AWS Key Management Service

API Reference

API Version 2014-11-01

AWS Key Management Service: API Reference

Table of Contents

Welcome

AWS Key Management Service (AWS KMS) is an encryption and key management web service. This guide describes the AWS KMS operations that you can call programmatically. For general information about AWS KMS, see the AWS Key Management Service Developer Guide.

> **Note**
> AWS provides SDKs that consist of libraries and sample code for various programming languages and platforms (Java, Ruby, .Net, iOS, Android, etc.). The SDKs provide a convenient way to create programmatic access to AWS KMS and other AWS services. For example, the SDKs take care of tasks such as signing requests (see below), 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 AWS KMS.

Clients must support TLS (Transport Layer Security) 1.0. We recommend TLS 1.2. Clients must also support cipher suites with Perfect Forward Secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Signing Requests

Requests must be signed by using an access key ID and a secret access key. We strongly recommend that you *do not* use your AWS account (root) access key ID and secret key for everyday work with AWS KMS. Instead, use the access key ID and secret access key for an IAM user, or you can use the AWS Security Token Service to generate temporary security credentials that you can use to sign requests.

All AWS KMS operations require Signature Version 4.

Logging API Requests

AWS KMS supports AWS CloudTrail, a service that logs AWS API calls and related events for your AWS account and delivers them to an Amazon S3 bucket that you specify. By using the information collected by CloudTrail, you can determine what requests were made to AWS KMS, who made the request, when it was made, and so on. To learn more about CloudTrail, including how to turn it on and find your log files, see the AWS CloudTrail User Guide.

Additional Resources

For more information about credentials and request signing, see the following:

- AWS Security Credentials - This topic provides general information about the types of credentials used for accessing AWS.
- Temporary Security Credentials - This section of the *IAM User Guide* describes how to create and use temporary security credentials.
- Signature Version 4 Signing Process - This set of topics walks you through the process of signing a request using an access key ID and a secret access key.

Commonly Used APIs

Of the APIs discussed in this guide, the following will prove the most useful for most applications. You will likely perform actions other than these, such as creating keys and assigning policies, by using the console.

- Encrypt (p. 33)
- Decrypt (p. 17)
- GenerateDataKey (p. 37)
- GenerateDataKeyWithoutPlaintext (p. 40)

This document was last published on December 9, 2016.

Actions

The following actions are supported:

- CancelKeyDeletion (p. 5)
- CreateAlias (p. 7)
- CreateGrant (p. 10)
- CreateKey (p. 14)
- Decrypt (p. 17)
- DeleteAlias (p. 20)
- DeleteImportedKeyMaterial (p. 21)
- DescribeKey (p. 23)
- DisableKey (p. 25)
- DisableKeyRotation (p. 27)
- EnableKey (p. 29)
- EnableKeyRotation (p. 31)
- Encrypt (p. 33)
- GenerateDataKey (p. 37)
- GenerateDataKeyWithoutPlaintext (p. 40)
- GenerateRandom (p. 43)
- GetKeyPolicy (p. 45)
- GetKeyRotationStatus (p. 47)
- GetParametersForImport (p. 49)
- ImportKeyMaterial (p. 52)
- ListAliases (p. 55)
- ListGrants (p. 57)
- ListKeyPolicies (p. 60)
- ListKeys (p. 63)
- ListRetirableGrants (p. 65)
- PutKeyPolicy (p. 68)
- ReEncrypt (p. 71)
- RetireGrant (p. 74)
- RevokeGrant (p. 76)
- ScheduleKeyDeletion (p. 78)

- UpdateAlias (p. 80)
- UpdateKeyDescription (p. 82)

CancelKeyDeletion

Cancels the deletion of a customer master key (CMK). When this operation is successful, the CMK is set to the `Disabled` state. To enable a CMK, use EnableKey (p. 29).

For more information about scheduling and canceling deletion of a CMK, see Deleting Customer Master Keys in the *AWS Key Management Service Developer Guide*.

Request Syntax

```
{
   "KeyId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

> **Note**
> In the following list, the required parameters are described first.

KeyId (p. 5)

The unique identifier for the customer master key (CMK) for which to cancel deletion.

To specify this value, use the unique key ID or the Amazon Resource Name (ARN) of the CMK. Examples:

- Unique key ID: 1234abcd-12ab-34cd-56ef-1234567890ab
- Key ARN: arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab

To obtain the unique key ID and key ARN for a given CMK, use ListKeys (p. 63) or DescribeKey (p. 23).

Type: String

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

Required: Yes

Response Syntax

```
{
   "KeyId": "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.

KeyId (p. 5)

The unique identifier of the master key for which deletion is canceled.

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. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

CreateAlias

Creates a display name for a customer master key. An alias can be used to identify a key and should be unique. The console enforces a one-to-one mapping between the alias and a key. An alias name can contain only alphanumeric characters, forward slashes (/), underscores (_), and dashes (-). An alias must start with the word "alias" followed by a forward slash (alias/). An alias that begins with "aws" after the forward slash (alias/aws...) is reserved by Amazon Web Services (AWS).

The alias and the key it is mapped to must be in the same AWS account and the same region.

To map an alias to a different key, call UpdateAlias (p. 80).

Request Syntax

```
{
   "AliasName": "string",
   "TargetKeyId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

Note
In the following list, the required parameters are described first.

AliasName (p. 7)

String that contains the display name. The name must start with the word "alias" followed by a forward slash (alias/). Aliases that begin with "alias/AWS" are reserved.

Type: String

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

Pattern: `^[a-zA-Z0-9:/_-]+$`

Required: Yes

TargetKeyId (p. 7)

An identifier of the key for which you are creating the alias. This value cannot be another alias but can be a globally unique identifier or a fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Type: String

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

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. 94).

AlreadyExistsException

The request was rejected because it attempted to create a resource that already exists.

HTTP Status Code: 400

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

InvalidAliasNameException

The request was rejected because the specified alias name is not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

LimitExceededException

The request was rejected because a limit was exceeded. For more information, see Limits in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

Examples

Sample Request

```
POST / HTTP/1.1
Host: kms.us-west-2.amazonaws.com
X-Amz-Target: TrentService.CreateAlias
X-Amz-Date: 20160517T204220Z
Content-Type: application/x-amz-json-1.1
Authorization: AWS4-HMAC-SHA256\
 Credential=AKIAI44QH8DHBEXAMPLE/20160517/us-west-2/kms/aws4_request,\
 SignedHeaders=content-type;host;x-amz-date;x-amz-target,\
 Signature=ca7bcf1e8d5364dc3f0d881c05bdadf36f498c6c6a8b576a060142d9b2199123

{
  "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "AliasName": "alias/example-alias"
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Tue, 17 May 2016 20:42:25 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 0
Connection: keep-alive
```

```
x-amzn-RequestId: dcb07ca7-1c6f-11e6-8540-77c363708b91
```

CreateGrant

Adds a grant to a key to specify who can use the key and under what conditions. Grants are alternate permission mechanisms to key policies.

For more information about grants, see Grants in the *AWS Key Management Service Developer Guide*.

Request Syntax

```
{
   "Constraints": {
      "EncryptionContextEquals": {
         "string" : "string"
      },
      "EncryptionContextSubset": {
         "string" : "string"
      }
   },
   "GranteePrincipal": "string",
   "GrantTokens": [ "string" ],
   "KeyId": "string",
   "Name": "string",
   "Operations": [ "string" ],
   "RetiringPrincipal": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

> **Note**
> In the following list, the required parameters are described first.

GranteePrincipal (p. 10)

The principal that is given permission to perform the operations that the grant permits.

To specify the principal, use the Amazon Resource Name (ARN) of an AWS principal. Valid AWS principals include AWS accounts (root), IAM users, federated users, and assumed role users. For examples of the ARN syntax to use for specifying a principal, see AWS Identity and Access Management (IAM) in the Example ARNs section of the *AWS General Reference*.

Type: String

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

Required: Yes

KeyId (p. 10)

The unique identifier for the customer master key (CMK) that the grant applies to.

To specify this value, use the globally unique key ID or the Amazon Resource Name (ARN) of the key. Examples:

- Globally unique key ID: 12345678-1234-1234-1234-123456789012
- Key ARN: arn:aws:kms:us-west-2:123456789012:key/12345678-1234-1234-1234-123456789012

Type: String

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

Required: Yes

Constraints (p. 10)

The conditions under which the operations permitted by the grant are allowed.

You can use this value to allow the operations permitted by the grant only when a specified encryption context is present. For more information, see Encryption Context in the *AWS Key Management Service Developer Guide*.

Type: GrantConstraints (p. 86) object

Required: No

GrantTokens (p. 10)

A list of grant tokens.

For more information, see Grant Tokens in the *AWS Key Management Service Developer Guide*.

Type: array of Strings

Array Members: Minimum number of 0 items. Maximum number of 10 items.

Length Constraints: Minimum length of 1. Maximum length of 8192.

Required: No

Name (p. 10)

A friendly name for identifying the grant. Use this value to prevent unintended creation of duplicate grants when retrying this request.

