# AWS Secrets Manager FAQs

## General

What is AWS Secrets Manager?

AWS Secrets Manager is a secrets management service that helps you protect access to your applications, services, and IT resources. This service enables you to easily rotate, manage, and retrieve database credentials, API keys, and other secrets throughout their lifecycle. Using Secrets Manager, you can secure and manage secrets used to access resources in the AWS Cloud, on third-party services, and on-premises.

Why should I use AWS Secrets Manager?

AWS Secrets Manager protects access to your applications, services, and IT resources, without the upfront investment and on-going maintenance costs of operating your own infrastructure.

Secrets Manager is for IT administrators looking for a secure and scalable method to store and manage secrets. Security administrators responsible for meeting regulatory and compliance requirements can use Secrets Manager to monitor secrets and rotate secrets without a risk of impacting applications. Developers who want to replace hardcoded secrets in their applications can retrieve secrets programmatically from Secrets Manager.

What can I do with AWS Secrets Manager?

AWS Secrets Manager enables you to store, retrieve, control access to, rotate, audit, and monitor secrets centrally.

You can encrypt secrets at rest to reduce the likelihood of unauthorized users viewing sensitive information. To retrieve secrets, you simply replace secrets in plain text in your applications with code to pull in those secrets programmatically using the Secrets Manager APIs. You use [AWS Identity and Access Management](https://aws.amazon.com/iam/) (IAM) policies to control which users and applications can access these secrets. You can rotate passwords, on a schedule or on demand, for supported database types hosted on AWS, without a risk of impacting applications. You can extend this functionality to rotate other secrets, such as passwords for Oracle databases hosted on Amazon EC2 or OAuth refresh tokens, by modifying sample Lambda functions. You can also audit and monitor secrets because Secrets Manager integrates with [AWS CloudTrail](https://aws.amazon.com/cloudtrail/), [Amazon CloudWatch](https://aws.amazon.com/cloudwatch/), and [Amazon Simple Notification Service](https://aws.amazon.com/sns/) (Amazon SNS).

What secrets can I manage in AWS Secrets Manager?

You can manage secrets such as database credentials, on-premises resource credentials, SaaS application credentials, third-party API keys, and Secure Shell (SSH) keys. Secrets Manager enables you to store a JSON document which allows you to manage any text blurb that is 7168 bytes or smaller.

What secrets can I rotate with AWS Secrets Manager?

You can natively rotate credentials for [Amazon Relational Database Service](https://aws.amazon.com/rds/) (RDS), [Amazon DocumentDB](https://aws.amazon.com/documentdb/), and [Amazon Redshift](https://aws.amazon.com/redshift/). You can extend Secrets Manager to rotate other secrets, such as credentials for Oracle databases hosted on EC2 or OAuth refresh tokens, by modifying sample [AWS Lambda](https://aws.amazon.com/lambda/) functions available in the [Secrets Manager documentation](https://aws.amazon.com/documentation/secretsmanager/).

How can my application use these secrets?

First, you must write an AWS Identity and Access Management (IAM) policy permitting your application to access specific secrets. Then, in the application source code, you can replace secrets in plain text with code to retrieve these secrets programmatically using the Secrets Manager APIs. For the complete details and examples, please see the [AWS Secrets Manager User Guide](https://docs.aws.amazon.com/secretsmanager/latest/userguide/).

How do I get started with AWS Secrets Manager?

To get started with AWS Secrets Manager:

1. Identify your secrets and locate where they are used in your applications.
2. Sign in to the [AWS Management Console](https://console.aws.amazon.com/console/home) using your AWS credentials and navigate to the [Secrets Manager console](https://console.aws.amazon.com/secretsmanager).
3. Use the Secrets Manager console to upload the secret you identified. Alternatively, you can use the [AWS SDK or AWS CLI](https://aws.amazon.com/tools/) to upload a secret (once per secret). You can also write a script to upload multiple secrets.
4. If your secret is not in use yet, follow the instructions on the console to configure automatic rotation. If applications are using your secret, complete steps (5) and (6) before configuring automatic rotation.
5. If other users or applications need to retrieve the secret, write an IAM policy to grant permissions to the secret.
6. Update your applications to retrieve secrets from Secrets Manager.

