AWS Secret Manager and Systems Manager Parameter Store

AWS Secrets Manager

AWS Secrets Manager helps you protect access to your applications, services, and IT resources without the upfront investment and on-going costs of operating your own infrastructure. Secrets Manager enables you to rotate, manage, and retrieve database credentials, API keys, and other secrets throughout their lifecycle.

Key Features:

1. **Secret Rotation:** AWS Secrets Manager can automatically rotate secrets, such as database credentials, based on a configurable schedule.
2. **Integration with AWS Services:** Easily integrate with other AWS services like Amazon RDS, Redshift, and DocumentDB for automatic secret rotation.
3. **Encryption:** All secrets are encrypted using AWS Key Management Service (KMS) keys, providing an additional layer of security.

Cost:

The cost of AWS Secrets Manager is based on the number of secrets stored and the number of API calls made.

- $0.40 per secret per month
- $0.05 per 10,000 API calls

Creating a Secret:

1. Open the AWS Management Console.
2. Navigate to Secrets Manager.
3. Choose "Store a new secret."
4. Select the type of secret you want to store (e.g., database credentials, API key).
5. Configure the settings and provide the required information.

6. Review and confirm, then click "Store" to save the secret.

Using Secrets in a Node.js Application:

To use a secret in a Node.js application, you can use the AWS SDK. Install the AWS SDK for Node.js using npm:

npm install aws-sdk

Example code to retrieve a secret in a Node.js application:

```
const AWS = require('aws-sdk');
const secretsManager = new AWS.SecretsManager();
const params = {
  SecretId: 'your-secret-name',
};
secretsManager.getSecretValue(params, (err, data) => {
  if (err) {
    console.error(err);
  } else {
    const secret = JSON.parse(data.SecretString);
    // Use the secret in your application
  }
});
```

AWS Systems Manager Parameter Store

AWS Systems Manager Parameter Store provides a secure and hierarchical storage for configuration data management and secrets management. It allows you to store parameters that can be easily referenced by other AWS services, such as EC2 Systems Manager, AWS Lambda, and AWS CloudFormation.

Key Features:

1. **Parameter Hierarchies:** Organize parameters hierarchically to simplify management and access control.
2. **Integration with AWS Services:** Easily reference parameters in other AWS services, facilitating dynamic configuration.
3. **Versioning:** Parameter Store supports versioning, allowing you to store multiple versions of a parameter.

Cost:

The cost of AWS Systems Manager Parameter Store is based on the number of parameters stored and the number of parameter reads or writes. Standard parameters are free of charge. Advanced parameters cost:

- $0.05 per advanced parameter per month
- $0.05 per 10,000 Parameter Store API interactions

Creating a Parameter:

1. Open the AWS Management Console.
2. Navigate to Systems Manager.
3. Choose "Parameter Store" in the left navigation pane.
4. Choose "Create parameter."
5. Configure the settings and provide the required information.
6. Review and confirm, then click "Create parameter" to save the parameter.

Using Parameters in a Node.js Application:

To use a parameter in a Node.js application, you can again use the AWS SDK. Install the AWS SDK for Node.js using npm:

npm install aws-sdk

Example code to retrieve a parameter in a Node.js application:

```
const AWS = require('aws-sdk');

const ssm = new AWS.SSM();
```

```
const params = {
  Name: 'your-parameter-name',
  WithDecryption: true, // Decrypt the parameter if it's encrypted
};
ssm.getParameter(params, (err, data) => {
  if (err) {
    console.error(err);
  } else {
    const parameterValue = data.Parameter.Value;
    // Use the parameter value in your application
  }
});
```

Key Differences

1. **Use Case:**
   - Secrets Manager: Primarily designed for managing and rotating secrets such as database credentials, API keys, and OAuth tokens.
   - Parameter Store: Suitable for storing a wide range of configuration data, including plaintext or encrypted parameters.
2. **Secret Rotation:**
   - Secrets Manager: Built-in support for automatic secret rotation.
   - Parameter Store: Requires manual rotation or custom implementation.
3. **Integration:**
   - Secrets Manager: Integrates well with AWS services like RDS for automatic rotation.
   - Parameter Store: Commonly used for dynamic configuration in applications.
4. **Encryption:**
   - Secrets Manager: Automatically encrypts secrets using AWS KMS.
   - Parameter Store: Provides an option for encryption but doesn't enforce it.