AWS Secrets Manager API Reference

API Reference

API Version 2017-10-17

AWS Secrets Manager API Reference: API Reference

The AWS Documentation website is getting a new look!

Try it now and let us know what you think. Switch to the new look >>

You can return to the original look by selecting English in the language selector above.

Table of Contents

Welcome

AWS Secrets Manager is a web service that enables you to store, manage, and retrieve, secrets.

This guide provides descriptions of the Secrets Manager API. For more information about using this service, see the AWS Secrets Manager User Guide.

API Version

This version of the Secrets Manager API Reference documents the Secrets Manager API version 2017-10-17.

> **Note**
> As an alternative to using the API directly, you can use one of the AWS SDKs, which consist of libraries and sample code for various programming languages and platforms (such as Java, Ruby, .NET, iOS, and Android). The SDKs provide a convenient way to create programmatic access to AWS Secrets Manager. For example, the SDKs take care of cryptographically signing requests, managing errors, and retrying requests automatically. For more information about the AWS SDKs, including how to download and install them, see Tools for Amazon Web Services.

We recommend that you use the AWS SDKs to make programmatic API calls to Secrets Manager. However, you also can use the Secrets Manager HTTP Query API to make direct calls to the Secrets Manager web service. To learn more about the Secrets Manager HTTP Query API, see Making Query Requests in the *AWS Secrets Manager User Guide*.

Secrets Manager supports GET and POST requests for all actions. That is, the API doesn't require you to use GET for some actions and POST for others. However, GET requests are subject to the limitation size of a URL. Therefore, for operations that require larger sizes, use a POST request.

Signing Requests

When you send HTTP requests to AWS, you must sign the requests so that AWS can identify who sent them. You sign requests with your AWS access key, which consists of an access key ID and a secret access key. We strongly recommend that you don't create an access key for your root account. Anyone who has the access key for your root account has unrestricted access to all the resources in your account. Instead, create an access key for an IAM user account that has the permissions required for the task at hand. As another option, use AWS Security Token Service to generate temporary security credentials, and use those credentials to sign requests.

To sign requests, you must use Signature Version 4. If you have an existing application that uses Signature Version 2, you must update it to use Signature Version 4.

When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you with the access key that you specify when you configure the tools.

Support and Feedback for AWS Secrets Manager

We welcome your feedback. Send your comments to awssecretsmanager-feedback@amazon.com, or post your feedback and questions in the AWS Secrets Manager Discussion Forum. For more information about the AWS Discussion Forums, see Forums Help.

How examples are presented

The JSON that AWS Secrets Manager expects as your request parameters and that the service returns as a response to HTTP query requests are single, long strings without line breaks or white space formatting.

The JSON shown in the examples is formatted with both line breaks and white space to improve readability. When example input parameters would also result in long strings that extend beyond the screen, we insert line breaks to enhance readability. You should always submit the input as a single JSON text string.

Logging API Requests

AWS Secrets Manager supports AWS CloudTrail, a service that records AWS API calls for your AWS account and delivers log files to an Amazon S3 bucket. By using information that's collected by AWS CloudTrail, you can determine which requests were successfully made to Secrets Manager, who made the request, when it was made, and so on. For more about AWS Secrets Manager and its support for AWS CloudTrail, see Logging AWS Secrets Manager Events with AWS CloudTrail in the *AWS Secrets Manager User Guide*. To learn more about CloudTrail, including how to turn it on and find your log files, see the AWS CloudTrail User Guide.

This document was last published on September 17, 2019.

Actions

The following actions are supported:

CancelRotateSecret

Disables automatic scheduled rotation and cancels the rotation of a secret if one is currently in progress.

To re-enable scheduled rotation, call RotateSecret (p. 64) with `AutomaticallyRotateAfterDays` set to a value greater than 0. This will immediately rotate your secret and then enable the automatic schedule.

> **Note**
> If you cancel a rotation that is in progress, it can leave the `VersionStage` labels in an unexpected state. Depending on what step of the rotation was in progress, you might need to remove the staging label `AWSPENDING` from the partially created version, specified by the `VersionId` response value. You should also evaluate the partially rotated new version to see if it should be deleted, which you can do by removing all staging labels from the new version's `VersionStage` field.

To successfully start a rotation, the staging label `AWSPENDING` must be in one of the following states:

- Not be attached to any version at all
- Attached to the same version as the staging label `AWSCURRENT`

If the staging label `AWSPENDING` is attached to a different version than the version with `AWSCURRENT` then the attempt to rotate fails.

Minimum permissions

To run this command, you must have the following permissions:

- secretsmanager:CancelRotateSecret

Related operations

- To configure rotation for a secret or to manually trigger a rotation, use RotateSecret (p. 64).
- To get the rotation configuration details for a secret, use DescribeSecret (p. 23).
- To list all of the currently available secrets, use ListSecrets (p. 41).
- To list all of the versions currently associated with a secret, use ListSecretVersionIds (p. 45).

Request Syntax

```
{
   "SecretId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 98).

The request accepts the following data in JSON format.

SecretId (p. 4)

Specifies the secret for which you want to cancel a rotation request. You can specify either the Amazon Resource Name (ARN) or the friendly name of the secret.

Note
If you specify an ARN, we generally recommend that you specify a complete ARN. You can specify a partial ARN too—for example, if you don't include the final hyphen and six random characters that Secrets Manager adds at the end of the ARN when you created the secret. A partial ARN match can work as long as it uniquely matches only one secret. However, if your secret has a name that ends in a hyphen followed by six characters (before Secrets Manager adds the hyphen and six characters to the ARN) and you try to use that as a partial ARN, then those characters cause Secrets Manager to assume that you're specifying a complete ARN. This confusion can cause unexpected results. To avoid this situation, we recommend that you don't create secret names that end with a hyphen followed by six characters.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

Response Syntax

```
{
   "ARN": "string",
   "Name": "string",
   "VersionId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ARN (p. 5)

The ARN of the secret for which rotation was canceled.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Name (p. 5)

The friendly name of the secret for which rotation was canceled.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

VersionId (p. 5)

The unique identifier of the version of the secret that was created during the rotation. This version might not be complete, and should be evaluated for possible deletion. At the very least, you should remove the `VersionStage` value `AWSPENDING` to enable this version to be deleted. Failing to clean up a cancelled rotation can block you from successfully starting future rotations.

Type: String

Length Constraints: Minimum length of 32. Maximum length of 64.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 100).

InternalServiceError

An error occurred on the server side.

HTTP Status Code: 500

InvalidParameterException

You provided an invalid value for a parameter.

HTTP Status Code: 400

InvalidRequestException

You provided a parameter value that is not valid for the current state of the resource.

Possible causes:

- You tried to perform the operation on a secret that's currently marked deleted.
- You tried to enable rotation on a secret that doesn't already have a Lambda function ARN configured and you didn't include such an ARN as a parameter in this call.

HTTP Status Code: 400

ResourceNotFoundException

We can't find the resource that you asked for.

HTTP Status Code: 400

Example

The following example shows how to cancel rotation for a secret. The RotationEnabled field is set to `false` and scheduled rotations are canceled. To resume scheduled rotations, you must reenable rotation by calling RotateSecret (p. 64).

Sample Request

```
POST / HTTP/1.1
Host: secretsmanager.region.domain
Accept-Encoding: identity
X-Amz-Target: secretsmanager.CancelRotateSecret
Content-Type: application/x-amz-json-1.1
User-Agent: <user-agent-string>
X-Amz-Date: <date>
Authorization: AWS4-HMAC-SHA256 Credential=<credentials>,SignedHeaders=<headers>,
 Signature=<signature>
Content-Length: <payload-size-bytes>

{
  "SecretId": "MyTestDatabaseSecret"
}
```

Sample Response

```
HTTP/1.1 200 OK
```

```
Date: <date>
Content-Type: application/x-amz-json-1.1
Content-Length: <response-size-bytes>
Connection: keep-alive
x-amzn-RequestId: <request-id-guid>

{
  "ARN":"arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
  "Name":"MyTestDatabaseSecret"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

CreateSecret

Creates a new secret. A secret in Secrets Manager consists of both the protected secret data and the important information needed to manage the secret.

Secrets Manager stores the encrypted secret data in one of a collection of "versions" associated with the secret. Each version contains a copy of the encrypted secret data. Each version is associated with one or more "staging labels" that identify where the version is in the rotation cycle. The `SecretVersionsToStages` field of the secret contains the mapping of staging labels to the active versions of the secret. Versions without a staging label are considered deprecated and are not included in the list.

You provide the secret data to be encrypted by putting text in either the `SecretString` parameter or binary data in the `SecretBinary` parameter, but not both. If you include `SecretString` or `SecretBinary` then Secrets Manager also creates an initial secret version and automatically attaches the staging label `AWSCURRENT` to the new version.

Note

- If you call an operation that needs to encrypt or decrypt the `SecretString` or `SecretBinary` for a secret in the same account as the calling user and that secret doesn't specify a AWS KMS encryption key, Secrets Manager uses the account's default AWS managed customer master key (CMK) with the alias `aws/secretsmanager`. If this key doesn't already exist in your account then Secrets Manager creates it for you automatically. All users and roles in the same AWS account automatically have access to use the default CMK. Note that if an Secrets Manager API call results in AWS having to create the account's AWS-managed CMK, it can result in a one-time significant delay in returning the result.
- If the secret is in a different AWS account from the credentials calling an API that requires encryption or decryption of the secret value then you must create and use a custom AWS KMS CMK because you can't access the default CMK for the account using credentials from a different AWS account. Store the ARN of the CMK in the secret when you create the secret or when you update it by including it in the `KMSKeyId`. If you call an API that must encrypt or decrypt `SecretString` or `SecretBinary` using credentials from a different account then the AWS KMS key policy must grant cross-account access to that other account's user or role for both the kms:GenerateDataKey and kms:Decrypt operations.

Minimum permissions

To run this command, you must have the following permissions:

- secretsmanager:CreateSecret
- kms:GenerateDataKey - needed only if you use a customer-managed AWS KMS key to encrypt the secret. You do not need this permission to use the account's default AWS managed CMK for Secrets Manager.
- kms:Encrypt - needed only if you use a customer-managed AWS KMS key to encrypt the secret. You do not need this permission to use the account's default AWS managed CMK for Secrets Manager.
- kms:Decrypt - needed only if you use a customer-managed AWS KMS key to encrypt the secret. You do not need this permission to use the account's default AWS managed CMK for Secrets Manager.
- secretsmanager:TagResource - needed only if you include the `Tags` parameter.

Related operations

- To delete a secret, use DeleteSecret (p. 19).
- To modify an existing secret, use UpdateSecret (p. 77).
- To create a new version of a secret, use PutSecretValue (p. 54).

- To retrieve the encrypted secure string and secure binary values, use GetSecretValue (p. 36).
- To retrieve all other details for a secret, use DescribeSecret (p. 23). This does not include the encrypted secure string and secure binary values.
- To retrieve the list of secret versions associated with the current secret, use DescribeSecret (p. 23) and examine the `SecretVersionsToStages` response value.

Request Syntax

```
{
   "ClientRequestToken": "string",
   "Description": "string",
   "KmsKeyId": "string",
   "Name": "string",
   "SecretBinary": blob,
   "SecretString": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 98).

The request accepts the following data in JSON format.

ClientRequestToken (p. 9)

(Optional) If you include `SecretString` or `SecretBinary`, then an initial version is created as part of the secret, and this parameter specifies a unique identifier for the new version.

> **Note**
> If you use the AWS CLI or one of the AWS SDK to call this operation, then you can leave this parameter empty. The CLI or SDK generates a random UUID for you and includes it as the value for this parameter in the request. If you don't use the SDK and instead generate a raw HTTP request to the Secrets Manager service endpoint, then you must generate a `ClientRequestToken` yourself for the new version and include that value in the request.

This value helps ensure idempotency. Secrets Manager uses this value to prevent the accidental creation of duplicate versions if there are failures and retries during a rotation. We recommend that you generate a UUID-type value to ensure uniqueness of your versions within the specified secret.

- If the `ClientRequestToken` value isn't already associated with a version of the secret then a new version of the secret is created.
- If a version with this value already exists and that version's `SecretString` and `SecretBinary` values are the same as those in the request, then the request is ignored (the operation is idempotent).
- If a version with this value already exists and that version's `SecretString` and `SecretBinary` values are different from those in the request then the request fails because you cannot modify an existing version. Instead, use PutSecretValue (p. 54) to create a new version.

This value becomes the `VersionId` of the new version.

Type: String

Length Constraints: Minimum length of 32. Maximum length of 64.

Required: No

Description (p. 9)

(Optional) Specifies a user-provided description of the secret.

Type: String

Length Constraints: Maximum length of 2048.

Required: No

KmsKeyId (p. 9)

(Optional) Specifies the ARN, Key ID, or alias of the AWS KMS customer master key (CMK) to be used to encrypt the `SecretString` or `SecretBinary` values in the versions stored in this secret.

You can specify any of the supported ways to identify a AWS KMS key ID. If you need to reference a CMK in a different account, you can use only the key ARN or the alias ARN.

If you don't specify this value, then Secrets Manager defaults to using the AWS account's default CMK (the one named `aws/secretsmanager`). If a AWS KMS CMK with that name doesn't yet exist, then Secrets Manager creates it for you automatically the first time it needs to encrypt a version's `SecretString` or `SecretBinary` fields.

> **Important**
> You can use the account's default CMK to encrypt and decrypt only if you call this operation using credentials from the same account that owns the secret. If the secret is in a different account, then you must create a custom CMK and specify the ARN in this field.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 2048.

Required: No

Name (p. 9)

Specifies the friendly name of the new secret.

The secret name must be ASCII letters, digits, or the following characters : /_+=.@-

> **Note**
> Don't end your secret name with a hyphen followed by six characters. If you do so, you risk confusion and unexpected results when searching for a secret by partial ARN. This is because Secrets Manager automatically adds a hyphen and six random characters at the end of the ARN.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 512.

Required: Yes

SecretBinary (p. 9)

(Optional) Specifies binary data that you want to encrypt and store in the new version of the secret. To use this parameter in the command-line tools, we recommend that you store your binary data in a file and then use the appropriate technique for your tool to pass the contents of the file as a parameter.

Either `SecretString` or `SecretBinary` must have a value, but not both. They cannot both be empty.

This parameter is not available using the Secrets Manager console. It can be accessed only by using the AWS CLI or one of the AWS SDKs.

Type: Base64-encoded binary data object

Length Constraints: Minimum length of 0. Maximum length of 7168.

Required: No

SecretString (p. 9)

(Optional) Specifies text data that you want to encrypt and store in this new version of the secret.

Either `SecretString` or `SecretBinary` must have a value, but not both. They cannot both be empty.

If you create a secret by using the Secrets Manager console then Secrets Manager puts the protected secret text in only the `SecretString` parameter. The Secrets Manager console stores the information as a JSON structure of key/value pairs that the Lambda rotation function knows how to parse.

