# Table of Contents

# Welcome

Amazon CloudWatch Events helps you to respond to state changes in your AWS resources. When your resources change state, they automatically send events into an event stream. You can create rules that match selected events in the stream and route them to targets to take action. You can also use rules to take action on a schedule. For example, you can configure rules to:

- Automatically invoke an AWS Lambda function to update DNS entries when an event notifies you that an EC2 instance entered the running state.
- Direct specific API records from CloudTrail to an Kinesis stream for detailed analysis of potential security or availability risks.
- Periodically invoke a built-in target to create a snapshot of an Amazon EBS volume.

For more information about CloudWatch Events features, see the Amazon CloudWatch Events User Guide.

Use the following links to get started using the CloudWatch Events Query API:

- Actions (p. 2): An alphabetical list of all CloudWatch Events actions.
- Data Types (p. 54): An alphabetical list of all CloudWatch Events data types.
- Common Parameters (p. 70): Parameters that all Query actions can use.
- Common Errors (p. 72): Client and server errors that all actions can return.
- Regions and Endpoints: Supported regions and endpoints for all AWS products.

Alternatively, you can use one of the AWS SDKs to access CloudWatch Events using an API tailored to your programming language or platform.

Developers in the AWS developer community also provide their own libraries, which you can find at the following AWS developer centers:

- Java Developer Center
- JavaScript Developer Center
- AWS Mobile Services
- PHP Developer Center
- Python Developer Center
- Ruby Developer Center
- Windows and .NET Developer Center

# Actions

The following actions are supported:

- DeleteRule (p. 3)
- DescribeEventBus (p. 5)
- DescribeRule (p. 8)
- DisableRule (p. 12)
- EnableRule (p. 14)
- ListRuleNamesByTarget (p. 16)
- ListRules (p. 19)
- ListTargetsByRule (p. 22)
- PutEvents (p. 26)
- PutPermission (p. 29)
- PutRule (p. 32)
- PutTargets (p. 36)
- RemovePermission (p. 46)
- RemoveTargets (p. 48)
- TestEventPattern (p. 51)

# DeleteRule

Deletes the specified rule.

You must remove all targets from a rule using RemoveTargets (p. 48) before you can delete the rule.

When you delete a rule, incoming events might continue to match to the deleted rule. Please allow a short period of time for changes to take effect.

## Request Syntax

```
{
   "Name": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 70).

The request accepts the following data in JSON format.

**Name (p. 3)**

The name of the rule.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

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. 72).

**ConcurrentModificationException**

There is concurrent modification on a rule or target.

HTTP Status Code: 400

**InternalException**

This exception occurs due to unexpected causes.

HTTP Status Code: 500

# Example

## Deletes a rule named "test"

The following is an example of a DeleteRule request.

### Sample Request

```
POST / HTTP/1.1
Host: events.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AWSEvents.DeleteRule

{
    "Name": "test"
}
```

### Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
```

# See Also

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

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

# DescribeEventBus

Displays the external AWS accounts that are permitted to write events to your account using your account's event bus, and the associated policy. To enable your account to receive events from other accounts, use PutPermission (p. 29).

## Response Syntax

```
{
   "Arn": "string",
   "Name": "string",
   "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.

**Arn (p. 5)**

The Amazon Resource Name (ARN) of the account permitted to write events to the current account.

Type: String

**Name (p. 5)**

The name of the event bus. Currently, this is always `default`.

Type: String

**Policy (p. 5)**

The policy that enables the external account to send events to your account.

Type: String

## Errors

For information about the errors that are common to all actions, see Common Errors (p. 72).

**InternalException**

This exception occurs due to unexpected causes.

HTTP Status Code: 500

**ResourceNotFoundException**

An entity that you specified does not exist.

HTTP Status Code: 400

## Examples

The following example is run in account `444455556666`, which has granted permission to AWS account `111122223333` to send events to `444455556666`.

## Example

### Sample Request

```
POST / HTTP/1.1
Host: events.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AWSEvents.DescribeEventBus
```

## Example

### Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>

{
   "Policy":
      {
        "Version":"2012-10-17",
        "Statement":[
            {
               "Sid": "mysid",
               "Effect": "Allow",
               "Principal":{
                    "AWS": "arn:aws:iam::111122223333:root"
               },
               "Action":"events:PutEvents",
               "Resource": "arn:aws:events:us-east-1:444455556666:event-bus/default"
                }
              ]
            },
   "Name": "default",
   "Arn": "arn:aws:events:us-east-1:444455556666:event-bus/default"
}
```

# See Also

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

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

- AWS SDK for Python
- AWS SDK for Ruby V2

# DescribeRule

Describes the specified rule.

## Request Syntax

```
{
   "Name": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 70).

The request accepts the following data in JSON format.

**Name (p. 8)**

The name of the rule.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[\.\-_A-Za-z0-9]+`

Required: Yes

## Response Syntax

```
{
   "Arn": "string",
   "Description": "string",
   "EventPattern": "string",
   "Name": "string",
   "RoleArn": "string",
   "ScheduleExpression": "string",
   "State": "string"
}
```

## Response Elements

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

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

**Arn (p. 8)**

The Amazon Resource Name (ARN) of the rule.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1600.

**Description (p. 8)**

The description of the rule.

Type: String

Length Constraints: Maximum length of 512.

**EventPattern (p. 8)**

The event pattern. For more information, see Events and Event Patterns in the *Amazon CloudWatch Events User Guide*.

Type: String

Length Constraints: Maximum length of 2048.

**Name (p. 8)**

The name of the rule.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[\.\-_A-Za-z0-9]+`

**RoleArn (p. 8)**

The Amazon Resource Name (ARN) of the IAM role associated with the rule.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1600.

**ScheduleExpression (p. 8)**

The scheduling expression. For example, "cron(0 20 * * ? *)", "rate(5 minutes)".

Type: String

Length Constraints: Maximum length of 256.

**State (p. 8)**

Specifies whether the rule is enabled or disabled.

Type: String

Valid Values: `ENABLED | DISABLED`

# Errors

For information about the errors that are common to all actions, see Common Errors (p. 72).

**InternalException**

This exception occurs due to unexpected causes.

HTTP Status Code: 500

**ResourceNotFoundException**

An entity that you specified does not exist.

HTTP Status Code: 400

# Example

## Describes a rule named "test"

The following is an example of a DescribeRule request and response.

### Sample Request

```
POST / HTTP/1.1
Host: events.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AWSEvents.DescribeRule

{
    "Name": "test"
}
```

### Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>

{
    "Name": "test",
    "EventPattern": "{\"source\":[\"aws.autoscaling\"],\"detail-type\":[\"EC2 Instance
 Launch Successful\",\"EC2 Instance Terminate Successful\",\"EC2 Instance Launch
 Unsuccessful\",\"EC2 Instance Terminate Unsuccessful\"]}",
    "State": "ENABLED",
    "Arn": "arn:aws:events:us-east-1:123456789012:rule/test",
    "Description": "Test rule for Auto Scaling events"
}
```

# See Also

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

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

- AWS SDK for Ruby V2

# DisableRule

Disables the specified rule. A disabled rule won't match any events, and won't self-trigger if it has a schedule expression.

