Amazon CloudWatch Logs API Reference

API Reference

API Version 2014-03-28

Amazon CloudWatch Logs API Reference: API Reference

Table of Contents

Welcome

This is the *Amazon CloudWatch Logs API Reference*. Amazon CloudWatch Logs enables you to monitor, store, and access your system, application, and custom log files. This guide provides detailed information about Amazon CloudWatch Logs actions, data types, parameters, and errors. For detailed information about Amazon CloudWatch Logs features and their associated API calls, go to the Amazon CloudWatch Developer Guide.

Use the following links to get started using the *Amazon CloudWatch Logs API Reference*:

- Actions: An alphabetical list of all Amazon CloudWatch Logs actions.
- Data Types: An alphabetical list of all Amazon CloudWatch Logs data types.
- Common Parameters: Parameters that all Query actions can use.
- Common Errors: Client and server errors that all actions can return.
- Regions and Endpoints: Itemized regions and endpoints for all AWS products.

In addition to using the Amazon CloudWatch Logs API, you can also use the following SDKs and third-party libraries to access Amazon CloudWatch Logs programmatically.

- AWS SDK for Java Documentation
- AWS SDK for .NET Documentation
- AWS SDK for PHP Documentation
- AWS SDK for Ruby Documentation

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

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

This document was last updated on December 10, 2015.

Actions

The following actions are supported:

CancelExportTask

Cancels an export task if it is in `PENDING` or `RUNNING` state.

Request Syntax

```
{
    "TaskId": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

TaskId
Id of the export task to cancel.

Type: String

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

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. 103).

InvalidOperationException
Returned if the operation is not valid on the specified resource

HTTP Status Code: 400

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

ResourceNotFoundException
Returned if the specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

Cancel an export task.

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.CancelExportTask
{
  "taskId": "exampleTaskId"
}
```

Sample Response

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

CreateExportTask

Creates an `ExportTask` which allows you to efficiently export data from a Log Group to your Amazon S3 bucket.

This is an asynchronous call. If all the required information is provided, this API will initiate an export task and respond with the task Id. Once started, `DescribeExportTasks` can be used to get the status of an export task.

You can export logs from multiple log groups or multiple time ranges to the same Amazon S3 bucket. To separate out log data for each export task, you can specify a prefix that will be used as the Amazon S3 key prefix for all exported objects.

Request Syntax

```
{
    "Destination": "string",
    "DestinationPrefix": "string",
    "From": number,
    "LogGroupName": "string",
    "LogStreamNamePrefix": "string",
    "TaskName": "string",
    "To": number
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

Destination
Name of Amazon S3 bucket to which the log data will be exported.

NOTE: Only buckets in the same AWS region are supported

Type: String

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

Required: Yes

DestinationPrefix
Prefix that will be used as the start of Amazon S3 key for every object exported. If not specified, this defaults to 'exportedlogs'.

Type: String

Required: No

From
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC. It indicates the start time of the range for the request. Events with a timestamp prior to this time will not be exported.

Type: Long

Valid range: Minimum value of 0.

Required: Yes

LogGroupName

The name of the log group to export.

Type: String

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

Pattern: `[\.\-_/#A-Za-z0-9]+`

Required: Yes

LogStreamNamePrefix

Will only export log streams that match the provided logStreamNamePrefix. If you don't specify a value, no prefix filter is applied.

Type: String

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

Pattern: `[^:*]*`

Required: No

TaskName

The name of the export task.

Type: String

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

Required: No

To

A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC. It indicates the end time of the range for the request. Events with a timestamp later than this time will not be exported.

Type: Long

Valid range: Minimum value of 0.

Required: Yes

Response Syntax

```
{
    "TaskId": "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.

TaskId
Id of the export task that got created.

Type: String

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 103).

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

LimitExceededException
Returned if you have reached the maximum number of resources that can be created.

HTTP Status Code: 400

OperationAbortedException
Returned if multiple requests to update the same resource were in conflict.

HTTP Status Code: 400

ResourceAlreadyExistsException
Returned if the specified resource already exists.

HTTP Status Code: 400

ResourceNotFoundException
Returned if the specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

Create export task to export data from a Log Group to Amazon S3 bucket.

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
```

```
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.CreateExportTask
{
  "taskName": "exampleTaskName",
  "logGroupName": "exampleLogGroupName",
  "from": 1437584472382,
  "to": 1437584472833,
  "destination": "exampleDestination",
  "destinationPrefix": "exampleDestinationPrefix"
}
```

Sample Response

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


{
  "taskId": "exampleTaskId"
}
```

CreateLogGroup

Creates a new log group with the specified name. The name of the log group must be unique within a region for an AWS account. You can create up to 500 log groups per account.

You must use the following guidelines when naming a log group:

- Log group names can be between 1 and 512 characters long.
- Allowed characters are a-z, A-Z, 0-9, '_' (underscore), '-' (hyphen), '/' (forward slash), and '.' (period).

Request Syntax

```
{
    "LogGroupName": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

LogGroupName
The name of the log group to create.

Type: String

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

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. 103).

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

LimitExceededException
Returned if you have reached the maximum number of resources that can be created.

HTTP Status Code: 400

OperationAbortedException
Returned if multiple requests to update the same resource were in conflict.

HTTP Status Code: 400

ResourceAlreadyExistsException
Returned if the specified resource already exists.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

Create a new Log Group

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.CreateLogGroup
{
  "logGroupName": "exampleLogGroupName"
}
```

Sample Response

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

CreateLogStream

Creates a new log stream in the specified log group. The name of the log stream must be unique within the log group. There is no limit on the number of log streams that can exist in a log group.

You must use the following guidelines when naming a log stream:

- Log stream names can be between 1 and 512 characters long.
- The ':' colon character is not allowed.

Request Syntax

```
{
    "LogGroupName": "string",
    "LogStreamName": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

LogGroupName
The name of the log group under which the log stream is to be created.

Type: String

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

Pattern: `[\.\-_/#A-Za-z0-9]+`

Required: Yes

LogStreamName
The name of the log stream to create.

Type: String

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

Pattern: `[^:*]*`

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. 103).

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

ResourceAlreadyExistsException
Returned if the specified resource already exists.

HTTP Status Code: 400

ResourceNotFoundException
Returned if the specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

Create a new Log Stream

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.CreateLogStream
{
  "logGroupName": "exampleLogGroupName",
  "logStreamName": "exampleLogStreamName"
}
```

Sample Response

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

DeleteDestination

Deletes the destination with the specified name and eventually disables all the subscription filters that publish to it. This will not delete the physical resource encapsulated by the destination.