For storing multiple values, we recommend that you use a JSON text string argument and specify key/value pairs. For information on how to format a JSON parameter for the various command line tool environments, see Using JSON for Parameters in the *AWS CLI User Guide*. For example:

```
[{"username":"bob"},{"password":"abc123xyz456"}]
```

If your command-line tool or SDK requires quotation marks around the parameter, you should use single quotes to avoid confusion with the double quotes required in the JSON text.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 7168.

Required: No

Tags (p. 9)

(Optional) Specifies a list of user-defined tags that are attached to the secret. Each tag is a "Key" and "Value" pair of strings. This operation only appends tags to the existing list of tags. To remove tags, you must use UntagResource (p. 74).

Important

- Secrets Manager tag key names are case sensitive. A tag with the key "ABC" is a different tag from one with key "abc".
- If you check tags in IAM policy `Condition` elements as part of your security strategy, then adding or removing a tag can change permissions. If the successful completion of this operation would result in you losing your permissions for this secret, then this operation is blocked and returns an `Access Denied` error.

This parameter requires a JSON text string argument. For information on how to format a JSON parameter for the various command line tool environments, see Using JSON for Parameters in the *AWS CLI User Guide*. For example:

```
[{"Key":"CostCenter","Value":"12345"},
{"Key":"environment","Value":"production"}]
```

If your command-line tool or SDK requires quotation marks around the parameter, you should use single quotes to avoid confusion with the double quotes required in the JSON text.

The following basic restrictions apply to tags:

- Maximum number of tags per secret—50
- Maximum key length—127 Unicode characters in UTF-8
- Maximum value length—255 Unicode characters in UTF-8
- Tag keys and values are case sensitive.
- Do not use the `aws:` prefix in your tag names or values because it is reserved for AWS use. You can't edit or delete tag names or values with this prefix. Tags with this prefix do not count against your tags per secret limit.
- If your tagging schema will be used across multiple services and resources, remember that other services might have restrictions on allowed characters. Generally allowed characters are: letters, spaces, and numbers representable in UTF-8, plus the following special characters: + - = . _ : / @.

Type: Array of Tag (p. 97) objects

Required: No

Response Syntax

```
{
   "ARN": "string",
   "Name": "string",
   "VersionId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ARN (p. 12)

The Amazon Resource Name (ARN) of the secret that you just created.

> **Note**
> Secrets Manager automatically adds several random characters to the name at the end of the ARN when you initially create a secret. This affects only the ARN and not the actual friendly name. This ensures that if you create a new secret with the same name as an old secret that you previously deleted, then users with access to the old secret *don't* automatically get access to the new secret because the ARNs are different.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Name (p. 12)

The friendly name of the secret that you just created.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

VersionId (p. 12)

The unique identifier that's associated with the version of the secret you just created.

Type: String

Length Constraints: Minimum length of 32. Maximum length of 64.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 100).

EncryptionFailure

Secrets Manager can't encrypt the protected secret text using the provided KMS key. Check that the customer master key (CMK) is available, enabled, and not in an invalid state. For more information, see How Key State Affects Use of a Customer Master Key.

HTTP Status Code: 400

InternalServiceError

An error occurred on the server side.

HTTP Status Code: 500

InvalidParameterException

You provided an invalid value for a parameter.

HTTP Status Code: 400

InvalidRequestException

You provided a parameter value that is not valid for the current state of the resource.

Possible causes:

- You tried to perform the operation on a secret that's currently marked deleted.
- You tried to enable rotation on a secret that doesn't already have a Lambda function ARN configured and you didn't include such an ARN as a parameter in this call.

HTTP Status Code: 400

LimitExceededException

The request failed because it would exceed one of the Secrets Manager internal limits.

HTTP Status Code: 400

MalformedPolicyDocumentException

The policy document that you provided isn't valid.

HTTP Status Code: 400

PreconditionNotMetException

The request failed because you did not complete all the prerequisite steps.

HTTP Status Code: 400

ResourceExistsException

A resource with the ID you requested already exists.

HTTP Status Code: 400

ResourceNotFoundException

We can't find the resource that you asked for.

HTTP Status Code: 400

Example

The following example shows how to create a secret. The credentials stored in the encrypted secret value are retrieved from a file on disk named mycreds.json. The JSON request string input and response output are shown formatted with white space and line breaks for better readability. Submit your input as a single line JSON string.

Sample Request

```
POST / HTTP/1.1
Host: secretsmanager.region.domain
Accept-Encoding: identity
X-Amz-Target: secretsmanager.CreateSecret
Content-Type: application/x-amz-json-1.1
User-Agent: <user-agent-string>
X-Amz-Date: <date>
Authorization: AWS4-HMAC-SHA256 Credential=<credentials>,SignedHeaders=<headers>,
 Signature=<signature>
Content-Length: <payload-size-bytes>

{
  "Name": "MyTestDatabaseSecret",
  "Description": "My test database secret created with the CLI",
  "SecretString": "{\"username\":\"david\",\"password\":\"BnQw!XDWgaEeT9XGTT29\"}",
 "ClientRequestToken": "EXAMPLE1-90ab-cdef-fedc-ba987SECRET1"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: <date>
Content-Type: application/x-amz-json-1.1
Content-Length: <response-size-bytes>
Connection: keep-alive
x-amzn-RequestId: <request-id-guid>

{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-
a1b2c3",
  "Name":"MyTestDatabaseSecret",
  "VersionId": "EXAMPLE1-90ab-cdef-fedc-ba987SECRET1"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java

- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DeleteResourcePolicy

Deletes the resource-based permission policy that's attached to the secret.

Minimum permissions

To run this command, you must have the following permissions:

- secretsmanager:DeleteResourcePolicy

Related operations

- To attach a resource policy to a secret, use PutResourcePolicy (p. 50).
- To retrieve the current resource-based policy that's attached to a secret, use GetResourcePolicy (p. 32).
- To list all of the currently available secrets, use ListSecrets (p. 41).

Request Syntax

```
{
   "SecretId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 98).

The request accepts the following data in JSON format.

SecretId (p. 16)

Specifies the secret that you want to delete the attached resource-based policy for. You can specify either the Amazon Resource Name (ARN) or the friendly name of the secret.

Note
If you specify an ARN, we generally recommend that you specify a complete ARN. You can specify a partial ARN too—for example, if you don't include the final hyphen and six random characters that Secrets Manager adds at the end of the ARN when you created the secret. A partial ARN match can work as long as it uniquely matches only one secret. However, if your secret has a name that ends in a hyphen followed by six characters (before Secrets Manager adds the hyphen and six characters to the ARN) and you try to use that as a partial ARN, then those characters cause Secrets Manager to assume that you're specifying a complete ARN. This confusion can cause unexpected results. To avoid this situation, we recommend that you don't create secret names that end with a hyphen followed by six characters.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

Response Syntax

```
{
   "ARN": "string",
   "Name": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ARN (p. 17)

The ARN of the secret that the resource-based policy was deleted for.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Name (p. 17)

The friendly name of the secret that the resource-based policy was deleted for.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 512.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 100).

InternalServiceError

An error occurred on the server side.

HTTP Status Code: 500

InvalidRequestException

You provided a parameter value that is not valid for the current state of the resource.

Possible causes:

- You tried to perform the operation on a secret that's currently marked deleted.
- You tried to enable rotation on a secret that doesn't already have a Lambda function ARN configured and you didn't include such an ARN as a parameter in this call.

HTTP Status Code: 400

ResourceNotFoundException

We can't find the resource that you asked for.

HTTP Status Code: 400

Example

The following example shows how to delete the resource-based policy that's attached to the specified secret.The JSON request string input and response output are shown formatted with white space and line breaks for better readability. Submit your input as a single line JSON string.

Sample Request

```
POST / HTTP/1.1
Host: secretsmanager.region.domain
Accept-Encoding: identity
X-Amz-Target: secretsmanager.DeleteResourcePolicy
Content-Type: application/x-amz-json-1.1
User-Agent: <user-agent-string>
X-Amz-Date: <date>
Authorization: AWS4-HMAC-SHA256 Credential=<credentials>,SignedHeaders=<headers>,
 Signature=<signature>
Content-Length: <payload-size-bytes>

{
  "SecretId": "MyTestDatabaseSecret"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: <date>
Content-Type: application/x-amz-json-1.1
Content-Length: <response-size-bytes>
Connection: keep-alive
x-amzn-RequestId: <request-id-guid>

{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-
a1b2c3",
  "Name": "MyTestDatabaseSecret"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DeleteSecret

Deletes an entire secret and all of its versions. You can optionally include a recovery window during which you can restore the secret. If you don't specify a recovery window value, the operation defaults to 30 days. Secrets Manager attaches a `DeletionDate` stamp to the secret that specifies the end of the recovery window. At the end of the recovery window, Secrets Manager deletes the secret permanently.

At any time before recovery window ends, you can use RestoreSecret (p. 61) to remove the `DeletionDate` and cancel the deletion of the secret.

You cannot access the encrypted secret information in any secret that is scheduled for deletion. If you need to access that information, you must cancel the deletion with RestoreSecret (p. 61) and then retrieve the information.

Note

- There is no explicit operation to delete a version of a secret. Instead, remove all staging labels from the `VersionStage` field of a version. That marks the version as deprecated and allows Secrets Manager to delete it as needed. Versions that do not have any staging labels do not show up in ListSecretVersionIds (p. 45) unless you specify `IncludeDeprecated`.
- The permanent secret deletion at the end of the waiting period is performed as a background task with low priority. There is no guarantee of a specific time after the recovery window for the actual delete operation to occur.

Minimum permissions

To run this command, you must have the following permissions:

- secretsmanager:DeleteSecret

Related operations

- To create a secret, use CreateSecret (p. 8).
- To cancel deletion of a version of a secret before the recovery window has expired, use RestoreSecret (p. 61).

Request Syntax

```
{
   "ForceDeleteWithoutRecovery": boolean,
   "RecoveryWindowInDays": number,
   "SecretId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 98).

The request accepts the following data in JSON format.

ForceDeleteWithoutRecovery (p. 19)

(Optional) Specifies that the secret is to be deleted without any recovery window. You can't use both this parameter and the `RecoveryWindowInDays` parameter in the same API call.

An asynchronous background process performs the actual deletion, so there can be a short delay before the operation completes. If you write code to delete and then immediately recreate a secret with the same name, ensure that your code includes appropriate back off and retry logic.

Important
Use this parameter with caution. This parameter causes the operation to skip the normal waiting period before the permanent deletion that AWS would normally impose with the `RecoveryWindowInDays` parameter. If you delete a secret with the `ForceDeleteWithouRecovery` parameter, then you have no opportunity to recover the secret. It is permanently lost.

Type: Boolean

Required: No

RecoveryWindowInDays (p. 19)

(Optional) Specifies the number of days that Secrets Manager waits before it can delete the secret. You can't use both this parameter and the `ForceDeleteWithoutRecovery` parameter in the same API call.

This value can range from 7 to 30 days. The default value is 30.

Type: Long

Required: No

SecretId (p. 19)

Specifies the secret that you want to delete. You can specify either the Amazon Resource Name (ARN) or the friendly name of the secret.

Note
If you specify an ARN, we generally recommend that you specify a complete ARN. You can specify a partial ARN too—for example, if you don't include the final hyphen and six random characters that Secrets Manager adds at the end of the ARN when you created the secret. A partial ARN match can work as long as it uniquely matches only one secret. However, if your secret has a name that ends in a hyphen followed by six characters (before Secrets Manager adds the hyphen and six characters to the ARN) and you try to use that as a partial ARN, then those characters cause Secrets Manager to assume that you're specifying a complete ARN. This confusion can cause unexpected results. To avoid this situation, we recommend that you don't create secret names that end with a hyphen followed by six characters.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

Response Syntax

```
{
   "ARN": "string",
   "DeletionDate": number,
   "Name": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ARN (p. 20)

The ARN of the secret that is now scheduled for deletion.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

DeletionDate (p. 20)

The date and time after which this secret can be deleted by Secrets Manager and can no longer be restored. This value is the date and time of the delete request plus the number of days specified in `RecoveryWindowInDays`.

Type: Timestamp

Name (p. 20)

The friendly name of the secret that is now scheduled for deletion.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 100).

InternalServiceError

An error occurred on the server side.

HTTP Status Code: 500

InvalidParameterException

You provided an invalid value for a parameter.

HTTP Status Code: 400

InvalidRequestException

You provided a parameter value that is not valid for the current state of the resource.

Possible causes:

- You tried to perform the operation on a secret that's currently marked deleted.
- You tried to enable rotation on a secret that doesn't already have a Lambda function ARN configured and you didn't include such an ARN as a parameter in this call.

HTTP Status Code: 400

ResourceNotFoundException

We can't find the resource that you asked for.

HTTP Status Code: 400

Example

The following example shows how to delete a secret. The secret stays in your account in a deprecated and inaccessible state until the recovery window ends. After the date and time in the `DeletionDate`

response field has passed, you can no longer recover this secret with RestoreSecret (p. 61). The JSON request string input and response output are shown formatted with white space and line breaks for better readability. Submit your input as a single line JSON string.

Sample Request

```
POST / HTTP/1.1
Host: secretsmanager.region.domain
Accept-Encoding: identity
X-Amz-Target: secretsmanager.DeleteSecret
Content-Type: application/x-amz-json-1.1
User-Agent: <user-agent-string>
X-Amz-Date: <date>
Authorization: AWS4-HMAC-SHA256 Credential=<credentials>,SignedHeaders=<headers>,
 Signature=<signature>
Content-Length: <payload-size-bytes>

{
  "SecretId": "MyTestDatabaseSecret",
  "RecoveryWindowInDays": 7
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: <date>
Content-Type: application/x-amz-json-1.1
Content-Length: <response-size-bytes>
Connection: keep-alive
x-amzn-RequestId: <request-id-guid>

{
  "ARN":"arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
  "DeletionDate":1.524085349095E9,
  "Name":"MyTestDatabaseSecret"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DescribeSecret

Retrieves the details of a secret. It does not include the encrypted fields. Only those fields that are populated with a value are returned in the response.

Minimum permissions

To run this command, you must have the following permissions:

- secretsmanager:DescribeSecret

Related operations

- To create a secret, use CreateSecret (p. 8).
- To modify a secret, use UpdateSecret (p. 77).
- To retrieve the encrypted secret information in a version of the secret, use GetSecretValue (p. 36).
- To list all of the secrets in the AWS account, use ListSecrets (p. 41).