When you disable a rule, incoming events might continue to match to the disabled rule. Please allow a short period of time for changes to take effect.

## Request Syntax

```
{
   "Name": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 70).

The request accepts the following data in JSON format.

**Name (p. 12)**

The name of the rule.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

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. 72).

**ConcurrentModificationException**

There is concurrent modification on a rule or target.

HTTP Status Code: 400

**InternalException**

This exception occurs due to unexpected causes.

HTTP Status Code: 500

**ResourceNotFoundException**

An entity that you specified does not exist.

HTTP Status Code: 400

# Example

## Disables a rule named "test"

The following is an example of a DisableRule request.

### Sample Request

```
POST / HTTP/1.1
Host: events.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AWSEvents.DisableRule

{
    "Name": "test"
}
```

### Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
```

# See Also

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

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

# EnableRule

Enables the specified rule. If the rule does not exist, the operation fails.

When you enable a rule, incoming events might not immediately start matching to a newly enabled rule. Please allow a short period of time for changes to take effect.

## Request Syntax

```
{
   "Name": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 70).

The request accepts the following data in JSON format.

**Name (p. 14)**

The name of the rule.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

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. 72).

**ConcurrentModificationException**

There is concurrent modification on a rule or target.

HTTP Status Code: 400

**InternalException**

This exception occurs due to unexpected causes.

HTTP Status Code: 500

**ResourceNotFoundException**

An entity that you specified does not exist.

HTTP Status Code: 400

# Example

## Enables a rule named "test"

The following is an example of an EnableRule request.

### Sample Request

```
POST / HTTP/1.1
Host: events.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AWSEvents.EnableRule

{
    "Name": "test"
}
```

### Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
```

# See Also

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

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

# ListRuleNamesByTarget

Lists the rules for the specified target. You can see which of the rules in Amazon CloudWatch Events can invoke a specific target in your account.

## Request Syntax

```
{
   "Limit": number,
   "NextToken": "string",
   "TargetArn": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 70).

The request accepts the following data in JSON format.

**Limit (p. 16)**

The maximum number of results to return.

Type: Integer

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

Required: No

**NextToken (p. 16)**

The token returned by a previous call to retrieve the next set of results.

Type: String

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

Required: No

**TargetArn (p. 16)**

The Amazon Resource Name (ARN) of the target resource.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1600.

Required: Yes

## Response Syntax

```
{
   "NextToken": "string",
   "RuleNames": [ "string" ]
}
```

# Response Elements

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

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

**NextToken (p. 16)**

Indicates whether there are additional results to retrieve. If there are no more results, the value is null.

Type: String

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

**RuleNames (p. 16)**

The names of the rules that can invoke the given target.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[\.\-_A-Za-z0-9]+`

# Errors

For information about the errors that are common to all actions, see Common Errors (p. 72).

**InternalException**

This exception occurs due to unexpected causes.

HTTP Status Code: 500

# Example

## Lists rule names by target with the specified ARN

The following is an example of a ListRuleNamesByTarget request and response.

### Sample Request

```
POST / HTTP/1.1
Host: events.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AWSEvents.ListRuleNamesByTarget

{
    "TargetArn": "arn:aws:lambda:us-east-1:123456789012:function:MyFunction",
    "NextToken": "",
    "Limit": 0
}
```

## Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>

{
    "RuleNames": [
        "test1",
        "test2",
        "test3",
        "test4",
        "test5"
    ]
}
```

# See Also

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

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

# ListRules

Lists your Amazon CloudWatch Events rules. You can either list all the rules or you can provide a prefix to match to the rule names.

## Request Syntax

```
{
   "Limit": number,
   "NamePrefix": "string",
   "NextToken": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 70).

The request accepts the following data in JSON format.

**Limit (p. 19)**

The maximum number of results to return.

Type: Integer

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

Required: No

**NamePrefix (p. 19)**

The prefix matching the rule name.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[\.\-_A-Za-z0-9]+`

Required: No

**NextToken (p. 19)**

The token returned by a previous call to retrieve the next set of results.

Type: String

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

Required: No

## Response Syntax

```
{
   "NextToken": "string",
   "Rules": [
      {
         "Arn": "string",
         "Description": "string",
         "EventPattern": "string",
         "Name": "string",
         "RoleArn": "string",
         "ScheduleExpression": "string",
         "State": "string"
      }
   ]
}
```

# Response Elements

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

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

### NextToken (p. 19)

Indicates whether there are additional results to retrieve. If there are no more results, the value is null.

Type: String

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

### Rules (p. 19)

The rules that match the specified criteria.

Type: Array of Rule (p. 62) objects

# Errors

For information about the errors that are common to all actions, see Common Errors (p. 72).

**InternalException**

This exception occurs due to unexpected causes.

HTTP Status Code: 500

# Example

## Lists all the rules that start with the letter "t" with a page size of 1

The following is an example of a ListRules request and response.

### Sample Request

```
POST / HTTP/1.1
Host: events.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signature=<Signature>
```

```
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AWSEvents.ListRules

{
    "NamePrefix": "t",
    "Limit": 1
}
```

## Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>

{
    "Rules": [
        {
            "EventPattern": "{\"source\":[\"aws.autoscaling\"],\"detail-type\":[\"EC2
 Instance Launch Successful\",\"EC2 Instance Terminate Successful\",\"EC2 Instance Launch
 Unsuccessful\",\"EC2 Instance Terminate Unsuccessful\"]}",
            "State": "DISABLED",
            "Name": "test",
            "Arn": "arn:aws:events:us-east-1:123456789012:rule/test",
            "Description": "Test rule for Auto Scaling events"
        }
    ],
    "NextToken": "ABCDEgAAAAAAAAAAQAAABCXtD8i7XlyFv5XFKH8GrudAAAAQIoQ0+7qXp63vQf1pvVklfHFd
+p2QgY36pjlAqsSsrkNbOtTePaCeJqN8O
+jbu66UhpJh7huA9r0iY9zjdtZ3vsAAAAgAAAAAAAAAAF5MZWktllmMuLd9gUjryM4sL9EG5IkcPUm60Vq1tzyYw=="
}
```

# See Also

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

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

# ListTargetsByRule

Lists the targets assigned to the specified rule.