When this value is absent, all `CreateGrant` requests result in a new grant with a unique `GrantId` even if all the supplied parameters are identical. This can result in unintended duplicates when you retry the `CreateGrant` request.

When this value is present, you can retry a `CreateGrant` request with identical parameters; if the grant already exists, the original `GrantId` is returned without creating a new grant. Note that the returned grant token is unique with every `CreateGrant` request, even when a duplicate `GrantId` is returned. All grant tokens obtained in this way can be used interchangeably.

Type: String

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

Pattern: `^[a-zA-Z0-9:/_-]+$`

Required: No

Operations (p. 10)

A list of operations that the grant permits. The list can contain any combination of one or more of the following values:

- Decrypt (p. 17)
- Encrypt (p. 33)
- GenerateDataKey (p. 37)
- GenerateDataKeyWithoutPlaintext (p. 40)
- ReEncryptFrom
- ReEncryptTo
- CreateGrant (p. 10)
- RetireGrant (p. 74)
- DescribeKey (p. 23)

Type: array of Strings

Valid Values: `Decrypt | Encrypt | GenerateDataKey | GenerateDataKeyWithoutPlaintext | ReEncryptFrom | ReEncryptTo | CreateGrant | RetireGrant | DescribeKey`

Required: No

RetiringPrincipal (p. 10)

The principal that is given permission to retire the grant by using RetireGrant (p. 74) operation.

To specify the principal, use the Amazon Resource Name (ARN) of an AWS principal. Valid AWS principals include AWS accounts (root), IAM users, federated users, and assumed role users. For examples of the ARN syntax to use for specifying a principal, see AWS Identity and Access Management (IAM) in the Example ARNs section of the *AWS General Reference*.

Type: String

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

Required: No

Response Syntax

```
{
   "GrantId": "string",
   "GrantToken": "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.

GrantId (p. 12)

The unique identifier for the grant.

You can use the `GrantId` in a subsequent RetireGrant (p. 74) or RevokeGrant (p. 76) operation.

Type: String

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

GrantToken (p. 12)

The grant token.

For more information, see Grant Tokens in the *AWS Key Management Service Developer Guide*.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 8192.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

DisabledException

The request was rejected because the specified CMK is not enabled.

HTTP Status Code: 400

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

InvalidGrantTokenException

The request was rejected because the specified grant token is not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

LimitExceededException

The request was rejected because a limit was exceeded. For more information, see Limits in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

CreateKey

Creates a customer master key (CMK).

You can use a CMK to encrypt small amounts of data (4 KiB or less) directly, but CMKs are more commonly used to encrypt data encryption keys (DEKs), which are used to encrypt raw data. For more information about DEKs and the difference between CMKs and DEKs, see the following:

- The GenerateDataKey (p. 37) operation
- AWS Key Management Service Concepts in the *AWS Key Management Service Developer Guide*

Request Syntax

```
{
   "BypassPolicyLockoutSafetyCheck": boolean,
   "Description": "string",
   "KeyUsage": "string",
   "Origin": "string",
   "Policy": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

> **Note**
> In the following list, the required parameters are described first.

BypassPolicyLockoutSafetyCheck (p. 14)

A flag to indicate whether to bypass the key policy lockout safety check.

> **Important**
> Setting this value to true increases the likelihood that the CMK becomes unmanageable. Do not set this value to true indiscriminately.
> For more information, refer to the scenario in the Default Key Policy section in the *AWS Key Management Service Developer Guide*.

Use this parameter only when you include a policy in the request and you intend to prevent the principal making the request from making a subsequent PutKeyPolicy (p. 68) request on the CMK.

The default value is false.

Type: Boolean

Required: No

Description (p. 14)

A description of the CMK.

Use a description that helps you decide whether the CMK is appropriate for a task.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 8192.

Required: No

KeyUsage (p. 14)

The intended use of the CMK.

You can use CMKs only for symmetric encryption and decryption.

Type: String

Valid Values: `ENCRYPT_DECRYPT`

Required: No

Origin (p. 14)

The source of the CMK's key material.

The default is `AWS_KMS`, which means AWS KMS creates the key material. When this parameter is set to `EXTERNAL`, the request creates a CMK without key material so that you can import key material from your existing key management infrastructure. For more information about importing key material into AWS KMS, see Importing Key Material in the *AWS Key Management Service Developer Guide*.

The CMK's `Origin` is immutable and is set when the CMK is created.

Type: String

Valid Values: `AWS_KMS | EXTERNAL`

Required: No

Policy (p. 14)

The key policy to attach to the CMK.

If you specify a policy and do not set `BypassPolicyLockoutSafetyCheck` to true, the policy must meet the following criteria:

- It must allow the principal making the `CreateKey` request to make a subsequent PutKeyPolicy (p. 68) request on the CMK. This reduces the likelihood that the CMK becomes unmanageable. For more information, refer to the scenario in the Default Key Policy section in the *AWS Key Management Service Developer Guide*.
- The principal(s) specified in the key policy must exist and be visible to AWS KMS. When you create a new AWS principal (for example, an IAM user or role), you might need to enforce a delay before specifying the new principal in a key policy because the new principal might not immediately be visible to AWS KMS. For more information, see Changes that I make are not always immediately visible in the *IAM User Guide*.

If you do not specify a policy, AWS KMS attaches a default key policy to the CMK. For more information, see Default Key Policy in the *AWS Key Management Service Developer Guide*.

The policy size limit is 32 KiB (32768 bytes).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 131072.

Pattern: `[\u0009\u000A\u000D\u0020-\u00FF]+`

Required: No

Response Syntax

```
{
   "KeyMetadata": {
      "Arn": "string",
      "AWSAccountId": "string",
      "CreationDate": number,
      "DeletionDate": number,
      "Description": "string",
      "Enabled": boolean,
      "ExpirationModel": "string",
      "KeyId": "string",
      "KeyState": "string",
      "KeyUsage": "string",
      "Origin": "string",
      "ValidTo": number
   }
}
```

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.

KeyMetadata (p. 15)

Metadata associated with the CMK.

Type: KeyMetadata (p. 90) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

LimitExceededException

The request was rejected because a limit was exceeded. For more information, see Limits in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

MalformedPolicyDocumentException

The request was rejected because the specified policy is not syntactically or semantically correct.

HTTP Status Code: 400

UnsupportedOperationException

The request was rejected because a specified parameter is not supported or a specified resource is not valid for this operation.

HTTP Status Code: 400

Decrypt

Decrypts ciphertext. Ciphertext is plaintext that has been previously encrypted by using any of the following functions:

- GenerateDataKey (p. 37)
- GenerateDataKeyWithoutPlaintext (p. 40)
- Encrypt (p. 33)

Note that if a caller has been granted access permissions to all keys (through, for example, IAM user policies that grant `Decrypt` permission on all resources), then ciphertext encrypted by using keys in other accounts where the key grants access to the caller can be decrypted. To remedy this, we recommend that you do not grant `Decrypt` access in an IAM user policy. Instead grant `Decrypt` access only in key policies. If you must grant `Decrypt` access in an IAM user policy, you should scope the resource to specific keys or to specific trusted accounts.

Request Syntax

```
{
   "CiphertextBlob": blob,
   "EncryptionContext": {
      "string" : "string"
   },
   "GrantTokens": [ "string" ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

> **Note**
> In the following list, the required parameters are described first.

CiphertextBlob (p. 17)

Ciphertext to be decrypted. The blob includes metadata.

Type: Base64-encoded binary data

Length Constraints: Minimum length of 1. Maximum length of 6144.

Required: Yes

EncryptionContext (p. 17)

The encryption context. If this was specified in the Encrypt (p. 33) function, it must be specified here or the decryption operation will fail. For more information, see Encryption Context.

Type: String to String map

Required: No

GrantTokens (p. 17)

A list of grant tokens.

For more information, see Grant Tokens in the *AWS Key Management Service Developer Guide*.

Type: array of Strings

Array Members: Minimum number of 0 items. Maximum number of 10 items.

Length Constraints: Minimum length of 1. Maximum length of 8192.

Required: No

Response Syntax

```
{
   "KeyId": "string",
   "Plaintext": blob
}
```

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.

KeyId (p. 18)

ARN of the key used to perform the decryption. This value is returned if no errors are encountered during the operation.

Type: String

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

Plaintext (p. 18)

Decrypted plaintext data. This value may not be returned if the customer master key is not available or if you didn't have permission to use it.

Type: Base64-encoded binary data

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. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

DisabledException

The request was rejected because the specified CMK is not enabled.

HTTP Status Code: 400

InvalidCiphertextException

The request was rejected because the specified ciphertext has been corrupted or is otherwise invalid.

HTTP Status Code: 400

InvalidGrantTokenException

The request was rejected because the specified grant token is not valid.

HTTP Status Code: 400

KeyUnavailableException

The request was rejected because the specified CMK was not available. The request can be retried.

HTTP Status Code: 500

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

Examples

Sample Request