Request Syntax

```
{
   "SecretId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 98).

The request accepts the following data in JSON format.

SecretId (p. 23)

The identifier of the secret whose details you want to retrieve. You can specify either the Amazon Resource Name (ARN) or the friendly name of the secret.

> **Note**
> If you specify an ARN, we generally recommend that you specify a complete ARN. You can specify a partial ARN too—for example, if you don't include the final hyphen and six random characters that Secrets Manager adds at the end of the ARN when you created the secret. A partial ARN match can work as long as it uniquely matches only one secret. However, if your secret has a name that ends in a hyphen followed by six characters (before Secrets Manager adds the hyphen and six characters to the ARN) and you try to use that as a partial ARN, then those characters cause Secrets Manager to assume that you're specifying a complete ARN. This confusion can cause unexpected results. To avoid this situation, we recommend that you don't create secret names that end with a hyphen followed by six characters.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

Response Syntax

```
{
   "ARN": "string",
   "DeletedDate": number,
   "Description": "string",
   "KmsKeyId": "string",
   "LastAccessedDate": number,
   "LastChangedDate": number,
   "LastRotatedDate": number,
   "Name": "string",
   "RotationEnabled": boolean,
   "RotationLambdaARN": "string",
   "RotationRules": {
      "AutomaticallyAfterDays": number
   },
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ],
   "VersionIdsToStages": {
      "string" : [ "string" ]
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ARN (p. 24)

The ARN of the secret.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

DeletedDate (p. 24)

This value exists if the secret is scheduled for deletion. Some time after the specified date and time, Secrets Manager deletes the secret and all of its versions.

If a secret is scheduled for deletion, then its details, including the encrypted secret information, is not accessible. To cancel a scheduled deletion and restore access, use RestoreSecret (p. 61).

Type: Timestamp

Description (p. 24)

The user-provided description of the secret.

Type: String

Length Constraints: Maximum length of 2048.

KmsKeyId (p. 24)

The ARN or alias of the AWS KMS customer master key (CMK) that's used to encrypt the `SecretString` or `SecretBinary` fields in each version of the secret. If you don't provide a key,

then Secrets Manager defaults to encrypting the secret fields with the default AWS KMS CMK (the one named `awssecretsmanager`) for this account.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 2048.

LastAccessedDate (p. 24)

The last date that this secret was accessed. This value is truncated to midnight of the date and therefore shows only the date, not the time.

Type: Timestamp

LastChangedDate (p. 24)

The last date and time that this secret was modified in any way.

Type: Timestamp

LastRotatedDate (p. 24)

The most recent date and time that the Secrets Manager rotation process was successfully completed. This value is null if the secret has never rotated.

Type: Timestamp

Name (p. 24)

The user-provided friendly name of the secret.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

RotationEnabled (p. 24)

Specifies whether automatic rotation is enabled for this secret.

To enable rotation, use RotateSecret (p. 64) with `AutomaticallyRotateAfterDays` set to a value greater than 0. To disable rotation, use CancelRotateSecret (p. 4).

Type: Boolean

RotationLambdaARN (p. 24)

The ARN of a Lambda function that's invoked by Secrets Manager to rotate the secret either automatically per the schedule or manually by a call to `RotateSecret`.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 2048.

RotationRules (p. 24)

A structure that contains the rotation configuration for this secret.

Type: RotationRulesType (p. 92) object

Tags (p. 24)

The list of user-defined tags that are associated with the secret. To add tags to a secret, use TagResource (p. 70). To remove tags, use UntagResource (p. 74).

Type: Array of Tag (p. 97) objects

VersionIdsToStages (p. 24)

A list of all of the currently assigned `VersionStage` staging labels and the `VersionId` that each is attached to. Staging labels are used to keep track of the different versions during the rotation process.

Note
A version that does not have any staging labels attached is considered deprecated and subject to deletion. Such versions are not included in this list.

Type: String to array of strings map

Key Length Constraints: Minimum length of 32. Maximum length of 64.

Array Members: Minimum number of 1 item. Maximum number of 20 items.

Length Constraints: Minimum length of 1. Maximum length of 256.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 100).

InternalServiceError

An error occurred on the server side.

HTTP Status Code: 500

ResourceNotFoundException

We can't find the resource that you asked for.

HTTP Status Code: 400

Example

The following example shows how to get the details about a secret. The JSON request string input and response output are shown formatted with white space and line breaks for better readability. Submit your input as a single line JSON string.

Sample Request

```
POST / HTTP/1.1
Host: secretsmanager.region.domain
Accept-Encoding: identity
X-Amz-Target: secretsmanager.DescribeSecret
Content-Type: application/x-amz-json-1.1
User-Agent: <user-agent-string>
X-Amz-Date: <date>
Authorization: AWS4-HMAC-SHA256 Credential=<credentials>,SignedHeaders=<headers>,
 Signature=<signature>
Content-Length: <payload-size-bytes>

{
  "SecretId": "MyTestDatabaseSecret"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: <date>
Content-Type: application/x-amz-json-1.1
Content-Length: <response-size-bytes>
Connection: keep-alive
x-amzn-RequestId: <request-id-guid>

{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-
a1b2c3",
  "Name": "MyTestDatabaseSecret",
  "Description": "My test database secret created with the CLI",
  "LastChangedDate": 1523477145.729,
  "RotationEnabled": true,
  "RotationLambdaARN": "arn:aws:lambda:us-
west-2:123456789012:function:MyTestRotationLambda",
  "RotationRules": {
    "AutomaticallyAfterDays": 30
  },
  "LastRotatedDate": 1525747253.72
  "Tags": [
    {
      "Key": "SecondTag",
      "Value": "AnotherValue"
    },
    {
      "Key": "FirstTag",
      "Value": "SomeValue"
    }
  ],
  "VersionIdsToStages": {
    "EXAMPLE1-90ab-cdef-fedc-ba987SECRET1": [
      "AWSPREVIOUS"
    ],
    "EXAMPLE2-90ab-cdef-fedc-ba987SECRET2": [
      "AWSCURRENT"
    ]
  }
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

GetRandomPassword

Generates a random password of the specified complexity. This operation is intended for use in the Lambda rotation function. Per best practice, we recommend that you specify the maximum length and include every character type that the system you are generating a password for can support.

Minimum permissions

To run this command, you must have the following permissions:

- secretsmanager:GetRandomPassword

Request Syntax

```
{
   "ExcludeCharacters": "string",
   "ExcludeLowercase": boolean,
   "ExcludeNumbers": boolean,
   "ExcludePunctuation": boolean,
   "ExcludeUppercase": boolean,
   "IncludeSpace": boolean,
   "PasswordLength": number,
   "RequireEachIncludedType": boolean
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 98).

The request accepts the following data in JSON format.

ExcludeCharacters (p. 28)

A string that includes characters that should not be included in the generated password. The default is that all characters from the included sets can be used.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 4096.

Required: No

ExcludeLowercase (p. 28)

Specifies that the generated password should not include lowercase letters. The default if you do not include this switch parameter is that lowercase letters can be included.

Type: Boolean

Required: No

ExcludeNumbers (p. 28)

Specifies that the generated password should not include digits. The default if you do not include this switch parameter is that digits can be included.

Type: Boolean

Required: No

ExcludePunctuation (p. 28)

Specifies that the generated password should not include punctuation characters. The default if you do not include this switch parameter is that punctuation characters can be included.

The following are the punctuation characters that *can* be included in the generated password if you don't explicitly exclude them with `ExcludeCharacters` or `ExcludePunctuation`:

```
! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [ \ ] ^ _ ` { | } ~
```

Type: Boolean

Required: No

ExcludeUppercase (p. 28)

Specifies that the generated password should not include uppercase letters. The default if you do not include this switch parameter is that uppercase letters can be included.

Type: Boolean

Required: No

IncludeSpace (p. 28)

Specifies that the generated password can include the space character. The default if you do not include this switch parameter is that the space character is not included.

Type: Boolean

Required: No

PasswordLength (p. 28)

The desired length of the generated password. The default value if you do not include this parameter is 32 characters.

Type: Long

Valid Range: Minimum value of 1. Maximum value of 4096.

Required: No

RequireEachIncludedType (p. 28)

A boolean value that specifies whether the generated password must include at least one of every allowed character type. The default value is `True` and the operation requires at least one of every character type.

Type: Boolean

Required: No

Response Syntax

```
{
   "RandomPassword": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

RandomPassword (p. 29)

A string with the generated password.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 4096.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 100).

InternalServiceError

An error occurred on the server side.

HTTP Status Code: 500

InvalidParameterException

You provided an invalid value for a parameter.

HTTP Status Code: 400

InvalidRequestException

You provided a parameter value that is not valid for the current state of the resource.

Possible causes:

- You tried to perform the operation on a secret that's currently marked deleted.
- You tried to enable rotation on a secret that doesn't already have a Lambda function ARN configured and you didn't include such an ARN as a parameter in this call.

HTTP Status Code: 400

Example

The following example shows how to request a randomly generated password. This example includes the optional flags to require spaces and at least one character of each included type. It specifies a length of 20 characters.

Sample Request

```
POST / HTTP/1.1
Host: secretsmanager.region.domain
Accept-Encoding: identity
X-Amz-Target: secretsmanager.GetRandomPassword
Content-Type: application/x-amz-json-1.1
User-Agent: <user-agent-string>
X-Amz-Date: <date>
```

```
Authorization: AWS4-HMAC-SHA256 Credential=<credentials>,SignedHeaders=<headers>,
 Signature=<signature>
Content-Length: <payload-size-bytes>

{
  "PasswordLength": 20,
  "IncludeSpace": true,
  "RequireEachIncludedType": true
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: <date>
Content-Type: application/x-amz-json-1.1
Content-Length: <response-size-bytes>
Connection: keep-alive
x-amzn-RequestId: <request-id-guid>

{
  "RandomPassword":"N+Z43a,>vx7j O8^*<8i3"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

GetResourcePolicy

Retrieves the JSON text of the resource-based policy document that's attached to the specified secret. The JSON request string input and response output are shown formatted with white space and line breaks for better readability. Submit your input as a single line JSON string.

Minimum permissions

To run this command, you must have the following permissions:

- secretsmanager:GetResourcePolicy

Related operations

- To attach a resource policy to a secret, use PutResourcePolicy (p. 50).
- To delete the resource-based policy that's attached to a secret, use DeleteResourcePolicy (p. 16).
- To list all of the currently available secrets, use ListSecrets (p. 41).

Request Syntax

```
{
   "SecretId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 98).

The request accepts the following data in JSON format.

SecretId (p. 32)

Specifies the secret that you want to retrieve the attached resource-based policy for. You can specify either the Amazon Resource Name (ARN) or the friendly name of the secret.

Note
If you specify an ARN, we generally recommend that you specify a complete ARN. You can specify a partial ARN too—for example, if you don't include the final hyphen and six random characters that Secrets Manager adds at the end of the ARN when you created the secret. A partial ARN match can work as long as it uniquely matches only one secret. However, if your secret has a name that ends in a hyphen followed by six characters (before Secrets Manager adds the hyphen and six characters to the ARN) and you try to use that as a partial ARN, then those characters cause Secrets Manager to assume that you're specifying a complete ARN. This confusion can cause unexpected results. To avoid this situation, we recommend that you don't create secret names that end with a hyphen followed by six characters.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

Response Syntax

```
{
   "ARN": "string",
   "Name": "string",
   "ResourcePolicy": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ARN (p. 33)

The ARN of the secret that the resource-based policy was retrieved for.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Name (p. 33)

The friendly name of the secret that the resource-based policy was retrieved for.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 512.

ResourcePolicy (p. 33)

A JSON-formatted string that describes the permissions that are associated with the attached secret. These permissions are combined with any permissions that are associated with the user or role that attempts to access this secret. The combined permissions specify who can access the secret and what actions they can perform. For more information, see Authentication and Access Control for AWS Secrets Manager in the *AWS Secrets Manager User Guide*.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 4096.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 100).

InternalServiceError

An error occurred on the server side.

HTTP Status Code: 500

InvalidRequestException

You provided a parameter value that is not valid for the current state of the resource.

Possible causes:

- You tried to perform the operation on a secret that's currently marked deleted.

- You tried to enable rotation on a secret that doesn't already have a Lambda function ARN configured and you didn't include such an ARN as a parameter in this call.

HTTP Status Code: 400

ResourceNotFoundException

We can't find the resource that you asked for.

HTTP Status Code: 400

Example

The following example shows how to retrieve the resource-based policy that's attached to the specified secret. The JSON request string input and response output are shown formatted with white space and line breaks for better readability. Submit your input as a single line JSON string.

Sample Request

```
POST / HTTP/1.1
Host: secretsmanager.region.domain
Accept-Encoding: identity
X-Amz-Target: secretsmanager.GetResourcePolicy
Content-Type: application/x-amz-json-1.1
User-Agent: <user-agent-string>
X-Amz-Date: <date>
Authorization: AWS4-HMAC-SHA256 Credential=<credentials>,SignedHeaders=<headers>,
 Signature=<signature>
Content-Length: <payload-size-bytes>

{
  "SecretId": "MyTestDatabaseSecret"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: <date>
Content-Type: application/x-amz-json-1.1
Content-Length: <response-size-bytes>
Connection: keep-alive
x-amzn-RequestId: <request-id-guid>

{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-
a1b2c3",
  "Name": "MyTestDatabaseSecret",
  "ResourcePolicy": "{\"Version\":\"2012-10-17\",\"Statement\":{\"Effect\":\"Allow\",
\"Principal\":{\"AWS\":[\"arn:aws:iam::111122223333:root\",\"arn:aws:iam::444455556666:root
\"]},\"Action\":[\"secretsmanager:GetSecretValue\"],\"Resource\":\"*\"}}"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

GetSecretValue

Retrieves the contents of the encrypted fields `SecretString` or `SecretBinary` from the specified version of a secret, whichever contains content.

Minimum permissions

To run this command, you must have the following permissions:

- secretsmanager:GetSecretValue
- kms:Decrypt - required only if you use a customer-managed AWS KMS key to encrypt the secret. You do not need this permission to use the account's default AWS managed CMK for Secrets Manager.

Related operations

- To create a new version of the secret with different encrypted information, use PutSecretValue (p. 54).
- To retrieve the non-encrypted details for the secret, use DescribeSecret (p. 23).

Request Syntax

```
{
   "SecretId": "string",
   "VersionId": "string",
   "VersionStage": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 98).

The request accepts the following data in JSON format.

SecretId (p. 36)

Specifies the secret containing the version that you want to retrieve. You can specify either the Amazon Resource Name (ARN) or the friendly name of the secret.

> **Note**
> If you specify an ARN, we generally recommend that you specify a complete ARN. You can specify a partial ARN too—for example, if you don't include the final hyphen and six random characters that Secrets Manager adds at the end of the ARN when you created the secret. A partial ARN match can work as long as it uniquely matches only one secret. However, if your secret has a name that ends in a hyphen followed by six characters (before Secrets Manager adds the hyphen and six characters to the ARN) and you try to use that as a partial ARN, then those characters cause Secrets Manager to assume that you're specifying a complete ARN. This confusion can cause unexpected results. To avoid this situation, we recommend that you don't create secret names that end with a hyphen followed by six characters.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

VersionId (p. 36)

Specifies the unique identifier of the version of the secret that you want to retrieve. If you specify this parameter then don't specify `VersionStage`. If you don't specify either a `VersionStage` or `VersionId` then the default is to perform the operation on the version with the `VersionStage` value of `AWSCURRENT`.

This value is typically a UUID-type value with 32 hexadecimal digits.

Type: String

Length Constraints: Minimum length of 32. Maximum length of 64.

Required: No

VersionStage (p. 36)

Specifies the secret version that you want to retrieve by the staging label attached to the version.