## Request Syntax

```
{
   "Limit": number,
   "NextToken": "string",
   "Rule": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 70).

The request accepts the following data in JSON format.

**Limit (p. 22)**

The maximum number of results to return.

Type: Integer

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

Required: No

**NextToken (p. 22)**

The token returned by a previous call to retrieve the next set of results.

Type: String

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

Required: No

**Rule (p. 22)**

The name of the rule.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[\.\-_A-Za-z0-9]+`

Required: Yes

## Response Syntax

```
{
   "NextToken": "string",
   "Targets": [
      {
         "Arn": "string",
         "EcsParameters": {
            "TaskCount": number,
            "TaskDefinitionArn": "string"
         },
         "Id": "string",
         "Input": "string",
         "InputPath": "string",
         "InputTransformer": {
            "InputPathsMap": {
               "string" : "string"
            },
            "InputTemplate": "string"
         },
         "KinesisParameters": {
            "PartitionKeyPath": "string"
         },
         "RoleArn": "string",
         "RunCommandParameters": {
            "RunCommandTargets": [
               {
                  "Key": "string",
                  "Values": [ "string" ]
               }
            ]
         }
      }
   ]
}
```

## Response Elements

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

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

**NextToken (p. 22)**

Indicates whether there are additional results to retrieve. If there are no more results, the value is null.

Type: String

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

**Targets (p. 22)**

The targets assigned to the rule.

Type: Array of Target (p. 66) objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

## Errors

For information about the errors that are common to all actions, see Common Errors (p. 72).

**InternalException**

This exception occurs due to unexpected causes.

HTTP Status Code: 500

**ResourceNotFoundException**

An entity that you specified does not exist.

HTTP Status Code: 400

# Example

## Lists the targets associated with a rule named "test"

The following is an example of a ListTargetsByRule request and response.

### Sample Request

```
POST / HTTP/1.1
Host: events.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AWSEvents.ListTargetsByRule

{
    "Rule": "test"
}
```

### Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>

{
    "Targets": [
        {
            "Id": "MyTargetId",
            "Arn": "arn:aws:lambda:us-east-1:123456789012:function:MyFunction"
        }
    ]
}
```

# See Also

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

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

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

# PutEvents

Sends custom events to Amazon CloudWatch Events so that they can be matched to rules.

## Request Syntax

```
{
   "Entries": [
      {
         "Detail": "string",
         "DetailType": "string",
         "Resources": [ "string" ],
         "Source": "string",
         "Time": number
      }
   ]
}
```

## Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 70).

The request accepts the following data in JSON format.

### Entries (p. 26)

The entry that defines an event in your system. You can specify several parameters for the entry such as the source and type of the event, resources associated with the event, and so on.

Type: Array of PutEventsRequestEntry (p. 58) objects

Array Members: Minimum number of 1 item. Maximum number of 10 items.

Required: Yes

## Response Syntax

```
{
   "Entries": [
      {
         "ErrorCode": "string",
         "ErrorMessage": "string",
         "EventId": "string"
      }
   ],
   "FailedEntryCount": 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.

**Entries (p. 26)**

The successfully and unsuccessfully ingested events results. If the ingestion was successful, the entry has the event ID in it. Otherwise, you can use the error code and error message to identify the problem with the entry.

Type: Array of PutEventsResultEntry (p. 59) objects

**FailedEntryCount (p. 26)**

The number of failed entries.

Type: Integer

# Errors

For information about the errors that are common to all actions, see Common Errors (p. 72).

**InternalException**

This exception occurs due to unexpected causes.

HTTP Status Code: 500

# Example

## Sends two custom events

The following is an example of a PutEvents request and response.

### Sample Request

```
POST / HTTP/1.1
Host: events.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AWSEvents.PutEvents

{
   "Entries":[
      {
         "Source":"com.mycompany.myapp",
         "Detail":"{ \"key1\": \"value1\", \"key2\": \"value2\" }",
         "Resources":[
            "resource1",
            "resource2"
         ],
         "DetailType":"myDetailType"
      },
      {
         "Source":"com.mycompany.myapp",
         "Detail":"{ \"key1\": \"value3\", \"key2\": \"value4\" }",
         "Resources":[
            "resource1",
            "resource2"
         ],
         "DetailType":"myDetailType"
      }
   ]
}
```

## Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>

{
    "FailedEntryCount": 0,
    "Entries": [
        {
            "EventId": "11710aed-b79e-4468-a20b-bb3c0c3b4860"
        },
        {
            "EventId": "d804d26a-88db-4b66-9eaf-9a11c708ae82"
        }
    ]
}
```

# See Also

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

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

# PutPermission

Running `PutPermission` permits the specified AWS account to put events to your account's default *event bus*. CloudWatch Events rules in your account are triggered by these events arriving to your default event bus.

For another account to send events to your account, that external account must have a CloudWatch Events rule with your account's default event bus as a target.

To enable multiple AWS accounts to put events to your default event bus, run `PutPermission` once for each of these accounts.

The permission policy on the default event bus cannot exceed 10KB in size.