```
POST / HTTP/1.1
Host: kms.us-west-2.amazonaws.com
X-Amz-Target: TrentService.Decrypt
X-Amz-Date: 20160517T204035Z
Content-Type: application/x-amz-json-1.1
Authorization: AWS4-HMAC-SHA256\
 Credential=AKIAI44QH8DHBEXAMPLE/20160517/us-west-2/kms/aws4_request,\
 SignedHeaders=content-type;host;x-amz-date;x-amz-target,\
 Signature=545b0c3bfd9223b8ef7e6293ef3ccac37a83d415ee3112d2e5c70727d2a49c46

{"CiphertextBlob":
 "CiDPoCH188S65r5Cy7pAhIFJMXDlU7mewhSlYUpuQIVBrhKmAQEBAgB4z6Ah9fPEuua
+Qsu6QISBSTFw5VO5nsIUpWFKbkCFQa4AAAB9MHsGCSqGSIb3DQEHBqBuMGwCAQAwZwYJKoZIhvcNAQcBMB4GCWCGSA
ZjYCARCAOt8la8qXLO5wB3JH2NlwWWzWRU2RKqpO9A/0psE5UWwkK6CnwoeC3Zj9Q0A66apZkbRglFfY1lTY
+Tc="}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Tue, 17 May 2016 20:40:40 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 146
Connection: keep-alive
x-amzn-RequestId: 9e02f41f-1c6f-11e6-af63-ab8791945da7

{
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "Plaintext": "VGhpcyBpcyBEYXkgMSBmb3IgdGhlIEludGVybmV0Cg=="
}
```

DeleteAlias

Deletes the specified alias. To map an alias to a different key, call UpdateAlias (p. 80).

Request Syntax

```
{
   "AliasName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

Note
In the following list, the required parameters are described first.

AliasName (p. 20)

The alias to be deleted. The name must start with the word "alias" followed by a forward slash (alias/). Aliases that begin with "alias/AWS" are reserved.

Type: String

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

Pattern: `^[a-zA-Z0-9:/_-]+$`

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. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

DeleteImportedKeyMaterial

Deletes key material that you previously imported and makes the specified customer master key (CMK) unusable. For more information about importing key material into AWS KMS, see Importing Key Material in the *AWS Key Management Service Developer Guide*.

When the specified CMK is in the `PendingDeletion` state, this operation does not change the CMK's state. Otherwise, it changes the CMK's state to `PendingImport`.

After you delete key material, you can use ImportKeyMaterial (p. 52) to reimport the same key material into the CMK.

Request Syntax

```
{
   "KeyId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

Note
In the following list, the required parameters are described first.

KeyId (p. 21)

The identifier of the CMK whose key material to delete. The CMK's `Origin` must be `EXTERNAL`.

A valid identifier is the unique key ID or the Amazon Resource Name (ARN) of the CMK. Examples:

- Unique key ID: `1234abcd-12ab-34cd-56ef-1234567890ab`
- Key ARN: `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`

Type: String

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

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. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

UnsupportedOperationException

The request was rejected because a specified parameter is not supported or a specified resource is not valid for this operation.

HTTP Status Code: 400

DescribeKey

Provides detailed information about the specified customer master key.

Request Syntax

```
{
   "GrantTokens": [ "string" ],
   "KeyId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

Note
In the following list, the required parameters are described first.

KeyId (p. 23)

A unique identifier for the customer master key. This value can be a globally unique identifier, a fully specified ARN to either an alias or a key, or an alias name prefixed by "alias/".

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Alias ARN Example - arn:aws:kms:us-east-1:123456789012:alias/MyAliasName
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012
- Alias Name Example - alias/MyAliasName

Type: String

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

Required: Yes

GrantTokens (p. 23)

A list of grant tokens.

For more information, see Grant Tokens in the *AWS Key Management Service Developer Guide*.

Type: array of Strings

Array Members: Minimum number of 0 items. Maximum number of 10 items.

Length Constraints: Minimum length of 1. Maximum length of 8192.

Required: No

Response Syntax

```
{
   "KeyMetadata": {
      "Arn": "string",
      "AWSAccountId": "string",
      "CreationDate": number,
      "DeletionDate": number,
      "Description": "string",
      "Enabled": boolean,
      "ExpirationModel": "string",
      "KeyId": "string",
      "KeyState": "string",
      "KeyUsage": "string",
      "Origin": "string",
      "ValidTo": number
   }
}
```

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.

KeyMetadata (p. 23)

Metadata associated with the key.

Type: KeyMetadata (p. 90) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

DisableKey

Sets the state of a customer master key (CMK) to disabled, thereby preventing its use for cryptographic operations. For more information about how key state affects the use of a CMK, see How Key State Affects the Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

Request Syntax

```
{
   "KeyId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

> **Note**
> In the following list, the required parameters are described first.

KeyId (p. 25)

A unique identifier for the CMK.

Use the CMK's unique identifier or its Amazon Resource Name (ARN). For example:

- Unique ID: 1234abcd-12ab-34cd-56ef-1234567890ab
- ARN: arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab

Type: String

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

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. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

DisableKeyRotation

Disables rotation of the specified key.

Request Syntax

```
{
   "KeyId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

> **Note**
> In the following list, the required parameters are described first.

KeyId (p. 27)

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Type: String

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

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. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

DisabledException

The request was rejected because the specified CMK is not enabled.

HTTP Status Code: 400

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

UnsupportedOperationException

The request was rejected because a specified parameter is not supported or a specified resource is not valid for this operation.

HTTP Status Code: 400

EnableKey

Marks a key as enabled, thereby permitting its use.

Request Syntax

```
{
   "KeyId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

> **Note**
> In the following list, the required parameters are described first.

KeyId (p. 29)

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Type: String

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

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. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

LimitExceededException

The request was rejected because a limit was exceeded. For more information, see Limits in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

EnableKeyRotation

Enables rotation of the specified customer master key.

Request Syntax

```
{
   "KeyId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

> **Note**
> In the following list, the required parameters are described first.

KeyId (p. 31)

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Type: String

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

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. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

DisabledException

The request was rejected because the specified CMK is not enabled.

HTTP Status Code: 400

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

UnsupportedOperationException

The request was rejected because a specified parameter is not supported or a specified resource is not valid for this operation.

HTTP Status Code: 400

Encrypt

Encrypts plaintext into ciphertext by using a customer master key. The `Encrypt` function has two primary use cases:

- You can encrypt up to 4 KB of arbitrary data such as an RSA key, a database password, or other sensitive customer information.
- If you are moving encrypted data from one region to another, you can use this API to encrypt in the new region the plaintext data key that was used to encrypt the data in the original region. This provides you with an encrypted copy of the data key that can be decrypted in the new region and used there to decrypt the encrypted data.

Unless you are moving encrypted data from one region to another, you don't use this function to encrypt a generated data key within a region. You retrieve data keys already encrypted by calling the GenerateDataKey (p. 37) or GenerateDataKeyWithoutPlaintext (p. 40) function. Data keys don't need to be encrypted again by calling `Encrypt`.

If you want to encrypt data locally in your application, you can use the `GenerateDataKey` function to return a plaintext data encryption key and a copy of the key encrypted under the customer master key (CMK) of your choosing.

Request Syntax

```
{
   "EncryptionContext": {
      "string" : "string"
   },
   "GrantTokens": [ "string" ],
   "KeyId": "string",
   "Plaintext": blob
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

Note
In the following list, the required parameters are described first.

KeyId (p. 33)

A unique identifier for the customer master key. This value can be a globally unique identifier, a fully specified ARN to either an alias or a key, or an alias name prefixed by "alias/".

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Alias ARN Example - arn:aws:kms:us-east-1:123456789012:alias/MyAliasName
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012
- Alias Name Example - alias/MyAliasName

Type: String

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

Required: Yes

Plaintext (p. 33)

Data to be encrypted.

Type: Base64-encoded binary data

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

Required: Yes

EncryptionContext (p. 33)

Name-value pair that specifies the encryption context to be used for authenticated encryption. If used here, the same value must be supplied to the `Decrypt` API or decryption will fail. For more information, see Encryption Context.

Type: String to String map

Required: No

GrantTokens (p. 33)

A list of grant tokens.

For more information, see Grant Tokens in the *AWS Key Management Service Developer Guide*.

Type: array of Strings

Array Members: Minimum number of 0 items. Maximum number of 10 items.

Length Constraints: Minimum length of 1. Maximum length of 8192.

Required: No

Response Syntax

```
{
   "CiphertextBlob": blob,
   "KeyId": "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.

CiphertextBlob (p. 34)

The encrypted plaintext. If you are using the CLI, the value is Base64 encoded. Otherwise, it is not encoded.

Type: Base64-encoded binary data

Length Constraints: Minimum length of 1. Maximum length of 6144.

KeyId (p. 34)

The ID of the key used during encryption.

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. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

DisabledException

The request was rejected because the specified CMK is not enabled.

HTTP Status Code: 400

InvalidGrantTokenException

The request was rejected because the specified grant token is not valid.