Staging labels are used to keep track of different versions during the rotation process. If you use this parameter then don't specify `VersionId`. If you don't specify either a `VersionStage` or `VersionId`, then the default is to perform the operation on the version with the `VersionStage` value of `AWSCURRENT`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Required: No

Response Syntax

```
{
   "ARN": "string",
   "CreatedDate": number,
   "Name": "string",
   "SecretBinary": blob,
   "SecretString": "string",
   "VersionId": "string",
   "VersionStages": [ "string" ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ARN (p. 37)

The ARN of the secret.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

CreatedDate (p. 37)

The date and time that this version of the secret was created.

Type: Timestamp

Name (p. 37)

The friendly name of the secret.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

SecretBinary (p. 37)

The decrypted part of the protected secret information that was originally provided as binary data in the form of a byte array. The response parameter represents the binary data as a base64-encoded string.

This parameter is not used if the secret is created by the Secrets Manager console.

If you store custom information in this field of the secret, then you must code your Lambda rotation function to parse and interpret whatever you store in the `SecretString` or `SecretBinary` fields.

Type: Base64-encoded binary data object

Length Constraints: Minimum length of 0. Maximum length of 7168.

SecretString (p. 37)

The decrypted part of the protected secret information that was originally provided as a string.

If you create this secret by using the Secrets Manager console then only the `SecretString` parameter contains data. Secrets Manager stores the information as a JSON structure of key/value pairs that the Lambda rotation function knows how to parse.

If you store custom information in the secret by using the CreateSecret (p. 8), UpdateSecret (p. 77), or PutSecretValue (p. 54) API operations instead of the Secrets Manager console, or by using the **Other secret type** in the console, then you must code your Lambda rotation function to parse and interpret those values.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 7168.

VersionId (p. 37)

The unique identifier of this version of the secret.

Type: String

Length Constraints: Minimum length of 32. Maximum length of 64.

VersionStages (p. 37)

A list of all of the staging labels currently attached to this version of the secret.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 20 items.

Length Constraints: Minimum length of 1. Maximum length of 256.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 100).

DecryptionFailure

Secrets Manager can't decrypt the protected secret text using the provided KMS key.

HTTP Status Code: 400

InternalServiceError

An error occurred on the server side.

HTTP Status Code: 500

InvalidParameterException

You provided an invalid value for a parameter.

HTTP Status Code: 400

InvalidRequestException

You provided a parameter value that is not valid for the current state of the resource.

Possible causes:

- You tried to perform the operation on a secret that's currently marked deleted.
- You tried to enable rotation on a secret that doesn't already have a Lambda function ARN configured and you didn't include such an ARN as a parameter in this call.

HTTP Status Code: 400

ResourceNotFoundException

We can't find the resource that you asked for.

HTTP Status Code: 400

Example

The following example shows how to retrieve the secret string value from the version of the secret that has the `AWSPREVIOUS` staging label attached. If you want to retrieve the `AWSCURRENT` version of the secret, then omit the `VersionStage` parameter because it defaults to `AWSCURRENT`. The JSON request string input and response output are shown formatted with white space and line breaks for better readability. Submit your input as a single line JSON string.

Sample Request

```
POST / HTTP/1.1
Host: secretsmanager.region.domain
Accept-Encoding: identity
X-Amz-Target: secretsmanager.GetSecretValue
Content-Type: application/x-amz-json-1.1
User-Agent: <user-agent-string>
X-Amz-Date: <date>
Authorization: AWS4-HMAC-SHA256 Credential=<credentials>,SignedHeaders=<headers>,
 Signature=<signature>
Content-Length: <payload-size-bytes>

{
  "SecretId": "MyTestDatabaseSecret",
  "VersionStage": "AWSPREVIOUS"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: <date>
Content-Type: application/x-amz-json-1.1
Content-Length: <response-size-bytes>
Connection: keep-alive
x-amzn-RequestId: <request-id-guid>

{
  "ARN":"arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
  "CreatedDate":1.523477145713E9,
  "Name":"MyTestDatabaseSecret",
  "SecretString":"{\n  \"username\":\"david\",\n  \"password\":
\"BnQw&XDWgaEeT9XGTT29\"\n}\n",
  "VersionId":"EXAMPLE1-90ab-cdef-fedc-ba987SECRET1",
  "VersionStages":["AWSPREVIOUS"]
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

ListSecrets

Lists all of the secrets that are stored by Secrets Manager in the AWS account. To list the versions currently stored for a specific secret, use ListSecretVersionIds (p. 45). The encrypted fields `SecretString` and `SecretBinary` are not included in the output. To get that information, call the GetSecretValue (p. 36) operation.

> **Note**
> Always check the `NextToken` response parameter when calling any of the `List*` operations. These operations can occasionally return an empty or shorter than expected list of results even when there are more results available. When this happens, the `NextToken` response parameter contains a value to pass to the next call to the same API to request the next part of the list.

Minimum permissions

To run this command, you must have the following permissions:

- secretsmanager:ListSecrets

Related operations

- To list the versions attached to a secret, use ListSecretVersionIds (p. 45).

Request Syntax

```
{
   "MaxResults": number,
   "NextToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 98).

The request accepts the following data in JSON format.

MaxResults (p. 41)

(Optional) Limits the number of results that you want to include in the response. If you don't include this parameter, it defaults to a value that's specific to the operation. If additional items exist beyond the maximum you specify, the `NextToken` response element is present and has a value (isn't null). Include that value as the `NextToken` request parameter in the next call to the operation to get the next part of the results. Note that Secrets Manager might return fewer results than the maximum even when there are more results available. You should check `NextToken` after every operation to ensure that you receive all of the results.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

NextToken (p. 41)

(Optional) Use this parameter in a request if you receive a `NextToken` response in a previous request that indicates that there's more output available. In a subsequent call, set it to the value of the previous call's `NextToken` response to indicate where the output should continue from.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 4096.

Required: No

Response Syntax

```
{
   "NextToken": "string",
   "SecretList": [
      {
         "ARN": "string",
         "DeletedDate": number,
         "Description": "string",
         "KmsKeyId": "string",
         "LastAccessedDate": number,
         "LastChangedDate": number,
         "LastRotatedDate": number,
         "Name": "string",
         "RotationEnabled": boolean,
         "RotationLambdaARN": "string",
         "RotationRules": {
            "AutomaticallyAfterDays": number
         },
         "SecretVersionsToStages": {
            "string" : [ "string" ]
         },
         "Tags": [
            {
               "Key": "string",
               "Value": "string"
            }
         ]
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken (p. 42)

If present in the response, this value indicates that there's more output available than what's included in the current response. This can occur even when the response includes no values at all, such as when you ask for a filtered view of a very long list. Use this value in the `NextToken` request parameter in a subsequent call to the operation to continue processing and get the next part of the output. You should repeat this until the `NextToken` response element comes back empty (as `null`).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 4096.

SecretList (p. 42)

A list of the secrets in the account.

Type: Array of SecretListEntry (p. 93) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 100).

InternalServiceError

An error occurred on the server side.

HTTP Status Code: 500

InvalidNextTokenException

You provided an invalid `NextToken` value.

HTTP Status Code: 400

InvalidParameterException

You provided an invalid value for a parameter.

HTTP Status Code: 400

Example

The following example shows how to list all of the secrets in the account. The JSON request string input and response output are shown formatted with white space and line breaks for better readability. Submit your input as a single line JSON string.

Sample Request

```
POST / HTTP/1.1
Host: secretsmanager.region.domain
Accept-Encoding: identity
X-Amz-Target: secretsmanager.ListSecrets
Content-Type: application/x-amz-json-1.1
User-Agent: <user-agent-string>
X-Amz-Date: <date>
Authorization: AWS4-HMAC-SHA256 Credential=<credentials>,SignedHeaders=<headers>,
 Signature=<signature>
Content-Length: <payload-size-bytes>

{}
```

Sample Response

```
HTTP/1.1 200 OK
Date: <date>
Content-Type: application/x-amz-json-1.1
Content-Length: <response-size-bytes>
Connection: keep-alive
x-amzn-RequestId: <request-id-guid>

{
  "SecretList":[
    {
      "ARN":"arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-
a1b2c3",
      "Description":"My test database secret",
      "LastChangedDate":1.523477145729E9,
```

```
      "Name":"MyTestDatabaseSecret",
      "SecretVersionsToStages":{
        "EXAMPLE2-90ab-cdef-fedc-ba987EXAMPLE":["AWSCURRENT"]
      }
    },
    {
      "ARN":"arn:aws:secretsmanager:us-west-2:123456789012:secret:AnotherDatabaseSecret-
d4e5f6",
      "Description":"Another secret created for a different database",
      "LastChangedDate":1.523482025685E9,
      "Name":"AnotherDatabaseSecret",
      "SecretVersionsToStages":{
        "EXAMPLE3-90ab-cdef-fedc-ba987EXAMPLE":["AWSCURRENT"]
      }
    }
  ]
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

ListSecretVersionIds

Lists all of the versions attached to the specified secret. The output does not include the `SecretString` or `SecretBinary` fields. By default, the list includes only versions that have at least one staging label in `VersionStage` attached.

> **Note**
> Always check the `NextToken` response parameter when calling any of the `List*` operations. These operations can occasionally return an empty or shorter than expected list of results even when there are more results available. When this happens, the `NextToken` response parameter contains a value to pass to the next call to the same API to request the next part of the list.

Minimum permissions

To run this command, you must have the following permissions:

- secretsmanager:ListSecretVersionIds

Related operations

- To list the secrets in an account, use ListSecrets (p. 41).

Request Syntax

```
{
   "IncludeDeprecated": boolean,
   "MaxResults": number,
   "NextToken": "string",
   "SecretId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 98).

The request accepts the following data in JSON format.

IncludeDeprecated (p. 45)

(Optional) Specifies that you want the results to include versions that do not have any staging labels attached to them. Such versions are considered deprecated and are subject to deletion by Secrets Manager as needed.

Type: Boolean

Required: No

MaxResults (p. 45)

(Optional) Limits the number of results that you want to include in the response. If you don't include this parameter, it defaults to a value that's specific to the operation. If additional items exist beyond the maximum you specify, the `NextToken` response element is present and has a value (isn't null). Include that value as the `NextToken` request parameter in the next call to the operation to get the next part of the results. Note that Secrets Manager might return fewer results than the maximum even when there are more results available. You should check `NextToken` after every operation to ensure that you receive all of the results.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

NextToken (p. 45)

(Optional) Use this parameter in a request if you receive a `NextToken` response in a previous request that indicates that there's more output available. In a subsequent call, set it to the value of the previous call's `NextToken` response to indicate where the output should continue from.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 4096.

Required: No

SecretId (p. 45)

The identifier for the secret containing the versions you want to list. You can specify either the Amazon Resource Name (ARN) or the friendly name of the secret.

> **Note**
> If you specify an ARN, we generally recommend that you specify a complete ARN. You can specify a partial ARN too—for example, if you don't include the final hyphen and six random characters that Secrets Manager adds at the end of the ARN when you created the secret. A partial ARN match can work as long as it uniquely matches only one secret. However, if your secret has a name that ends in a hyphen followed by six characters (before Secrets Manager adds the hyphen and six characters to the ARN) and you try to use that as a partial ARN, then those characters cause Secrets Manager to assume that you're specifying a complete ARN. This confusion can cause unexpected results. To avoid this situation, we recommend that you don't create secret names that end with a hyphen followed by six characters.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

Response Syntax

```
{
   "ARN": "string",
   "Name": "string",
   "NextToken": "string",
   "Versions": [
      {
         "CreatedDate": number,
         "LastAccessedDate": number,
         "VersionId": "string",
         "VersionStages": [ "string" ]
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ARN (p. 46)

The Amazon Resource Name (ARN) for the secret.

> **Note**
> Secrets Manager automatically adds several random characters to the name at the end of the ARN when you initially create a secret. This affects only the ARN and not the actual friendly name. This ensures that if you create a new secret with the same name as an old secret that you previously deleted, then users with access to the old secret *don't* automatically get access to the new secret because the ARNs are different.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Name (p. 46)

The friendly name of the secret.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

NextToken (p. 46)

If present in the response, this value indicates that there's more output available than what's included in the current response. This can occur even when the response includes no values at all, such as when you ask for a filtered view of a very long list. Use this value in the `NextToken` request parameter in a subsequent call to the operation to continue processing and get the next part of the output. You should repeat this until the `NextToken` response element comes back empty (as `null`).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 4096.

Versions (p. 46)

The list of the currently available versions of the specified secret.

Type: Array of SecretVersionsListEntry (p. 96) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 100).

InternalServiceError

An error occurred on the server side.

HTTP Status Code: 500

InvalidNextTokenException

You provided an invalid `NextToken` value.

HTTP Status Code: 400

ResourceNotFoundException

We can't find the resource that you asked for.

HTTP Status Code: 400

Example

The following example shows how to retrieve a list of all of the versions of a secret, including those without any staging labels. The JSON request string input and response output are shown formatted with white space and line breaks for better readability. Submit your input as a single line JSON string.

Sample Request

```
POST / HTTP/1.1
Host: secretsmanager.region.domain
Accept-Encoding: identity
X-Amz-Target: secretsmanager.ListSecretVersionIds
Content-Type: application/x-amz-json-1.1
User-Agent: <user-agent-string>
X-Amz-Date: <date>
Authorization: AWS4-HMAC-SHA256 Credential=<credentials>,SignedHeaders=<headers>,
 Signature=<signature>
Content-Length: <payload-size-bytes>

{
  "SecretId": "MyTestDatabaseSecret",
  "IncludeDeprecated": true
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: <date>
Content-Type: application/x-amz-json-1.1
Content-Length: <response-size-bytes>
Connection: keep-alive
x-amzn-RequestId: <request-id-guid>

{
  "ARN":"arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
  "Name":"MyTestDatabaseSecret",
  "Versions":[
    {
      "CreatedDate":1.523477145713E9,
      "VersionId":"EXAMPLE1-90ab-cdef-fedc-ba987SECRET1",
      "VersionStages":["AWSPREVIOUS"]
    },
    {
      "CreatedDate":1.523486221391E9,
      "VersionId":"EXAMPLE2-90ab-cdef-fedc-ba987SECRET2",
      "VersionStages":["AWSCURRENT"]
    },
    {
      "CreatedDate": 1.51197446236E9,
      "VersionId": "EXAMPLE3-90ab-cdef-fedc-ba987SECRET3"
    }
  ]
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

PutResourcePolicy

Attaches the contents of the specified resource-based permission policy to a secret. A resource-based policy is optional. Alternatively, you can use IAM identity-based policies that specify the secret's Amazon Resource Name (ARN) in the policy statement's `Resources` element. You can also use a combination of both identity-based and resource-based policies. The affected users and roles receive the permissions that are permitted by all of the relevant policies. For more information, see Using Resource-Based Policies for AWS Secrets Manager. For the complete description of the AWS policy syntax and grammar, see IAM JSON Policy Reference in the *IAM User Guide*.

