



AWS Secrets Manager

AWS Secrets Manager is a service provided by Amazon Web Services (AWS) that helps you securely manage and rotate secrets, such as database credentials, API keys, and other sensitive information. Here's an overview of its key features:

Key Features:

1. **Secret Storage:**
 - Stores secrets securely using encryption.
 - Secrets are encrypted at rest using AWS KMS (Key Management Service) keys.
2. **Automatic Rotation:**
 - Allows you to automatically rotate secrets according to a schedule you define, reducing the risk of expired or compromised credentials.
3. **Fine-Grained Access Control:**
 - Uses AWS IAM (Identity and Access Management) to control access to secrets, ensuring only authorized users or services can retrieve them.
4. **Audit Logging:**
 - Integrates with AWS CloudTrail to log and monitor all actions taken on secrets, providing an audit trail for compliance and security purposes.
5. **Versioning:**
 - Supports multiple versions of a secret, allowing you to manage and reference different versions during secret updates or rotations.
6. **Integration with Other AWS Services:**
 - Easily integrates with AWS services like Amazon RDS, Amazon Redshift, and Amazon ECS, enabling seamless management of secrets used by these services.

Common Use Cases:

- **Database Credentials:** Securely store and automatically rotate credentials for databases like MySQL, PostgreSQL, and more.
- **API Keys:** Manage and rotate API keys for third-party services.
- **Application Secrets:** Store sensitive application configuration data, such as tokens or credentials, securely.

Benefits:

- **Enhanced Security:** Secrets are encrypted, rotated, and access is tightly controlled.
- **Reduced Risk:** Automated rotation and versioning help reduce the risks associated with stale or compromised secrets.

- **Compliance Support:** Logging and monitoring through CloudTrail support compliance with security and regulatory requirements.

AWS Secrets Manager is particularly useful in DevOps environments, where secure and automated management of sensitive information is critical for maintaining security and operational efficiency.

You're securely saving important information like a username and password in AWS Secrets Manager. You create a new secret by entering your details, giving it a name, and then storing it. Once saved, you can easily view the secret whenever needed if you have the appropriate permission, and you'll also get some code to help use it in your application. The main purpose is to keep your sensitive information safe and easily accessible when needed.

😊 To begin with the Lab:

1. In your Amazon console search and navigate to Secrets Manager. Below is the dashboard for Secrets Manager. Click on store a new secret.

2. In the secret type you can see multiple options but for the simplicity choose other type of secrets.

3. Then in the Key Value give your user and password as you can see below and click on next.

Key/value pairs [Info](#)

Key/value Plaintext

| | |
|------------|----------------|
| admin-user | admin-password |
|------------|----------------|

+ Add row

Encryption key [Info](#)

You can encrypt using the KMS key that Secrets Manager creates or a customer managed KMS key that you create.

aws/secretsmanager [▼](#) [C](#)

Add new key [\[+\]](#)

Cancel [Next](#)

4. Now you need give your secret a name and click on next. After that skip step 3 and move to review page to store your secret.

Configure secret

Secret name and description [Info](#)

Secret name

A descriptive name that helps you find your secret later.

demo-secret-credentials

Secret name must contain only alphanumeric characters and the characters /_+=.@[

Description - optional

Access to MySQL prod database for my AppBeta

Maximum 250 characters.

5. Below you can see that our Secret has been created.

⌚ You successfully stored the secret demo-secret-credentials. To show it in the list, choose Refresh.
Use the sample code to update your applications to retrieve this secret.

AWS Secrets Manager > Secrets

Secrets

Filter secrets by name, description, tag key, tag value, owning service or primary Region

| Secret name | Description | Last retrieved (UTC) |
|-------------------------|-------------|----------------------|
| demo-secret-credentials | - | - |

View details See sample code X

Store a new secret

6. Now if you go inside of your secret then you can have the overview of your Secret.

AWS Secrets Manager > Secrets > demo-secret-credentials

demo-secret-credentials

Secret details

| | |
|---|-------------------------|
| Encryption key aws/secretsmanager | Secret description - |
| Secret name demo-secret-credentials | |
| Secret ARN arn:aws:secretsmanager:us-east-1:878893308172:secret:demo-secret-credentials-qjDyl7 | |

7. If you click on retrieve secret value, you can view your secret.

Overview Rotation Versions Replication Tags

Secret value Info Retrieve secret value

Resource permissions - optional Info Edit permissions

Overview Rotation Versions Replication Tags

Secret value Info Close Edit

Retrieve and view the secret value.

Key/value Plaintext

| | |
|--------------------------|--------------------------------|
| Secret key admin-user | Secret value admin-password |
|--------------------------|--------------------------------|

8. Also, you can see that you have the sample code which you can use for your application.

Sample code

Use these code samples to retrieve the secret in your application.

Java | JavaScript | C# | Python3 | Ruby | Go | Rust

```
1 // Use this code snippet in your app.
2 // If you need more information about configurations or implementing the sample
3 // code, visit the AWS docs:
4 // https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/home.html
5
6 // Make sure to import the following packages in your code
7 // import software.amazon.awssdk.regions.Region;
8 // import software.amazon.awssdk.services.secretsmanager.SecretsManagerClient;
9 // import software.amazon.awssdk.services.secretsmanager.model.GetSecretValueRequest;
10 // import software.amazon.awssdk.services.secretsmanager.model.GetSecretValueResponse;
11
12 public static void getSecret() {
13
14     String secretName = "demo-secret-credentials";
15     Region region = Region.of("us-east-1");
16 }
```

Java Line 1, Column 1 Errors: 0 Warnings: 0

[Download AWS SDK for Java](#)

9. You can also view your Secret in CLI. You have to configure CLI in your laptop then run the below commands.
10. The first command will describe your secret and the second command will show you're the value of your secrete such as your username and password.

aws secretsmanager describe-secret --secret-id demo-secret-credentials

aws secretsmanager get-secret-value --secret-id demo-secret-credentials --version-stage AWSCURRENT

```
C:\>aws secretsmanager describe-secret --secret-id demo-secret-credentials
{
    "ARN": "arn:aws:secretsmanager:us-east-1:878893308172:secret:demo-secret-credentials-qjDyI7",
    "Name": "demo-secret-credentials",
    "LastChangedDate": "2024-09-03T16:40:01.220000+05:30",
    "LastAccessedDate": "2024-09-03T05:30:00+05:30",
    "Tags": [],
    "VersionIdsToStages": {
        "9377147d-e135-421e-99af-36ceb6878e1e": [
            "AWSCURRENT"
        ]
    },
    "CreatedDate": "2024-09-03T16:40:01.183000+05:30"
}
```

C:\>

```
C:\>aws secretsmanager get-secret-value --secret-id demo-secret-credentials --version-stage AWSCURRENT
{
    "ARN": "arn:aws:secretsmanager:us-east-1:878893308172:secret:demo-secret-credentials-qjDyI7",
    "Name": "demo-secret-credentials",
    "VersionId": "9377147d-e135-421e-99af-36ceb6878e1e",
    "SecretString": "{\"admin-user\":\"admin-password\"}",
    "VersionStages": [
        "AWSCURRENT"
    ],
    "CreatedDate": "2024-09-03T16:40:01.215000+05:30"
}
```

C:\>