HTTP Status Code: 400

InvalidKeyUsageException

The request was rejected because the specified `KeySpec` value is not valid.

HTTP Status Code: 400

KeyUnavailableException

The request was rejected because the specified CMK was not available. The request can be retried.

HTTP Status Code: 500

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

Examples

Sample Request

```
POST / HTTP/1.1
Host: kms.us-west-2.amazonaws.com
X-Amz-Target: TrentService.Encrypt
X-Amz-Date: 20160517T203825Z
Content-Type: application/x-amz-json-1.1
Authorization: AWS4-HMAC-SHA256\
 Credential=AKIAI44QH8DHBEXAMPLE/20160517/us-west-2/kms/aws4_request,\
 SignedHeaders=content-type;host;x-amz-date;x-amz-target,\
 Signature=67ccaa73c1af7fe83973ce8139104d55f3bdcebee323d2f2e65996d99015ace2

{
  "Plaintext": "VGhpcyBpcyBEYXkgMSBmb3IgdGhlIEludGVybmV0Cg==",
  "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Tue, 17 May 2016 20:38:30 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 379
Connection: keep-alive
x-amzn-RequestId: 50a0c603-1c6f-11e6-bb9e-3fadde80ce75
```

```
{
  "CiphertextBlob":
 "CiDPoCH188S65r5Cy7pAhIFJMXDlU7mewhSlYUpuQIVBrhKmAQEBAgB4z6Ah9fPEuua
+Qsu6QISBSTFw5VO5nsIUpWFKbkCFQa4AAAB9MHsGCSqGSIb3DQEHBqBuMGwCAQAwZwYJKoZIhvcNAQcBMB4GCWCGSA
ZjYCARCAOt8la8qXLO5wB3JH2NlwWWzWRU2RKqpO9A/0psE5UWwkK6CnwoeC3Zj9Q0A66apZkbRglFfY1lTY
+Tc=",
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

GenerateDataKey

Returns a data encryption key that you can use in your application to encrypt data locally.

You must specify the customer master key (CMK) under which to generate the data key. You must also specify the length of the data key using either the `KeySpec` or `NumberOfBytes` field. You must specify one field or the other, but not both. For common key lengths (128-bit and 256-bit symmetric keys), we recommend that you use `KeySpec`.

This operation returns a plaintext copy of the data key in the `Plaintext` field of the response, and an encrypted copy of the data key in the `CiphertextBlob` field. The data key is encrypted under the CMK specified in the `KeyId` field of the request.

We recommend that you use the following pattern to encrypt data locally in your application:

1. Use this operation (`GenerateDataKey`) to retrieve a data encryption key.
2. Use the plaintext data encryption key (returned in the `Plaintext` field of the response) to encrypt data locally, then erase the plaintext data key from memory.
3. Store the encrypted data key (returned in the `CiphertextBlob` field of the response) alongside the locally encrypted data.

To decrypt data locally:

1. Use the Decrypt (p. 17) operation to decrypt the encrypted data key into a plaintext copy of the data key.
2. Use the plaintext data key to decrypt data locally, then erase the plaintext data key from memory.

To return only an encrypted copy of the data key, use GenerateDataKeyWithoutPlaintext (p. 40). To return an arbitrary unpredictable byte string, use GenerateRandom (p. 43).

If you use the optional `EncryptionContext` field, you must store at least enough information to be able to reconstruct the full encryption context when you later send the ciphertext to the Decrypt (p. 17) operation. It is a good practice to choose an encryption context that you can reconstruct on the fly to better secure the ciphertext. For more information, see Encryption Context in the *AWS Key Management Service Developer Guide*.

Request Syntax

```
{
   "EncryptionContext": {
      "string" : "string"
   },
   "GrantTokens": [ "string" ],
   "KeyId": "string",
   "KeySpec": "string",
   "NumberOfBytes": number
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

> **Note**
> In the following list, the required parameters are described first.

KeyId (p. 37)

The identifier of the CMK under which to generate and encrypt the data encryption key.

A valid identifier is the unique key ID or the Amazon Resource Name (ARN) of the CMK, or the alias name or ARN of an alias that points to the CMK. Examples:

- Unique key ID: `1234abcd-12ab-34cd-56ef-1234567890ab`
- CMK ARN: `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`
- Alias name: `alias/ExampleAlias`
- Alias ARN: `arn:aws:kms:us-west-2:111122223333:alias/ExampleAlias`

Type: String

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

Required: Yes

EncryptionContext (p. 37)

A set of key-value pairs that represents additional authenticated data.

For more information, see Encryption Context in the *AWS Key Management Service Developer Guide*.

Type: String to String map

Required: No

GrantTokens (p. 37)

A list of grant tokens.

For more information, see Grant Tokens in the *AWS Key Management Service Developer Guide*.

Type: array of Strings

Array Members: Minimum number of 0 items. Maximum number of 10 items.

Length Constraints: Minimum length of 1. Maximum length of 8192.

Required: No

KeySpec (p. 37)

The length of the data encryption key. Use `AES_128` to generate a 128-bit symmetric key, or `AES_256` to generate a 256-bit symmetric key.

Type: String

Valid Values: `AES_256 | AES_128`

Required: No

NumberOfBytes (p. 37)

The length of the data encryption key in bytes. For example, use the value 64 to generate a 512-bit data key (64 bytes is 512 bits). For common key lengths (128-bit and 256-bit symmetric keys), we recommend that you use the `KeySpec` field instead of this one.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1024.

Required: No

Response Syntax

```
{
   "CiphertextBlob": blob,
   "KeyId": "string",
   "Plaintext": blob
}
```

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.

CiphertextBlob (p. 38)

The encrypted data encryption key.

Type: Base64-encoded binary data

Length Constraints: Minimum length of 1. Maximum length of 6144.

KeyId (p. 38)

The identifier of the CMK under which the data encryption key was generated and encrypted.

Type: String

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

Plaintext (p. 38)

The data encryption key. Use this data key for local encryption and decryption, then remove it from memory as soon as possible.

Type: Base64-encoded binary data

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. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

DisabledException

The request was rejected because the specified CMK is not enabled.

HTTP Status Code: 400

InvalidGrantTokenException

The request was rejected because the specified grant token is not valid.

HTTP Status Code: 400

InvalidKeyUsageException

The request was rejected because the specified `KeySpec` value is not valid.

HTTP Status Code: 400

KeyUnavailableException

The request was rejected because the specified CMK was not available. The request can be retried.

HTTP Status Code: 500

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

GenerateDataKeyWithoutPlaintext

Returns a data encryption key encrypted under a customer master key (CMK). This operation is identical to GenerateDataKey (p. 37) but returns only the encrypted copy of the data key.

This operation is useful in a system that has multiple components with different degrees of trust. For example, consider a system that stores encrypted data in containers. Each container stores the encrypted data and an encrypted copy of the data key. One component of the system, called the *control plane*, creates new containers. When it creates a new container, it uses this operation (`GenerateDataKeyWithoutPlaintext`) to get an encrypted data key and then stores it in the container. Later, a different component of the system, called the *data plane*, puts encrypted data into the containers. To do this, it passes the encrypted data key to the Decrypt (p. 17) operation, then uses the returned plaintext data key to encrypt data, and finally stores the encrypted data in the container. In this system, the control plane never sees the plaintext data key.

Request Syntax

```
{
   "EncryptionContext": { 
      "string" : "string" 
   },
   "GrantTokens": [ "string" ],
   "KeyId": "string",
   "KeySpec": "string",
   "NumberOfBytes": number
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

Note
In the following list, the required parameters are described first.

KeyId (p. 40)

The identifier of the CMK under which to generate and encrypt the data encryption key.

A valid identifier is the unique key ID or the Amazon Resource Name (ARN) of the CMK, or the alias name or ARN of an alias that points to the CMK. Examples:

- Unique key ID: `1234abcd-12ab-34cd-56ef-1234567890ab`
- CMK ARN: `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`
- Alias name: `alias/ExampleAlias`
- Alias ARN: `arn:aws:kms:us-west-2:111122223333:alias/ExampleAlias`

Type: String

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

Required: Yes

EncryptionContext (p. 40)

A set of key-value pairs that represents additional authenticated data.

For more information, see Encryption Context in the *AWS Key Management Service Developer Guide*.

Type: String to String map

Required: No

GrantTokens (p. 40)

A list of grant tokens.

For more information, see Grant Tokens in the *AWS Key Management Service Developer Guide*.

Type: array of Strings

Array Members: Minimum number of 0 items. Maximum number of 10 items.

Length Constraints: Minimum length of 1. Maximum length of 8192.

Required: No

KeySpec (p. 40)

The length of the data encryption key. Use `AES_128` to generate a 128-bit symmetric key, or `AES_256` to generate a 256-bit symmetric key.

Type: String

Valid Values: `AES_256 | AES_128`

Required: No

NumberOfBytes (p. 40)

The length of the data encryption key in bytes. For example, use the value 64 to generate a 512-bit data key (64 bytes is 512 bits). For common key lengths (128-bit and 256-bit symmetric keys), we recommend that you use the `KeySpec` field instead of this one.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1024.

Required: No

Response Syntax

```
{
   "CiphertextBlob": blob,
   "KeyId": "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.