Minimum permissions

To run this command, you must have the following permissions:

- secretsmanager:PutResourcePolicy

Related operations

- To retrieve the resource policy that's attached to a secret, use GetResourcePolicy (p. 32).
- To delete the resource-based policy that's attached to a secret, use DeleteResourcePolicy (p. 16).
- To list all of the currently available secrets, use ListSecrets (p. 41).

Request Syntax

```
{
   "ResourcePolicy": "string",
   "SecretId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 98).

The request accepts the following data in JSON format.

ResourcePolicy (p. 50)

A JSON-formatted string that's constructed according to the grammar and syntax for an AWS resource-based policy. The policy in the string identifies who can access or manage this secret and its versions. For information on how to format a JSON parameter for the various command line tool environments, see Using JSON for Parameters in the *AWS CLI User Guide*.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 4096.

Required: Yes

SecretId (p. 50)

Specifies the secret that you want to attach the resource-based policy to. You can specify either the ARN or the friendly name of the secret.

Note
If you specify an ARN, we generally recommend that you specify a complete ARN. You can specify a partial ARN too—for example, if you don't include the final hyphen and six

random characters that Secrets Manager adds at the end of the ARN when you created the secret. A partial ARN match can work as long as it uniquely matches only one secret. However, if your secret has a name that ends in a hyphen followed by six characters (before Secrets Manager adds the hyphen and six characters to the ARN) and you try to use that as a partial ARN, then those characters cause Secrets Manager to assume that you're specifying a complete ARN. This confusion can cause unexpected results. To avoid this situation, we recommend that you don't create secret names that end with a hyphen followed by six characters.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

Response Syntax

```
{
   "ARN": "string",
   "Name": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ARN (p. 51)

The ARN of the secret that the resource-based policy was retrieved for.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Name (p. 51)

The friendly name of the secret that the resource-based policy was retrieved for.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 512.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 100).

InternalServiceError

An error occurred on the server side.

HTTP Status Code: 500

InvalidParameterException

You provided an invalid value for a parameter.

HTTP Status Code: 400

InvalidRequestException

You provided a parameter value that is not valid for the current state of the resource.

Possible causes:

- You tried to perform the operation on a secret that's currently marked deleted.
- You tried to enable rotation on a secret that doesn't already have a Lambda function ARN configured and you didn't include such an ARN as a parameter in this call.

HTTP Status Code: 400

MalformedPolicyDocumentException

The policy document that you provided isn't valid.

HTTP Status Code: 400

ResourceNotFoundException

We can't find the resource that you asked for.

HTTP Status Code: 400

Example

The following example shows how to attach a resource-based policy to the specified secret. The JSON request string input and response output are shown formatted with white space and line breaks for better readability. Submit your input as a single line JSON string.

Sample Request

```
POST / HTTP/1.1
Host: secretsmanager.region.domain
Accept-Encoding: identity
X-Amz-Target: secretsmanager.GetResourcePolicy
Content-Type: application/x-amz-json-1.1
User-Agent: <user-agent-string>
X-Amz-Date: <date>
Authorization: AWS4-HMAC-SHA256 Credential=<credentials>,SignedHeaders=<headers>,
 Signature=<signature>
Content-Length: <payload-size-bytes>

{
  "SecretId": "MyTestDatabaseSecret",
  "ResourcePolicy": "{\"Version\":\"2012-10-17\",\"Statement\":{\"Effect\":\"Allow\",
\"Principal\":{\"AWS\":[\"arn:aws:iam::111122223333:root\",\"arn:aws:iam::444455556666:root
\"]},\"Action\":[\"secretsmanager:GetSecretValue\"],\"Resource\":\"*\"}}"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: <date>
Content-Type: application/x-amz-json-1.1
Content-Length: <response-size-bytes>
Connection: keep-alive
x-amzn-RequestId: <request-id-guid>
```

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-
a1b2c3",
  "Name": "MyTestDatabaseSecret"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

PutSecretValue

Stores a new encrypted secret value in the specified secret. To do this, the operation creates a new version and attaches it to the secret. The version can contain a new `SecretString` value or a new `SecretBinary` value. You can also specify the staging labels that are initially attached to the new version.

> **Note**
> The Secrets Manager console uses only the `SecretString` field. To add binary data to a secret with the `SecretBinary` field you must use the AWS CLI or one of the AWS SDKs.

- If this operation creates the first version for the secret then Secrets Manager automatically attaches the staging label `AWSCURRENT` to the new version.
- If another version of this secret already exists, then this operation does not automatically move any staging labels other than those that you explicitly specify in the `VersionStages` parameter.
- If this operation moves the staging label `AWSCURRENT` from another version to this version (because you included it in the `StagingLabels` parameter) then Secrets Manager also automatically moves the staging label `AWSPREVIOUS` to the version that `AWSCURRENT` was removed from.
- This operation is idempotent. If a version with a `VersionId` with the same value as the `ClientRequestToken` parameter already exists and you specify the same secret data, the operation succeeds but does nothing. However, if the secret data is different, then the operation fails because you cannot modify an existing version; you can only create new ones.

> **Note**
>
> - If you call an operation that needs to encrypt or decrypt the `SecretString` or `SecretBinary` for a secret in the same account as the calling user and that secret doesn't specify a AWS KMS encryption key, Secrets Manager uses the account's default AWS managed customer master key (CMK) with the alias `aws/secretsmanager`. If this key doesn't already exist in your account then Secrets Manager creates it for you automatically. All users and roles in the same AWS account automatically have access to use the default CMK. Note that if an Secrets Manager API call results in AWS having to create the account's AWS-managed CMK, it can result in a one-time significant delay in returning the result.
> - If the secret is in a different AWS account from the credentials calling an API that requires encryption or decryption of the secret value then you must create and use a custom AWS KMS CMK because you can't access the default CMK for the account using credentials from a different AWS account. Store the ARN of the CMK in the secret when you create the secret or when you update it by including it in the `KMSKeyId`. If you call an API that must encrypt or decrypt `SecretString` or `SecretBinary` using credentials from a different account then the AWS KMS key policy must grant cross-account access to that other account's user or role for both the kms:GenerateDataKey and kms:Decrypt operations.

Minimum permissions

To run this command, you must have the following permissions:

- secretsmanager:PutSecretValue
- kms:GenerateDataKey - needed only if you use a customer-managed AWS KMS key to encrypt the secret. You do not need this permission to use the account's default AWS managed CMK for Secrets Manager.

Related operations

- To retrieve the encrypted value you store in the version of a secret, use GetSecretValue (p. 36).

- To create a secret, use CreateSecret (p. 8).
- To get the details for a secret, use DescribeSecret (p. 23).
- To list the versions attached to a secret, use ListSecretVersionIds (p. 45).

Request Syntax

```
{
   "ClientRequestToken": "string",
   "SecretBinary": blob,
   "SecretId": "string",
   "SecretString": "string",
   "VersionStages": [ "string" ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 98).

The request accepts the following data in JSON format.

ClientRequestToken (p. 55)

(Optional) Specifies a unique identifier for the new version of the secret.

> **Note**
> If you use the AWS CLI or one of the AWS SDK to call this operation, then you can leave this parameter empty. The CLI or SDK generates a random UUID for you and includes that in the request. If you don't use the SDK and instead generate a raw HTTP request to the Secrets Manager service endpoint, then you must generate a `ClientRequestToken` yourself for new versions and include that value in the request.

This value helps ensure idempotency. Secrets Manager uses this value to prevent the accidental creation of duplicate versions if there are failures and retries during the Lambda rotation function's processing. We recommend that you generate a UUID-type value to ensure uniqueness within the specified secret.

- If the `ClientRequestToken` value isn't already associated with a version of the secret then a new version of the secret is created.
- If a version with this value already exists and that version's `SecretString` or `SecretBinary` values are the same as those in the request then the request is ignored (the operation is idempotent).
- If a version with this value already exists and that version's `SecretString` and `SecretBinary` values are different from those in the request then the request fails because you cannot modify an existing secret version. You can only create new versions to store new secret values.

This value becomes the `VersionId` of the new version.

Type: String

Length Constraints: Minimum length of 32. Maximum length of 64.

Required: No

SecretBinary (p. 55)

(Optional) Specifies binary data that you want to encrypt and store in the new version of the secret. To use this parameter in the command-line tools, we recommend that you store your binary data

in a file and then use the appropriate technique for your tool to pass the contents of the file as a parameter. Either `SecretBinary` or `SecretString` must have a value, but not both. They cannot both be empty.

This parameter is not accessible if the secret using the Secrets Manager console.

Type: Base64-encoded binary data object

Length Constraints: Minimum length of 0. Maximum length of 7168.

Required: No

SecretId (p. 55)

Specifies the secret to which you want to add a new version. You can specify either the Amazon Resource Name (ARN) or the friendly name of the secret. The secret must already exist.

> **Note**
> If you specify an ARN, we generally recommend that you specify a complete ARN. You can specify a partial ARN too—for example, if you don't include the final hyphen and six random characters that Secrets Manager adds at the end of the ARN when you created the secret. A partial ARN match can work as long as it uniquely matches only one secret. However, if your secret has a name that ends in a hyphen followed by six characters (before Secrets Manager adds the hyphen and six characters to the ARN) and you try to use that as a partial ARN, then those characters cause Secrets Manager to assume that you're specifying a complete ARN. This confusion can cause unexpected results. To avoid this situation, we recommend that you don't create secret names that end with a hyphen followed by six characters.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

SecretString (p. 55)

(Optional) Specifies text data that you want to encrypt and store in this new version of the secret. Either `SecretString` or `SecretBinary` must have a value, but not both. They cannot both be empty.

If you create this secret by using the Secrets Manager console then Secrets Manager puts the protected secret text in only the `SecretString` parameter. The Secrets Manager console stores the information as a JSON structure of key/value pairs that the default Lambda rotation function knows how to parse.

For storing multiple values, we recommend that you use a JSON text string argument and specify key/value pairs. For information on how to format a JSON parameter for the various command line tool environments, see Using JSON for Parameters in the *AWS CLI User Guide*.

For example:

```
[{"username":"bob"},{"password":"abc123xyz456"}]
```

If your command-line tool or SDK requires quotation marks around the parameter, you should use single quotes to avoid confusion with the double quotes required in the JSON text.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 7168.

Required: No

VersionStages (p. 55)

(Optional) Specifies a list of staging labels that are attached to this version of the secret. These staging labels are used to track the versions through the rotation process by the Lambda rotation function.

A staging label must be unique to a single version of the secret. If you specify a staging label that's already associated with a different version of the same secret then that staging label is automatically removed from the other version and attached to this version.

If you do not specify a value for `VersionStages` then Secrets Manager automatically moves the staging label `AWSCURRENT` to this new version.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 20 items.

Length Constraints: Minimum length of 1. Maximum length of 256.

Required: No

Response Syntax

```
{
   "ARN": "string",
   "Name": "string",
   "VersionId": "string",
   "VersionStages": [ "string" ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ARN (p. 57)

The Amazon Resource Name (ARN) for the secret for which you just created a version.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Name (p. 57)

The friendly name of the secret for which you just created or updated a version.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

VersionId (p. 57)

The unique identifier of the version of the secret you just created or updated.

Type: String

Length Constraints: Minimum length of 32. Maximum length of 64.

VersionStages (p. 57)

The list of staging labels that are currently attached to this version of the secret. Staging labels are used to track a version as it progresses through the secret rotation process.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 20 items.

Length Constraints: Minimum length of 1. Maximum length of 256.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 100).

EncryptionFailure

Secrets Manager can't encrypt the protected secret text using the provided KMS key. Check that the customer master key (CMK) is available, enabled, and not in an invalid state. For more information, see How Key State Affects Use of a Customer Master Key.

HTTP Status Code: 400

InternalServiceError

An error occurred on the server side.

HTTP Status Code: 500

InvalidParameterException

You provided an invalid value for a parameter.

HTTP Status Code: 400

InvalidRequestException

You provided a parameter value that is not valid for the current state of the resource.

Possible causes:

- You tried to perform the operation on a secret that's currently marked deleted.
- You tried to enable rotation on a secret that doesn't already have a Lambda function ARN configured and you didn't include such an ARN as a parameter in this call.

HTTP Status Code: 400

LimitExceededException

The request failed because it would exceed one of the Secrets Manager internal limits.

HTTP Status Code: 400

ResourceExistsException

A resource with the ID you requested already exists.

HTTP Status Code: 400

ResourceNotFoundException

We can't find the resource that you asked for.

HTTP Status Code: 400

Example

The following example shows how to create a new version of the secret. The `ClientRequestToken` becomes the `VersionId` of the new version. Alternatively, you can use UpdateSecret (p. 77). The JSON request string input and response output are shown formatted with white space and line breaks for better readability. Submit your input as a single line JSON string.

Sample Request

```
POST / HTTP/1.1
Host: secretsmanager.region.domain
Accept-Encoding: identity
X-Amz-Target: secretsmanager.PutSecretValue
Content-Type: application/x-amz-json-1.1
User-Agent: <user-agent-string>
X-Amz-Date: <date>
Authorization: AWS4-HMAC-SHA256 Credential=<credentials>,SignedHeaders=<headers>,
 Signature=<signature>
Content-Length: <payload-size-bytes>

{
  "SecretId": "MyTestDatabaseSecret",
  "SecretString": "{\"username\":\"david\",\"password\":\"BnQw!XDWgaEeT9XGTT29\"}",
  "ClientRequestToken": "EXAMPLE2-90ab-cdef-fedc-ba987EXAMPLE"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: <date>
Content-Type: application/x-amz-json-1.1
Content-Length: <response-size-bytes>
Connection: keep-alive
x-amzn-RequestId: <request-id-guid>

{
  "ARN":"arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
  "Name":"MyTestDatabaseSecret",
  "VersionId":"EXAMPLE2-90ab-cdef-fedc-ba987EXAMPLE",
  "VersionStages":[
      "AWSCURRENT"
  ]
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript

- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

RestoreSecret

Cancels the scheduled deletion of a secret by removing the `DeletedDate` time stamp. This makes the secret accessible to query once again.

Minimum permissions

To run this command, you must have the following permissions:

- secretsmanager:RestoreSecret

Related operations

- To delete a secret, use DeleteSecret (p. 19).

Request Syntax

```
{
   "SecretId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 98).

The request accepts the following data in JSON format.

SecretId (p. 61)

Specifies the secret that you want to restore from a previously scheduled deletion. You can specify either the Amazon Resource Name (ARN) or the friendly name of the secret.