Request Syntax

```
{
    "DestinationName": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

DestinationName
The name of destination to delete.

Type: String

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

Pattern: `[^:*]*`

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. 103).

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

OperationAbortedException
Returned if multiple requests to update the same resource were in conflict.

HTTP Status Code: 400

ResourceNotFoundException
Returned if the specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

Delete a Destination

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DeleteDestination
{
  "destinationName": "exampleDestinationName"
}
```

Sample Response

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

DeleteLogGroup

Deletes the log group with the specified name and permanently deletes all the archived log events associated with it.

Request Syntax

```
{
    "LogGroupName": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

LogGroupName
The name of the log group to delete.

Type: String

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

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. 103).

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

OperationAbortedException
Returned if multiple requests to update the same resource were in conflict.

HTTP Status Code: 400

ResourceNotFoundException
Returned if the specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

Delete a Log Group

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DeleteLogGroup
{
  "logGroupName": "exampleLogGroupName"
}
```

Sample Response

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

DeleteLogStream

Deletes a log stream and permanently deletes all the archived log events associated with it.

Request Syntax

```
{
    "LogGroupName": "string",
    "LogStreamName": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

LogGroupName
The name of the log group under which the log stream to delete belongs.

Type: String

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

Pattern: `[\.\-_/#A-Za-z0-9]+`

Required: Yes

LogStreamName
The name of the log stream to delete.

Type: String

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

Pattern: `[^:*]*`

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. 103).

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

OperationAbortedException
Returned if multiple requests to update the same resource were in conflict.

HTTP Status Code: 400

ResourceNotFoundException
Returned if the specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

Delete a Log Stream

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DeleteLogStream
{
  "logGroupName": "exampleLogGroupName",
  "logStreamName": "exampleLogStreamName"
}
```

Sample Response

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

DeleteMetricFilter

Deletes a metric filter associated with the specified log group.

Request Syntax

```
{
    "FilterName": "string",
    "LogGroupName": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

FilterName
The name of the metric filter to delete.

Type: String

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

Pattern: `[^:*]*`

Required: Yes

LogGroupName
The name of the log group that is associated with the metric filter to delete.

Type: String

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

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. 103).

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

OperationAbortedException
Returned if multiple requests to update the same resource were in conflict.

HTTP Status Code: 400

ResourceNotFoundException
Returned if the specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

Delete a metric filter

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DeleteMetricFilter
{
  "logGroupName": "exampleLogGroupName",
  "filterName": "exampleMetricFilterName"
}
```

Sample Response

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

DeleteRetentionPolicy

Deletes the retention policy of the specified log group. Log events would not expire if they belong to log groups without a retention policy.

Request Syntax

```
{
    "LogGroupName": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

LogGroupName
The name of the log group that is associated with the retention policy to delete.

Type: String

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

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. 103).

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

OperationAbortedException
Returned if multiple requests to update the same resource were in conflict.

HTTP Status Code: 400

ResourceNotFoundException
Returned if the specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

Deletes the retention policy of a log group

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DeleteRetentionPolicy
{
  "logGroupName": "exampleLogGroupName"
}
```

Sample Response

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

DeleteSubscriptionFilter

Deletes a subscription filter associated with the specified log group.

Request Syntax

```
{
    "FilterName": "string",
    "LogGroupName": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

FilterName
The name of the subscription filter to delete.

Type: String

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

Pattern: `[^:*]*`

Required: Yes

LogGroupName
The name of the log group that is associated with the subscription filter to delete.

Type: String

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

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. 103).

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

OperationAbortedException
Returned if multiple requests to update the same resource were in conflict.

HTTP Status Code: 400

ResourceNotFoundException
Returned if the specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

Delete a subscription filter

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DeleteSubscriptionFilter
{
  "logGroupName": "exampleLogGroupName",
  "filterName": "exampleSubscriptionFilterName"
}
```

Sample Response

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

DescribeDestinations

Returns all the destinations that are associated with the AWS account making the request. The list returned in the response is ASCII-sorted by destination name.

By default, this operation returns up to 50 destinations. If there are more destinations to list, the response would contain a `nextToken` value in the response body. You can also limit the number of destinations returned in the response by specifying the `limit` parameter in the request.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

DestinationNamePrefix
Will only return destinations that match the provided destinationNamePrefix. If you don't specify a value, no prefix is applied.

Type: String

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

Pattern: `[^:*]*`

Required: No

Limit
The maximum number of results to return.

Type: Number

Valid range: Minimum value of 1. Maximum value of 50.

Required: No

NextToken
A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous request. The token expires after 24 hours.

Type: String

Length constraints: Minimum length of 1.

Required: No

Response Syntax

```
{
    "Destinations": [
        {
            "AccessPolicy": "string",
            "Arn": "string",
            "CreationTime": number,
            "DestinationName": "string",
            "RoleArn": "string",
            "TargetArn": "string"
        }
    ],
    "NextToken": "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.

Destinations
Type: array of Destination (p. 86) objects

NextToken
A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous request. The token expires after 24 hours.

Type: String

Length constraints: Minimum length of 1.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 103).

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

List all the destinations

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DescribeDestinations
{
  "destinationNamePrefix": "exampleDestination"
}
```

Sample Response

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


{
  "destination": [
    {
      "destinationName": "exampleDestinationName",
      "targetArn": "arn:aws:kinesis:us-east-1:123456789:stream/kinesisStream
Name",
      "roleArn": "arn:aws:iam::123456789:role/subscriptionRoleName",
      "arn": "arn:aws:logs:us-east-1:123456789:destination:exampleDestination
Name",
      "creationTime": 1437584472382
    }
  ]
}
```

DescribeExportTasks

Returns all the export tasks that are associated with the AWS account making the request. The export tasks can be filtered based on `TaskId` or `TaskStatus`.

By default, this operation returns up to 50 export tasks that satisfy the specified filters. If there are more export tasks to list, the response would contain a `nextToken` value in the response body. You can also limit the number of export tasks returned in the response by specifying the `limit` parameter in the request.

Request Syntax

```
{
    "Limit": number,
    "NextToken": "string",
    "StatusCode": "string",
    "TaskId": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

Limit
The maximum number of items returned in the response. If you don't specify a value, the request would return up to 50 items.

Type: Number

Valid range: Minimum value of 1. Maximum value of 50.

Required: No

NextToken
A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous `DescribeExportTasks` request.

Type: String

Length constraints: Minimum length of 1.

Required: No

StatusCode
All export tasks that matches the specified status code will be returned. This can return zero or more export tasks.