CiphertextBlob (p. 41)

The encrypted data encryption key.

Type: Base64-encoded binary data

Length Constraints: Minimum length of 1. Maximum length of 6144.

KeyId (p. 41)

The identifier of the CMK under which the data encryption key was generated and encrypted.

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. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

DisabledException

The request was rejected because the specified CMK is not enabled.

HTTP Status Code: 400

InvalidGrantTokenException

The request was rejected because the specified grant token is not valid.

HTTP Status Code: 400

InvalidKeyUsageException

The request was rejected because the specified `KeySpec` value is not valid.

HTTP Status Code: 400

KeyUnavailableException

The request was rejected because the specified CMK was not available. The request can be retried.

HTTP Status Code: 500

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

GenerateRandom

Generates an unpredictable byte string.

Request Syntax

```
{
   "NumberOfBytes": number
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

> **Note**
> In the following list, the required parameters are described first.

NumberOfBytes (p. 43)

The length of the byte string.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1024.

Required: No

Response Syntax

```
{
   "Plaintext": blob
}
```

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.

Plaintext (p. 43)

The unpredictable byte string.

Type: Base64-encoded binary data

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. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

GetKeyPolicy

Retrieves a policy attached to the specified key.

Request Syntax

```
{
   "KeyId": "string",
   "PolicyName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

> **Note**
> In the following list, the required parameters are described first.

KeyId (p. 45)

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Type: String

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

Required: Yes

PolicyName (p. 45)

String that contains the name of the policy. Currently, this must be "default". Policy names can be discovered by calling ListKeyPolicies (p. 60).

Type: String

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

Pattern: `[\w]+`

Required: Yes

Response Syntax

```
{
   "Policy": "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.

Policy (p. 45)

A policy document in JSON format.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 131072.

Pattern: `[\u0009\u000A\u000D\u0020-\u00FF]+`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

GetKeyRotationStatus

Retrieves a Boolean value that indicates whether key rotation is enabled for the specified key.

Request Syntax

```
{
   "KeyId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

> **Note**
> In the following list, the required parameters are described first.

KeyId (p. 47)

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Type: String

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

Required: Yes

Response Syntax

```
{
   "KeyRotationEnabled": boolean
}
```

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.

KeyRotationEnabled (p. 47)

A Boolean value that specifies whether key rotation is enabled.

Type: Boolean

Errors

For information about the errors that are common to all actions, see Common Errors (p. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

UnsupportedOperationException

The request was rejected because a specified parameter is not supported or a specified resource is not valid for this operation.

HTTP Status Code: 400

GetParametersForImport

Returns the items you need in order to import key material into AWS KMS from your existing key management infrastructure. For more information about importing key material into AWS KMS, see Importing Key Material in the *AWS Key Management Service Developer Guide*.

You must specify the key ID of the customer master key (CMK) into which you will import key material. This CMK's `Origin` must be `EXTERNAL`. You must also specify the wrapping algorithm and type of wrapping key (public key) that you will use to encrypt the key material.

This operation returns a public key and an import token. Use the public key to encrypt the key material. Store the import token to send with a subsequent ImportKeyMaterial (p. 52) request. The public key and import token from the same response must be used together. These items are valid for 24 hours, after which they cannot be used for a subsequent ImportKeyMaterial (p. 52) request. To retrieve new ones, send another `GetParametersForImport` request.

Request Syntax

```
{
   "KeyId": "string",
   "WrappingAlgorithm": "string",
   "WrappingKeySpec": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

> **Note**
> In the following list, the required parameters are described first.

KeyId (p. 49)

The identifier of the CMK into which you will import key material. The CMK's `Origin` must be `EXTERNAL`.

A valid identifier is the unique key ID or the Amazon Resource Name (ARN) of the CMK. Examples:

- Unique key ID: `1234abcd-12ab-34cd-56ef-1234567890ab`
- Key ARN: `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`

Type: String

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

Required: Yes

WrappingAlgorithm (p. 49)

The algorithm you will use to encrypt the key material before importing it with ImportKeyMaterial (p. 52). For more information, see Encrypt the Key Material in the *AWS Key Management Service Developer Guide*.

Type: String

Valid Values: `RSAES_PKCS1_V1_5 | RSAES_OAEP_SHA_1 | RSAES_OAEP_SHA_256`

Required: Yes

WrappingKeySpec (p. 49)

The type of wrapping key (public key) to return in the response. Only 2048-bit RSA public keys are supported.

Type: String

Valid Values: `RSA_2048`

Required: Yes

Response Syntax

```
{
   "ImportToken": blob,
   "KeyId": "string",
   "ParametersValidTo": number,
   "PublicKey": blob
}
```

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.

ImportToken (p. 50)

The import token to send in a subsequent ImportKeyMaterial (p. 52) request.

Type: Base64-encoded binary data

Length Constraints: Minimum length of 1. Maximum length of 6144.

KeyId (p. 50)

The identifier of the CMK to use in a subsequent ImportKeyMaterial (p. 52) request. This is the same CMK specified in the `GetParametersForImport` request.

Type: String

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

ParametersValidTo (p. 50)

The time at which the import token and public key are no longer valid. After this time, you cannot use them to make an ImportKeyMaterial (p. 52) request and you must send another `GetParametersForImport` request to retrieve new ones.

Type: Timestamp

PublicKey (p. 50)

The public key to use to encrypt the key material before importing it with ImportKeyMaterial (p. 52).

Type: Base64-encoded binary data

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. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

UnsupportedOperationException

The request was rejected because a specified parameter is not supported or a specified resource is not valid for this operation.

HTTP Status Code: 400

ImportKeyMaterial

Imports key material into an AWS KMS customer master key (CMK) from your existing key management infrastructure. For more information about importing key material into AWS KMS, see Importing Key Material in the *AWS Key Management Service Developer Guide*.

You must specify the key ID of the CMK to import the key material into. This CMK's `Origin` must be `EXTERNAL`. You must also send an import token and the encrypted key material. Send the import token that you received in the same GetParametersForImport (p. 49) response that contained the public key that you used to encrypt the key material. You must also specify whether the key material expires and if so, when. When the key material expires, AWS KMS deletes the key material and the CMK becomes unusable. To use the CMK again, you can reimport the same key material. If you set an expiration date, you can change it only by reimporting the same key material and specifying a new expiration date.

When this operation is successful, the specified CMK's key state changes to `Enabled`, and you can use the CMK.

After you successfully import key material into a CMK, you can reimport the same key material into that CMK, but you cannot import different key material.

Request Syntax

```
{
   "EncryptedKeyMaterial": blob,
   "ExpirationModel": "string",
   "ImportToken": blob,
   "KeyId": "string",
   "ValidTo": number
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

> **Note**
> In the following list, the required parameters are described first.

EncryptedKeyMaterial (p. 52)

The encrypted key material to import. It must be encrypted with the public key that you received in the response to a previous GetParametersForImport (p. 49) request, using the wrapping algorithm that you specified in that request.

Type: Base64-encoded binary data

Length Constraints: Minimum length of 1. Maximum length of 6144.

Required: Yes

ImportToken (p. 52)

The import token that you received in the response to a previous GetParametersForImport (p. 49) request. It must be from the same response that contained the public key that you used to encrypt the key material.

Type: Base64-encoded binary data

Length Constraints: Minimum length of 1. Maximum length of 6144.

Required: Yes

KeyId (p. 52)

The identifier of the CMK to import the key material into. The CMK's `Origin` must be `EXTERNAL`.

A valid identifier is the unique key ID or the Amazon Resource Name (ARN) of the CMK. Examples:

- Unique key ID: `1234abcd-12ab-34cd-56ef-1234567890ab`
- Key ARN: `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`

Type: String

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

Required: Yes

ExpirationModel (p. 52)

Specifies whether the key material expires. The default is `KEY_MATERIAL_EXPIRES`, in which case you must include the `ValidTo` parameter. When this parameter is set to `KEY_MATERIAL_DOES_NOT_EXPIRE`, you must omit the `ValidTo` parameter.

Type: String

Valid Values: `KEY_MATERIAL_EXPIRES | KEY_MATERIAL_DOES_NOT_EXPIRE`

Required: No

ValidTo (p. 52)

The time at which the imported key material expires. When the key material expires, AWS KMS deletes the key material and the CMK becomes unusable. You must omit this parameter when the `ExpirationModel` parameter is set to `KEY_MATERIAL_DOES_NOT_EXPIRE`. Otherwise it is required.

Type: Timestamp

Required: No

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. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

ExpiredImportTokenException

The request was rejected because the provided import token is expired. Use GetParametersForImport (p. 49) to retrieve a new import token and public key, use the new public key to encrypt the key material, and then try the request again.

HTTP Status Code: 400

IncorrectKeyMaterialException

The request was rejected because the provided key material is invalid or is not the same key material that was previously imported into this customer master key (CMK).

