Google Cloud Platform - IAM (Identity and Access Management)

Done by: Malleeswari D

1. Introduction to IAM

IAM in GCP (Identity and Access Management) is a framework that allows you to manage who has what access to which resources in Google Cloud. It ensures that the right people and services have the appropriate level of access.

2. Why is IAM Important?

- Security: Controls unauthorized access to resources.
- Compliance: Ensures access meets security policies.
- Control: Allows fine-grained access management.
- Auditability: Tracks who accessed what resources and when.

3. Core IAM Concepts:

- Identity: The user or service accessing resources (e.g., user, service account).
- Role: A collection of permissions assigned to identities.
- Permission: Actions that can be performed on resources.
- Policy: A document that binds identities to roles.
- Binding: Connects members (identities) to roles.
- Resource: Any GCP service like VM, Bucket, BigQuery dataset, etc.

4. Types of Identities:

- Google Account (e.g., user@gmail.com)
- Google Group (e.g., malleeswari@company.com)
- Service Account (used by applications)
- G Suite / Cloud Identity Domain
- All Authenticated Users
- All Users (public access)

5. IAM Roles:

Basic Roles:

- Viewer: Read-only access
- Editor: Read and write access
- Owner: Full control including IAM changes

Predefined Roles:

- roles/storage.objectViewer
- roles/bigquery.dataViewer
- roles/compute.instanceAdmin

Custom Roles:

- User-defined roles with specific permissions

6. IAM Resource Hierarchy:

Organization → Folder → Project → Resource
IAM policies are inherited top-down.

7. Assigning IAM Roles:

Using Console:
1. Go to IAM & Admin → IAM
2. Click '+ Grant Access'
3. Enter identity and choose role

Using gcloud CLI:

```
gcloud projects add-iam-policy-binding PROJECT_ID \
 --member='user:email@example.com' \
 --role='roles/viewer'
```

Using Terraform:

```
resource "google_project_iam_member" "example" {
 project = "my-gcp-project"
 role    = "roles/editor"
 member  = "user:email@example.com"
}
```

8. Best Practices:

- Follow principle of least privilege
- Use predefined roles over basic roles
- Create separate service accounts per application
- Monitor IAM changes using audit logs
- Avoid public access unless required

9. Real-Time Example:

Example: Assign roles in a data project

dataanalyst@xyz.com → roles/bigquery.dataViewer (BigQuery)
dataengineer@xyz.com → roles/storage.admin (Cloud Storage)
vm-app@project.iam.gserviceaccount.com → roles/compute.instanceAdmin (Compute Engine)

10. Monitoring and Auditing IAM:

- Use Cloud Audit Logs to track changes
- Use IAM Policy Analyzer for over-permission detection
- Use Recommender for cleanup suggestions

11. Summary:

IAM in GCP is essential for controlling access securely and efficiently. Understanding identities, roles, and permissions helps build a secure cloud environment. Always follow best practices to maintain security and compliance.