## Request Syntax

```
{
   "Action": "string",
   "Principal": "string",
   "StatementId": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 70).

The request accepts the following data in JSON format.

### Action (p. 29)

The action that you are enabling the other account to perform. Currently, this must be `events:PutEvents`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `events:[a-zA-Z]+`

Required: Yes

### Principal (p. 29)

The 12-digit AWS account ID that you are permitting to put events to your default event bus. Specify "*" to permit any account to put events to your default event bus.

If you specify "*", avoid creating rules that may match undesirable events. To create more secure rules, make sure that the event pattern for each rule contains an `account` field with a specific account ID from which to receive events. Rules with an account field do not match any events sent from other accounts.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 12.

Pattern: `(\d{12}|\*)`

Required: Yes

**StatementId (p. 29)**

An identifier string for the external account that you are granting permissions to. If you later want to revoke the permission for this external account, specify this `StatementId` when you run RemovePermission (p. 46).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

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. 72).

**InternalException**

This exception occurs due to unexpected causes.

HTTP Status Code: 500

**PolicyLengthExceededException**

The event bus policy is too long. For more information, see the limits.

HTTP Status Code: 400

**ResourceNotFoundException**

An entity that you specified does not exist.

HTTP Status Code: 400

# Example

The following example enables the current account to receive events from account `111122223333`.

## Sample Request

```
POST / HTTP/1.1
Host: events.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AWSEvents.PutPermission
```

```
{
   "Action":"events:PutEvents"
   "Principal":"111122223333"
   "StatementId":"MyStatement"

}
```

## See Also

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

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

# PutRule

Creates or updates the specified rule. Rules are enabled by default, or based on value of the state. You can disable a rule using DisableRule (p. 12).

If you are updating an existing rule, the rule is completely replaced with what you specify in this `PutRule` command. If you omit arguments in `PutRule`, the old values for those arguments are not kept. Instead, they are replaced with null values.

When you create or update a rule, incoming events might not immediately start matching to new or updated rules. Please allow a short period of time for changes to take effect.

A rule must contain at least an EventPattern or ScheduleExpression. Rules with EventPatterns are triggered when a matching event is observed. Rules with ScheduleExpressions self-trigger based on the given schedule. A rule can have both an EventPattern and a ScheduleExpression, in which case the rule triggers on matching events as well as on a schedule.

Most services in AWS treat : or / as the same character in Amazon Resource Names (ARNs). However, CloudWatch Events uses an exact match in event patterns and rules. Be sure to use the correct ARN characters when creating event patterns so that they match the ARN syntax in the event you want to match.

## Request Syntax

```
{
   "Description": "string",
   "EventPattern": "string",
   "Name": "string",
   "RoleArn": "string",
   "ScheduleExpression": "string",
   "State": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 70).

The request accepts the following data in JSON format.

**Description (p. 32)**

A description of the rule.

Type: String

Length Constraints: Maximum length of 512.

Required: No

**EventPattern (p. 32)**

The event pattern. For more information, see Events and Event Patterns in the *Amazon CloudWatch Events User Guide*.

Type: String

Length Constraints: Maximum length of 2048.

Required: No

**Name (p. 32)**

The name of the rule that you are creating or updating.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[\.\-_A-Za-z0-9]+`

Required: Yes

**RoleArn (p. 32)**

The Amazon Resource Name (ARN) of the IAM role associated with the rule.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1600.

Required: No

**ScheduleExpression (p. 32)**

The scheduling expression. For example, "cron(0 20 * * ? *)" or "rate(5 minutes)".

Type: String

Length Constraints: Maximum length of 256.

Required: No

**State (p. 32)**

Indicates whether the rule is enabled or disabled.

Type: String

Valid Values: `ENABLED | DISABLED`

Required: No

## Response Syntax

```
{
   "RuleArn": "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.

**RuleArn (p. 33)**

The Amazon Resource Name (ARN) of the rule.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1600.

# Errors

For information about the errors that are common to all actions, see Common Errors (p. 72).

**ConcurrentModificationException**

There is concurrent modification on a rule or target.

HTTP Status Code: 400

**InternalException**

This exception occurs due to unexpected causes.

HTTP Status Code: 500

**InvalidEventPatternException**

The event pattern is not valid.

HTTP Status Code: 400

**LimitExceededException**

You tried to create more rules or add more targets to a rule than is allowed.

HTTP Status Code: 400

# Example

## Creates a rule named "test" that matches events from Amazon EC2

The following is an example of a PutRule request and response.

### Sample Request

```
POST / HTTP/1.1
Host: events.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AWSEvents.PutRule

{
    "Name": "test",
    "EventPattern": "{ \"source\": [\"aws.ec2\"] }"
}
```

### Sample Response

```
HTTP/1.1 200 OK
```

```
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>

{
    "RuleArn": "arn:aws:events:us-east-1:123456789012:rule/test"
}
```

# See Also

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

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

# PutTargets

Adds the specified targets to the specified rule, or updates the targets if they are already associated with the rule.

Targets are the resources that are invoked when a rule is triggered.

You can configure the following as targets for CloudWatch Events:

- EC2 instances
- AWS Lambda functions
- Streams in Amazon Kinesis Streams
- Delivery streams in Amazon Kinesis Firehose
- Amazon ECS tasks
- AWS Step Functions state machines
- Pipelines in Amazon Code Pipeline
- Amazon Inspector assessment templates
- Amazon SNS topics
- Amazon SQS queues
- The default event bus of another AWS account

Note that creating rules with built-in targets is supported only in the AWS Management Console.

For some target types, `PutTargets` provides target-specific parameters. If the target is an Amazon Kinesis stream, you can optionally specify which shard the event goes to by using the `KinesisParameters` argument. To invoke a command on multiple EC2 instances with one rule, you can use the `RunCommandParameters` field.

To be able to make API calls against the resources that you own, Amazon CloudWatch Events needs the appropriate permissions. For AWS Lambda and Amazon SNS resources, CloudWatch Events relies on resource-based policies. For EC2 instances, Amazon Kinesis streams, and AWS Step Functions state machines, CloudWatch Events relies on IAM roles that you specify in the `RoleARN` argument in `PutTargets`. For more information, see Authentication and Access Control in the *Amazon CloudWatch Events User Guide*.

If another AWS account is in the same region and has granted you permission (using `PutPermission`), you can send events to that account by setting that account's event bus as a target of the rules in your account. To send the matched events to the other account, specify that account's event bus as the `Arn` when you run `PutTargets`. If your account sends events to another account, your account is charged for each sent event. Each event sent to antoher account is charged as a custom event. The account receiving the event is not charged. For more information on pricing, see Amazon CloudWatch Pricing.

For more information about enabling cross-account events, see PutPermission (p. 29).

**Input**, **InputPath** and **InputTransformer** are mutually exclusive and optional parameters of a target. When a rule is triggered due to a matched event:

- If none of the following arguments are specified for a target, then the entire event is passed to the target in JSON form (unless the target is Amazon EC2 Run Command or Amazon ECS task, in which case nothing from the event is passed to the target).
- If **Input** is specified in the form of valid JSON, then the matched event is overridden with this constant.
- If **InputPath** is specified in the form of JSONPath (for example, `$.detail`), then only the part of the event specified in the path is passed to the target (for example, only the detail part of the event is passed).

- If **InputTransformer** is specified, then one or more specified JSONPaths are extracted from the event and used as values in a template that you specify as the input to the target.

When you specify `InputPath` or `InputTransformer`, you must use JSON dot notation, not bracket notation.

When you add targets to a rule and the associated rule triggers soon after, new or updated targets might not be immediately invoked. Please allow a short period of time for changes to take effect.

This action can partially fail if too many requests are made at the same time. If that happens, `FailedEntryCount` is non-zero in the response and each entry in `FailedEntries` provides the ID of the failed target and the error code.

## Request Syntax