HTTP Status Code: 400

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

InvalidCiphertextException

The request was rejected because the specified ciphertext has been corrupted or is otherwise invalid.

HTTP Status Code: 400

InvalidImportTokenException

The request was rejected because the provided import token is invalid or is associated with a different customer master key (CMK).

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

UnsupportedOperationException

The request was rejected because a specified parameter is not supported or a specified resource is not valid for this operation.

HTTP Status Code: 400

ListAliases

Lists all of the key aliases in the account.

Request Syntax

```
{
   "Limit": number,
   "Marker": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

> **Note**
> In the following list, the required parameters are described first.

Limit (p. 55)

When paginating results, specify the maximum number of items to return in the response. If additional items exist beyond the number you specify, the `Truncated` element in the response is set to true.

This value is optional. If you include a value, it must be between 1 and 100, inclusive. If you do not include a value, it defaults to 50.

Type: Integer

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

Required: No

Marker (p. 55)

Use this parameter only when paginating results and only in a subsequent request after you receive a response with truncated results. Set it to the value of `NextMarker` from the response you just received.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 320.

Pattern: `[\u0020-\u00FF]*`

Required: No

Response Syntax

```
{
   "Aliases": [
      {
         "AliasArn": "string",
         "AliasName": "string",
         "TargetKeyId": "string"
      }
   ],
   "NextMarker": "string",
   "Truncated": boolean
}
```

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.

Aliases (p. 55)

A list of key aliases in the user's account.

Type: array of AliasListEntry (p. 85) objects

NextMarker (p. 55)

When `Truncated` is true, this value is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 320.

Pattern: `[\u0020-\u00FF]*`

Truncated (p. 55)

A flag that indicates whether there are more items in the list. If your results were truncated, you can use the `Marker` parameter to make a subsequent pagination request to retrieve more items in the list.

Type: Boolean

Errors

For information about the errors that are common to all actions, see Common Errors (p. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

InvalidMarkerException

The request was rejected because the marker that specifies where pagination should next begin is not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

ListGrants

List the grants for a specified key.

Request Syntax

```
{
   "KeyId": "string",
   "Limit": number,
   "Marker": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

> **Note**
> In the following list, the required parameters are described first.

KeyId (p. 57)

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Type: String

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

Required: Yes

Limit (p. 57)

When paginating results, specify the maximum number of items to return in the response. If additional items exist beyond the number you specify, the `Truncated` element in the response is set to true.

This value is optional. If you include a value, it must be between 1 and 100, inclusive. If you do not include a value, it defaults to 50.

Type: Integer

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

Required: No

Marker (p. 57)

Use this parameter only when paginating results and only in a subsequent request after you receive a response with truncated results. Set it to the value of `NextMarker` from the response you just received.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 320.

Pattern: `[\u0020-\u00FF]*`

Required: No

Response Syntax

```
{
   "Grants": [
      {
         "Constraints": {
            "EncryptionContextEquals": {
               "string" : "string"
            },
            "EncryptionContextSubset": {
               "string" : "string"
            }
         },
         "CreationDate": number,
         "GranteePrincipal": "string",
         "GrantId": "string",
         "IssuingAccount": "string",
         "KeyId": "string",
         "Name": "string",
         "Operations": [ "string" ],
         "RetiringPrincipal": "string"
      }
   ],
   "NextMarker": "string",
   "Truncated": boolean
}
```

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.

Grants (p. 57)

A list of grants.

Type: array of GrantListEntry (p. 87) objects

NextMarker (p. 57)

When `Truncated` is true, this value is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 320.

Pattern: `[\u0020-\u00FF]*`

Truncated (p. 57)

A flag that indicates whether there are more items in the list. If your results were truncated, you can use the `Marker` parameter to make a subsequent pagination request to retrieve more items in the list.

Type: Boolean

Errors

For information about the errors that are common to all actions, see Common Errors (p. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

InvalidMarkerException

The request was rejected because the marker that specifies where pagination should next begin is not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

ListKeyPolicies

Retrieves a list of policies attached to a key.

Request Syntax

```
{
   "KeyId": "string",
   "Limit": number,
   "Marker": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

Note
In the following list, the required parameters are described first.

KeyId (p. 60)

A unique identifier for the customer master key. This value can be a globally unique identifier, a fully specified ARN to either an alias or a key, or an alias name prefixed by "alias/".

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Alias ARN Example - arn:aws:kms:us-east-1:123456789012:alias/MyAliasName
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012
- Alias Name Example - alias/MyAliasName

Type: String

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

Required: Yes

Limit (p. 60)

When paginating results, specify the maximum number of items to return in the response. If additional items exist beyond the number you specify, the `Truncated` element in the response is set to true.

This value is optional. If you include a value, it must be between 1 and 1000, inclusive. If you do not include a value, it defaults to 100.

Currently only 1 policy can be attached to a key.

Type: Integer

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

Required: No

Marker (p. 60)

Use this parameter only when paginating results and only in a subsequent request after you receive a response with truncated results. Set it to the value of `NextMarker` from the response you just received.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 320.

Pattern: `[\u0020-\u00FF]*`

Required: No

Response Syntax

```
{
   "NextMarker": "string",
   "PolicyNames": [ "string" ],
   "Truncated": boolean
}
```

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.

NextMarker (p. 61)

When `Truncated` is true, this value is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 320.

Pattern: `[\u0020-\u00FF]*`

PolicyNames (p. 61)

A list of policy names. Currently, there is only one policy and it is named "Default".

Type: array of Strings

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

Pattern: `[\w]+`

Truncated (p. 61)

A flag that indicates whether there are more items in the list. If your results were truncated, you can use the `Marker` parameter to make a subsequent pagination request to retrieve more items in the list.

Type: Boolean

Errors

For information about the errors that are common to all actions, see Common Errors (p. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

ListKeys

Lists the customer master keys.

Request Syntax

```
{
   "Limit": number,
   "Marker": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

> **Note**
> In the following list, the required parameters are described first.

Limit (p. 63)

When paginating results, specify the maximum number of items to return in the response. If additional items exist beyond the number you specify, the `Truncated` element in the response is set to true.

This value is optional. If you include a value, it must be between 1 and 1000, inclusive. If you do not include a value, it defaults to 100.

Type: Integer

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

Required: No

Marker (p. 63)

Use this parameter only when paginating results and only in a subsequent request after you receive a response with truncated results. Set it to the value of `NextMarker` from the response you just received.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 320.

Pattern: `[\u0020-\u00FF]*`

Required: No

Response Syntax

```
{
   "Keys": [
      {
         "KeyArn": "string",
         "KeyId": "string"
      }
   ],
   "NextMarker": "string",
   "Truncated": boolean
}
```

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.

Keys (p. 63)

A list of keys.

Type: array of KeyListEntry (p. 89) objects

NextMarker (p. 63)

When `Truncated` is true, this value is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 320.

Pattern: `[\u0020-\u00FF]*`

Truncated (p. 63)

A flag that indicates whether there are more items in the list. If your results were truncated, you can use the `Marker` parameter to make a subsequent pagination request to retrieve more items in the list.

Type: Boolean

Errors

For information about the errors that are common to all actions, see Common Errors (p. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

InvalidMarkerException

The request was rejected because the marker that specifies where pagination should next begin is not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

ListRetirableGrants

Returns a list of all grants for which the grant's `RetiringPrincipal` matches the one specified.

A typical use is to list all grants that you are able to retire. To retire a grant, use RetireGrant (p. 74).

Request Syntax

```
{
   "Limit": number,
   "Marker": "string",
   "RetiringPrincipal": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

> **Note**
> In the following list, the required parameters are described first.

RetiringPrincipal (p. 65)

The retiring principal for which to list grants.

To specify the retiring principal, use the Amazon Resource Name (ARN) of an AWS principal. Valid AWS principals include AWS accounts (root), IAM users, federated users, and assumed role users. For examples of the ARN syntax for specifying a principal, see AWS Identity and Access Management (IAM) in the Example ARNs section of the *Amazon Web Services General Reference*.

Type: String

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

Required: Yes

Limit (p. 65)

When paginating results, specify the maximum number of items to return in the response. If additional items exist beyond the number you specify, the `Truncated` element in the response is set to true.

This value is optional. If you include a value, it must be between 1 and 100, inclusive. If you do not include a value, it defaults to 50.

Type: Integer

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

Required: No

Marker (p. 65)

Use this parameter only when paginating results and only in a subsequent request after you receive a response with truncated results. Set it to the value of `NextMarker` from the response you just received.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 320.

Pattern: `[\u0020-\u00FF]*`

Required: No

Response Syntax

```
{
   "Grants": [
      {
         "Constraints": {
            "EncryptionContextEquals": {
               "string" : "string"
            },
            "EncryptionContextSubset": {
               "string" : "string"
            }
         },
         "CreationDate": number,
         "GranteePrincipal": "string",
         "GrantId": "string",
         "IssuingAccount": "string",
         "KeyId": "string",
         "Name": "string",
         "Operations": [ "string" ],
         "RetiringPrincipal": "string"
      }
   ],
   "NextMarker": "string",
   "Truncated": boolean
}
```

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.

Grants (p. 66)

A list of grants.

Type: array of GrantListEntry (p. 87) objects

NextMarker (p. 66)

When `Truncated` is true, this value is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 320.

Pattern: `[\u0020-\u00FF]*`

Truncated (p. 66)

A flag that indicates whether there are more items in the list. If your results were truncated, you can use the `Marker` parameter to make a subsequent pagination request to retrieve more items in the list.

Type: Boolean

Errors

For information about the errors that are common to all actions, see Common Errors (p. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

InvalidMarkerException

The request was rejected because the marker that specifies where pagination should next begin is not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

PutKeyPolicy

Attaches a key policy to the specified customer master key (CMK).

For more information about key policies, see Key Policies in the *AWS Key Management Service Developer Guide*.

Request Syntax

```
{
   "BypassPolicyLockoutSafetyCheck": boolean,
   "KeyId": "string",
   "Policy": "string",
   "PolicyName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

> **Note**
> In the following list, the required parameters are described first.

