**🌐 What is a Service Account in GCP?**

A **Service Account** is a **special kind of Google identity** (like a robot account) used by applications, virtual machines, and services—not by human users—to authenticate and perform actions on Google Cloud resources.

**🔍 Why Do We Need a Service Account?**

**✅ 1. Automation Needs Identity**

When you run a program (like a VM, Cloud Function, or Kubernetes Pod) that needs to talk to GCP services (e.g., Cloud Storage, BigQuery), it needs an identity to **authenticate** and **authorize** requests.

📌 **You can't use a human email account in production automation.** That’s where service accounts come in.

**💡 Real-World Analogy**

Imagine:

* Human users log in with usernames & passwords.
* Applications (code, VMs) don’t have hands to type credentials.
* So we create **service accounts**: like giving them an ID card to interact with Google Cloud.

**🧱 Structure of a Service Account**

A service account has:

* **Name**: my-app-service-account
* **Email**: [my-app@my-project.iam.gserviceaccount.com](mailto:my-app@my-project.iam.gserviceaccount.com)
* **Unique ID**: used internally by GCP
* **Optional Key**: used for authentication from outside GCP (like from a CI/CD system)

**📦 Types of Service Accounts**

| **Type** | **Purpose** |
| --- | --- |
| **User-managed** | Created and maintained by users |
| **Default service account** | Automatically created by GCP for some services (e.g., Compute Engine) |
| **Google-managed** | Used internally by GCP; not directly visible or modifiable |

**🔐 How Do Service Accounts Work?**

They:

1. **Authenticate** using:
   * Built-in credentials (when running in GCP)
   * JSON key files (outside GCP, not recommended)
2. **Authorize** using IAM roles assigned to them.
3. **Act as an identity** when calling GCP APIs.

**🚀 Use Cases for Service Accounts**

| **Scenario** | **Why Service Account?** |
| --- | --- |
| A VM uploading files to Cloud Storage | Needs identity to access storage |
| Cloud Function querying BigQuery | Needs read access to datasets |
| CI/CD pipeline deploying resources to GCP | Needs permission to create/update infrastructure |
| External app accessing GCP via APIs | Needs secure authentication |

**🛠️ How to Create and Use a Service Account**

**1. Create Service Account**

**Using GCP Console:**

1. Go to **IAM & Admin > Service Accounts**
2. Click **“Create Service Account”**
3. Give a name and description
4. Assign roles (e.g., Storage Admin)
5. (Optional) Create a key (JSON file)

**Using gcloud CLI:**

gcloud iam service-accounts create my-service-account \

--display-name="My Application SA"

**2. Grant Permissions to Service Account**

gcloud projects add-iam-policy-binding my-project-id \

--member="[serviceAccount:my-service-account@my-project.iam.gserviceaccount.com](mailto:serviceAccount%3Amy-service-account@my-project.iam.gserviceaccount.com)" \

--role="roles/storage.objectViewer"

**3. Assign Service Account to a Resource**

E.g., for a VM:

gcloud compute instances create vm-instance \

--service-account=[my-service-account@my-project.iam.gserviceaccount.com](mailto:my-service-account@my-project.iam.gserviceaccount.com) \

--scopes=<https://www.googleapis.com/auth/cloud-platform>

**🔒 Authentication Methods**

| **Method** | **When to Use** | **Caution** |
| --- | --- | --- |
| **Built-in metadata server** | For apps running inside GCP (best practice) | Secure and easy |
| **Service Account Key (JSON)** | For external tools (CI/CD, local dev) | Needs to be stored and rotated securely |
| **Workload Identity Federation** | For external identity providers (AWS, GitHub) | No key needed, secure & modern |

**🧠 Expert Tips**

| **Tip** | **Why It Matters** |
| --- | --- |
| Avoid creating and using service account keys | Keys can be leaked or misused |
| Use least privilege principle | Don't give full access unless required |
| Rotate service account keys regularly | If used, rotate and monitor them |
| Use one SA per application/service purpose | Easier to manage and audit |
| Use labels and descriptions for traceability | Better visibility in complex environments |
| Enable audit logging | To monitor who used the service account |

**🧪 How to Become Expert**

| **Learning Path** | **Activity** |
| --- | --- |
| 💻 Practice | Create VMs/functions with SAs, assign IAM roles |
| 📚 Study IAM deeply | Understand roles, policies, inheritance, custom roles |
| 🔐 Security Awareness | Learn how to avoid key leakage and use workload identity |
| 📈 Monitor usage | Use Cloud Audit Logs to see how SAs are used |
| 🛠️ Automate with Terraform | Manage service accounts and permissions as code |

**🧾 Summary Cheatsheet**

| **Term** | **Meaning** |
| --- | --- |
| Service Account | Identity used by apps/services |
| IAM Role | Set of permissions (e.g., Viewer, Admin) |
| Principal | User or service account |
| Key | JSON credentials (use sparingly) |
| Binding | Association between identity & role |