripts, and utilities** required by all Hadoop components.
- Acts as a **shared set of tools** to support HDFS, MapReduce, and YARN.

In the diagram: Connected with “Java Library and utilities (Java Scripts)”, showing its support function.

---

4. Summary of Interactions

- **HDFS** stores data → **MapReduce** processes it in parallel.
- **YARN** allocates resources and schedules tasks for processing.
- **Hadoop Common** provides shared tools and libraries for the entire ecosystem.

---

5. Advantages of Hadoop Architecture

- **Scalable:** Can scale to thousands of nodes.
- **Fault Tolerant:** Replicates data to prevent loss.
- **Cost-Effective:** Works on commodity hardware.
- **Efficient:** Handles large-scale data processing quickly.

2.Explain HDFC files it's working model, name, node and how to process large dataset with example?

1. Introduction to HDFS (Hadoop Distributed File System)

- **HDFS** is the **primary storage system** of **Hadoop**, designed to **store and manage large datasets** across multiple machines in a distributed environment.
- It follows a **master-slave architecture** and provides **fault tolerance**, **scalability**, and **high throughput** access to data.

---

2. Working Model of HDFS

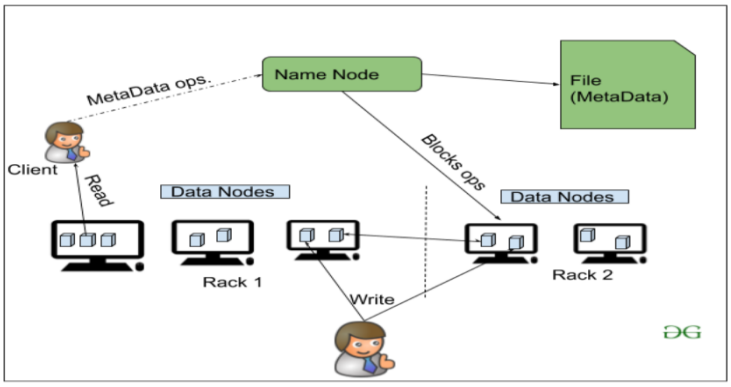
➢ Master-Slave Architecture

- **NameNode (Master):**
  - Manages **metadata** (file names, directories, block locations).
  - Does **not store actual data**, only information about where data blocks are stored.
  - Single point of control for HDFS.
- **DataNodes (Slaves):**
  - Store **actual data blocks**.
  - Perform read/write operations as instructed by the NameNode.
  - Periodically send **heartbeat** and **block reports** to the NameNode.

➢ Block-Based Storage

- Large files are **split into blocks** (default size: 128 MB).
- Each block is **replicated** (default: 3 times) and stored on **different DataNodes** to ensure **fault tolerance**.

Prathamesh Arvind Jadhav



---

3. Data Processing with HDFS

- HDFS works in coordination with **MapReduce** or **YARN** to process data.
- Data is stored in a **distributed manner** across DataNodes.
- Processing (e.g., using MapReduce) is sent to where the data resides to **reduce data movement** (data locality principle).

---

4. Step-by-Step Working Example

 Example Use Case: Processing a 300 MB Log File

Step 1: File Upload

- A 300 MB log file is uploaded to HDFS.
- The file is split into **3 blocks** of **128 MB, 128 MB, and 44 MB**.
- Blocks are distributed across **different DataNodes** (e.g., D1, D2, D3).
- **Metadata** (e.g., file name, block IDs, locations) is stored in the **NameNode**.

Step 2: Replication

- Each block is **replicated 3 times** for fault tolerance.
- So, block 1 may be on D1, D2, D3; block 2 on D2, D3, D4; etc.

Step 3: Processing

- A MapReduce job is run to analyze log patterns.
- The **Map** function runs on each DataNode that holds a block of data.
- The **Reduce** function aggregates results and outputs final data.
- Since processing is done **where the data resides**, this improves performance.