```
{
   "Rule": "string",
   "Targets": [
      {
         "Arn": "string",
         "EcsParameters": {
            "TaskCount": number,
            "TaskDefinitionArn": "string"
         },
         "Id": "string",
         "Input": "string",
         "InputPath": "string",
         "InputTransformer": {
            "InputPathsMap": {
               "string" : "string"
            },
            "InputTemplate": "string"
         },
         "KinesisParameters": {
            "PartitionKeyPath": "string"
         },
         "RoleArn": "string",
         "RunCommandParameters": {
            "RunCommandTargets": [
               {
                  "Key": "string",
                  "Values": [ "string" ]
               }
            ]
         }
      }
   ]
}
```

## Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 70).

The request accepts the following data in JSON format.

### Rule (p. 37)

The name of the rule.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[\.\-_A-Za-z0-9]+`

Required: Yes

**Targets (p. 37)**

The targets to update or add to the rule.

Type: Array of Target (p. 66) objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Required: Yes

## Response Syntax

```
{
   "FailedEntries": [
      {
         "ErrorCode": "string",
         "ErrorMessage": "string",
         "TargetId": "string"
      }
   ],
   "FailedEntryCount": 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.

**FailedEntries (p. 38)**

The failed target entries.

Type: Array of PutTargetsResultEntry (p. 60) objects

**FailedEntryCount (p. 38)**

The number of failed entries.

Type: Integer

## Errors

For information about the errors that are common to all actions, see Common Errors (p. 72).

**ConcurrentModificationException**

There is concurrent modification on a rule or target.

HTTP Status Code: 400

**InternalException**

This exception occurs due to unexpected causes.

HTTP Status Code: 500

**LimitExceededException**

You tried to create more rules or add more targets to a rule than is allowed.

HTTP Status Code: 400

**ResourceNotFoundException**

An entity that you specified does not exist.

HTTP Status Code: 400

# Examples

## Adds a target to a Lambda function with the ID "MyTargetId" to the rule named "test"

The following is an example of a PutTargets request.

### Sample Request

```
POST / HTTP/1.1
Host: events.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AWSEvents.PutTargets

{
    "Rule": "test",
    "Targets": [
        {
            "Id": "MyTargetId",
            "Arn": "arn:aws:lambda:us-east-1:123456789012:function:MyFunction"
        }
    ]
}
```

### Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
```

```
{
    "FailedEntries": [],
    "FailedEntryCount": 0
}
```

## Uses Input Transformer to extract data from an event and input that data to the target

This example extracts the instance and state from an event, puts them into a simple text template, and passes this data to a Lambda function called `MyFunction`.

### Sample Request

```
POST / HTTP/1.1
Host: events.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AWSEvents.PutTargets

{
    "Rule": "testrule",
    "Targets": [
        {
            "Id": "MyTargetId",
            "Arn": "arn:aws:lambda:us-east-1:123456789012:function:MyFunction"
            "InputTransformer":
              {
                "InputPathsMap": {“instance”: “$.detail.instance”,”status”:
 "$.detail.status"},
                "InputTemplate": "<instance> is in state <status>"
              }
    ]
}
```

## Example

Here is another sample request using `InputTransformer`. The input to the Lambda function is in JSON format, with an array substituted. Below that sample request are examples of an event and the resulting output to the target, using this sample request.

```
POST / HTTP/1.1
Host: events.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AWSEvents.PutTargets

{
    "Rule": "testrule",
    "Targets": [
        {
            "Id": "MyTargetId",
            "Arn": "arn:aws:lambda:us-east-1:123456789012:function:MyFunction"
            "InputTransformer":
              {
                "InputPathsMap": {"commandstToRun": "$.detail.commands"}
                "InputTemplate": "{\"commands\": <commandsToRun>}"
        }
    ]
}
```

```
                  Incoming event:

{
    "Time": 1225864800,
    "Source": "foo",
    "Resources": ["foo", "foo"],
    "DetailType": "foo",
    "Detail": {
        "commands": ["ls -lrt", "echo HelloWorld!"]
    }
}
```

```
                  Output sent to the target:

{
    "commands" : ["ls -lrt", "echo HelloWorld!"]
}
```

## Sends a command to a list of EC2 instances specified by InstanceIds, using Amazon EC2 Run Command

```
POST / HTTP/1.1
Host: events.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AWSEvents.PutTargets

{
    "Rule": "testrule",
    "Targets": [
        {
            "Id": "id123456789",
            "Arn": "arn:aws:ssm:us-east-1:12345679012:document/RestartLinuxService",
            "RoleArn": "arn:aws:iam::123456789012:role/MyRoleToAccessEC2"
            "RunCommandProperties": {
                 "RunCommandTargets": [
                    {
                        "Key":"InstanceIds",
                        "Values":["i-123456789012", "i-098765432109"]
                    }
                                       ]
    }

    ]
}
```

## Uses KinesisParameters to control shard assignment

In this example, `KinesisParameters` is used to specify that events related to status changes are sent to a shard specific to the affected instance ID.

```
POST / HTTP/1.1
Host: events.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AWSEvents.PutTargets

{
    "Rule": "StatusChangeRule",
    "Targets": [
           {
        "Id" : "1",
        "Arn": "arn:aws:kinesis:us-east-1:123456789012:function:stream/mystream",
        "KinesisParameters":{
             "PartitionKeyPath":"$.detail.instance-id"}'
    }
]
}
```

## Adds an Amazon Kinesis Firehose delivery stream as a target

This example sets an Amazon Kinesis Firehose delivery stream named `target-stream-name` as a target.