KeyId (p. 68)

A unique identifier for the CMK.

Use the CMK's unique identifier or its Amazon Resource Name (ARN). For example:

- Unique ID: 1234abcd-12ab-34cd-56ef-1234567890ab
- ARN: arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab

Type: String

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

Required: Yes

Policy (p. 68)

The key policy to attach to the CMK.

If you do not set `BypassPolicyLockoutSafetyCheck` to true, the policy must meet the following criteria:

- It must allow the principal making the `PutKeyPolicy` request to make a subsequent `PutKeyPolicy` request on the CMK. This reduces the likelihood that the CMK becomes unmanageable. For more information, refer to the scenario in the Default Key Policy section in the *AWS Key Management Service Developer Guide*.
- The principal(s) specified in the key policy must exist and be visible to AWS KMS. When you create a new AWS principal (for example, an IAM user or role), you might need to enforce a delay before specifying the new principal in a key policy because the new principal might not immediately be visible to AWS KMS. For more information, see Changes that I make are not always immediately visible in the *IAM User Guide*.

The policy size limit is 32 KiB (32768 bytes).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 131072.

Pattern: `[\u0009\u000A\u000D\u0020-\u00FF]+`

Required: Yes

PolicyName (p. 68)

The name of the key policy.

This value must be `default`.

Type: String

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

Pattern: `[\w]+`

Required: Yes

BypassPolicyLockoutSafetyCheck (p. 68)

A flag to indicate whether to bypass the key policy lockout safety check.

> **Important**
> Setting this value to true increases the likelihood that the CMK becomes unmanageable. Do not set this value to true indiscriminately.
> For more information, refer to the scenario in the Default Key Policy section in the *AWS Key Management Service Developer Guide*.

Use this parameter only when you intend to prevent the principal making the request from making a subsequent `PutKeyPolicy` request on the CMK.

The default value is false.

Type: Boolean

Required: No

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. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

LimitExceededException

The request was rejected because a limit was exceeded. For more information, see Limits in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

MalformedPolicyDocumentException

The request was rejected because the specified policy is not syntactically or semantically correct.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

UnsupportedOperationException

The request was rejected because a specified parameter is not supported or a specified resource is not valid for this operation.

HTTP Status Code: 400

ReEncrypt

Encrypts data on the server side with a new customer master key without exposing the plaintext of the data on the client side. The data is first decrypted and then encrypted. This operation can also be used to change the encryption context of a ciphertext.

Unlike other actions, `ReEncrypt` is authorized twice - once as `ReEncryptFrom` on the source key and once as `ReEncryptTo` on the destination key. We therefore recommend that you include the `"action":"kms:ReEncrypt*"` statement in your key policies to permit re-encryption from or to the key. The statement is included automatically when you authorize use of the key through the console but must be included manually when you set a policy by using the PutKeyPolicy (p. 68) function.

Request Syntax

```
{
   "CiphertextBlob": blob,
   "DestinationEncryptionContext": {
      "string" : "string"
   },
   "DestinationKeyId": "string",
   "GrantTokens": [ "string" ],
   "SourceEncryptionContext": {
      "string" : "string"
   }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

Note
In the following list, the required parameters are described first.

CiphertextBlob (p. 71)

Ciphertext of the data to re-encrypt.

Type: Base64-encoded binary data

Length Constraints: Minimum length of 1. Maximum length of 6144.

Required: Yes

DestinationKeyId (p. 71)

A unique identifier for the customer master key used to re-encrypt the data. This value can be a globally unique identifier, a fully specified ARN to either an alias or a key, or an alias name prefixed by "alias/".

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Alias ARN Example - arn:aws:kms:us-east-1:123456789012:alias/MyAliasName
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012
- Alias Name Example - alias/MyAliasName

Type: String

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

Required: Yes

DestinationEncryptionContext (p. 71)

Encryption context to be used when the data is re-encrypted.

Type: String to String map

Required: No

GrantTokens (p. 71)

A list of grant tokens.

For more information, see Grant Tokens in the *AWS Key Management Service Developer Guide*.

Type: array of Strings

Array Members: Minimum number of 0 items. Maximum number of 10 items.

Length Constraints: Minimum length of 1. Maximum length of 8192.

Required: No

SourceEncryptionContext (p. 71)

Encryption context used to encrypt and decrypt the data specified in the `CiphertextBlob` parameter.

Type: String to String map

Required: No

Response Syntax

```
{
   "CiphertextBlob": blob,
   "KeyId": "string",
   "SourceKeyId": "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.

CiphertextBlob (p. 72)

The re-encrypted data. If you are using the CLI, the value is Base64 encoded. Otherwise, it is not encoded.

Type: Base64-encoded binary data

Length Constraints: Minimum length of 1. Maximum length of 6144.

KeyId (p. 72)

Unique identifier of the key used to re-encrypt the data.

Type: String

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

SourceKeyId (p. 72)

Unique identifier of the key used to originally encrypt the data.

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. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

DisabledException

The request was rejected because the specified CMK is not enabled.

HTTP Status Code: 400

InvalidCiphertextException

The request was rejected because the specified ciphertext has been corrupted or is otherwise invalid.

HTTP Status Code: 400

InvalidGrantTokenException

The request was rejected because the specified grant token is not valid.

HTTP Status Code: 400

InvalidKeyUsageException

The request was rejected because the specified `KeySpec` value is not valid.

HTTP Status Code: 400

KeyUnavailableException

The request was rejected because the specified CMK was not available. The request can be retried.

HTTP Status Code: 500

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

RetireGrant

Retires a grant. You can retire a grant when you're done using it to clean up. You should revoke a grant when you intend to actively deny operations that depend on it. The following are permitted to call this API:

- The account that created the grant
- The `RetiringPrincipal`, if present
- The `GranteePrincipal`, if `RetireGrant` is a grantee operation

The grant to retire must be identified by its grant token or by a combination of the key ARN and the grant ID. A grant token is a unique variable-length base64-encoded string. A grant ID is a 64 character unique identifier of a grant. Both are returned by the `CreateGrant` function.

Request Syntax

```
{
   "GrantId": "string",
   "GrantToken": "string",
   "KeyId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

> **Note**
> In the following list, the required parameters are described first.

GrantId (p. 74)

Unique identifier of the grant to be retired. The grant ID is returned by the `CreateGrant` function.

- Grant ID Example - 0123456789012345678901234567890123456789012345678901234567890123

Type: String

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

Required: No

GrantToken (p. 74)

Token that identifies the grant to be retired.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 8192.

Required: No

KeyId (p. 74)

A unique identifier for the customer master key associated with the grant. This value can be a globally unique identifier or a fully specified ARN of the key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Type: String

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

Required: No

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. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

InvalidGrantIdException

The request was rejected because the specified `GrantId` is not valid.

HTTP Status Code: 400

InvalidGrantTokenException

The request was rejected because the specified grant token is not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

RevokeGrant

Revokes a grant. You can revoke a grant to actively deny operations that depend on it.

Request Syntax

```
{
   "GrantId": "string",
   "KeyId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

GrantId (p. 76)

Identifier of the grant to be revoked.

Type: String

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

Required: Yes

KeyId (p. 76)

A unique identifier for the customer master key associated with the grant. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Type: String

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

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. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

InvalidGrantIdException

The request was rejected because the specified `GrantId` is not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

ScheduleKeyDeletion

Schedules the deletion of a customer master key (CMK). You may provide a waiting period, specified in days, before deletion occurs. If you do not provide a waiting period, the default period of 30 days is used. When this operation is successful, the state of the CMK changes to `PendingDeletion`. Before the waiting period ends, you can use CancelKeyDeletion (p. 5) to cancel the deletion of the CMK. After the waiting period ends, AWS KMS deletes the CMK and all AWS KMS data associated with it, including all aliases that point to it.

> **Important**
> Deleting a CMK is a destructive and potentially dangerous operation. When a CMK is deleted, all data that was encrypted under the CMK is rendered unrecoverable. To restrict the use of a CMK without deleting it, use DisableKey (p. 25).