Step 4: Output

- Output of the Reduce task is stored back into HDFS or downloaded by the user.

---

5. Key Features of HDFS

Feature	Description
Fault Tolerance	Data is replicated to prevent data loss.
Scalability	Easily scales to thousands of nodes.
High Throughput	Optimized for batch processing of large datasets.
Data Locality	Computation is moved to data for efficiency.
Reliability	Handles hardware failures automatically.

3.Explain MapReduce with it's architecture and give the steps for small applications like word count?

1. Introduction to MapReduce

- **MapReduce** is a **programming model** and **processing technique** used in **Hadoop** for processing large data sets.
- It breaks down tasks into **two major phases**:
  - **Map Phase**
  - **Reduce Phase**
- Designed to run **in parallel** across distributed systems, improving **efficiency and scalability**.

---

2. MapReduce Architecture

▶□ Client

- The user submits a **job** (like Word Count) to the Hadoop system.

▶□ Job

- A **job** is the complete program or application (e.g., Word Count).
- The job is submitted to the **MapReduce Master (JobTracker)**.

▶□ Hadoop MapReduce Master

- Splits the job into **smaller sub-tasks** called **job parts**.
- Assigns **Map tasks** and **Reduce tasks** to **worker nodes**.
- Monitors progress and manages task failures.

▶□ Map Phase

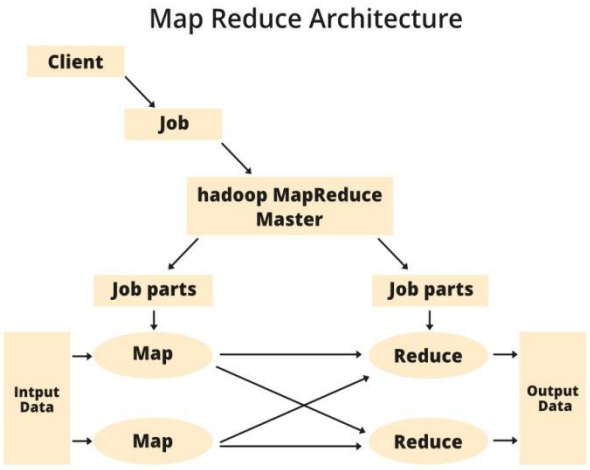
- Each Map task processes a **split of input data** (e.g., one text file).
- Produces intermediate key-value pairs.
  - Example: ("word", 1)

▶□ Shuffle and Sort (Intermediate Step)

- Intermediate key-value pairs are **shuffled and sorted** by key.
- All values for the same key are **grouped together** and sent to Reduce phase.

▶□ Reduce Phase

- Reducers **aggregate** the values for each key.
  - Example: For ("Hadoop", [1, 1, 1]), it outputs ("Hadoop", 3).
- Final output is stored in **HDFS**.



---

3. Example: Word Count Application using MapReduce

□ Objective: Count occurrences of each word in a text file.

Step-by-Step Execution

 Step 1: Input

- Text File (in HDFS):
  "Hadoop is open source. Hadoop is fast."

 Step 2: Map Phase

Each line is split into words, and the map function emits key-value pairs:

Input: "Hadoop is open source"
Output:
("Hadoop", 1)
("is", 1)
("open", 1)
("source", 1)

Input: "Hadoop is fast"
Output:
("Hadoop", 1)
("is", 1)
("fast", 1)

 Step 3: Shuffle and Sort

Intermediate data is grouped by key:

("Hadoop", [1, 1])
("is", [1, 1])
("open", [1])
("source", [1])
("fast", [1])

 Step 4: Reduce Phase

Reducer adds the values for each key:

("Hadoop", 2)

("is", 2)
("open", 1)
("source", 1)
("fast", 1)

 Step 5: Output

The result is written back to HDFS:

Hadoop 2
is 2
open 1
source 1
fast 1

---

4. Key Advantages of MapReduce

Feature	Description
Parallelism	Tasks run in parallel across nodes.
Scalability	Handles petabytes of data.
Fault Tolerance	Retries failed tasks automatically.
Data Locality	Code is moved to data, not vice versa.

