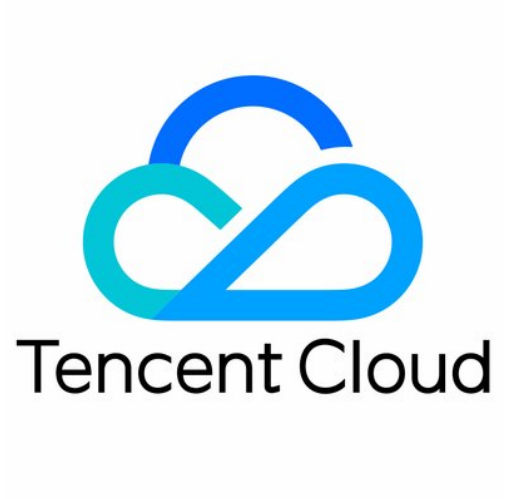
Secrets Manager

Product Introduction

Product Documentation



Contents

Product Introduction
Overview

Last updated：2024-01-02 15:07:13

Secrets Manager Overview

Secrets Manager (SSM) is a management service that enables you to create, retrieve, update, and delete secrets through their lifecycle. You can use SSM together with resource-level role authorization and comprehensive audit control to centrally manage sensitive secrets easily. Users and applications can call SSM APIs to avoid risks of sensitive configuration and sensitive secret hardcoding, and avoid sensitive information leakage as well as business risks caused by out-of-control permissions.

Strengths

Enterprise-level secret management

SSM facilitates the management of sensitive secrets, avoiding plaintext leakage caused by hardcoding and business risks caused by out-of-control permissions.

Full lifecycle management

You can use SSM to easily manage secrets through their lifecycle, such as creating, retrieving, updating, deleting, and managing permissions for secrets. You can use SSM together with resource-level role authorization and comprehensive audit control to centrally manage sensitive secrets easily.

High security and reliability

SSM adopts the clustered deployment mode. It uses the distributed database storage system to implement data storage, disaster recovery, and backup. With SSM, business users can create the same secrets in different regions to achieve cross-region disaster recovery.

Encrypted storage

Secrets are encrypted and stored by Tencent Cloud Key Management Service (KMS). Encryption keys are generated and protected by a hardware security module (HSM) certified by third parties. During secret retrieval, secrets are securely transferred by TLS to the local server.

Pay-as-you-go

SSM offers pay-as-you-go pricing. You will be billed based on the number of managed secrets and API calls in SSM. No minimum fee or setting fee is required. For more information, please see Purchase Guide.

Features

Last updated：2024-01-02 15:07:13

To avoid risks posed by sensitive configuration and hardcoded sensitive secrets, all secrets are encrypted and protected by KMS. Also, easy-to-use APIs and SDK are provided to reduce service and management costs. Users can use SSM to centrally retrieve, manage, and store encrypted database credentials, API keys, and other keys as well as sensitive configuration, avoiding plaintext leakage caused by hardcoding and business risks caused by out-of-control permissions.

Secure Secret Retrieval

With SSM, hardcoded secrets are deleted from the application source code and replaced by calls to Secrets Manager API, so you can dynamically retrieve and manage secrets programmatically.

Encrypted Secret Storage and Transfer

SSM allows encrypted storage of your secrets using a Key Management Service (KMS)-protected Customer Master Key (CMK) as encryption key, and transmit them to your local server securely over TLS.

Application-layer Secret Rotation

SSM helps you rotate and manage your secrets. It can routinely update sensitive secrets and automatically sync the update across all applications, ensuring that your applications are using the latest version of your secrets and guaranteeing business continuity.

Multi-Type Secret Storage

Multiple types of data can be stored in the format of Name-Value pairs. "Value" can be up to 4096 bytes, storing data such as database credentials, account passwords, and IP ports.

Resource-level Access Authorization

Integrating with Tencent Cloud Cloud Access Management (CAM), SSM allows only granted users to access and modify secrets via identity management and policy management, and to specify users or roles who can access these secrets by associating with these policies.

Refined Regulation and Audit

SSM combines with CloudAudit to perform supervision, compliance checks, operational reviews, and risk reviews on your Tencent Cloud accounts. All management operations and usage of the secrets can be recorded.

High-availability Disaster Recovery and Backup

SSM utilizes cluster deployment and a distributed database storage system for data storage and disaster recovery. You can create the same secrets in multiple regions to implement cross-region disaster recovery for secrets.

Security Compliance

SSM is linked to KMS. SSM uses a hardware security module (HSM) certified by third parties at its underlying layer to generate and protect secrets, meeting the supervision and compliance requirements.