Note
If you specify an ARN, we generally recommend that you specify a complete ARN. You can specify a partial ARN too—for example, if you don't include the final hyphen and six random characters that Secrets Manager adds at the end of the ARN when you created the secret. A partial ARN match can work as long as it uniquely matches only one secret. However, if your secret has a name that ends in a hyphen followed by six characters (before Secrets Manager adds the hyphen and six characters to the ARN) and you try to use that as a partial ARN, then those characters cause Secrets Manager to assume that you're specifying a complete ARN. This confusion can cause unexpected results. To avoid this situation, we recommend that you don't create secret names that end with a hyphen followed by six characters.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

Response Syntax

```
{
```

```
   "ARN": "string",
   "Name": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ARN (p. 61)

The ARN of the secret that was restored.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Name (p. 61)

The friendly name of the secret that was restored.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 100).

InternalServiceError

An error occurred on the server side.

HTTP Status Code: 500

InvalidParameterException

You provided an invalid value for a parameter.

HTTP Status Code: 400

InvalidRequestException

You provided a parameter value that is not valid for the current state of the resource.

Possible causes:

- You tried to perform the operation on a secret that's currently marked deleted.
- You tried to enable rotation on a secret that doesn't already have a Lambda function ARN configured and you didn't include such an ARN as a parameter in this call.

HTTP Status Code: 400

ResourceNotFoundException

We can't find the resource that you asked for.

HTTP Status Code: 400

Example

The following example shows how to restore a secret that was previously scheduled for deletion. The JSON request string input and response output are shown formatted with white space and line breaks for better readability. Submit your input as a single line JSON string.

Sample Request

```
POST / HTTP/1.1
Host: secretsmanager.region.domain
Accept-Encoding: identity
X-Amz-Target: secretsmanager.RestoreSecret
Content-Type: application/x-amz-json-1.1
User-Agent: <user-agent-string>
X-Amz-Date: <date>
Authorization: AWS4-HMAC-SHA256 Credential=<credentials>,SignedHeaders=<headers>,
 Signature=<signature>
Content-Length: <payload-size-bytes>

{
  "SecretId": "MyTestDatabaseSecret"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: <date>
Content-Type: application/x-amz-json-1.1
Content-Length: <response-size-bytes>
Connection: keep-alive
x-amzn-RequestId: <request-id-guid>

{
  "ARN":"arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
  "Name":"MyTestDatabaseSecret"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

RotateSecret

Configures and starts the asynchronous process of rotating this secret. If you include the configuration parameters, the operation sets those values for the secret and then immediately starts a rotation. If you do not include the configuration parameters, the operation starts a rotation with the values already stored in the secret. After the rotation completes, the protected service and its clients all use the new version of the secret.

This required configuration information includes the ARN of an AWS Lambda function and the time between scheduled rotations. The Lambda rotation function creates a new version of the secret and creates or updates the credentials on the protected service to match. After testing the new credentials, the function marks the new secret with the staging label `AWSCURRENT` so that your clients all immediately begin to use the new version. For more information about rotating secrets and how to configure a Lambda function to rotate the secrets for your protected service, see Rotating Secrets in AWS Secrets Manager in the *AWS Secrets Manager User Guide*.

Secrets Manager schedules the next rotation when the previous one is complete. Secrets Manager schedules the date by adding the rotation interval (number of days) to the actual date of the last rotation. The service chooses the hour within that 24-hour date window randomly. The minute is also chosen somewhat randomly, but weighted towards the top of the hour and influenced by a variety of factors that help distribute load.

The rotation function must end with the versions of the secret in one of two states:

- The `AWSPENDING` and `AWSCURRENT` staging labels are attached to the same version of the secret, or
- The `AWSPENDING` staging label is not attached to any version of the secret.

If instead the `AWSPENDING` staging label is present but is not attached to the same version as `AWSCURRENT` then any later invocation of `RotateSecret` assumes that a previous rotation request is still in progress and returns an error.

Minimum permissions

To run this command, you must have the following permissions:

- secretsmanager:RotateSecret
- lambda:InvokeFunction (on the function specified in the secret's metadata)

Related operations

- To list the secrets in your account, use ListSecrets (p. 41).
- To get the details for a version of a secret, use DescribeSecret (p. 23).
- To create a new version of a secret, use CreateSecret (p. 8).
- To attach staging labels to or remove staging labels from a version of a secret, use UpdateSecretVersionStage (p. 85).

Request Syntax

```
{
   "ClientRequestToken": "string",
   "RotationLambdaARN": "string",
   "RotationRules": {
      "AutomaticallyAfterDays": number
   },
```

```
   "SecretId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 98).

The request accepts the following data in JSON format.

ClientRequestToken (p. 64)

(Optional) Specifies a unique identifier for the new version of the secret that helps ensure idempotency.

If you use the AWS CLI or one of the AWS SDK to call this operation, then you can leave this parameter empty. The CLI or SDK generates a random UUID for you and includes that in the request for this parameter. If you don't use the SDK and instead generate a raw HTTP request to the Secrets Manager service endpoint, then you must generate a `ClientRequestToken` yourself for new versions and include that value in the request.

You only need to specify your own value if you are implementing your own retry logic and want to ensure that a given secret is not created twice. We recommend that you generate a UUID-type value to ensure uniqueness within the specified secret.

Secrets Manager uses this value to prevent the accidental creation of duplicate versions if there are failures and retries during the function's processing. This value becomes the `VersionId` of the new version.

Type: String

Length Constraints: Minimum length of 32. Maximum length of 64.

Required: No

RotationLambdaARN (p. 64)

(Optional) Specifies the ARN of the Lambda function that can rotate the secret.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 2048.

Required: No

RotationRules (p. 64)

A structure that defines the rotation configuration for this secret.

Type: RotationRulesType (p. 92) object

Required: No

SecretId (p. 64)

Specifies the secret that you want to rotate. You can specify either the Amazon Resource Name (ARN) or the friendly name of the secret.

> **Note**
> If you specify an ARN, we generally recommend that you specify a complete ARN. You can specify a partial ARN too—for example, if you don't include the final hyphen and six random characters that Secrets Manager adds at the end of the ARN when you created the secret. A partial ARN match can work as long as it uniquely matches only one secret.

However, if your secret has a name that ends in a hyphen followed by six characters (before Secrets Manager adds the hyphen and six characters to the ARN) and you try to use that as a partial ARN, then those characters cause Secrets Manager to assume that you're specifying a complete ARN. This confusion can cause unexpected results. To avoid this situation, we recommend that you don't create secret names that end with a hyphen followed by six characters.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

Response Syntax

```
{
   "ARN": "string",
   "Name": "string",
   "VersionId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ARN (p. 66)

The ARN of the secret.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Name (p. 66)

The friendly name of the secret.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

VersionId (p. 66)

The ID of the new version of the secret created by the rotation started by this request.

Type: String

Length Constraints: Minimum length of 32. Maximum length of 64.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 100).

InternalServiceError

An error occurred on the server side.

HTTP Status Code: 500

InvalidParameterException

You provided an invalid value for a parameter.

HTTP Status Code: 400

InvalidRequestException

You provided a parameter value that is not valid for the current state of the resource.

Possible causes:

- You tried to perform the operation on a secret that's currently marked deleted.
- You tried to enable rotation on a secret that doesn't already have a Lambda function ARN configured and you didn't include such an ARN as a parameter in this call.

HTTP Status Code: 400

ResourceNotFoundException

We can't find the resource that you asked for.

HTTP Status Code: 400

Examples

Example

The following example configures rotation for a secret by providing the ARN of a AWS Lambda rotation function (that must already exist) and the number of days between rotation. The first rotation happens immediately after the changes are stored in the secret. The `ClientRequestToken` field becomes the `VersionId` of the new version created during the rotation. The rotation function runs asynchronously in the background. The JSON request string input and response output are shown formatted with white space and line breaks for better readability. Submit your input as a single line JSON string.

Sample Request

```
POST / HTTP/1.1
Host: secretsmanager.region.domain
Accept-Encoding: identity
X-Amz-Target: secretsmanager.RotateSecret
Content-Type: application/x-amz-json-1.1
User-Agent: <user-agent-string>
X-Amz-Date: <date>
Authorization: AWS4-HMAC-SHA256 Credential=<credentials>,SignedHeaders=<headers>,
 Signature=<signature>
Content-Length: <payload-size-bytes>

{
  "SecretId": "MyTestDatabaseSecret",
  "RotationLambdaARN": "arn:aws:lambda:us-
west-2:123456789012:function:MyTestDatabaseRotationLambda",
  "RotationRules": {"AutomaticallyAfterDays": 30},
  "ClientRequestToken": "EXAMPLE2-90ab-cdef-fedc-ba987SECRET2"
}
```

Sample Response

```
HTTP/1.1 200 OK
```

```
Date: <date>
Content-Type: application/x-amz-json-1.1
Content-Length: <response-size-bytes>
Connection: keep-alive
x-amzn-RequestId: <request-id-guid>

{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-
a1b2c3",
  "Name": "MyTestDatabaseSecret",
  "VersionId": "EXAMPLE2-90ab-cdef-fedc-ba987SECRET2"
}
```

Example

The following example requests an immediate invocation of the secret's AWS Lambda rotation function. It assumes that the specified secret already has rotation configured. The `ClientRequestToken` field becomes the `VersionId` of the new version created during the rotation. The rotation function runs asynchronously in the background. The JSON request string input and response output are shown formatted with white space and line breaks for better readability. Submit your input as a single line JSON string.

Sample Request

```
POST / HTTP/1.1
Host: secretsmanager.region.domain
Accept-Encoding: identity
X-Amz-Target: secretsmanager.RotateSecret
Content-Type: application/x-amz-json-1.1
User-Agent: <user-agent-string>
X-Amz-Date: <date>
Authorization: AWS4-HMAC-SHA256 Credential=<credentials>,SignedHeaders=<headers>,
 Signature=<signature>
Content-Length: <payload-size-bytes>

{
  "SecretId": "MyTestDatabaseSecret",
  "ClientRequestToken": "EXAMPLE2-90ab-cdef-fedc-ba987SECRET2"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: <date>
Content-Type: application/x-amz-json-1.1
Content-Length: <response-size-bytes>
Connection: keep-alive
x-amzn-RequestId: <request-id-guid>

{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-
a1b2c3",
  "Name": "MyTestDatabaseSecret",
  "VersionId": "EXAMPLE2-90ab-cdef-fedc-ba987SECRET2"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

TagResource

Attaches one or more tags, each consisting of a key name and a value, to the specified secret. Tags are part of the secret's overall metadata, and are not associated with any specific version of the secret. This operation only appends tags to the existing list of tags. To remove tags, you must use UntagResource (p. 74).

The following basic restrictions apply to tags:

- Maximum number of tags per secret—50
- Maximum key length—127 Unicode characters in UTF-8
- Maximum value length—255 Unicode characters in UTF-8
- Tag keys and values are case sensitive.
- Do not use the `aws:` prefix in your tag names or values because it is reserved for AWS use. You can't edit or delete tag names or values with this prefix. Tags with this prefix do not count against your tags per secret limit.
- If your tagging schema will be used across multiple services and resources, remember that other services might have restrictions on allowed characters. Generally allowed characters are: letters, spaces, and numbers representable in UTF-8, plus the following special characters: + - = . _ : / @.

> **Important**
> If you use tags as part of your security strategy, then adding or removing a tag can change permissions. If successfully completing this operation would result in you losing your permissions for this secret, then the operation is blocked and returns an Access Denied error.

Minimum permissions

To run this command, you must have the following permissions:

- secretsmanager:TagResource

Related operations

- To remove one or more tags from the collection attached to a secret, use UntagResource (p. 74).
- To view the list of tags attached to a secret, use DescribeSecret (p. 23).

Request Syntax

```
{
   "SecretId": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 98).

The request accepts the following data in JSON format.

SecretId (p. 70)

The identifier for the secret that you want to attach tags to. You can specify either the Amazon Resource Name (ARN) or the friendly name of the secret.

Note
If you specify an ARN, we generally recommend that you specify a complete ARN. You can specify a partial ARN too—for example, if you don't include the final hyphen and six random characters that Secrets Manager adds at the end of the ARN when you created the secret. A partial ARN match can work as long as it uniquely matches only one secret. However, if your secret has a name that ends in a hyphen followed by six characters (before Secrets Manager adds the hyphen and six characters to the ARN) and you try to use that as a partial ARN, then those characters cause Secrets Manager to assume that you're specifying a complete ARN. This confusion can cause unexpected results. To avoid this situation, we recommend that you don't create secret names that end with a hyphen followed by six characters.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

Tags (p. 70)

The tags to attach to the secret. Each element in the list consists of a `Key` and a `Value`.

This parameter to the API requires a JSON text string argument. For information on how to format a JSON parameter for the various command line tool environments, see Using JSON for Parameters in the *AWS CLI User Guide*. For the AWS CLI, you can also use the syntax: `--Tags Key="Key1",Value="Value1",Key="Key2",Value="Value2"[,…]`

Type: Array of Tag (p. 97) objects

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 100).

InternalServiceError

An error occurred on the server side.

HTTP Status Code: 500

InvalidParameterException

You provided an invalid value for a parameter.

HTTP Status Code: 400

InvalidRequestException

You provided a parameter value that is not valid for the current state of the resource.

Possible causes:

- You tried to perform the operation on a secret that's currently marked deleted.
- You tried to enable rotation on a secret that doesn't already have a Lambda function ARN configured and you didn't include such an ARN as a parameter in this call.

HTTP Status Code: 400

ResourceNotFoundException

We can't find the resource that you asked for.

HTTP Status Code: 400

Example

The following example shows how to attach two tags each with a Key and Value to a secret. There is no output from this API. To see the result, use the DescribeSecret (p. 23) operation.

Sample Request

```
POST / HTTP/1.1
Host: secretsmanager.region.domain
Accept-Encoding: identity
X-Amz-Target: secretsmanager.TagResource
Content-Type: application/x-amz-json-1.1
User-Agent: <user-agent-string>
X-Amz-Date: <date>
Authorization: AWS4-HMAC-SHA256 Credential=<credentials>,SignedHeaders=<headers>,
 Signature=<signature>
Content-Length: <payload-size-bytes>

{
  "SecretId": "MyExampleSecret",
  "Tags": [
    {
      "Key": "FirstTag",
      "Value": "SomeValue"
    },
    {
      "Key": "SecondTag",
      "Value": "AnotherValue"
    }
  ]
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: <date>
Content-Type: application/x-amz-json-1.1
Content-Length: <response-size-bytes>
Connection: keep-alive
x-amzn-RequestId: <request-id-guid>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

UntagResource

Removes one or more tags from the specified secret.

This operation is idempotent. If a requested tag is not attached to the secret, no error is returned and the secret metadata is unchanged.

> **Important**
> If you use tags as part of your security strategy, then removing a tag can change permissions. If successfully completing this operation would result in you losing your permissions for this secret, then the operation is blocked and returns an Access Denied error.

Minimum permissions

To run this command, you must have the following permissions:

- secretsmanager:UntagResource

Related operations

- To add one or more tags to the collection attached to a secret, use TagResource (p. 70).
- To view the list of tags attached to a secret, use DescribeSecret (p. 23).

Request Syntax

```
{
   "SecretId": "string",
   "TagKeys": [ "string" ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 98).

The request accepts the following data in JSON format.

SecretId (p. 74)

The identifier for the secret that you want to remove tags from. You can specify either the Amazon Resource Name (ARN) or the friendly name of the secret.