4.Compare & Contrast Public & Private Cloud with the context to Openstack / Trystack environment?

Public vs Private Cloud (with OpenStack/TryStack Context)

---

1. Introduction to Cloud Models

- **Public Cloud**: Cloud services offered over the internet to multiple users by third-party providers. Example: TryStack.
- **Private Cloud**: Dedicated cloud infrastructure for a single organization, often deployed on-premises. Example: Private OpenStack setup.

---

2. Comparison Table (Core Differences)

Feature	Public Cloud (TryStack)	Private Cloud (OpenStack - Private Setup)
Access	Open to general public via the internet	Restricted to a specific organization
Ownership	Owned and maintained by cloud providers (e.g., TryStack)	Owned and controlled by the organization
Cost	Pay-as-you-go or free for demo (TryStack)	High initial cost for setup and maintenance
Scalability	Highly scalable with elastic resource provisioning	Limited scalability depending on internal resources
Security	Less control over data & compliance	Full control over data, access policies, and compliance
Customization	Limited customization (predefined templates)	High level of customization as per internal needs
Maintenance	Done by provider (no user responsibility)	Handled by internal IT team
Performance	May vary due to multi-tenancy	Consistent, as resources are dedicated
Example Environment	TryStack (OpenStack Public Trial Platform)	Private OpenStack deployment (in university/enterprise)

---

3. TryStack (Public Cloud using OpenStack)

- **TryStack** is a **public testbed** for OpenStack, allowing users to explore cloud features **without local setup**.
- Users can **create VMs**, test services, and simulate real cloud usage.
- It's shared among users, hence **limited customization and performance**.

---

4. OpenStack Private Cloud

- A company or institution may deploy **OpenStack privately** on their own servers.
- Offers **fine-grained control**, **security**, and integration with **internal systems**.
- Suitable for organizations with **strict data control policies** (banks, universities, research labs).

---

5. Use-Case Based Comparison

Scenario	Preferred Cloud Type
Student learning/practice	Public (TryStack)
Secure healthcare data storage	Private (OpenStack)
Quick app testing/deployment	Public (TryStack)
Customized enterprise workflow	Private (OpenStack)

---

6. Advantages and Disadvantages

 Public Cloud (TryStack)

Advantages:

- No installation or infrastructure required
- Free for learning
- Quick to start

Disadvantages:

- Not secure for sensitive data
- Limited features and quota

 Private Cloud (OpenStack)

Advantages:

- Full control over infrastructure
- Better performance and security
- Customizable

Disadvantages:

- High setup and maintenance cost
- Requires skilled personnel

5.Explain about Federation of cloud in details with example?

Definition:

Cloud Federation refers to the **interconnection of multiple cloud service providers** (CSPs) who collaborate and share their resources (compute, storage, services) dynamically to meet the users' requirements efficiently. It allows clouds to operate beyond their individual limitations by cooperating with other clouds.

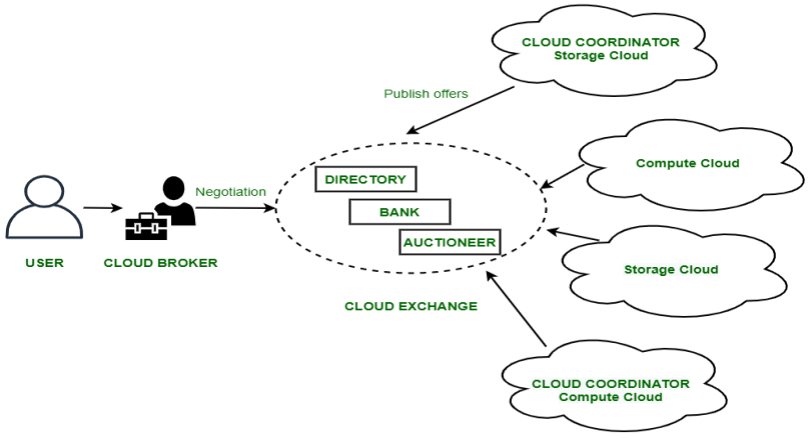
Purpose of Cloud Federation:

- To **enhance resource availability**.
- To **improve performance and scalability**.
- To **reduce cost** by selecting the best-priced offers.
- To **ensure service continuity** even during failures.

Key Components in Cloud Federation :

1. **User**: The end consumer who requires cloud services.
2. **Cloud Broker**: Acts as a **middleman** between the user and cloud providers. It negotiates the best service deal based on the user's requirements.
3. **Cloud Exchange**: A **virtual marketplace** where services are bought and sold. It includes:
    - **Directory**: Stores the list of available offers.
    - **Bank**: Handles transactions (payments, billing).
    - **Auctioneer**: Matches service demands with offers through auction-based mechanisms.
4. **Cloud Coordinator**: Manages offers from clouds and communicates with the exchange. It **publishes offers** like availability, pricing, and service level.
5. **Storage/Compute Clouds**: The actual infrastructure providers offering computing and storage services.

Prathamesh Arvind Jadhav



Working Process:

1. The **User** requests services through the **Cloud Broker**.
2. The **Broker** sends this requirement to the **Cloud Exchange**.
3. The **Exchange** receives published offers from **Cloud Coordinators**.
4. The **Auctioneer** selects the best match based on pricing, performance, and availability.
5. Once selected, the service is provisioned from the chosen cloud to the user.
6. The **Bank** module handles payments and billing.

Example Scenario:

Suppose a company wants to run a **big data analytics job** that requires large amounts of storage and high-performance computing:

- Their private cloud can't handle the current workload.
- The broker sends a request to the **Cloud Exchange**.
- Multiple CSPs (say, AWS, Azure, Google Cloud) publish their offers.
- The auctioneer matches the company's need with Azure for compute and Google Cloud for storage.
- Resources are allocated from multiple clouds in the federation.
- The company pays through the exchange's banking system.

Benefits of Cloud Federation:

1. **Resource Optimization**: Utilizes idle resources from other clouds.
2. **Cost Efficiency**: Competitive pricing through auctions or comparisons.
3. **Scalability**: Burst workloads can be offloaded to federated clouds.

4. **Reliability and Redundancy**: Alternative cloud support during failures.

Challenges:

- **Security and Trust** among different CSPs.
- **Standardization and Interoperability**.
- **Complex SLAs (Service Level Agreements)**.
- **Billing Complexity** when services span multiple providers.

Unit-6

1.Explain Hyper Jacking its attacks and its preventing?

Hyperjacking: Attacks and Prevention

---

1. What is Hyperjacking?

Hyperjacking is a **type of cyberattack where a hacker takes control of the hypervisor**, the software layer that manages virtual machines (VMs) in a virtualized environment. Once compromised, the attacker can **control all virtual machines** running on the hypervisor, making it a highly dangerous and stealthy attack vector in cloud and data center environments.

---

2. How Hyperjacking Works:

- **Hypervisor Layer Compromise:** Attackers exploit vulnerabilities in the hypervisor (e.g., VMware, Hyper-V, Xen) to gain administrative control.
- **VM Escaping:** A malicious VM breaks out of isolation to access the hypervisor.
- **Hidden Hypervisor Installation (Type 0 Attack):** Attackers install a rogue hypervisor below the existing OS to monitor or manipulate all operations.

---

3. Types of Hyperjacking Attacks:

Attack Type	Explanation
VM Escape	Malicious code escapes from a guest VM to control the host hypervisor.