Automatic Rotation

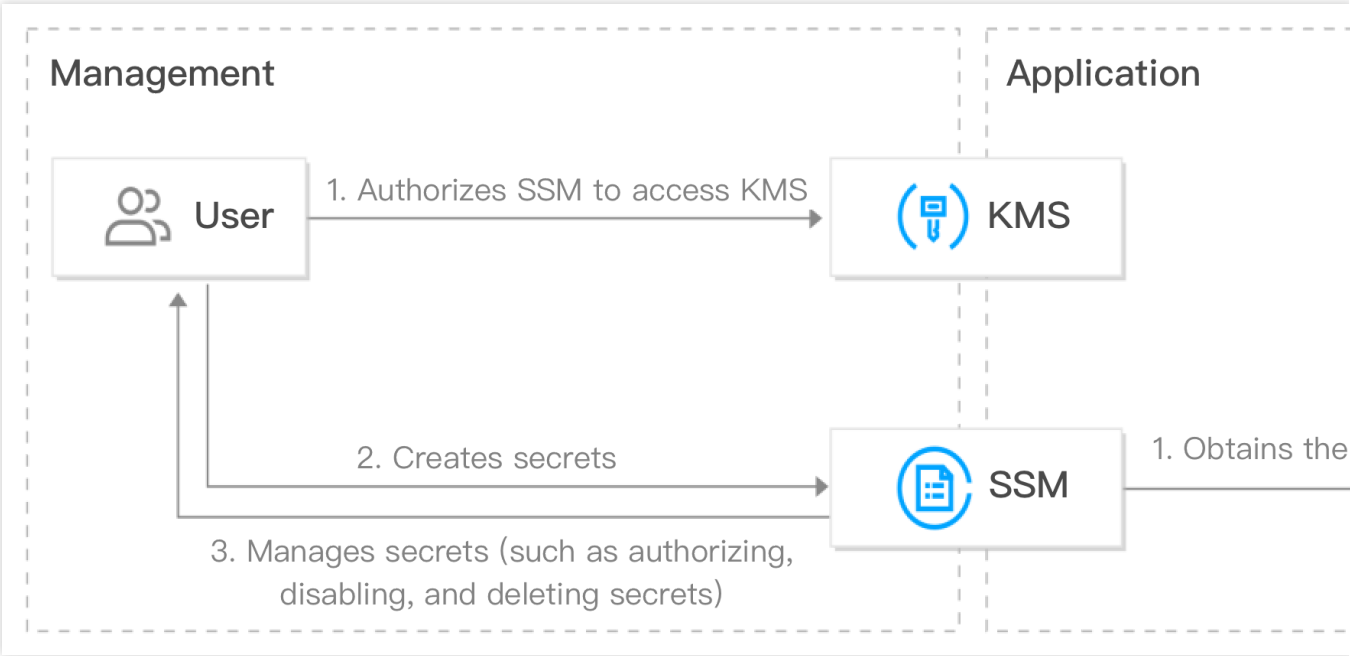
There are security challenges your account may face, such as improper use of management permission, your password unchanged for a long time, and key information in plaintext, leading to a loss of digital assets.
Given these risks, database credentials will be periodically rotated to create strong passwords and manage sensitive configuration information, securing your data while reducing security risks and threats to your account.