In what regions is AWS Secrets Manager available?

Please visit the [AWS Region Table](https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services/) to see the current region availability for AWS services.

## Rotation

How does AWS Secrets Manager implement database credential rotation without impacting applications?

AWS Secrets Manager enables you to configure database credential rotation on a schedule. This enables you to follow security best practices and rotate your database credentials safely. When Secrets Manager initiates a rotation, it uses the super database credentials provided by you to create a clone user with the same privileges, but with a different password. Secrets Manager then communicates the clone user information to databases and applications retrieving the database credentials. To learn more about rotation, refer to [AWS Secrets Manager Rotation Guide.](https://docs.aws.amazon.com/secretsmanager/latest/userguide/rotating-secrets.html)

Will rotating database credentials impact open connections?

No. Authentication happens when a connection is established. When AWS Secrets Manager rotates a database credential, the open database connection is not re-authenticated.

How do I know when AWS Secrets Manager rotates a database credential?

You can configure Amazon CloudWatch Events to receive a notification when AWS Secrets Manager rotates a secret. You can also see when Secrets Manager last rotated a secret using the Secrets Manager console or APIs.

## Security

How does AWS Secrets Manager keep my secrets secure?

AWS Secrets Manager encrypts at rest using encryption keys that you own and store in AWS Key Management Service (KMS). You can control access to the secret using AWS Identity and Access Management (IAM) policies. When you retrieve a secret, Secrets Manager decrypts the secret and transmits it securely over TLS to your local environment. By default, Secrets Manager does not write or cache the secret to persistent storage.

Who can use and manage secrets in AWS Secrets Manager?

You can use AWS Identity and Access Management (IAM) policies to control the access permissions of users and applications to retrieve or manage specific secrets. For example, you can create a policy that only enables developers to retrieve secrets used for the development environment. To learn more, visit [Authentication and Access Control for AWS Secrets Manager](https://docs.aws.amazon.com/secretsmanager/latest/userguide/auth-and-access.html).

How does AWS Secrets Manager encrypt my secrets?

AWS Secrets Manager uses envelope encryption (AES-256 encryption algorithm) to encrypt your secrets in AWS Key Management Service (KMS).

When you first use Secrets Manager, you can specify the Customer Master Keys (CMKs) to encrypt secrets. If you do not provide a CMK, Secrets Manager creates AWS KMS default keys for your account automatically. When a secret is stored, Secrets Manager requests a plaintext and an encrypted data key from KMS. Secrets Manager uses the plaintext data key to encrypt the secret in memory. AWS Secrets Manager stores and maintains the encrypted secret and encrypted data key. When a secret is retrieved, Secrets Manager decrypts the data key (using the AWS KMS default keys) and uses the plaintext data key to decrypt the secret. The data key is stored encrypted and is never written to disk in plaintext. Also, Secrets Manager does not write or cache the plaintext secret to persistent storage.

## Billing

How will I be charged and billed for my use of AWS Secrets Manager?

With Secrets Manager, you pay only for what you use, there is no minimum fee. There are no set-up fees or commitments to begin using the service. At the end of the month, your credit card will automatically be charged for that month’s usage. You are charged for number of secrets you store and for API requests made to the service each month.

For current pricing information, visit [AWS Secrets Manager pricing](https://aws.amazon.com/secrets-manager/pricing/).

Is there a free trial?

Yes, you can try Secrets Manager at no additional charge through the AWS Secrets Manager 30-day free trial. The free trial enables you to rotate, manage, and retrieve secrets over the 30-day period. The free trial starts when you store your first secret.