Type: String

Valid Values: `CANCELLED | COMPLETED | FAILED | PENDING | PENDING_CANCEL | RUNNING`

Required: No

TaskId
Export task that matches the specified task Id will be returned. This can result in zero or one export task.

Type: String

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

Required: No

Response Syntax

```
{
    "ExportTasks": [
        {
            "Destination": "string",
            "DestinationPrefix": "string",
            "ExecutionInfo": {
                "CompletionTime": number,
                "CreationTime": number
            },
            "From": number,
            "LogGroupName": "string",
            "Status": {
                "Code": "string",
                "Message": "string"
            },
            "TaskId": "string",
            "TaskName": "string",
            "To": number
        }
    ],
    "NextToken": "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.

ExportTasks
A list of export tasks.

Type: array of ExportTask (p. 88) objects

NextToken
A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous request. The token expires after 24 hours.

Type: String

Length constraints: Minimum length of 1.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 103).

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

List all the export tasks that are COMPLETE.

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DescribeExportTasks
{
  "statusCode": "COMPLETE"
}
```

Sample Response

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


{
  "exportTasks": [
    {
      "taskId": "exampleTaskId_1",
      "taskName": "exampleTaskName_1",
      "logGroupName": "exampleLogGroupName",
      "from": 1437584472382,
      "to": 1437584472833,
      "destination": "exampleDestination",
```

```
      "destinationPrefix": "exampleDestinationPrefix",
      "status":
        {
          "code": "COMPLETE",
          "message": "exampleMessage"
        },
      "executionInfo":
        {
          "creationTime": 1437584472856,
          "completionTime" : 1437584472986
        }
    },

    {
      "taskId": "exampleTaskId_2",
      "taskName": "exampleTaskName_2",
      "logGroupName": "exampleLogGroupName",
      "from": 1437584472382,
      "to": 1437584472833,
      "destination": "exampleDestination",
      "destinationPrefix": "exampleDestinationPrefix",
      "status":
        {
          "code": "COMPLETE",
          "message": "exampleMessage"
        },
      "executionInfo":
        {
          "creationTime": 1437584472856,
          "completionTime" : 1437584472986
        }
    }
  ]
}
```

DescribeLogGroups

Returns all the log groups that are associated with the AWS account making the request. The list returned in the response is ASCII-sorted by log group name.

By default, this operation returns up to 50 log groups. If there are more log groups to list, the response would contain a `nextToken` value in the response body. You can also limit the number of log groups returned in the response by specifying the `limit` parameter in the request.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

Limit
The maximum number of items returned in the response. If you don't specify a value, the request would return up to 50 items.

Type: Number

Valid range: Minimum value of 1. Maximum value of 50.

Required: No

LogGroupNamePrefix
Will only return log groups that match the provided logGroupNamePrefix. If you don't specify a value, no prefix filter is applied.

Type: String

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

Pattern: `[\.\-_/#A-Za-z0-9]+`

Required: No

NextToken
A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous `DescribeLogGroups` request.

Type: String

Length constraints: Minimum length of 1.

Required: No

Response Syntax

```
{
    "LogGroups": [
        {
            "Arn": "string",
            "CreationTime": number,
            "LogGroupName": "string",
            "MetricFilterCount": number,
            "RetentionInDays": number,
            "StoredBytes": number
        }
    ],
    "NextToken": "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.

LogGroups
A list of log groups.

Type: array of LogGroup (p. 91) objects

NextToken
A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous request. The token expires after 24 hours.

Type: String

Length constraints: Minimum length of 1.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 103).

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

List all the log groups

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DescribeLogGroups
```

Sample Response

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


{
  "logGroups": [
    {
      "storageBytes": 1048576,
     "arn": "arn:aws:logs:us-east-1:123456789:log-group:exampleLogGroupName1:*",

      "creationTime": 1393545600000,
      "logGroupName": "exampleLogGroupName1",
      "metricFilterCount": 0,
      "retentionInDays": 14
    },
    {
      "storageBytes": 5242880,
     "arn": "arn:aws:logs:us-east-1:123456789:log-group:exampleLogGroupName2:*",

      "creationTime": 1396224000000,
      "logGroupName": "exampleLogGroupName2",
      "metricFilterCount": 0,
      "retentionInDays": 30
    }
  ]
```

```
}
```

DescribeLogStreams

Returns all the log streams that are associated with the specified log group. The list returned in the response is ASCII-sorted by log stream name.

By default, this operation returns up to 50 log streams. If there are more log streams to list, the response would contain a `nextToken` value in the response body. You can also limit the number of log streams returned in the response by specifying the `limit` parameter in the request. This operation has a limit of five transactions per second, after which transactions are throttled.

Request Syntax

```
{
    "Descending": boolean,
    "Limit": number,
    "LogGroupName": "string",
    "LogStreamNamePrefix": "string",
    "NextToken": "string",
    "OrderBy": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

Descending
If set to true, results are returned in descending order. If you don't specify a value or set it to false, results are returned in ascending order.

Type: Boolean

Required: No

Limit
The maximum number of items returned in the response. If you don't specify a value, the request would return up to 50 items.

Type: Number

Valid range: Minimum value of 1. Maximum value of 50.

Required: No

LogGroupName
The log group name for which log streams are to be listed.

Type: String

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

Pattern: `[\.\-_/#A-Za-z0-9]+`

Required: Yes

LogStreamNamePrefix
Will only return log streams that match the provided logStreamNamePrefix. If you don't specify a value, no prefix filter is applied.

Type: String

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

Pattern: `[^:*]*`

Required: No

NextToken
A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous `DescribeLogStreams` request.

Type: String

Length constraints: Minimum length of 1.

Required: No

OrderBy
Specifies what to order the returned log streams by. Valid arguments are 'LogStreamName' or 'LastEventTime'. If you don't specify a value, results are ordered by LogStreamName. If 'LastEventTime' is chosen, the request cannot also contain a logStreamNamePrefix.

Type: String

Valid Values: `LogStreamName | LastEventTime`

Required: No

Response Syntax

```
{
    "LogStreams": [
        {
            "Arn": "string",
            "CreationTime": number,
            "FirstEventTimestamp": number,
            "LastEventTimestamp": number,
            "LastIngestionTime": number,
            "LogStreamName": "string",
            "StoredBytes": number,
            "UploadSequenceToken": "string"
        }
    ],
    "NextToken": "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.

LogStreams
A list of log streams.

Type: array of LogStream (p. 92) objects

NextToken
A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous request. The token expires after 24 hours.

