Secrets Manager Workshop

AWS Secrets Manager - Overview

- Allows you to manage, retrieve, and rotate credentials.
- Can assist you in meeting key security controls associated with credential management.
- Includes native support for RDS PostgreSQL, MySQL, and Aurora.
- Additional credential sources can be rotated via AWS Lambda functions.
- Keeps track of different password versions.

What is a secret?

- Definition: Something that is meant to be kept unknown or unseen by others.
- In our context we'll limit our consideration of secrets to those related to securing information
- Many different types:
  - Authenticators
    - Passwords
    - API keys
  - Encryption keys
    - Symmetric
    - Asymmetric
  - Etc…

aws

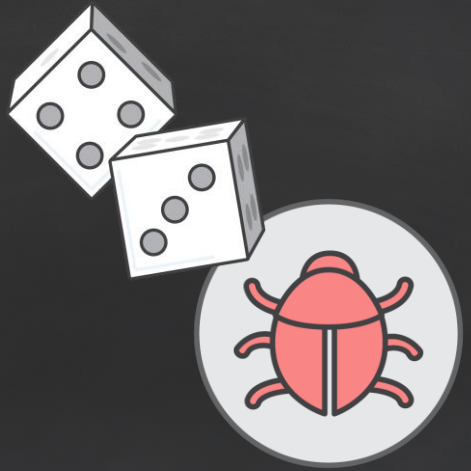
But what is the most important aspect of a secret?

- Definition: Something that is meant to be kept unseen by others.

- The problem:
  - How do you keep something unknown or unseen if it has to be shared in order to be used?

What are the challenges?

Too many humans with unnecessary access to secrets



Unreliable rotation processes

What do you need to do?

Connect to databases, APIs, and other resources, using the secrets that existing resources require.

Rotate secrets regularly without breaking stuff.

Maintain control and visibility over where, how, and by whom secrets are used.

Typical Use Cases

Connect to database from application code

- DBA loads application specific database credentials into AWS Secrets Manager.
- DevOps engineer deploys application with an attached AWS IAM role.
- Application bootstrapping calls Secrets Manager using permissions provided by the IAM role, retrieves credentials, and connects to the database.

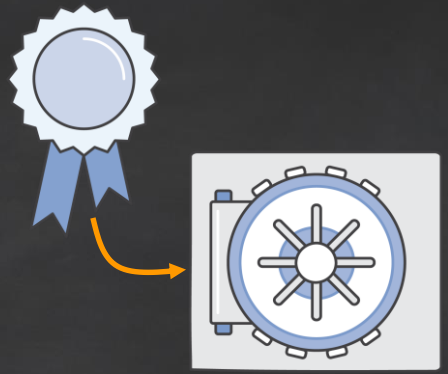
Typical Use Cases

Rotate database credentials used by application code without interruption

- Secrets Manager creates a new credential with equivalent permissions.
- The new credential is promoted and returned via subsequent Secrets Manager API calls.
- Secrets Manager safely disables the original credential.

But won’t my code break if I change the secret while it’s running?

Two scenarios:



- If your application supports only one set of credentials, your code needs to handle retry logic.
- If your code supports more than one set of credentials, use two sets of credentials and stagger the rotation.

Encryption

All secrets protected at-rest and in-transit

At-rest

- Secrets encrypted at rest using AWS Key Management Service (KMS).
- Choose your desired Customer Master Key (CMK) or AWS managed default encryption key.

In-transit

- Secrets encrypted in transit using Transport Layer Security (TLS).
- All API calls authenticated by SigV4 verification.

aws

Secrets Manager Workshop - Overview

The workshop is designed to support multiple rounds.

There is currently only one round:

Using Secrets Manager with Amazon RDS & AWS Fargate

Secrets Manager Workshop – RDS & Fargate Round

There are two phases to this round

Amazon RDS for MySQL phase (required)
AWS Fargate phase (optional)

Secrets Manager Workshop – Working with JSON

When you retrieve a secret for RDS, you get a JSON string:

```
{
   "username":”ABCDEFGH",
   "password":“IJKLMNOPQ",
   "engine":"mysql",
   "host":“aaaaaaaa.us-east-1.rds.amazonaws.com",
   "port":3306,
   "dbname":“mydbname",
   "dbInstanceIdentifier":“aaaaaaaa
 }
```

Secrets Manager Workshop – Shell Scripts + RDS

You will be using shell scripts that fetch the value of an Amazon RDS Secret using the AWS CLI and parse the JSON string using the jq command and storing the values in environment variables.

THIS IS NOT A BEST PRACTICE! You should not store credentials in environment variables. This is, however, a good way to visualize AWS Secrets Manager.

aws

Secrets Manager Workshop - Fargate

You can pass secrets to AWS Fargate using the Task Definition capability of AWS Elastic Container Service (ECS).

Modify the JSON configuration of the Task Definition to contain the secret name and an environment variable.

```
"secrets": [
    {
        "valueFrom": "arn:aws:secretsmanager:us-east-1:2[redacted]0:secret:SECRETNAME",
        "name": "TASKDEF_SECRET"
    }
],
```


aws

Secrets Manager Workshop - Fargate

When the Fargate task is started, the Task Definition retrieves the secret string and passes it to the environment variable you specify.

The task is responsible for parsing the string.

The task creates a script in /etc/profile.d that propagates the value to the login shell. More information is available in the workshop site.

aws

Secrets Manager Workshop – Get started!

Here is the workshop link:

https://secrets-manager.awssecworkshops.com

Questions?