```
POST / HTTP/1.1
Host: events.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AWSEvents.PutTargets

{
    "Rule": "FirehoseExample",
    "Targets": [
           {
        "Id" : "FirehoseStream",
        "Arn": "arn:aws:firehose:us-east-1:123456789012:deliverystream/target-stream-name",
    }
]
}
```

## Adds a Step Functions state machine as a target

This example targets a state machine called "HelloWorld", and sends the input constant "Hello World!" to that target.

```
POST / HTTP/1.1
Host: events.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AWSEvents.PutTargets

{
    "Rule": "testrule",
    "Targets": [
           {
        "RoleArn": "arn:aws:iam::123456789012:role/MyRoleToAccessStepFunctions"
        "Arn":"arn:aws:states:us-east-1:123456789012:stateMachine:HelloWorld",
        "Input":"HelloWorld!"
    }
]
}
```

## Adds a target that creates three Amazon ECS tasks based on a task definition.

This example uses Amazon ECS as the target. You must have already created the task definition and cluster in Amazon ECS.

```
POST / HTTP/1.1
Host: events.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AWSEvents.PutTargets

{
    "Rule": "test",
    "Targets": [
        {
            "Id": "Target1",
            "RoleArn": "arn:aws:iam::123456789012:role/MyRoleToAccessECS"
            "Arn": "arn:aws:ecs:us-east-1:123456789012:cluster/example-cluster",
            "ECSParameters": {
                  "TaskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-definition/
example",
                  "TaskCount": 3
             }
         }
    ]
}
```

## Specifying two targets with one command

This example sets two simple targets with one command. In this example, both targets are AWS Lambda functions, but the two targets could be different AWS services as well.

### Sample Request

```
POST / HTTP/1.1
Host: events.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AWSEvents.PutTargets

{
    "Rule": "test",
    "Targets": [
        {
            "Id": "MyTargetId",
            "Arn": "arn:aws:lambda:us-east-1:123456789012:function:MyFunction"
        }
        {
            "Id": "MyTargetId2",
            "Arn": "arn:aws:lambda:us-east-1:123456789012:function:MyFunction2"
        }
    ]
}
```

## Specifying another AWS account as a target

This example shows cross-account event delivery. The target being added is the event bus of a separate AWS account, which has the AWS account ID of `444455556666`.

### Sample Request

```
POST / HTTP/1.1
Host: events.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AWSEvents.PutTargets

{
    "Rule": "producer-rule",
    "Targets": [
        {
            "Id": "CrossAccountTargetId",
            "Arn": "arn:aws:events:us-east-1:444455556666:event-bus/default"
        }
    ]
}
```

# See Also

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

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

# RemovePermission

Revokes the permission of another AWS account to be able to put events to your default event bus. Specify the account to revoke by the `StatementId` value that you associated with the account when you granted it permission with `PutPermission`. You can find the `StatementId` by using DescribeEventBus (p. 5).

## Request Syntax

```
{
   "StatementId": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 70).

The request accepts the following data in JSON format.

**StatementId (p. 46)**

The statement ID corresponding to the account that is no longer allowed to put events to the default event bus.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

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. 72).

**InternalException**

This exception occurs due to unexpected causes.

HTTP Status Code: 500

**ResourceNotFoundException**

An entity that you specified does not exist.

HTTP Status Code: 400

## See Also

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

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

# RemoveTargets

Removes the specified targets from the specified rule. When the rule is triggered, those targets are no longer be invoked.

When you remove a target, when the associated rule triggers, removed targets might continue to be invoked. Please allow a short period of time for changes to take effect.

This action can partially fail if too many requests are made at the same time. If that happens, `FailedEntryCount` is non-zero in the response and each entry in `FailedEntries` provides the ID of the failed target and the error code.

## Request Syntax

```
{
   "Ids": [ "string" ],
   "Rule": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 70).

The request accepts the following data in JSON format.

**Ids (p. 48)**

The IDs of the targets to remove from the rule.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[\.\-_A-Za-z0-9]+`

Required: Yes

**Rule (p. 48)**

The name of the rule.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[\.\-_A-Za-z0-9]+`

Required: Yes

## Response Syntax

```
{
   "FailedEntries": [
      {
         "ErrorCode": "string",
         "ErrorMessage": "string",
         "TargetId": "string"
      }
   ],
   "FailedEntryCount": 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.

**FailedEntries (p. 48)**

The failed target entries.

Type: Array of RemoveTargetsResultEntry (p. 61) objects

**FailedEntryCount (p. 48)**

The number of failed entries.

Type: Integer

# Errors

For information about the errors that are common to all actions, see Common Errors (p. 72).

**ConcurrentModificationException**

There is concurrent modification on a rule or target.

HTTP Status Code: 400

**InternalException**

This exception occurs due to unexpected causes.

HTTP Status Code: 500

**ResourceNotFoundException**

An entity that you specified does not exist.

HTTP Status Code: 400

# Example

## Removes a target with ID "MyTargetId" from a rule named "test"

The following is an example of a RemoveTargets request.

### Sample Request

```
POST / HTTP/1.1
```

```
Host: events.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AWSEvents.RemoveTargets

{
    "Rule": "test",
    "Ids": [
        "MyTargetId"
    ]
}
```

### Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>

{
    "FailedEntries": [],
    "FailedEntryCount": 0
}
```

## See Also

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

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

# TestEventPattern

Tests whether the specified event pattern matches the provided event.

Most services in AWS treat : or / as the same character in Amazon Resource Names (ARNs). However, CloudWatch Events uses an exact match in event patterns and rules. Be sure to use the correct ARN characters when creating event patterns so that they match the ARN syntax in the event you want to match.

## Request Syntax

```
{
   "Event": "string",
   "EventPattern": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 70).

The request accepts the following data in JSON format.

**Event (p. 51)**

The event, in JSON format, to test against the event pattern.

Type: String

Required: Yes

**EventPattern (p. 51)**

The event pattern. For more information, see Events and Event Patterns in the *Amazon CloudWatch Events User Guide*.

Type: String

Length Constraints: Maximum length of 2048.