Attack Type	Explanation
Direct Hypervisor Exploitation	Using vulnerabilities or misconfigurations to hijack the hypervisor.
Rootkit-based Hyperjacking	Rootkits are installed to replace or hijack the hypervisor undetected.
Firmware or BIOS-level Attack	Malware is injected into system firmware to load rogue hypervisors on boot.

---

4. Consequences of Hyperjacking:

- Complete control over all VMs.
- Data theft across multiple virtual machines.
- Creation of stealthy backdoors.
- Difficult to detect due to operation beneath the OS level.
- Interruption or manipulation of virtual workloads.

---

5. Prevention of Hyperjacking:

Preventive Measure	Description
Hypervisor Hardening	Keep hypervisors updated with latest security patches and limit unnecessary features.
Access Control & Least Privilege	Restrict admin access to hypervisors and use role-based access control (RBAC).
Use Trusted Hypervisors	Prefer enterprise-grade hypervisors with strong security features.
Firmware and BIOS Security	Use secure boot, firmware integrity checks, and update BIOS regularly.
Monitoring & Logging	Enable hypervisor-level logging and anomaly detection systems.
Isolation and Segmentation	Isolate critical workloads and use network segmentation to reduce lateral movement.
Hypervisor Attestation	Verify the integrity of the hypervisor during boot using TPM (Trusted Platform Module).

---

6. Real-World Example :

In research labs, **Blue Pill** was a proof-of-concept rootkit that installed a rogue hypervisor without detection, showing how attackers could theoretically implement hyperjacking.

Prathamesh Arvind Jadhav

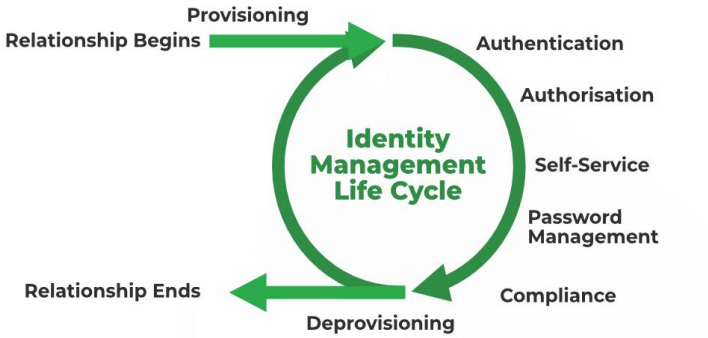
2.Explain IAM Components & Risk Mitigation Steps?

Identity and Access Management (IAM): Components & Risk Mitigation Steps

---

1. What is IAM?

Identity and Access Management (IAM) is a **framework of policies, technologies, and processes** that ensures **the right individuals access the right resources at the right time** for the right reasons in an organization. It helps manage **user identities, authentication, and authorization** securely and efficiently.



---

2. Key Components of IAM:

Component	Description
1. Identity Management	Creation, maintenance, and deletion of user identities across systems.
2. Authentication	Verifying the identity of users (e.g., passwords, biometrics, OTPs).
3. Authorization	Granting access rights to users based on their roles and policies.
4. Role-Based Access Control (RBAC)	Assigning permissions based on predefined user roles (admin, HR, user, etc.).
5. Single Sign-On (SSO)	Allowing users to access multiple services with one login credential.
6. Multi-Factor Authentication (MFA)	Adding extra layers of verification (e.g., OTP + password).
7. Privileged Access Management (PAM)	Monitoring and controlling access for users with elevated privileges.

Component	Description
8. Audit & Reporting	Tracking user activities, logins, and changes for compliance and security.

