Name:- Preeti chouhan

Class:- mca 2st sem

Topic:-Secret manger

# SECRET MANAGER

## Introduction:-

In Google Cloud Platform (GCP), Secret Manager is a fully managed service designed to securely store and manage sensitive information such as API keys, passwords, certificates, and other confidential data. Secret Manager lets you store, manage, and access secrets as binary blobs or text strings. With the appropriate permissions, you can view the contents of the secret.

Secret Manager works well for storing configuration information such as database passwords, API keys, or TLS certificates needed by an application at runtime.

A key management system, such as Cloud KMS, lets you manage cryptographic keys and to use them to encrypt or decrypt data. However, you cannot view, extract, or export the key material itself.

Similarly, you can use a key management system to encrypt sensitive data before transmitting it or storing it. You can then decrypt the sensitive data before using it. Using a key management system to protect a secret in this way is more complex and less efficient than using Secret Manager.

Here's a detailed overview of its key features and how it works:

## Key Features

1. **Secure Storage**: Secret Manager encrypts your secrets both at rest and in transit using Google-managed encryption keys or your own Customer-Managed Encryption Keys (CMEK).
2. **Access Control**: It integrates with Google Cloud Identity and Access Management (IAM) to provide fine-grained access control. You can specify which users, service accounts, or applications can access specific secrets.
3. **Versioning**: Secrets are versioned, allowing you to manage and access multiple versions of a secret. This helps in scenarios where you need to rotate secrets without downtime.
4. **Audit Logging**: Secret Manager integrates with Cloud Logging, enabling you to track access and changes to your secrets. This is crucial for compliance and security monitoring.

5. **Automatic Secret Rotation**: While Secret Manager itself doesn't automate rotation, you can use Google Cloud's operations suite or custom scripts to manage and rotate secrets.
6. **Simple API and Integration**: It provides a straightforward API for managing secrets and integrates with other Google Cloud services seamlessly. You can access secrets programmatically or via the GCP Console.
7. **Secret Retrieval**: You can retrieve secrets in real-time via API calls or use the GCP Console. Secrets can be accessed in a way that ensures they are not exposed in plaintext unnecessarily.

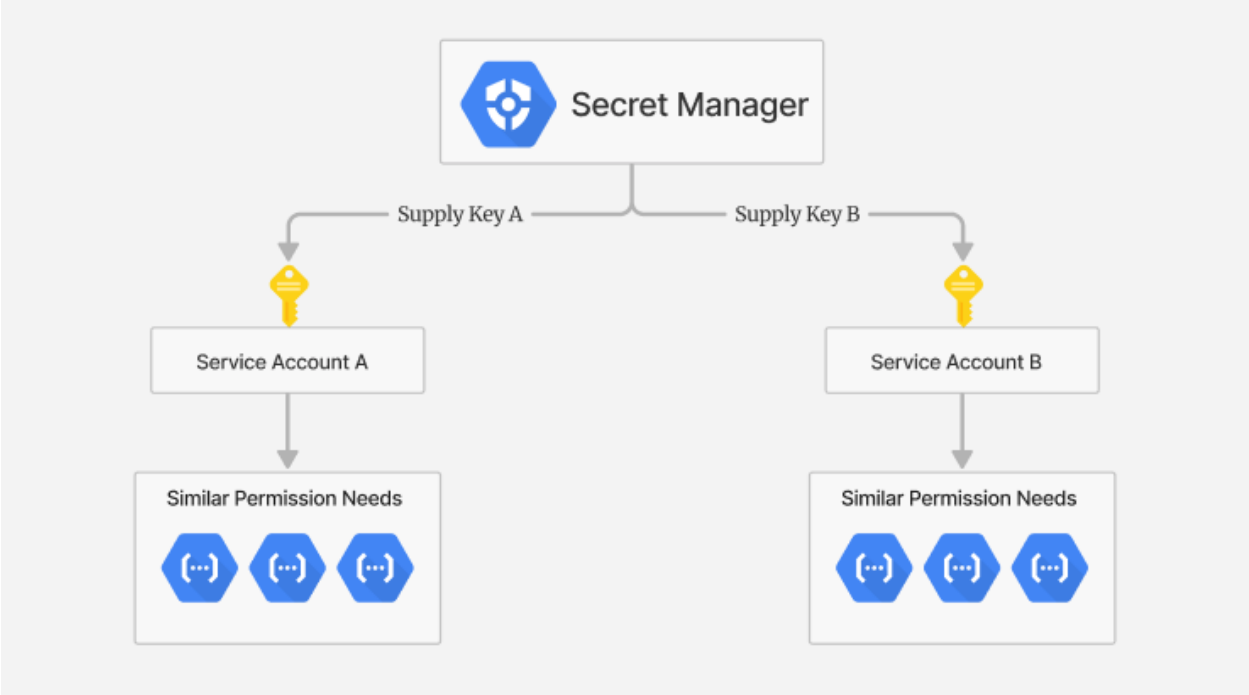
## How It Works

1. **Creating Secrets**: You create a secret by specifying its name and the secret data. This data is then encrypted and stored in the Secret Manager.
2. **Managing Versions**: When you need to update a secret, you create a new version. The old versions are retained (based on your retention policy) and can be accessed if needed.
3. **Accessing Secrets**: Applications or users access secrets by calling the Secret Manager API, which retrieves the requested secret's value. Access is controlled by IAM policies.
4. **Deleting Secrets**: You can delete secrets when they are no longer needed. Deletion can be done immediately or can be scheduled with a delay to allow for recovery if needed.

## Common Use Cases

- **Storing Credentials**: Store database passwords, API keys, or service account keys securely.
- **Configuration Management**: Manage application configuration values that need to be kept confidential.
- **Certificate Management**: Store and manage SSL/TLS certificates securely.

By using Secret Manager, you ensure that sensitive data is managed securely, reducing the risk of accidental exposure and simplifying secret management processes in your GCP environment.


Secret Manager
Supply Key A
Supply Key B
Service Account A
Service Account B
Similar Permission Needs
Similar Permission Needs