Required: Yes

## Response Syntax

```
{
   "Result": 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.

**Result (p. 51)**

Indicates whether the event matches the event pattern.

Type: Boolean

# Errors

For information about the errors that are common to all actions, see Common Errors (p. 72).

**InternalException**

This exception occurs due to unexpected causes.

HTTP Status Code: 500

**InvalidEventPatternException**

The event pattern is not valid.

HTTP Status Code: 400

# Example

## Tests that a given event matches a given event pattern

The following is an example of a TestEventPattern request and response.

### Sample Request

```
POST / HTTP/1.1
Host: events.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AWSEvents.TestEventPattern

{
    "EventPattern": "{\"source\": [\"com.mycompany.myapp\"]}",
    "Event": "{\"id\": \"e00c66cb-fe7a-4fcc-81ad-58eb60f5d96b\", \"detail-type\":
 \"myDetailType\", \"source\": \"com.mycompany.myapp\", \"account\": \"123456789012\",
 \"time\": \"2016-01-10T01:29:23Z\", \"region\": \"us-east-1\", \"resources\":
 [\"resource1\",  \"resource2\"], \"detail\": {{\"key1\": \"value1\", \"key2\":
 \"value2\"}}"
  }
}
```

### Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
```

```
{
    "Result": true
}
```

# See Also

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

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

# Data Types

The Amazon CloudWatch Events 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:

# EcsParameters

The custom parameters to be used when the target is an Amazon ECS cluster.

## Contents

**TaskCount**

The number of tasks to create based on the `TaskDefinition`. The default is one.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

**TaskDefinitionArn**

The ARN of the task definition to use if the event target is an Amazon ECS cluster.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1600.

Required: Yes

## See Also

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

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

# InputTransformer

Contains the parameters needed for you to provide custom input to a target based on one or more pieces of data extracted from the event.

## Contents

**InputPathsMap**

Map of JSON paths to be extracted from the event. These are key-value pairs, where each value is a JSON path. You must use JSON dot notation, not bracket notation.

Type: String to string map

Length Constraints: Maximum length of 256.

Required: No

**InputTemplate**

Input template where you can use the values of the keys from `InputPathsMap` to customize the data sent to the target.

Type: String

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

Required: Yes

## See Also

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

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

# KinesisParameters

This object enables you to specify a JSON path to extract from the event and use as the partition key for the Amazon Kinesis stream, so that you can control the shard to which the event goes. If you do not include this parameter, the default is to use the `eventId` as the partition key.

## Contents

**PartitionKeyPath**

The JSON path to be extracted from the event and used as the partition key. For more information, see Amazon Kinesis Streams Key Concepts in the *Amazon Kinesis Streams Developer Guide*.

Type: String

Length Constraints: Maximum length of 256.

Required: Yes

## See Also

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

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

# PutEventsRequestEntry

Represents an event to be submitted.

## Contents

**Detail**

In the JSON sense, an object containing fields, which may also contain nested subobjects. No constraints are imposed on its contents.

Type: String

Required: No

**DetailType**

Free-form string used to decide what fields to expect in the event detail.

Type: String

Required: No

**Resources**

AWS resources, identified by Amazon Resource Name (ARN), which the event primarily concerns. Any number, including zero, may be present.

Type: Array of strings

Required: No

**Source**

The source of the event.

Type: String

Required: No

**Time**

The timestamp of the event, per RFC3339. If no timestamp is provided, the timestamp of the PutEvents (p. 26) call is used.

Type: Timestamp

Required: No

## See Also

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

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

# PutEventsResultEntry

Represents an event that failed to be submitted.

## Contents

**ErrorCode**

The error code that indicates why the event submission failed.

Type: String

Required: No

**ErrorMessage**

The error message that explains why the event submission failed.

Type: String

Required: No

**EventId**

The ID of the event.

Type: String

Required: No

## See Also

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

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

# PutTargetsResultEntry

Represents a target that failed to be added to a rule.

## Contents

**ErrorCode**

The error code that indicates why the target addition failed. If the value is `ConcurrentModificationException`, too many requests were made at the same time.

Type: String

Required: No

**ErrorMessage**

The error message that explains why the target addition failed.

Type: String

Required: No

**TargetId**

The ID of the target.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[\.\-_A-Za-z0-9]+`

Required: No

## See Also

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

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

# RemoveTargetsResultEntry

Represents a target that failed to be removed from a rule.

## Contents

**ErrorCode**

The error code that indicates why the target removal failed. If the value is `ConcurrentModificationException`, too many requests were made at the same time.

Type: String

Required: No

**ErrorMessage**

The error message that explains why the target removal failed.

Type: String

Required: No

**TargetId**

The ID of the target.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[\.\-_A-Za-z0-9]+`

Required: No

## See Also

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

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

# Rule

Contains information about a rule in Amazon CloudWatch Events.

## Contents

**Arn**

The Amazon Resource Name (ARN) of the rule.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1600.

Required: No

**Description**

The description of the rule.

Type: String

Length Constraints: Maximum length of 512.

Required: No

**EventPattern**

The event pattern of the rule. For more information, see Events and Event Patterns in the *Amazon CloudWatch Events User Guide*.

Type: String

Length Constraints: Maximum length of 2048.

Required: No

**Name**

The name of the rule.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[\.\-_A-Za-z0-9]+`

Required: No

**RoleArn**

The Amazon Resource Name (ARN) of the role that is used for target invocation.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1600.

Required: No

**ScheduleExpression**

The scheduling expression. For example, "cron(0 20 * * ? *)", "rate(5 minutes)".

Type: String

Length Constraints: Maximum length of 256.

Required: No

**State**

The state of the rule.

Type: String

Valid Values: `ENABLED | DISABLED`

Required: No

# See Also

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

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

# RunCommandParameters

This parameter contains the criteria (either InstanceIds or a tag) used to specify which EC2 instances are to be sent the command.

## Contents

**RunCommandTargets**

Currently, we support including only one RunCommandTarget block, which specifies either an array of InstanceIds or a tag.

Type: Array of RunCommandTarget (p. 65) objects

Array Members: Minimum number of 1 item. Maximum number of 5 items.

Required: Yes

## See Also

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

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

# RunCommandTarget

Information about the EC2 instances that are to be sent the command, specified as key-value pairs. Each `RunCommandTarget` block can include only one key, but this key may specify multiple values.

## Contents

**Key**

Can be either `tag:` *tag-key* or `InstanceIds`.