For more information about scheduling a CMK for deletion, see Deleting Customer Master Keys in the *AWS Key Management Service Developer Guide*.

Request Syntax

```
{
   "KeyId": "string",
   "PendingWindowInDays": number
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

> **Note**
> In the following list, the required parameters are described first.

KeyId (p. 78)

The unique identifier for the customer master key (CMK) to delete.

To specify this value, use the unique key ID or the Amazon Resource Name (ARN) of the CMK. Examples:

- Unique key ID: 1234abcd-12ab-34cd-56ef-1234567890ab
- Key ARN: arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab

To obtain the unique key ID and key ARN for a given CMK, use ListKeys (p. 63) or DescribeKey (p. 23).

Type: String

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

Required: Yes

PendingWindowInDays (p. 78)

The waiting period, specified in number of days. After the waiting period ends, AWS KMS deletes the customer master key (CMK).

This value is optional. If you include a value, it must be between 7 and 30, inclusive. If you do not include a value, it defaults to 30.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 365.

Required: No

Response Syntax

```
{
   "DeletionDate": number,
   "KeyId": "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.

DeletionDate (p. 79)

The date and time after which AWS KMS deletes the customer master key (CMK).

Type: Timestamp

KeyId (p. 79)

The unique identifier of the customer master key (CMK) for which deletion is scheduled.

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. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

UpdateAlias

Updates an alias to map it to a different key.

An alias is not a property of a key. Therefore, an alias can be mapped to and unmapped from an existing key without changing the properties of the key.

An alias name can contain only alphanumeric characters, forward slashes (/), underscores (_), and dashes (-). An alias must start with the word "alias" followed by a forward slash (alias/). An alias that begins with "aws" after the forward slash (alias/aws...) is reserved by Amazon Web Services (AWS).

The alias and the key it is mapped to must be in the same AWS account and the same region.

Request Syntax

```
{
   "AliasName": "string",
   "TargetKeyId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

> **Note**
> In the following list, the required parameters are described first.

AliasName (p. 80)

String that contains the name of the alias to be modified. The name must start with the word "alias" followed by a forward slash (alias/). Aliases that begin with "alias/aws" are reserved.

Type: String

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

Pattern: `^[a-zA-Z0-9:/_-]+$`

Required: Yes

TargetKeyId (p. 80)

Unique identifier of the customer master key to be mapped to the alias. This value can be a globally unique identifier or the fully specified ARN of a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

You can call ListAliases (p. 55) to verify that the alias is mapped to the correct `TargetKeyId`.

Type: String

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

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. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

UpdateKeyDescription

Updates the description of a key.

Request Syntax

```
{
   "Description": "string",
   "KeyId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 92).

The request accepts the following data in JSON format.

Note
In the following list, the required parameters are described first.

Description (p. 82)

New description for the key.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 8192.

Required: Yes

KeyId (p. 82)

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Type: String

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

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. 94).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 500

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource is not valid for this request.

For more information about how key state affects the use of a CMK, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

Data Types

The AWS Key Management Service 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:

- AliasListEntry (p. 85)
- GrantConstraints (p. 86)
- GrantListEntry (p. 87)
- KeyListEntry (p. 89)
- KeyMetadata (p. 90)

AliasListEntry

Contains information about an alias.

Contents

Note
In the following list, the required parameters are described first.

AliasArn

String that contains the key ARN.

Type: String

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

Required: No

AliasName

String that contains the alias.

Type: String

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

Pattern: `^[a-zA-Z0-9:/_-]+$`

Required: No

TargetKeyId

String that contains the key identifier pointed to by the alias.

Type: String

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

Required: No

GrantConstraints

A structure for specifying the conditions under which the operations permitted by the grant are allowed.

You can use this structure to allow the operations permitted by the grant only when a specified encryption context is present. For more information about encryption context, see Encryption Context in the *AWS Key Management Service Developer Guide*.

Contents

Note
In the following list, the required parameters are described first.

EncryptionContextEquals

Contains a list of key-value pairs that must be present in the encryption context of a subsequent operation permitted by the grant. When a subsequent operation permitted by the grant includes an encryption context that matches this list, the grant allows the operation. Otherwise, the operation is not allowed.

Type: String to String map

Required: No

EncryptionContextSubset

Contains a list of key-value pairs, a subset of which must be present in the encryption context of a subsequent operation permitted by the grant. When a subsequent operation permitted by the grant includes an encryption context that matches this list or is a subset of this list, the grant allows the operation. Otherwise, the operation is not allowed.

Type: String to String map

Required: No

GrantListEntry

Contains information about an entry in a list of grants.

Contents

Note
In the following list, the required parameters are described first.

Constraints

The conditions under which the grant's operations are allowed.

Type: GrantConstraints (p. 86) object

Required: No

CreationDate

The date and time when the grant was created.

Type: Timestamp

Required: No

GranteePrincipal

The principal that receives the grant's permissions.

Type: String

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

Required: No

GrantId

The unique identifier for the grant.

Type: String

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

Required: No

IssuingAccount

The AWS account under which the grant was issued.

Type: String

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

Required: No

KeyId

The unique identifier for the customer master key (CMK) to which the grant applies.

Type: String

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

Required: No

Name

The friendly name that identifies the grant. If a name was provided in the CreateGrant (p. 10) request, that name is returned. Otherwise this value is null.

Type: String

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

Pattern: `^[a-zA-Z0-9:/_-]+$`

Required: No

Operations

The list of operations permitted by the grant.

Type: array of Strings

Valid Values: `Decrypt | Encrypt | GenerateDataKey | GenerateDataKeyWithoutPlaintext | ReEncryptFrom | ReEncryptTo | CreateGrant | RetireGrant | DescribeKey`

Required: No

RetiringPrincipal

The principal that can retire the grant.

Type: String

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

Required: No

KeyListEntry

Contains information about each entry in the key list.

Contents

Note
In the following list, the required parameters are described first.

KeyArn

ARN of the key.

Type: String

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

Required: No

KeyId

Unique identifier of the key.

Type: String

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

Required: No

KeyMetadata

Contains metadata about a customer master key (CMK).

This data type is used as a response element for the CreateKey (p. 14) and DescribeKey (p. 23) operations.

Contents

Note
In the following list, the required parameters are described first.

KeyId
The globally unique identifier for the CMK.

Type: String

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

Required: Yes

Arn
The Amazon Resource Name (ARN) of the CMK. For examples, see AWS Key Management Service (AWS KMS) in the Example ARNs section of the *AWS General Reference*.

Type: String

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

Required: No

AWSAccountId
The twelve-digit account ID of the AWS account that owns the CMK.

Type: String

Required: No

CreationDate
The date and time when the CMK was created.

Type: Timestamp

Required: No

DeletionDate
The date and time after which AWS KMS deletes the CMK. This value is present only when `KeyState` is `PendingDeletion`, otherwise this value is omitted.

Type: Timestamp

Required: No

Description
The description of the CMK.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 8192.

Required: No

Enabled
Specifies whether the CMK is enabled. When `KeyState` is `Enabled` this value is true, otherwise it is false.

Type: Boolean

Required: No

ExpirationModel
Specifies whether the CMK's key material expires. This value is present only when `Origin` is `EXTERNAL`, otherwise this value is omitted.

Type: String

Valid Values: `KEY_MATERIAL_EXPIRES | KEY_MATERIAL_DOES_NOT_EXPIRE`

Required: No

KeyState

The state of the CMK.

For more information about how key state affects the use of a CMK, see How Key State Affects the Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

Type: String

Valid Values: `Enabled | Disabled | PendingDeletion | PendingImport`

Required: No

KeyUsage

The cryptographic operations for which you can use the CMK. Currently the only allowed value is `ENCRYPT_DECRYPT`, which means you can use the CMK for the Encrypt (p. 33) and Decrypt (p. 17) operations.

Type: String

Valid Values: `ENCRYPT_DECRYPT`

Required: No

Origin

The source of the CMK's key material. When this value is `AWS_KMS`, AWS KMS created the key material. When this value is `EXTERNAL`, the key material was imported from your existing key management infrastructure or the CMK lacks key material.

Type: String

Valid Values: `AWS_KMS | EXTERNAL`

Required: No

ValidTo

The time at which the imported key material expires. When the key material expires, AWS KMS deletes the key material and the CMK becomes unusable. This value is present only for CMKs whose `Origin` is `EXTERNAL` and whose `ExpirationModel` is `KEY_MATERIAL_EXPIRES`, otherwise this value is omitted.

Type: Timestamp

Required: No

Common Parameters

The following table lists the parameters that all actions use for signing Signature Version 4 requests. Any action-specific parameters are listed in the topic for that action. To view sample requests, see Examples of Signed Signature Version 4 Requests or Signature Version 4 Test Suite 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. For a list of services that support AWS Security Token Service, go to Using Temporary Security Credentials to Access AWS in *Using Temporary Security Credentials*.

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 common errors that all actions return. Any action-specific errors are listed in the topic for the action.

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

Throttling

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