> **Note**
> If you specify an ARN, we generally recommend that you specify a complete ARN. You can specify a partial ARN too—for example, if you don't include the final hyphen and six random characters that Secrets Manager adds at the end of the ARN when you created the secret. A partial ARN match can work as long as it uniquely matches only one secret. However, if your secret has a name that ends in a hyphen followed by six characters (before Secrets Manager adds the hyphen and six characters to the ARN) and you try to use that as a partial ARN, then those characters cause Secrets Manager to assume that you're specifying a complete ARN. This confusion can cause unexpected results. To avoid this situation, we recommend that you don't create secret names that end with a hyphen followed by six characters.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

TagKeys (p. 74)

A list of tag key names to remove from the secret. You don't specify the value. Both the key and its associated value are removed.

This parameter to the API requires a JSON text string argument. For information on how to format a JSON parameter for the various command line tool environments, see Using JSON for Parameters in the *AWS CLI User Guide*.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 100).

InternalServiceError

An error occurred on the server side.

HTTP Status Code: 500

InvalidParameterException

You provided an invalid value for a parameter.

HTTP Status Code: 400

InvalidRequestException

You provided a parameter value that is not valid for the current state of the resource.

Possible causes:

- You tried to perform the operation on a secret that's currently marked deleted.
- You tried to enable rotation on a secret that doesn't already have a Lambda function ARN configured and you didn't include such an ARN as a parameter in this call.

HTTP Status Code: 400

ResourceNotFoundException

We can't find the resource that you asked for.

HTTP Status Code: 400

Example

The following example shows how to remove two tags from a secret's metadata. For each, both the tag and the associated value are removed. There is no output from this API. To see the result, use the DescribeSecret (p. 23) operation.

Sample Request

```
POST / HTTP/1.1
Host: secretsmanager.region.domain
Accept-Encoding: identity
X-Amz-Target: secretsmanager.UntagResource
Content-Type: application/x-amz-json-1.1
User-Agent: <user-agent-string>
X-Amz-Date: <date>
Authorization: AWS4-HMAC-SHA256 Credential=<credentials>,SignedHeaders=<headers>,
 Signature=<signature>
Content-Length: <payload-size-bytes>

{
  "SecretId": "MyTestDatabaseSecret",
  "TagKeys": [
    "FirstTag", "SecondTag"
  ]
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: <date>
Content-Type: application/x-amz-json-1.1
Content-Length: <response-size-bytes>
Connection: keep-alive
x-amzn-RequestId: <request-id-guid>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

UpdateSecret

Modifies many of the details of the specified secret. If you include a `ClientRequestToken` and *either* `SecretString` or `SecretBinary` then it also creates a new version attached to the secret.

To modify the rotation configuration of a secret, use RotateSecret (p. 64) instead.

> **Note**
> The Secrets Manager console uses only the `SecretString` parameter and therefore limits you to encrypting and storing only a text string. To encrypt and store binary data as part of the version of a secret, you must use either the AWS CLI or one of the AWS SDKs.

- If a version with a `VersionId` with the same value as the `ClientRequestToken` parameter already exists, the operation results in an error. You cannot modify an existing version, you can only create a new version.
- If you include `SecretString` or `SecretBinary` to create a new secret version, Secrets Manager automatically attaches the staging label `AWSCURRENT` to the new version.

> **Note**
>
> - If you call an operation that needs to encrypt or decrypt the `SecretString` or `SecretBinary` for a secret in the same account as the calling user and that secret doesn't specify a AWS KMS encryption key, Secrets Manager uses the account's default AWS managed customer master key (CMK) with the alias `aws/secretsmanager`. If this key doesn't already exist in your account then Secrets Manager creates it for you automatically. All users and roles in the same AWS account automatically have access to use the default CMK. Note that if an Secrets Manager API call results in AWS having to create the account's AWS-managed CMK, it can result in a one-time significant delay in returning the result.
> - If the secret is in a different AWS account from the credentials calling an API that requires encryption or decryption of the secret value then you must create and use a custom AWS KMS CMK because you can't access the default CMK for the account using credentials from a different AWS account. Store the ARN of the CMK in the secret when you create the secret or when you update it by including it in the `KMSKeyId`. If you call an API that must encrypt or decrypt `SecretString` or `SecretBinary` using credentials from a different account then the AWS KMS key policy must grant cross-account access to that other account's user or role for both the kms:GenerateDataKey and kms:Decrypt operations.

Minimum permissions

To run this command, you must have the following permissions:

- secretsmanager:UpdateSecret
- kms:GenerateDataKey - needed only if you use a custom AWS KMS key to encrypt the secret. You do not need this permission to use the account's AWS managed CMK for Secrets Manager.
- kms:Decrypt - needed only if you use a custom AWS KMS key to encrypt the secret. You do not need this permission to use the account's AWS managed CMK for Secrets Manager.

Related operations

- To create a new secret, use CreateSecret (p. 8).
- To add only a new version to an existing secret, use PutSecretValue (p. 54).
- To get the details for a secret, use DescribeSecret (p. 23).
- To list the versions contained in a secret, use ListSecretVersionIds (p. 45).

Request Syntax

```
{
   "ClientRequestToken": "string",
   "Description": "string",
   "KmsKeyId": "string",
   "SecretBinary": blob,
   "SecretId": "string",
   "SecretString": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 98).

The request accepts the following data in JSON format.

ClientRequestToken (p. 78)

(Optional) If you want to add a new version to the secret, this parameter specifies a unique identifier for the new version that helps ensure idempotency.

If you use the AWS CLI or one of the AWS SDK to call this operation, then you can leave this parameter empty. The CLI or SDK generates a random UUID for you and includes that in the request. If you don't use the SDK and instead generate a raw HTTP request to the Secrets Manager service endpoint, then you must generate a `ClientRequestToken` yourself for new versions and include that value in the request.

You typically only need to interact with this value if you implement your own retry logic and want to ensure that a given secret is not created twice. We recommend that you generate a UUID-type value to ensure uniqueness within the specified secret.

Secrets Manager uses this value to prevent the accidental creation of duplicate versions if there are failures and retries during the Lambda rotation function's processing.

- If the `ClientRequestToken` value isn't already associated with a version of the secret then a new version of the secret is created.
- If a version with this value already exists and that version's `SecretString` and `SecretBinary` values are the same as those in the request then the request is ignored (the operation is idempotent).
- If a version with this value already exists and that version's `SecretString` and `SecretBinary` values are different from the request then an error occurs because you cannot modify an existing secret value.

This value becomes the `VersionId` of the new version.

Type: String

Length Constraints: Minimum length of 32. Maximum length of 64.

Required: No

Description (p. 78)

(Optional) Specifies an updated user-provided description of the secret.

Type: String

Length Constraints: Maximum length of 2048.

Required: No

KmsKeyId (p. 78)

(Optional) Specifies an updated ARN or alias of the AWS KMS customer master key (CMK) to be used to encrypt the protected text in new versions of this secret.

> **Important**
> You can only use the account's default CMK to encrypt and decrypt if you call this operation using credentials from the same account that owns the secret. If the secret is in a different account, then you must create a custom CMK and provide the ARN of that CMK in this field. The user making the call must have permissions to both the secret and the CMK in their respective accounts.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 2048.

Required: No

SecretBinary (p. 78)

(Optional) Specifies updated binary data that you want to encrypt and store in the new version of the secret. To use this parameter in the command-line tools, we recommend that you store your binary data in a file and then use the appropriate technique for your tool to pass the contents of the file as a parameter. Either `SecretBinary` or `SecretString` must have a value, but not both. They cannot both be empty.

This parameter is not accessible using the Secrets Manager console.

Type: Base64-encoded binary data object

Length Constraints: Minimum length of 0. Maximum length of 7168.

Required: No

SecretId (p. 78)

Specifies the secret that you want to modify or to which you want to add a new version. You can specify either the Amazon Resource Name (ARN) or the friendly name of the secret.

> **Note**
> If you specify an ARN, we generally recommend that you specify a complete ARN. You can specify a partial ARN too—for example, if you don't include the final hyphen and six random characters that Secrets Manager adds at the end of the ARN when you created the secret. A partial ARN match can work as long as it uniquely matches only one secret. However, if your secret has a name that ends in a hyphen followed by six characters (before Secrets Manager adds the hyphen and six characters to the ARN) and you try to use that as a partial ARN, then those characters cause Secrets Manager to assume that you're specifying a complete ARN. This confusion can cause unexpected results. To avoid this situation, we recommend that you don't create secret names that end with a hyphen followed by six characters.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

SecretString (p. 78)

(Optional) Specifies updated text data that you want to encrypt and store in this new version of the secret. Either `SecretBinary` or `SecretString` must have a value, but not both. They cannot both be empty.

If you create this secret by using the Secrets Manager console then Secrets Manager puts the protected secret text in only the `SecretString` parameter. The Secrets Manager console stores the information as a JSON structure of key/value pairs that the default Lambda rotation function knows how to parse.

For storing multiple values, we recommend that you use a JSON text string argument and specify key/value pairs. For information on how to format a JSON parameter for the various command line tool environments, see Using JSON for Parameters in the *AWS CLI User Guide*. For example:

```
[{"username":"bob"},{"password":"abc123xyz456"}]
```

If your command-line tool or SDK requires quotation marks around the parameter, you should use single quotes to avoid confusion with the double quotes required in the JSON text. You can also 'escape' the double quote character in the embedded JSON text by prefacing each with a backslash. For example, the following string is surrounded by double-quotes. All of the embedded double quotes are escaped:

```
"[{\"username\":\"bob\"},{\"password\":\"abc123xyz456\"}]"
```

Type: String

Length Constraints: Minimum length of 0. Maximum length of 7168.

Required: No

Response Syntax

```
{
   "ARN": "string",
   "Name": "string",
   "VersionId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ARN (p. 80)

The ARN of the secret that was updated.

Note
Secrets Manager automatically adds several random characters to the name at the end of the ARN when you initially create a secret. This affects only the ARN and not the actual friendly name. This ensures that if you create a new secret with the same name as an old secret that you previously deleted, then users with access to the old secret *don't* automatically get access to the new secret because the ARNs are different.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Name (p. 80)

The friendly name of the secret that was updated.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

VersionId (p. 80)

If a new version of the secret was created by this operation, then `VersionId` contains the unique identifier of the new version.

Type: String

Length Constraints: Minimum length of 32. Maximum length of 64.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 100).

EncryptionFailure

Secrets Manager can't encrypt the protected secret text using the provided KMS key. Check that the customer master key (CMK) is available, enabled, and not in an invalid state. For more information, see How Key State Affects Use of a Customer Master Key.

HTTP Status Code: 400

InternalServiceError

An error occurred on the server side.

HTTP Status Code: 500

InvalidParameterException

You provided an invalid value for a parameter.

HTTP Status Code: 400

InvalidRequestException

You provided a parameter value that is not valid for the current state of the resource.

Possible causes:

- You tried to perform the operation on a secret that's currently marked deleted.
- You tried to enable rotation on a secret that doesn't already have a Lambda function ARN configured and you didn't include such an ARN as a parameter in this call.

HTTP Status Code: 400

LimitExceededException

The request failed because it would exceed one of the Secrets Manager internal limits.

HTTP Status Code: 400

MalformedPolicyDocumentException

The policy document that you provided isn't valid.

HTTP Status Code: 400

PreconditionNotMetException

The request failed because you did not complete all the prerequisite steps.

HTTP Status Code: 400

ResourceExistsException

A resource with the ID you requested already exists.

HTTP Status Code: 400

ResourceNotFoundException

We can't find the resource that you asked for.

HTTP Status Code: 400

Examples

The following examples show how to modify individual components of the secret. Alternatively, you can combine all of the parameters into a single command to do them all in one operation.

Example

The following example shows how to modify the description of a secret. The JSON request string input and response output are shown formatted with white space and line breaks for better readability. Submit your input as a single line JSON string.

Sample Request

```
POST / HTTP/1.1
Host: secretsmanager.region.domain
Accept-Encoding: identity
X-Amz-Target: secretsmanager.UpdateSecret
Content-Type: application/x-amz-json-1.1
User-Agent: <user-agent-string>
X-Amz-Date: <date>
Authorization: AWS4-HMAC-SHA256 Credential=<credentials>,SignedHeaders=<headers>,
 Signature=<signature>
Content-Length: <payload-size-bytes>

{
  "SecretId": "MyTestDatabaseSecret",
  "Description": "This is a new description for the secret.",
  "ClientRequestToken": "EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: <date>
Content-Type: application/x-amz-json-1.1
Content-Length: <response-size-bytes>
Connection: keep-alive
x-amzn-RequestId: <request-id-guid>

{
  "ARN":"arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
  "Name":"MyTestDatabaseSecret"
}
```

Example

This example shows how to update the AWS KMS customer managed key (CMK) used to encrypt the secret value. The AWS KMS CMK must be in the same region as the secret. The JSON request string input

and response output are shown formatted with white space and line breaks for better readability. Submit your input as a single line JSON string.

Sample Request

```
POST / HTTP/1.1
Host: secretsmanager.region.domain
Accept-Encoding: identity
X-Amz-Target: secretsmanager.UpdateSecret
Content-Type: application/x-amz-json-1.1
User-Agent: <user-agent-string>
X-Amz-Date: <date>
Authorization: AWS4-HMAC-SHA256 Credential=<credentials>,SignedHeaders=<headers>,
 Signature=<signature>
Content-Length: <payload-size-bytes>

{
  "SecretId": "MyTestDatabaseSecret",
  "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/EXAMPLE2-90ab-cdef-fedc-ba987EXAMPLE"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: <date>
Content-Type: application/x-amz-json-1.1
Content-Length: <response-size-bytes>
Connection: keep-alive
x-amzn-RequestId: <request-id-guid>

{
  "ARN":"arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
  "Name":"MyTestDatabaseSecret"
}
```

Example

The following example shows how to create a new version of the secret by updating the `SecretString` field. The `ClientRequestToken` parameter becomes the `VersionId` of the new version. Alternatively, you can use the PutSecretValue (p. 54) operation. The JSON request string input and response output are shown formatted with white space and line breaks for better readability. Submit your input as a single line JSON string.

Sample Request

```
POST / HTTP/1.1
Host: secretsmanager.region.domain
Accept-Encoding: identity
X-Amz-Target: secretsmanager.UpdateSecret
Content-Type: application/x-amz-json-1.1
User-Agent: <user-agent-string>
X-Amz-Date: <date>
Authorization: AWS4-HMAC-SHA256 Credential=<credentials>,SignedHeaders=<headers>,
 Signature=<signature>
Content-Length: <payload-size-bytes>

{
  "SecretId": "MyTestDatabaseSecret",
  "SecretString": "{<JSON STRING WITH CREDENTIALS>}",
  "ClientRequestToken": "EXAMPLE1-90ab-cdef-fedc-ba987SECRET1"
```

```
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: <date>
Content-Type: application/x-amz-json-1.1
Content-Length: <response-size-bytes>
Connection: keep-alive
x-amzn-RequestId: <request-id-guid>

{
  "ARN":"arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
  "Name":"MyTestDatabaseSecret",
  "VersionId":"EXAMPLE1-90ab-cdef-fedc-ba987SECRET1"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

UpdateSecretVersionStage

Modifies the staging labels attached to a version of a secret. Staging labels are used to track a version as it progresses through the secret rotation process. You can attach a staging label to only one version of a secret at a time. If a staging label to be added is already attached to another version, then it is moved--removed from the other version first and then attached to this one. For more information about staging labels, see Staging Labels in the *AWS Secrets Manager User Guide*.

