AWS — Difference between Secrets Manager and Parameter Store (Systems Manager)

Ashish Patel

Comparisons: AWS Secrets Manager vs Systems Manager Parameter Store

Awesome Cloud — AWS Secrets Manager vs Parameter Store (Systems Manager)

TL;DR:

AWS gives you two ways to store and manage application configuration data centrally:

- Secrets Manager: It was designed specifically for confidential information (like database credentials, API keys) that needs to be encrypted, so the creation of a secret entry has encryption enabled by

default. It also gives additional functionality like rotation of keys.
- Systems Manager Parameter Store: It was designed to cater to a wider use case, not just secrets or passwords, but also application configuration variables like URLs, Custom settings, AMI IDs, License keys, etc.

Similarities

Encryption

Both Secrets Manager and Parameter Store can leverage AWS KMS to encrypt values. By using KMS, IAM policies can be configured to control permissions on which IAM users and roles have permission to decrypt the value. Though access to the values can be restricted through IAM, encryption provides an additional layer of security and is sometimes required for compliance.

Key/Value Store

Both services allow you to store values under a name or key.
Both allow the keys to having prefixes. For example, parameters or secrets can be put in the following prefix schema `application/environment/parametername` or any other combination of prefixes that meets the need of the application. This is useful since the deployment of the application can reference different parameters/secrets based on the deployment environment.

CloudFormation Integration

CloudFormation is used as an Infrastructure as a code (IaC) model, and storing secrets in CloudFormation is a bad security practice. You can store the secrets (e.g. Database username and password) in Parameter Store or Secrets Manager which can be referenced in the CloudFormation template so that you just have a pointer to the value in your template instead of containing the secrets in plaintext.

Versioning

Both services support versioning of secret values. This allows you to view previous versions of your parameters of secret in case you needed them. You can choose to restore the older version of the parameter.
Parameter Store only allows one version of the parameter to be active at any given time.
Secrets Manager allows multiple versions to exist at the same time when you are performing a secret rotation using the staging labels.

Key Differences

Cost

Secrets Manager: It is paid. The storage cost is $0.40 per secret per month and API interactions cost is $0.05 per 10,000 API calls.

Parameter Store: For Standard parameters, No additional charge for storage and standard throughput. For higher throughput, API interactions cost is $0.05 per 10,000 API calls.
For Advanced parameters, storage cost is $0.05 per advanced parameter per month and API interactions cost is $0.05 per 10,000 API calls.

Secrets Rotation

Secrets Manager: It offers the ability to switch secrets at any given time and can be configured to regularly rotate depending on your requirements. It provides full key rotation integration with few AWS service like RDS, Redshift, DocumentDB. For other services, AWS allows you to write custom key rotation logic using an AWS Lambda function.

Parameter Store: You can write your own function that updates credentials managed by Parameter Store, and invoking it via a CloudWatch scheduled event or Eventbridge.

Cross-account Access

Secrets Manager: Secrets can be accessed from another AWS account. It easier to share the secrets cross-accounts. This is useful if secrets are centrally managed from another AWS account or beneficial for use cases where a customer needs to share a particular secret with a partner.

Parameter Store: Not supported.

Secret Size

Secrets Manager: It can store up to 10KB secret size.

Parameter Store: Standard Parameters can store up to 4096 characters (4KB size) for each entry, and Advanced Parameters can store up to 8KB entries.

Limits

Secrets Manager: It has a limitation of storing 500,000 secrets per region per account.

Parameter Store: It has a limitation of storing 10,000 standard parameters per region per account.

Multiple Regions Replication

Secrets Manager: It lets you easily replicate your secrets in multiple AWS Regions to support applications spread across those Regions as well as disaster recovery scenarios.

Parameter Store: It doesn't support cross region replication out of the box.

Use Cases

Choose Secrets Manager if:

- You want to store only encrypted values and super easy way to manage the rotation of the secrets. For instance, for organizations that have to be PCI compliant where the mandate is to rotate your passwords every 90d, AWS Secrets Manager makes that a very easy and seamless process.

Choose Parameter Store if:

- You want cheaper option to store encrypted or unencrypted secrets.

Alternatives

- Hashicorp Vault
- Azure Key Vault
- Azure App Configuration

Summary

Parameter Store can be used to store the secrets encrypted or unencrypted fashion. It helps you optimize and streamline application deployments by storing environmental config data, other parameters and it is free. AWS Secrets Manager takes it up by few notches by providing additional functionality such as rotation of keys, cross-account access and tighter integration with AWS services.

Recommendation: Use Secrets Manager to store confidential secrets like database credentials, API keys, OAuth tokens. *Use Parameter Store to store* *other application settings,* environmental *config data, License codes, etc.*

> *Prefer storing secrets in Secrets Manager or Parameter Store instead of storing them in a config file or hard-coded in applications.*

View more from *Awesome Cloud*

- Difference between SQS and SNS
- Difference between Application load balancer and Network load balancer
- Difference between Security Groups and NACL
- Difference between Amazon Aurora and Amazon RDS
- Difference between Internet Gateway and NAT Gateway
- Difference between EKS and ECS

Happy Clouding!!!