Type: String

Length constraints: Minimum length of 1.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 103).

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

ResourceNotFoundException
Returned if the specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

List the log streams associated with a log group

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DescribeLogStreams
{
  "logGroupName": "exampleLogGroupName"
}
```

Sample Response

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


{
  "logStreams": [
    {
      "storageBytes": 1048576,
      "arn": "arn:aws:logs:us-east-1:123456789:log-group:exampleLogGroup
Name1:log-stream:exampleLogStreamName1",
      "creationTime": 1393545600000,
      "firstEventTimestamp": 1393545600000,
      "lastEventTimestamp": 1393567800000,
      "lastIngestionTime": 1393589200000,
      "logStreamName": "exampleLogStreamName1",
      "uploadSequenceToken":
"88602967394531410094953670125156212707622379445839968487"
    },
    {
      "storageBytes": 5242880,
      "arn": "arn:aws:logs:us-east-1:123456789:log-group:exampleLogGroup
Name2:log-stream:exampleLogStreamName2",
      "creationTime": 1396224000000,
      "firstEventTimestamp": 1396224000000,
      "lastEventTimestamp": 1396235500000,
      "lastIngestionTime": 1396225560000,
      "logStreamName": "exampleLogStreamName2",
      "uploadSequenceToken":
"07622379445839968487886029673945314100949536701251562127"
    }
  ]
}
```

DescribeMetricFilters

Returns all the metrics filters associated with the specified log group. The list returned in the response is ASCII-sorted by filter name.

By default, this operation returns up to 50 metric filters. If there are more metric filters to list, the response would contain a `nextToken` value in the response body. You can also limit the number of metric filters returned in the response by specifying the `limit` parameter in the request.

Request Syntax

```
{
    "FilterNamePrefix": "string",
    "Limit": number,
    "LogGroupName": "string",
    "NextToken": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

FilterNamePrefix
Will only return metric filters that match the provided filterNamePrefix. If you don't specify a value, no prefix filter is applied.

Type: String

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

Pattern: `[^:*]*`

Required: No

Limit
The maximum number of items returned in the response. If you don't specify a value, the request would return up to 50 items.

Type: Number

Valid range: Minimum value of 1. Maximum value of 50.

Required: No

LogGroupName
The log group name for which metric filters are to be listed.

Type: String

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

Pattern: `[\.\-_/#A-Za-z0-9]+`

Required: Yes

NextToken
A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous `DescribeMetricFilters` request.

Type: String

Length constraints: Minimum length of 1.

Required: No

Response Syntax

```
{
    "MetricFilters": [
        {
            "CreationTime": number,
            "FilterName": "string",
            "FilterPattern": "string",
            "MetricTransformations": [
                {
                    "MetricName": "string",
                    "MetricNamespace": "string",
                    "MetricValue": "string"
                }
            ]
        }
    ],
    "NextToken": "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.

MetricFilters
Type: array of MetricFilter (p. 93) objects

NextToken
A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous request. The token expires after 24 hours.

Type: String

Length constraints: Minimum length of 1.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 103).

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

ResourceNotFoundException
Returned if the specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

List the metric filters associated with a log group

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DescribeMetricFilters
{
  "logGroupName": "exampleLogGroupName"
}
```

Sample Response

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


{
  "metricFilters": [
    {
      "creationTime": 1396224000000,
      "filterName": "exampleFilterName",
     "filterPattern": "[ip, identity, user_id, timestamp, request, status_code,
 size]",
      "metricTransformations": [
```

```
        {
          "metricValue": "$size",
          "metricNamespace": "MyApp",
          "metricName": "Volume"
        }
      ]
    }
  ]
}
```

DescribeSubscriptionFilters

Returns all the subscription filters associated with the specified log group. The list returned in the response is ASCII-sorted by filter name.

By default, this operation returns up to 50 subscription filters. If there are more subscription filters to list, the response would contain a `nextToken` value in the response body. You can also limit the number of subscription filters returned in the response by specifying the `limit` parameter in the request.

Request Syntax

```
{
    "FilterNamePrefix": "string",
    "Limit": number,
    "LogGroupName": "string",
    "NextToken": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

FilterNamePrefix
Will only return subscription filters that match the provided filterNamePrefix. If you don't specify a value, no prefix filter is applied.

Type: String

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

Pattern: `[^:*]*`

Required: No

Limit
The maximum number of results to return.

Type: Number

Valid range: Minimum value of 1. Maximum value of 50.

Required: No

LogGroupName
The log group name for which subscription filters are to be listed.

Type: String

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

Pattern: `[\.\-_/#A-Za-z0-9]+`

Required: Yes

NextToken
A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous request. The token expires after 24 hours.

Type: String

Length constraints: Minimum length of 1.

Required: No

Response Syntax

```
{
    "NextToken": "string",
    "SubscriptionFilters": [
        {
            "CreationTime": number,
            "DestinationArn": "string",
            "FilterName": "string",
            "FilterPattern": "string",
            "LogGroupName": "string",
            "RoleArn": "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
A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous request. The token expires after 24 hours.

Type: String

Length constraints: Minimum length of 1.

SubscriptionFilters
Type: array of SubscriptionFilter (p. 97) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 103).

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

ResourceNotFoundException
Returned if the specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

List the subscription filters associated with a log group

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DescribeSubscriptionFilters
{
  "logGroupName": "exampleLogGroupName"
}
```

Sample Response

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


{
  "subscriptionFilters": [
    {
      "creationTime": 1396224000000,
      "logGroupName": "exampleLogGroupName",
      "filterName": "exampleSubscriptionFilterName",
     "filterPattern": "[ip, identity, user_id, timestamp, request, status_code
 = 500, size]",
     "destinationArn": "arn:aws:kinesis:us-east-1:123456789:stream/kinesisStream
Name",
      "roleArn": "arn:aws:iam::123456789:role/subscriptionRoleName"
    }
```

```
  ]
}
```

FilterLogEvents

Retrieves log events, optionally filtered by a filter pattern from the specified log group. You can provide an optional time range to filter the results on the event `timestamp`. You can limit the streams searched to an explicit list of `logStreamNames`.

By default, this operation returns as much matching log events as can fit in a response size of 1MB, up to 10,000 log events, or all the events found within a time-bounded scan window. If the response includes a `nextToken`, then there is more data to search, and the search can be resumed with a new request providing the nextToken. The response will contain a list of `searchedLogStreams` that contains information about which streams were searched in the request and whether they have been searched completely or require further pagination. The `limit` parameter in the request. can be used to specify the maximum number of events to return in a page.