Type: String

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

Pattern: `^[\p{L}\p{Z}\p{N}_.:/=+\-@]*$`

Required: Yes

**Values**

If `Key` is `tag:` *tag-key*, `Values` is a list of tag values. If `Key` is `InstanceIds`, `Values` is a list of Amazon EC2 instance IDs.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 50 items.

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

Required: Yes

## See Also

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

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

# Target

Targets are the resources to be invoked when a rule is triggered. Target types include EC2 instances, AWS Lambda functions, Amazon Kinesis streams, Amazon ECS tasks, AWS Step Functions state machines, Run Command, and built-in targets.

## Contents

**Arn**

The Amazon Resource Name (ARN) of the target.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1600.

Required: Yes

**EcsParameters**

Contains the Amazon ECS task definition and task count to be used, if the event target is an Amazon ECS task. For more information about Amazon ECS tasks, see Task Definitions in the *Amazon EC2 Container Service Developer Guide*.

Type: EcsParameters (p. 55) object

Required: No

**Id**

The ID of the target.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[\.\-_A-Za-z0-9]+`

Required: Yes

**Input**

Valid JSON text passed to the target. In this case, nothing from the event itself is passed to the target. For more information, see The JavaScript Object Notation (JSON) Data Interchange Format.

Type: String

Length Constraints: Maximum length of 8192.

Required: No

**InputPath**

The value of the JSONPath that is used for extracting part of the matched event when passing it to the target. You must use JSON dot notation, not bracket notation. For more information about JSON paths, see JSONPath.

Type: String

Length Constraints: Maximum length of 256.

Required: No

**InputTransformer**

Settings to enable you to provide custom input to a target based on certain event data. You can extract one or more key-value pairs from the event and then use that data to send customized input to the target.

Type: InputTransformer (p. 56) object

Required: No

**KinesisParameters**

The custom parameter you can use to control shard assignment, when the target is an Amazon Kinesis stream. If you do not include this parameter, the default is to use the `eventId` as the partition key.

Type: KinesisParameters (p. 57) object

Required: No

**RoleArn**

The Amazon Resource Name (ARN) of the IAM role to be used for this target when the rule is triggered. If one rule triggers multiple targets, you can use a different IAM role for each target.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1600.

Required: No

**RunCommandParameters**

Parameters used when you are using the rule to invoke Amazon EC2 Run Command.

Type: RunCommandParameters (p. 64) object

Required: No

## See Also

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

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

# Making API Requests

Query requests used with CloudWatch Events are HTTP or HTTPS requests that use the HTTP verb GET or POST and a Query parameter named `Action` or `Operation`. This documentation uses `Action`, although `Operation` is supported for backward compatibility.

## CloudWatch Events Endpoints

An endpoint is a URL that serves as an entry point for a web service. You can select a regional endpoint when you make your requests to reduce latency. For information about the endpoints used with CloudWatch Events, see Regions and Endpoints in the *Amazon Web Services General Reference*.

## Query Parameters

Each query request must include some common parameters to handle authentication and selection of an action. For more information, see Common Parameters (p. 70).

Some API operations take lists of parameters. These lists are specified using the following notation: `param.member.n`. Values of n are integers starting from 1. All lists of parameters must follow this notation, including lists that contain only one parameter. For example, a Query parameter list looks like this:

```
&attribute.member.1=this
&attribute.member.2=that
```

## Request Identifiers

In every response from an AWS Query API, there is a `ResponseMetadata` element, which contains a `RequestId` element. This string is a unique identifier that AWS assigns to provide tracking information. Although `RequestId` is included as part of every response, it is not listed on the individual API documentation pages to improve readability and to reduce redundancy.

# Query API Authentication

You can send query requests over either HTTP or HTTPS. Regardless of which protocol you use, you must include a signature in every query request. For more information about creating and including a signature, see Signing AWS API Requests in the *Amazon Web Services General Reference*.

# Available Libraries

AWS provides libraries, sample code, tutorials, and other resources for software developers who prefer to build applications using language-specific APIs instead of the command-line tools and Query API. These libraries provide basic functions (not included in the APIs), such as request authentication, request retries, and error handling so that it is easier to get started. Libraries and resources are available for the following languages and platforms:

- AWS Mobile SDK for Android
- AWS SDK for Go
- AWS Mobile SDK for iOS
- AWS SDK for Java 2.0
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for JavaScript in Node.js
- AWS SDK for .NET
- AWS SDK for PHP
- AWS SDK for Python (Boto)
- AWS SDK for Ruby

For libraries and sample code in all languages, see Sample Code & Libraries.

# Common Parameters

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

**Action**

The action to be performed.

Type: string

Required: Yes

**Version**

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

Type: string

Required: Yes

**X-Amz-Algorithm**

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

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

Type: string

Valid Values: `AWS4-HMAC-SHA256`

Required: Conditional

**X-Amz-Credential**

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

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

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

Type: string

Required: Conditional

**X-Amz-Date**

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

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see Handling Dates in Signature Version 4 in the *Amazon Web Services General Reference*.

Type: string

Required: Conditional

**X-Amz-Security-Token**

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

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

Type: string

Required: Conditional

**X-Amz-Signature**

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

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

Type: string

Required: Conditional

**X-Amz-SignedHeaders**

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

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

Type: string

Required: Conditional

# Common Errors

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

**AccessDeniedException**

You do not have sufficient access to perform this action.

HTTP Status Code: 400

**IncompleteSignature**

The request signature does not conform to AWS standards.

HTTP Status Code: 400

**InternalFailure**

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

HTTP Status Code: 500

**InvalidAction**

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

HTTP Status Code: 400

**InvalidClientTokenId**

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

HTTP Status Code: 403

**InvalidParameterCombination**

Parameters that must not be used together were used together.

HTTP Status Code: 400

**InvalidParameterValue**

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

HTTP Status Code: 400

**InvalidQueryParameter**

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

HTTP Status Code: 400

**MalformedQueryString**

The query string contains a syntax error.

HTTP Status Code: 404

**MissingAction**

The request is missing an action or a required parameter.

HTTP Status Code: 400

**MissingAuthenticationToken**

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

HTTP Status Code: 403

**MissingParameter**

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

**OptInRequired**

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

HTTP Status Code: 403

**RequestExpired**

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

HTTP Status Code: 400

**ServiceUnavailable**

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

HTTP Status Code: 503

**ThrottlingException**

The request was denied due to request throttling.

HTTP Status Code: 400

**ValidationError**

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

HTTP Status Code: 400