Course 6: Cloud Databases & Secrets Management

---

1. Introduction

In this course, we focus on two fundamental building blocks of cloud-native applications:

- **Managed Cloud Databases**: why we use them, their benefits, and how they integrate into cloud networking.
- **Secrets Management**: how to store and secure sensitive information like passwords, API keys, and tokens in a cloud-native way.

---

2. Cloud Databases

2.1 What Are Managed SQL Databases?

- A **Managed SQL Database** is a fully managed relational database service provided by cloud providers.

- Instead of manually running a database server (e.g. MySQL, PostgreSQL, SQL Server) on a VM, the cloud provider:

    - Handles provisioning, scaling, patching, and backups.
    - Provides automated replication and high availability options.
    - Integrates with the cloud VPC for private, secure connectivity.
    - Offer native backup, permissions and lifecycle policies management.

2.2 Services

- **Google Cloud** → **Cloud SQL** (PostgreSQL, MySQL, SQL Server).

- **AWS** → **Amazon RDS** (PostgreSQL, MySQL, MariaDB, Oracle, SQL Server).

2.3 Networking Integration

- Databases are deployed **inside the VPC**.

- Applications running in Kubernetes clusters or VMs connect through **private IPs** or **service endpoints**.

- Public IPs can be enabled but are not recommended for production.

- Cloud IAM and firewall rules control **who and what can access** the database.

---

3. Cloud Secrets Management

3.1 What Is Secrets Management?

- **Secrets** = sensitive information needed by applications:
    - Database credentials
    - API keys
    - Certificates
    - OAuth tokens
    - Etc, anything sensible

Storing these in code or config files is insecure.
A **Secrets Manager** provides a central, secure place to store, retrieve them and manage permissions.

3.2 Services

- **Google Cloud → Secret Manager**.
- **AWS → Secrets Manager**.

3.3 Why Use Them?

- Centralized and **encrypted at rest** (using KMS).
- Secrets are **rotatable** (credentials can be automatically updated).
- **IAM** for access permissions and **Audit logging** for access events.
- Remove the need to hardcode credentials in source code or configs.

3.4 Integration with Cloud IAM

- Secrets Managers integrate seamlessly with **applications and services**:
    - Kubernetes pods, serverless functions, VMs.
    - Access is controlled using **IAM** (fine-grained permissions).
- Example:
    - An app running in GKE or EKS requests a DB password from Secret Manager.
    - IAM verifies the app's identity.
    - The secret is delivered securely at runtime.

---

4. Key Takeaways

- **Managed SQL Databases** simplify database operations: automated scaling, patching, and backups, while integrating securely into the VPC.
- **Cloud Secrets Managers** provide a secure, centralized way to handle sensitive information.
- Both rely on **IAM integration** to enforce secure, least-privilege access.