Request Syntax

```
{
    "EndTime": number,
    "FilterPattern": "string",
    "Interleaved": boolean,
    "Limit": number,
    "LogGroupName": "string",
    "LogStreamNames": [
        "string"
    ],
    "NextToken": "string",
    "StartTime": number
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

EndTime
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC. If provided, events with a timestamp later than this time are not returned.

Type: Long

Valid range: Minimum value of 0.

Required: No

FilterPattern
A valid CloudWatch Logs filter pattern to use for filtering the response. If not provided, all the events are matched.

Type: String

Length constraints: Minimum length of 0. Maximum length of 512.

Required: No

Interleaved

If provided, the API will make a best effort to provide responses that contain events from multiple log streams within the log group interleaved in a single response. If not provided, all the matched log events in the first log stream will be searched first, then those in the next log stream, etc.

Type: Boolean

Required: No

Limit

The maximum number of events to return in a page of results. Default is 10,000 events.

Type: Number

Valid range: Minimum value of 1. Maximum value of 10000.

Required: No

LogGroupName

The name of the log group to query.

Type: String

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

Pattern: `[\.\-_/#A-Za-z0-9]+`

Required: Yes

LogStreamNames

Optional list of log stream names within the specified log group to search. Defaults to all the log streams in the log group.

Type: array of Strings

Length constraints: Minimum of 1 item(s) in the list. Maximum of 100 item(s) in the list.

Required: No

NextToken

A pagination token obtained from a `FilterLogEvents` response to continue paginating the FilterLogEvents results. This token is omitted from the response when there are no other events to display.

Type: String

Length constraints: Minimum length of 1.

Required: No

StartTime

A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC. If provided, events with a timestamp prior to this time are not returned.

Type: Long

Valid range: Minimum value of 0.

Required: No

Response Syntax

```
{
    "Events": [
        {
            "EventId": "string",
            "IngestionTime": number,
            "LogStreamName": "string",
            "Message": "string",
            "Timestamp": number
        }
    ],
    "NextToken": "string",
    "SearchedLogStreams": [
        {
            "LogStreamName": "string",
            "SearchedCompletely": 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.

Events
A list of `FilteredLogEvent` objects representing the matched events from the request.

Type: array of FilteredLogEvent (p. 90) objects

NextToken
A pagination token obtained from a `FilterLogEvents` response to continue paginating the FilterLogEvents results. This token is omitted from the response when there are no other events to display.

Type: String

Length constraints: Minimum length of 1.

SearchedLogStreams
A list of `SearchedLogStream` objects indicating which log streams have been searched in this request and whether each has been searched completely or still has more to be paginated.

Type: array of SearchedLogStream (p. 96) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 103).

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

ResourceNotFoundException
Returned if the specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

Retrieves all the events containing 'ERROR' from a log group

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.FilterLogEvents
{
  "logGroupName": "exampleLogGroupName",
  "filterPattern": "ERROR"
}
```

Sample Response

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


{
  "events": [
    {
      "ingestionTime": 1396035394997,
      "timestamp": 1396035378988,
      "message": "ERROR Event 1",
      "logStreamName": "exampleLogStreamName1",
```

```
      "eventId": "31132629274945519779805322857203735586714454643391594505"
    },
    {
      "ingestionTime": 1396035394997,
      "timestamp": 1396035378988,
      "message": "ERROR Event 2",
      "logStreamName": "exampleLogStreamName2",
      "eventId": "31132629274945519779805322857203735586814454643391594505"
    },
    {
      "ingestionTime": 1396035394997,
      "timestamp": 1396035378989,
      "message": "ERROR Event 3",
      "logStreamName": "exampleLogStreamName3"
      "eventId": "31132629274945519779805322857203735586824454643391594505"
    }
  ],
  "searchedLogStreams": [
    {
      "searchedCompletely": true,
      "logStreamName": "exampleLogStreamName1"
    },
    {
      "searchedCompletely": true,
      "logStreamName": "exampleLogStreamName2"
    },
    {
      "searchedCompletely": false,
      "logStreamName": "exampleLogStreamName3"
    },
  ],
  "nextToken": "ZNUEPl7FcQuXbIH4Swk9D9eFu2XBg-ijZIZlvzz4ea9zZRjw-MMtQtvcoMd
mq4T29K7Q6Y1e_KvyfpcT_f_tUw"
}
```

GetLogEvents

Retrieves log events from the specified log stream. You can provide an optional time range to filter the results on the event `timestamp`.

By default, this operation returns as much log events as can fit in a response size of 1MB, up to 10,000 log events. The response will always include a `nextForwardToken` and a `nextBackwardToken` in the response body. You can use any of these tokens in subsequent `GetLogEvents` requests to paginate through events in either forward or backward direction. You can also limit the number of log events returned in the response by specifying the `limit` parameter in the request.

Request Syntax

```
{
    "EndTime": number,
    "Limit": number,
    "LogGroupName": "string",
    "LogStreamName": "string",
    "NextToken": "string",
    "StartFromHead": boolean,
    "StartTime": number
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

EndTime
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

Type: Long

Valid range: Minimum value of 0.

Required: No

Limit
The maximum number of log events returned in the response. If you don't specify a value, the request would return as many log events as can fit in a response size of 1MB, up to 10,000 log events.

Type: Number

Valid range: Minimum value of 1. Maximum value of 10000.

Required: No

LogGroupName
The name of the log group to query.

Type: String

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

Pattern: `[\.\-_/#A-Za-z0-9]+`

Required: Yes

LogStreamName

The name of the log stream to query.

Type: String

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

Pattern: `[^:*]*`

Required: Yes

NextToken

A string token used for pagination that points to the next page of results. It must be a value obtained from the `nextForwardToken` or `nextBackwardToken` fields in the response of the previous `GetLogEvents` request.

Type: String

Length constraints: Minimum length of 1.

Required: No

StartFromHead

If set to true, the earliest log events would be returned first. The default is false (the latest log events are returned first).

Type: Boolean

Required: No

StartTime

A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

Type: Long

Valid range: Minimum value of 0.

Required: No

Response Syntax

```
{
    "Events": [
        {
            "IngestionTime": number,
            "Message": "string",
            "Timestamp": number
        }
    ],
    "NextBackwardToken": "string",
    "NextForwardToken": "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.

Events
Type: array of OutputLogEvent (p. 95) objects

NextBackwardToken
A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous request. The token expires after 24 hours.

Type: String

Length constraints: Minimum length of 1.

NextForwardToken
A string token used for pagination that points to the next page of results. It must be a value obtained from the response of the previous request. The token expires after 24 hours.