---

3. Risks Associated with IAM:

- **Weak Passwords or Poor Authentication**
- **Excessive or Unused Privileges**
- **Identity Theft and Impersonation**
- **Lack of Access Reviews**
- **Poorly Managed Lifecycle of Identities (e.g., not deactivating ex-employee accounts)**

---

4. IAM Risk Mitigation Steps:

Step	Description
1. Enforce Strong Authentication	Use MFA (Multi-Factor Authentication) to prevent unauthorized access.
2. Implement RBAC and Least Privilege	Provide users only the access needed for their roles.
3. Regular Access Reviews and Audits	Periodically review user access rights and remove unnecessary privileges.
4. Automate Identity Lifecycle Management	Automate provisioning and de-provisioning of users during onboarding/offboarding.
5. Monitor and Log Activities	Enable logging and real-time monitoring of user activities and access patterns.
6. Secure Privileged Accounts	Use PAM tools to monitor and control high-risk admin accounts.
7. Educate Users	Conduct regular security awareness training about phishing and credential risks.

---

5. Real-World Use Case :

In large enterprises like Google or Amazon, IAM tools such as **AWS IAM** or **Azure AD** are used to manage millions of user permissions securely using **role-based policies, MFA, and audit logging**.

Prathamesh Arvind Jadhav

3.Explain Cloud attacks, issues & its defects?

Cloud Attacks, Issues, and Defects

---

1. Introduction to Cloud Computing Security

Cloud computing provides **on-demand services** over the internet like storage, applications, and processing power. While it brings flexibility and scalability, it also introduces **new security threats and vulnerabilities** due to its shared and remote nature.

---

2. Common Cloud Attacks:

Attack Type	Description
1. Data Breaches	Unauthorized access to sensitive cloud-stored data (e.g., passwords, customer data).
2. Denial of Service (DoS/DDoS)	Overwhelming cloud services to make them unavailable to legitimate users.
3. Man-in-the-Middle (MitM)	Attackers intercept data in transit between users and cloud servers.
4. Account Hijacking	Attackers steal login credentials to gain unauthorized access to cloud accounts.
5. Insecure APIs	Exploiting poorly secured APIs to gain access or control over cloud services.
6. Insider Threats	Malicious insiders misusing their access to compromise cloud systems or data.
7. Cloud Malware Injection	Injecting malicious software or services into the cloud system.

---

3. Key Issues in Cloud Computing:

Issue	Explanation
1. Data Security & Privacy	Ensuring confidentiality of user data stored and processed in the cloud.
2. Data Loss	Accidental deletion or corruption of data without backup.
3. Compliance & Legal Issues	Meeting industry standards (e.g., GDPR, HIPAA) for data protection.
4. Lack of Visibility &	Limited control over data and infrastructure by cloud users.

Issue	Explanation
Control	
5. Multi-Tenancy Risks	Shared resources can lead to data leakage or improper isolation.
6. Service Availability	Dependence on internet and provider uptime for access to cloud services.

---

4. Defects in Cloud Security Models:

Defect	Explanation
1. Misconfigured Cloud Settings	Leads to exposed data, open ports, and unrestricted access (e.g., S3 bucket leaks).
2. Weak Authentication Mechanisms	Use of weak passwords or lack of MFA makes systems vulnerable to breaches.
3. Inadequate Logging & Monitoring	Delays in detecting and responding to attacks.
4. Poor Encryption Practices	Data not encrypted at rest or in transit increases risk of exposure.
5. Third-Party Risks	Use of third-party vendors can introduce vulnerabilities if not properly vetted.

---

5. Mitigation Strategies :

- Implement **Multi-Factor Authentication (MFA)**
- Use **end-to-end encryption**
- Regular **vulnerability assessments**
- Monitor APIs and enforce **secure API gateways**
- Conduct regular **audits and compliance checks**

4.What is meant by VM Migration & what are the risk during Migration?

VM Migration and Its Risks

---

1. What is VM Migration?

Virtual Machine (VM) Migration refers to the process of **moving a virtual machine from one physical host/server to another** without shutting it down. It is commonly used in cloud and virtualized environments to improve **load balancing, maintenance, fault tolerance, energy efficiency**, and **resource optimization**.