The staging labels that you specify in the `VersionStage` parameter are added to the existing list of staging labels--they don't replace it.

You can move the `AWSCURRENT` staging label to this version by including it in this call.

> **Note**
> Whenever you move `AWSCURRENT`, Secrets Manager automatically moves the label `AWSPREVIOUS` to the version that `AWSCURRENT` was removed from.

If this action results in the last label being removed from a version, then the version is considered to be 'deprecated' and can be deleted by Secrets Manager.

Minimum permissions

To run this command, you must have the following permissions:

- secretsmanager:UpdateSecretVersionStage

Related operations

- To get the list of staging labels that are currently associated with a version of a secret, use `DescribeSecret (p. 23)` and examine the `SecretVersionsToStages` response value.

Request Syntax

```
{
   "MoveToVersionId": "string",
   "RemoveFromVersionId": "string",
   "SecretId": "string",
   "VersionStage": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 98).

The request accepts the following data in JSON format.

MoveToVersionId (p. 85)

(Optional) The secret version ID that you want to add the staging label to. If you want to remove a label from a version, then do not specify this parameter.

If the staging label is already attached to a different version of the secret, then you must also specify the `RemoveFromVersionId` parameter.

Type: String

Length Constraints: Minimum length of 32. Maximum length of 64.

Required: No

RemoveFromVersionId (p. 85)

Specifies the secret version ID of the version that the staging label is to be removed from. If the staging label you are trying to attach to one version is already attached to a different version, then you must include this parameter and specify the version that the label is to be removed from. If the label is attached and you either do not specify this parameter, or the version ID does not match, then the operation fails.

Type: String

Length Constraints: Minimum length of 32. Maximum length of 64.

Required: No

SecretId (p. 85)

Specifies the secret with the version whose list of staging labels you want to modify. You can specify either the Amazon Resource Name (ARN) or the friendly name of the secret.

Note
If you specify an ARN, we generally recommend that you specify a complete ARN. You can specify a partial ARN too—for example, if you don't include the final hyphen and six random characters that Secrets Manager adds at the end of the ARN when you created the secret. A partial ARN match can work as long as it uniquely matches only one secret. However, if your secret has a name that ends in a hyphen followed by six characters (before Secrets Manager adds the hyphen and six characters to the ARN) and you try to use that as a partial ARN, then those characters cause Secrets Manager to assume that you're specifying a complete ARN. This confusion can cause unexpected results. To avoid this situation, we recommend that you don't create secret names that end with a hyphen followed by six characters.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

VersionStage (p. 85)

The staging label to add to this version.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Required: Yes

Response Syntax

```
{
   "ARN": "string",
   "Name": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ARN (p. 86)

The ARN of the secret with the staging label that was modified.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Name (p. 86)

The friendly name of the secret with the staging label that was modified.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 100).

InternalServiceError

An error occurred on the server side.

HTTP Status Code: 500

InvalidParameterException

You provided an invalid value for a parameter.

HTTP Status Code: 400

InvalidRequestException

You provided a parameter value that is not valid for the current state of the resource.

Possible causes:

- You tried to perform the operation on a secret that's currently marked deleted.
- You tried to enable rotation on a secret that doesn't already have a Lambda function ARN configured and you didn't include such an ARN as a parameter in this call.

HTTP Status Code: 400

LimitExceededException

The request failed because it would exceed one of the Secrets Manager internal limits.

HTTP Status Code: 400

ResourceNotFoundException

We can't find the resource that you asked for.

HTTP Status Code: 400

Examples

Example

The following example shows you how to add a staging label to a version of a secret. You can review the results by calling ListSecretVersionIds (p. 45). The JSON request string input and response output are shown formatted with white space and line breaks for better readability. Submit your input as a single line JSON string.

Sample Request

```
POST / HTTP/1.1
Host: secretsmanager.region.domain
Accept-Encoding: identity
X-Amz-Target: secretsmanager.UpdateSecretVersionStage
Content-Type: application/x-amz-json-1.1
User-Agent: <user-agent-string>
X-Amz-Date: <date>
Authorization: AWS4-HMAC-SHA256 Credential=<credentials>,SignedHeaders=<headers>,
 Signature=<signature>
Content-Length: <payload-size-bytes>

{
  "SecretId": "MyTestDatabaseSecret",
  "VersionStage": "STAGINGLABEL1",
  "MoveToVersionId": "EXAMPLE1-90ab-cdef-fedc-ba987SECRET1"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: <date>
Content-Type: application/x-amz-json-1.1
Content-Length: <response-size-bytes>
Connection: keep-alive
x-amzn-RequestId: <request-id-guid>

{
  "ARN":"arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
  "Name":"MyTestDatabaseSecret"
}
```

Example

The following example shows you how to remove a staging label from a version of a secret. You can review the results by calling ListSecretVersionIds (p. 45). The JSON request string input and response output are shown formatted with white space and line breaks for better readability. Submit your input as a single line JSON string.

Sample Request

```
POST / HTTP/1.1
Host: secretsmanager.region.domain
Accept-Encoding: identity
X-Amz-Target: secretsmanager.UpdateSecretVersionStage
Content-Type: application/x-amz-json-1.1
User-Agent: <user-agent-string>
X-Amz-Date: <date>
```

```
Authorization: AWS4-HMAC-SHA256 Credential=<credentials>,SignedHeaders=<headers>,
 Signature=<signature>
Content-Length: <payload-size-bytes>

{
  "SecretId": "MyTestDatabaseSecret",
  "VersionStage": "STAGINGLABEL1",
  "RemoveFromVersionId": "EXAMPLE1-90ab-cdef-fedc-ba987SECRET1"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: <date>
Content-Type: application/x-amz-json-1.1
Content-Length: <response-size-bytes>
Connection: keep-alive
x-amzn-RequestId: <request-id-guid>

{
  "ARN":"arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
  "Name":"MyTestDatabaseSecret"
}
```

Example

The following example shows you how to move a staging label from one version of a secret to another. You can review the results by calling ListSecretVersionIds (p. 45). The JSON request string input and response output are shown formatted with white space and line breaks for better readability. Submit your input as a single line JSON string.

Sample Request

```
POST / HTTP/1.1
Host: secretsmanager.region.domain
Accept-Encoding: identity
X-Amz-Target: secretsmanager.UpdateSecretVersionStage
Content-Type: application/x-amz-json-1.1
User-Agent: <user-agent-string>
X-Amz-Date: <date>
Authorization: AWS4-HMAC-SHA256 Credential=<credentials>,SignedHeaders=<headers>,
 Signature=<signature>
Content-Length: <payload-size-bytes>

{
  "SecretId": "MyTestDatabaseSecret",
  "VersionStage": "AWSCURRENT",
  "RemoveFromVersionId": "EXAMPLE1-90ab-cdef-fedc-ba987SECRET1",
  "MoveToVersionId": "EXAMPLE2-90ab-cdef-fedc-ba987SECRET2"
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: <date>
Content-Type: application/x-amz-json-1.1
Content-Length: <response-size-bytes>
Connection: keep-alive
x-amzn-RequestId: <request-id-guid>
```

```
{
  "ARN":"arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
  "Name":"MyTestDatabaseSecret"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

Data Types

The AWS Secrets Manager API contains several data types that various actions use. This section describes each data type in detail.

> **Note**
> The order of each element in a data type structure is not guaranteed. Applications should not assume a particular order.

The following data types are supported:

- RotationRulesType (p. 92)
- SecretListEntry (p. 93)
- SecretVersionsListEntry (p. 96)
- Tag (p. 97)

RotationRulesType

A structure that defines the rotation configuration for the secret.

Contents

AutomaticallyAfterDays

Specifies the number of days between automatic scheduled rotations of the secret.

Secrets Manager schedules the next rotation when the previous one is complete. Secrets Manager schedules the date by adding the rotation interval (number of days) to the actual date of the last rotation. The service chooses the hour within that 24-hour date window randomly. The minute is also chosen somewhat randomly, but weighted towards the top of the hour and influenced by a variety of factors that help distribute load.

Type: Long

Valid Range: Minimum value of 1. Maximum value of 1000.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

SecretListEntry

A structure that contains the details about a secret. It does not include the encrypted `SecretString` and `SecretBinary` values. To get those values, use the GetSecretValue (p. 36) operation.

Contents

ARN

The Amazon Resource Name (ARN) of the secret.

For more information about ARNs in Secrets Manager, see Policy Resources in the *AWS Secrets Manager User Guide*.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Required: No

DeletedDate

The date and time on which this secret was deleted. Not present on active secrets. The secret can be recovered until the number of days in the recovery window has passed, as specified in the `RecoveryWindowInDays` parameter of the DeleteSecret (p. 19) operation.

Type: Timestamp

Required: No

Description

The user-provided description of the secret.

Type: String

Length Constraints: Maximum length of 2048.

Required: No

KmsKeyId

The ARN or alias of the AWS KMS customer master key (CMK) that's used to encrypt the `SecretString` and `SecretBinary` fields in each version of the secret. If you don't provide a key, then Secrets Manager defaults to encrypting the secret fields with the default KMS CMK (the one named `awssecretsmanager`) for this account.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 2048.

Required: No

LastAccessedDate

The last date that this secret was accessed. This value is truncated to midnight of the date and therefore shows only the date, not the time.

Type: Timestamp

Required: No

LastChangedDate

The last date and time that this secret was modified in any way.

Type: Timestamp

Required: No

LastRotatedDate

The last date and time that the rotation process for this secret was invoked.

Type: Timestamp

Required: No

Name

The friendly name of the secret. You can use forward slashes in the name to represent a path hierarchy. For example, `/prod/databases/dbserver1` could represent the secret for a server named `dbserver1` in the folder `databases` in the folder `prod`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Required: No

RotationEnabled

Indicated whether automatic, scheduled rotation is enabled for this secret.

Type: Boolean

Required: No

RotationLambdaARN

The ARN of an AWS Lambda function that's invoked by Secrets Manager to rotate and expire the secret either automatically per the schedule or manually by a call to RotateSecret (p. 64).

Type: String

Length Constraints: Minimum length of 0. Maximum length of 2048.

Required: No

RotationRules

A structure that defines the rotation configuration for the secret.

Type: RotationRulesType (p. 92) object

Required: No

SecretVersionsToStages

A list of all of the currently assigned `SecretVersionStage` staging labels and the `SecretVersionId` that each is attached to. Staging labels are used to keep track of the different versions during the rotation process.

> **Note**
> A version that does not have any `SecretVersionStage` is considered deprecated and subject to deletion. Such versions are not included in this list.

Type: String to array of strings map

Key Length Constraints: Minimum length of 32. Maximum length of 64.

Array Members: Minimum number of 1 item. Maximum number of 20 items.

Length Constraints: Minimum length of 1. Maximum length of 256.

Required: No

Tags

The list of user-defined tags that are associated with the secret. To add tags to a secret, use TagResource (p. 70). To remove tags, use UntagResource (p. 74).

Type: Array of Tag (p. 97) objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

SecretVersionsListEntry

A structure that contains information about one version of a secret.

Contents

CreatedDate

The date and time this version of the secret was created.

Type: Timestamp

Required: No

LastAccessedDate

The date that this version of the secret was last accessed. Note that the resolution of this field is at the date level and does not include the time.

Type: Timestamp

Required: No

VersionId

The unique version identifier of this version of the secret.

Type: String

Length Constraints: Minimum length of 32. Maximum length of 64.

Required: No

VersionStages

An array of staging labels that are currently associated with this version of the secret.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 20 items.

Length Constraints: Minimum length of 1. Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

Tag

A structure that contains information about a tag.

Contents

Key

The key identifier, or name, of the tag.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: No

Value

The string value that's associated with the key of the tag.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

Common Parameters

The following list contains the parameters that all actions use for signing Signature Version 4 requests with a query string. Any action-specific parameters are listed in the topic for that action. For more information about Signature Version 4, see Signature Version 4 Signing Process in the *Amazon Web Services General Reference*.

Action

The action to be performed.

Type: string

Required: Yes

Version

The API version that the request is written for, expressed in the format YYYY-MM-DD.

Type: string

Required: Yes

X-Amz-Algorithm

The hash algorithm that you used to create the request signature.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Valid Values: `AWS4-HMAC-SHA256`

Required: Conditional

X-Amz-Credential

The credential scope value, which is a string that includes your access key, the date, the region you are targeting, the service you are requesting, and a termination string ("aws4_request"). The value is expressed in the following format: *access_key*/*YYYYMMDD*/*region*/*service*/aws4_request.

For more information, see Task 2: Create a String to Sign for Signature Version 4 in the *Amazon Web Services General Reference*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-Date

The date that is used to create the signature. The format must be ISO 8601 basic format (YYYYMMDD'T'HHMMSS'Z'). For example, the following date time is a valid X-Amz-Date value: `20120325T120000Z`.

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is

not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see Handling Dates in Signature Version 4 in the *Amazon Web Services General Reference*.

Type: string

Required: Conditional

X-Amz-Security-Token

The temporary security token that was obtained through a call to AWS Security Token Service (AWS STS). For a list of services that support temporary security credentials from AWS Security Token Service, go to AWS Services That Work with IAM in the *IAM User Guide*.

Condition: If you're using temporary security credentials from the AWS Security Token Service, you must include the security token.

Type: string

Required: Conditional

X-Amz-Signature

Specifies the hex-encoded signature that was calculated from the string to sign and the derived signing key.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-SignedHeaders

Specifies all the HTTP headers that were included as part of the canonical request. For more information about specifying signed headers, see Task 1: Create a Canonical Request For Signature Version 4 in the *Amazon Web Services General Reference*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

Common Errors

This section lists the errors common to the API actions of all AWS services. For errors specific to an API action for this service, see the topic for that API action.

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

IncompleteSignature

The request signature does not conform to AWS standards.

HTTP Status Code: 400

InternalFailure

The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

InvalidAction

The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 400

InvalidClientTokenId

The X.509 certificate or AWS access key ID provided does not exist in our records.

HTTP Status Code: 403

InvalidParameterCombination

Parameters that must not be used together were used together.

HTTP Status Code: 400

InvalidParameterValue

An invalid or out-of-range value was supplied for the input parameter.

HTTP Status Code: 400

InvalidQueryParameter

The AWS query string is malformed or does not adhere to AWS standards.

HTTP Status Code: 400

MalformedQueryString

The query string contains a syntax error.

HTTP Status Code: 404

MissingAction

The request is missing an action or a required parameter.

HTTP Status Code: 400

MissingAuthenticationToken

The request must contain either a valid (registered) AWS access key ID or X.509 certificate.

HTTP Status Code: 403

MissingParameter

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

OptInRequired

The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

RequestExpired

The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

ServiceUnavailable

The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 400

ValidationError

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400