Type: String

Length constraints: Minimum length of 1.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 103).

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

ResourceNotFoundException
Returned if the specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

Retrieves all the events from a log stream

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
```

```
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.GetLogEvents
{
  "logGroupName": "exampleLogGroupName",
  "logStreamName": "exampleLogStreamName"
}
```

Sample Response

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


{
  "events": [
    {
      "ingestionTime": 1396035394997,
      "timestamp": 1396035378988,
      "message": "Example Event 1"
    },
    {
      "ingestionTime": 1396035394997,
      "timestamp": 1396035378988,
      "message": "Example Event 2"
    },
    {
      "ingestionTime": 1396035394997,
      "timestamp": 1396035378989,
      "message": "Example Event 3"
    }
  ],
  "nextBackwardToken":
"b/31132629274945519779805322857203735586714454643391594505",
  "nextForwardToken":
"f/31132629323784151764587387538205132201699397759403884544"
}
```

PutDestination

Creates or updates a `Destination`. A destination encapsulates a physical resource (such as a Kinesis stream) and allows you to subscribe to a real-time stream of log events of a different account, ingested through `PutLogEvents` requests. Currently, the only supported physical resource is a Amazon Kinesis stream belonging to the same account as the destination.

A destination controls what is written to its Amazon Kinesis stream through an access policy. By default, PutDestination does not set any access policy with the destination, which means a cross-account user will not be able to call `PutSubscriptionFilter` against this destination. To enable that, the destination owner must call `PutDestinationPolicy` after PutDestination.

Request Syntax

```
{
    "DestinationName": "string",
    "RoleArn": "string",
    "TargetArn": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

DestinationName
A name for the destination.

Type: String

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

Pattern: `[^:*]*`

Required: Yes

RoleArn
The ARN of an IAM role that grants Amazon CloudWatch Logs permissions to do Amazon Kinesis PutRecord requests on the desitnation stream.

Type: String

Length constraints: Minimum length of 1.

Required: Yes

TargetArn
The ARN of an Amazon Kinesis stream to deliver matching log events to.

Type: String

Length constraints: Minimum length of 1.

Required: Yes

Response Syntax

```
{
    "Destination": {
        "AccessPolicy": "string",
        "Arn": "string",
        "CreationTime": number,
        "DestinationName": "string",
        "RoleArn": "string",
        "TargetArn": "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.

Destination
A cross account destination that is the recipient of subscription log events.

Type: Destination (p. 86) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 103).

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

OperationAbortedException
Returned if multiple requests to update the same resource were in conflict.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

Create or update a destination

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.PutDestination
{
  "destinationName": "exampleDestinationName",
  "targetArn": "arn:aws:kinesis:us-east-1:123456789:stream/kinesisStreamName",

  "roleArn": "arn:aws:iam::123456789:role/subscriptionRoleName"
}
```

Sample Response

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


{
  "destination": [
    {
      "destinationName": "exampleDestinationName",
      "targetArn": "arn:aws:kinesis:us-east-1:123456789:stream/kinesisStream
Name",
      "roleArn": "arn:aws:iam::123456789:role/subscriptionRoleName",
      "arn": "arn:aws:logs:us-east-1:123456789:destination:exampleDestination
Name",
      "creationTime": 1437584472382
    }
  ]
}
```

PutDestinationPolicy

Creates or updates an access policy associated with an existing `Destination`. An access policy is an IAM policy document that is used to authorize claims to register a subscription filter against a given destination.

Request Syntax

```
{
    "AccessPolicy": "string",
    "DestinationName": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

AccessPolicy
An IAM policy document that authorizes cross-account users to deliver their log events to associated destination.

Type: String

Length constraints: Minimum length of 1.

Required: Yes

DestinationName
A name for an existing destination.

Type: String

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

Pattern: `[^:*]*`

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. 103).

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

OperationAbortedException
Returned if multiple requests to update the same resource were in conflict.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

Create or update an access policy of a destination

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.PutDestinationPolicy
{
  "destinationName": "exampleDestinationName",
  "accessPolicy": "{ \"Version\": \"2012-10-17\", \"Statement\": [{ \"Sid\":
\"\", \"Effect\": \"Allow\", \"Principal\": { \"AWS\": \"logs.us-east-
1.amazonaws.com\"}, \"Action\": \"logs:PutSubscriptionFilter\",\"Resource\":
\"arn:aws:logs:us-east-1:123456789:destination:exampleDestinationName\"}]}"
}
```

Sample Response

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

PutLogEvents

Uploads a batch of log events to the specified log stream.

Every PutLogEvents request must include the `sequenceToken` obtained from the response of the previous request. An upload in a newly created log stream does not require a `sequenceToken`.

The batch of events must satisfy the following constraints:

- The maximum batch size is 1,048,576 bytes, and this size is calculated as the sum of all event messages in UTF-8, plus 26 bytes for each log event.
- None of the log events in the batch can be more than 2 hours in the future.
- None of the log events in the batch can be older than 14 days or the retention period of the log group.
- The log events in the batch must be in chronological ordered by their `timestamp`.
- The maximum number of log events in a batch is 10,000.
- A batch of log events in a single PutLogEvents request cannot span more than 24 hours. Otherwise, the PutLogEvents operation will fail.

Request Syntax

```
{
    "LogEvents": [
        {
            "Message": "string",
            "Timestamp": number
        }
    ],
    "LogGroupName": "string",
    "LogStreamName": "string",
    "SequenceToken": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

LogEvents
A list of log events belonging to a log stream.

Type: array of InputLogEvent (p. 91) objects

Length constraints: Minimum of 1 item(s) in the list. Maximum of 10000 item(s) in the list.

Required: Yes

LogGroupName
The name of the log group to put log events to.

Type: String

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

Pattern: `[\.\-_/#A-Za-z0-9]+`

Required: Yes

LogStreamName

The name of the log stream to put log events to.

Type: String

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

Pattern: `[^:*]*`

Required: Yes

SequenceToken

A string token that must be obtained from the response of the previous `PutLogEvents` request.

Type: String

Length constraints: Minimum length of 1.

Required: No

Response Syntax

```
{
    "NextSequenceToken": "string",
    "RejectedLogEventsInfo": {
        "ExpiredLogEventEndIndex": number,
        "TooNewLogEventStartIndex": number,
        "TooOldLogEventEndIndex": 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.

NextSequenceToken

A string token used for making PutLogEvents requests. A `sequenceToken` can only be used once, and PutLogEvents requests must include the `sequenceToken` obtained from the response of the previous request.

Type: String

Length constraints: Minimum length of 1.