Prathamesh Arvind Jadhav

There are two main types:

- **Live Migration:** The VM continues running during migration with minimal downtime.
- **Cold Migration:** The VM is shut down and then moved.

---

2. Objectives of VM Migration:

- **Load balancing:** Distribute VMs across servers to optimize performance.
- **Fault tolerance:** Prevent VM loss due to hardware failure.
- **Server maintenance:** Migrate VMs while updating or repairing hosts.
- **Energy saving:** Power down idle servers by consolidating VMs.

---

3. Risks During VM Migration:

Risk	Explanation
1. Data Loss or Corruption	During migration, incomplete transfers or errors may lead to partial or lost data.
2. Network Congestion	Migration traffic can consume high bandwidth, affecting other services.
3. Downtime or Service Disruption	Even live migration may cause slight delays or disruptions to running applications.
4. Security Vulnerabilities	Data in transit can be intercepted if not encrypted properly.
5. Resource Overload	The target server may become overloaded if not provisioned properly.
6. Compatibility Issues	Incompatible hardware or hypervisors may result in migration failures.
7. Synchronization Errors	Differences in memory state or disk storage can lead to VM instability.
8. Configuration Mismatch	IP address conflicts, misconfigured firewalls, or policies can affect VM performance.

---

4. Risk Mitigation Strategies:

Strategy	How It Helps
Use Secure Migration Protocols	Encrypt data during transfer to avoid interception.
Pre-Migration Testing	Simulate migration to ensure compatibility and prevent crashes.

Strategy	How It Helps
Resource Monitoring	Monitor CPU, memory, and network bandwidth before and after migration.
Load Balancing Tools	Automate decision-making for safe migration timing and destination.
Post-Migration Validation	Verify data consistency and VM performance after migration.
Scheduled Maintenance Windows	Perform migrations during off-peak hours to minimize user impact.

5.Compare IAM services provided by different cloud providers with merits and demerits?

Comparison of IAM Services by Different Cloud Providers

---

1. Introduction

Identity and Access Management (IAM) services are crucial for managing **user identities, roles, and permissions** in cloud environments. Major cloud providers like **AWS, Microsoft Azure, and Google Cloud Platform (GCP)** offer IAM solutions tailored to their ecosystems. Each has **strengths and weaknesses** based on features, usability, and integration.

---

2. IAM Services Overview

Cloud Provider	IAM Service Name	Key Focus
AWS	AWS IAM	Fine-grained control over AWS resources
Azure	Azure Active Directory (AAD)	Identity services + enterprise integration
GCP	Google Cloud IAM	Resource-level access control

---

3. Feature Comparison

Feature	AWS IAM	Azure AD	GCP IAM
Role Management	Custom & predefined roles	Built-in + conditional access policies	Predefined + custom roles
Multi-Factor Auth (MFA)	Supported, strong integration	Native, enterprise-grade MFA	Supported with Google Authenticator
Single Sign-On (SSO)	Supported via AWS SSO	Advanced SSO with 3rd party integrations	Supported via Cloud Identity
User Federation	Limited (can integrate with AD)	Native integration with on-prem AD	G Suite and LDAP support
Policy Language	JSON-based, powerful and flexible	GUI-based + Conditional Access policies	IAM policy hierarchy (Org > Project > Res)
Audit & Logging	CloudTrail for tracking IAM actions	Azure Monitor, Security Center	Stackdriver (now Cloud Operations)

---

4. Merits and Demerits

Provider	Merits	Demerits
AWS IAM	- Highly granular policy control - Strong ecosystem integration - Free tier included	- Steep learning curve - Complex policy management
Azure AD	- Excellent for hybrid cloud & enterprise environments - Rich SSO support - Seamless with Office 365	- Costlier for premium features - Overhead in managing external users
GCP IAM	- Simple policy inheritance - Tight integration with G Suite - Easy-to-use interface	- Limited advanced conditional access - Slightly less mature than AWS/Azure

Prathamesh Arvind Jadhav