AWS Secrets Manager

Overview

- AWS Secrets Manager is a **secure vault** for storing sensitive information like:
  - Database passwords
  - API keys
  - Tokens
  - Credentials
- Instead of hardcoding secrets in code or config files, applications **fetch secrets securely at runtime**.
- Runtime Retrieval: Fetching secrets securely via API at execution time, eliminating the risk of exposure in code or configuration.
- Pricing is:
  - $0.40 per secret per month
  - $0.05 per 10,000 API calls

Why AWS Secrets Manager Is Needed

- Problems without Secrets Manager:
  - Passwords hardcoded in code
  - Credentials exposed in GitHub
  - Manual password rotation
  - Security risk during breaches
- Secrets Manager solves:
  - Secure storage
  - Automatic rotation
  - Auditability
  - Fine-grained access control

AWS Secrets Manager

Common Use Cases

- Store DB credentials: Secure RDS passwords.
- Store API keys: Avoid hardcoding in code.
- Credential rotation: Automatically change passwords.
- Multi-account secrets: Central secret management.

Key Terminologies

- Secret - Stored sensitive value.
- Secret Value - Username, password, token, etc.
- Version - Each update creates a new version.
- Rotation - Automatic password update.
- KMS Key - Encrypts the secret.
- IAM Policy - Controls access to secrets.

Architecture Flow

- A secret is stored and encrypted using KMS.
- Application requests secret.
- IAM permissions are checked.
- Secrets Manager decrypts secret.
- Secret is returned securely to the app.

Execution Steps

- Create a Secret
    - Go to **Secrets Manager**

  - Click **Store a new secret**
  - Select secret type (e.g., RDS credentials)
  - Enter username and password
  - Choose KMS key
  - Name the secret
  - Disable rotation (optional)
  - Create secret
- Retrieve Secret Manually
  - Open secret
  - Click **Retrieve secret value**
  - View JSON or plain text
  - Use secret ARN in application
- EC2 Fetching DB Credentials from Secrets Manager
  - Create IAM role for EC2
  - Attach policy: secretsmanager:GetSecretValue
  - Attach role to EC2
  - Application uses AWS SDK
  - Secret fetched at runtime
  - DB connection established securely
  - No passwords stored on EC2
- Automatic Rotation
  - Enable rotation
  - Select rotation interval (e.g., 30 days)
  - AWS Lambda rotates password
  - Database updated automatically
  - Application continues working without change

Best Practices

- Never hardcode secrets.

AWS Secrets Manager

- Enable rotation for production.
- Use IAM roles, not access keys.
- Separate secrets by environment.
- Monitor access via CloudTrail.
- Use CMK for encryption.