RejectedLogEventsInfo

Type: RejectedLogEventsInfo (p. 96) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 103).

DataAlreadyAcceptedException
HTTP Status Code: 400

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

InvalidSequenceTokenException
HTTP Status Code: 400

OperationAbortedException
Returned if multiple requests to update the same resource were in conflict.

HTTP Status Code: 400

ResourceNotFoundException
Returned if the specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

Upload a batch of log events into a log stream

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.PutLogEvents
{
  "logGroupName": "exampleLogGroupName",
  "logStreamName": "exampleLogStreamName",
  "logEvents": [
    {
      "timestamp": 1396035378988,
      "message": "Example Event 1"
    },
    {
      "timestamp": 1396035378988,
      "message": "Example Event 2"
    },
```

```
    {
      "timestamp": 1396035378989,
      "message": "Example Event 3"
    }
  ]
}
```

Sample Response

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


{
  "nextSequenceToken": "49536701251539826331025683274032969384950891766572122113"
}
```

PutMetricFilter

Creates or updates a metric filter and associates it with the specified log group. Metric filters allow you to configure rules to extract metric data from log events ingested through `PutLogEvents` requests.

The maximum number of metric filters that can be associated with a log group is 100.

Request Syntax

```
{
    "FilterName": "string",
    "FilterPattern": "string",
    "LogGroupName": "string",
    "MetricTransformations": [
        {
            "MetricName": "string",
            "MetricNamespace": "string",
            "MetricValue": "string"
        }
    ]
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

FilterName
A name for the metric filter.

Type: String

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

Pattern: `[^:*]*`

Required: Yes

FilterPattern
A valid CloudWatch Logs filter pattern for extracting metric data out of ingested log events.

Type: String

Length constraints: Minimum length of 0. Maximum length of 512.

Required: Yes

LogGroupName
The name of the log group to associate the metric filter with.

Type: String

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

Pattern: `[\.\-_/#A-Za-z0-9]+`

Required: Yes

MetricTransformations
A collection of information needed to define how metric data gets emitted.

Type: array of MetricTransformation (p. 95) objects

Length constraints: Minimum of 1 item(s) in the list. Maximum of 1 item(s) in the list.

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. 103).

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

LimitExceededException
Returned if you have reached the maximum number of resources that can be created.

HTTP Status Code: 400

OperationAbortedException
Returned if multiple requests to update the same resource were in conflict.

HTTP Status Code: 400

ResourceNotFoundException
Returned if the specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

Create or update a metric filter

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
```

```
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.PutMetricFilter
{
  "logGroupName": "exampleLogGroupName",
  "filterName": "exampleMetricFilterName",
  "filterPattern": "[ip, identity, user_id, timestamp, request, status_code,
size]",
  "metricTransformations": [
    {
      "metricValue": "$size",
      "metricNamespace": "MyApp",
      "metricName": "Volume"
    }
  ]
}
```

Sample Response

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

PutRetentionPolicy

Sets the retention of the specified log group. A retention policy allows you to configure the number of days you want to retain log events in the specified log group.

Request Syntax

```
{
    "LogGroupName": "string",
    "RetentionInDays": number
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

LogGroupName
The name of the log group to associate the retention policy with.

Type: String

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

Pattern: `[\.\-_/#A-Za-z0-9]+`

Required: Yes

RetentionInDays
Specifies the number of days you want to retain log events in the specified log group. Possible values are: 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 3653.

Type: Number

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. 103).

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

OperationAbortedException
Returned if multiple requests to update the same resource were in conflict.

HTTP Status Code: 400

ResourceNotFoundException
Returned if the specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

Creates or updates a 30 day retention policy for a log group

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.PutRetentionPolicy
{
  "logGroupName": "exampleLogGroupName",
  "retentionInDays": 30
}
```

Sample Response

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

PutSubscriptionFilter

Creates or updates a subscription filter and associates it with the specified log group. Subscription filters allow you to subscribe to a real-time stream of log events ingested through `PutLogEvents` requests and have them delivered to a specific destination. Currently, the supported destinations are:

- A Amazon Kinesis stream belonging to the same account as the subscription filter, for same-account delivery.
- A logical destination (used via an ARN of `Destination`) belonging to a different account, for cross-account delivery.

Currently there can only be one subscription filter associated with a log group.

Request Syntax

```
{
    "DestinationArn": "string",
    "FilterName": "string",
    "FilterPattern": "string",
    "LogGroupName": "string",
    "RoleArn": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

DestinationArn

The ARN of the destination to deliver matching log events to. Currently, the supported destinations are:

- A Amazon Kinesis stream belonging to the same account as the subscription filter, for same-account delivery.
- A logical destination (used via an ARN of `Destination`) belonging to a different account, for cross-account delivery.

Type: String

Length constraints: Minimum length of 1.

Required: Yes

FilterName

A name for the subscription filter.

Type: String

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

Pattern: `[^:*]*`

Required: Yes

FilterPattern
A valid CloudWatch Logs filter pattern for subscribing to a filtered stream of log events.

Type: String

Length constraints: Minimum length of 0. Maximum length of 512.

Required: Yes

LogGroupName
The name of the log group to associate the subscription filter with.

Type: String

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

Pattern: `[\.\-_/#A-Za-z0-9]+`

Required: Yes

RoleArn
The ARN of an IAM role that grants Amazon CloudWatch Logs permissions to deliver ingested log events to the destination stream. You don't need to provide the ARN when you are working with a logical destination (used via an ARN of `Destination`) for cross-account delivery.

Type: String

Length constraints: Minimum length of 1.

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. 103).

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

LimitExceededException
Returned if you have reached the maximum number of resources that can be created.

HTTP Status Code: 400

OperationAbortedException
Returned if multiple requests to update the same resource were in conflict.

HTTP Status Code: 400

ResourceNotFoundException
Returned if the specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

Create or update a subscription filter

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.PutSubscriptionFilter
{
  "logGroupName": "exampleLogGroupName",
  "filterName": "exampleSubscriptionFilterName",
  "filterPattern": "[ip, identity, user_id, timestamp, request, status_code =
500, size]",
  "destinationArn": "arn:aws:kinesis:us-east-1:123456789:stream/kinesisStream
Name",
  "roleArn": "arn:aws:iam::123456789:role/subscriptionRoleName"
}
```

Sample Response

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

TestMetricFilter

Tests the filter pattern of a metric filter against a sample of log event messages. You can use this operation to validate the correctness of a metric filter pattern.