Use Cases

Last updated：2024-01-02 15:07:14

Managing Secrets Centrally

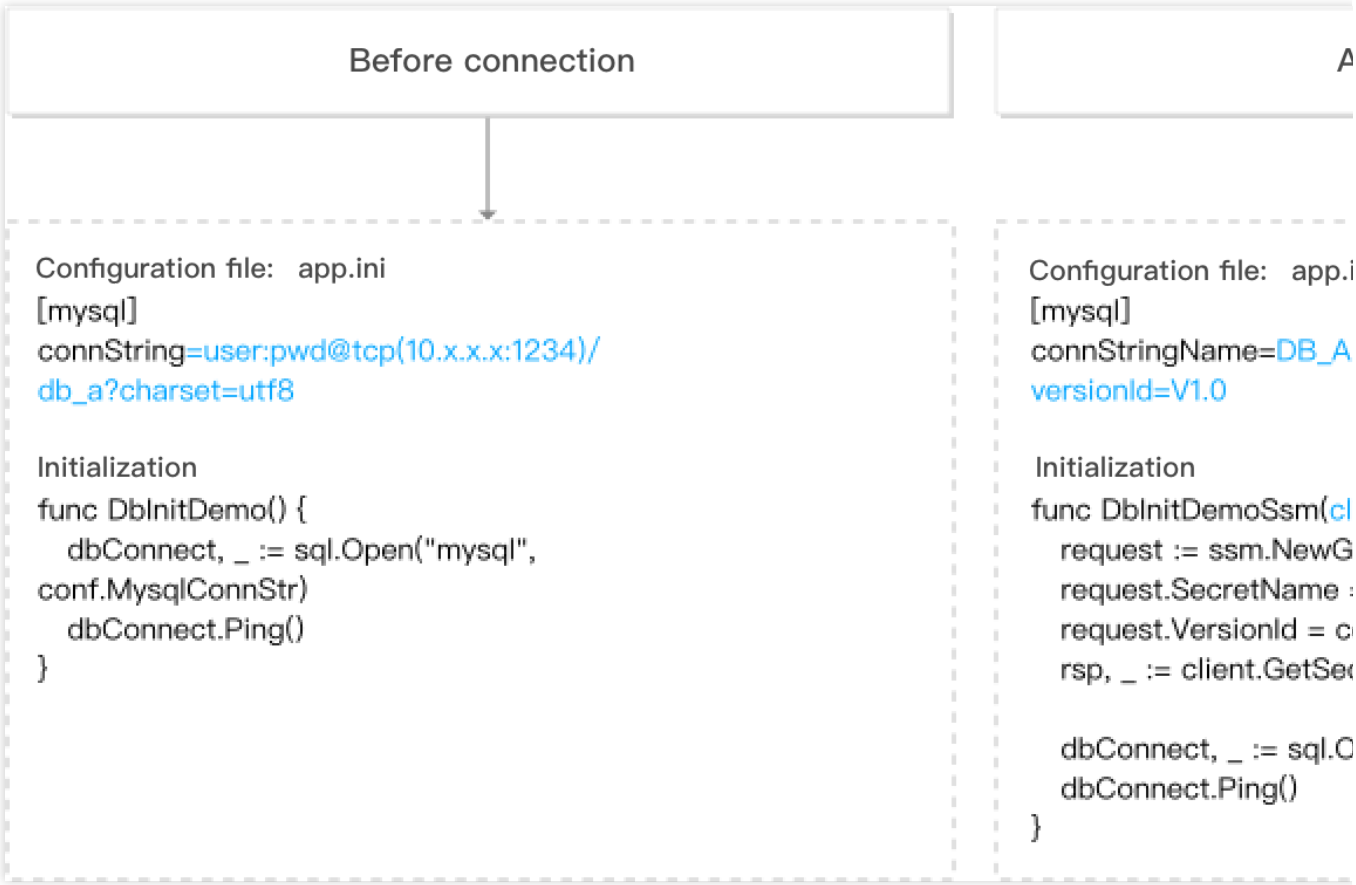
Use case: to achieve agile development, there will be lots of sensitive information (i.e., account information, tokens, certificates, SSH keys, and API keys) in the system. Therefore, there is a need to store, retrieve, use, and manage sensitive secrets through their lifecycle.
Use case example: managing secrets through their lifecycle, such as storing encrypted secrets of sensitive configuration for multiple applications, and querying and managing secrets.
Risk: hardcoding of sensitive secrets, disorganized permission management, and difficult management of hosted secrets.
Solution: developers can go to the SSM console or use the SDK or CLI to create, use, and store secrets of sensitive configuration. By using SSM together with CAM and CloudAudit, business users can manage enterprise secrets centrally through their lifecycle.



Managing Sensitive Secret Retrieval

Use case: during access to an application or service, users need to create certificates (i.e., passwords, tokens, certificates, SSH keys, or API keys) for authentication. Normally, confidential information is embedded in the configuration file of the application, which offers lower security. SSM enables you to effectively avoid risks such as the hardcoding of sensitive secrets.

Use case example: replacing database credentials, API keys, and account passwords.
Risk: information leakage of sensitive secrets.
Solution: users can replace hard-coded secrets (including passwords) with SSM APIs in the code to facilitate dynamic secret queries. Since the secret does not contain sensitive information, keys will not be leaked.



Rotating Secrets

Use case: to improve system security, sensitive secrets need to be updated periodically. Users can update secrets with SSM.
Use case example: rotating secrets at the application layer.
Risk: during secret rotation, the update needs to be synced across dependent applications and configurations. For a multi-application system, it is easy to miss an application, possibly resulting in application interruption.
Solution: you can add a secret version on the SSM console, or call APIs to update the content of the target secret. Users can decide whether to rotate secrets fully or for beta tests to sync the update across all dependent application points.