Request Syntax

```
{
    "FilterPattern": "string",
    "LogEventMessages": [
        "string"
    ]
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 99).

The request requires the following data in JSON format.

FilterPattern
A symbolic description of how Amazon CloudWatch Logs should interpret the data in each log event. For example, a log event may contain timestamps, IP addresses, strings, and so on. You use the filter pattern to specify what to look for in the log event message.

Type: String

Length constraints: Minimum length of 0. Maximum length of 512.

Required: Yes

LogEventMessages
A list of log event messages to test.

Type: array of Strings

Length constraints: Minimum of 1 item(s) in the list. Maximum of 50 item(s) in the list.

Required: Yes

Response Syntax

```
{
    "Matches": [
        {
            "EventMessage": "string",
            "EventNumber": number,
            "ExtractedValues":
                {
                    "string" :
                        "string"
                }
```

```
        }
    ]
}
```

Response Elements

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

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

Matches
Type: array of MetricFilterMatchRecord (p. 94) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 103).

InvalidParameterException
Returned if a parameter of the request is incorrectly specified.

HTTP Status Code: 400

ServiceUnavailableException
Returned if the service cannot complete the request.

HTTP Status Code: 500

Examples

Test a metric filter pattern on Apache access.log events

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.TestMetricFilter
{
  "filterPattern": "[ip, identity, user_id, timestamp, request, status_code,
size]",
  "logEventMessages": [
    "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET /apache_pb.gif HT
```

```
TP/1.0\" 200 1534",
    "127.0.0.1 - frank [10/Oct/2000:13:35:22 -0700] \"GET /apache_pb.gif HT
TP/1.0\" 500 5324",
    "127.0.0.1 - frank [10/Oct/2000:13:50:35 -0700] \"GET /apache_pb.gif HT
TP/1.0\" 200 4355"
  ]
}
```

Sample Response

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


{
  "matches": [
    {
      "eventNumber": 0,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET
/apache_pb.gif HTTP/1.0\" 200 1534",
      "extractedValues": {
        "$status_code": "200",
        "$identity": "-",
        "$request": "GET /apache_pb.gif HTTP/1.0",
        "$size": "1534,",
        "$user_id": "frank",
        "$ip": "127.0.0.1",
        "$timestamp": "10/Oct/2000:13:25:15 -0700"
      }
    },
    {
      "eventNumber": 1,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:35:22 -0700] \"GET
/apache_pb.gif HTTP/1.0\" 500 5324",
      "extractedValues": {
        "$status_code": "500",
        "$identity": "-",
        "$request": "GET /apache_pb.gif HTTP/1.0",
        "$size": "5324,",
        "$user_id": "frank",
        "$ip": "127.0.0.1",
        "$timestamp": "10/Oct/2000:13:35:22 -0700"
      }
    },
    {
      "eventNumber": 2,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:50:35 -0700] \"GET
/apache_pb.gif HTTP/1.0\" 200 4355",
      "extractedValues": {
        "$status_code": "200",
```

```
        "$identity": "-",
        "$request": "GET /apache_pb.gif HTTP/1.0",
        "$size": "4355",
        "$user_id": "frank",
        "$ip": "127.0.0.1",
        "$timestamp": "10/Oct/2000:13:50:35 -0700"
      }
    }
  ]
}
```

Test a metric filter pattern on Apache access.log events without specifying all the fields

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.TestMetricFilter
{
  "filterPattern": "[..., size]",
  "logEventMessages": [
    "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET /apache_pb.gif HT
TP/1.0\" 200 1534",
    "127.0.0.1 - frank [10/Oct/2000:13:35:22 -0700] \"GET /apache_pb.gif HT
TP/1.0\" 500 5324",
    "127.0.0.1 - frank [10/Oct/2000:13:50:35 -0700] \"GET /apache_pb.gif HT
TP/1.0\" 200 4355"
  ]
}
```

Sample Response

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

```
Date: <Date>


{
  "matches": [
    {
      "eventNumber": 0,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET
/apache_pb.gif HTTP/1.0\" 200 1534",
      "extractedValues": {
        "$size": "1534",
        "$6": "200",
        "$4": "10/Oct/2000:13:25:15 -0700",
        "$5": "GET /apache_pb.gif HTTP/1.0",
        "$2": "-",
        "$3": "frank",
        "$1": "127.0.0.1"
      }
    },
    {
      "eventNumber": 1,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:35:22 -0700] \"GET
/apache_pb.gif HTTP/1.0\" 500 5324",
      "extractedValues": {
        "$size": "5324",
        "$6": "500",
        "$4": "10/Oct/2000:13:35:22 -0700",
        "$5": "GET /apache_pb.gif HTTP/1.0",
        "$2": "-",
        "$3": "frank",
        "$1": "127.0.0.1"
      }
    },
    {
      "eventNumber": 2,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:50:35 -0700] \"GET
/apache_pb.gif HTTP/1.0\" 200 4355",
      "extractedValues": {
        "$size": "4355",
        "$6": "200",
        "$4": "10/Oct/2000:13:50:35 -0700",
        "$5": "GET /apache_pb.gif HTTP/1.0",
        "$2": "-",
        "$3": "frank",
        "$1": "127.0.0.1"
      }
    }
  ]
}
```

Test a metric filter pattern on Apache access.log events without specifying any fields

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.TestMetricFilter
{
  "filterPattern": "[]",
  "logEventMessages": [
    "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET /apache_pb.gif HT
TP/1.0\" 200 1534",
    "127.0.0.1 - frank [10/Oct/2000:13:35:22 -0700] \"GET /apache_pb.gif HT
TP/1.0\" 500 5324",
    "127.0.0.1 - frank [10/Oct/2000:13:50:35 -0700] \"GET /apache_pb.gif HT
TP/1.0\" 200 4355"
  ]
}
```

Sample Response

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


{
  "matches": [
    {
      "eventNumber": 0,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET
/apache_pb.gif HTTP/1.0\" 200 1534",
      "extractedValues": {
        "$7": "1534",
        "$6": "200",
        "$4": "10/Oct/2000:13:25:15 -0700",
        "$5": "GET /apache_pb.gif HTTP/1.0",
        "$2": "-",
        "$3": "frank",
        "$1": "127.0.0.1"
      }
    },
    {
```

```
      "eventNumber": 1,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:35:22 -0700] \"GET
/apache_pb.gif HTTP/1.0\" 500 5324",
      "extractedValues": {
        "$7": "5324",
        "$6": "500",
        "$4": "10/Oct/2000:13:35:22 -0700",
        "$5": "GET /apache_pb.gif HTTP/1.0",
        "$2": "-",
        "$3": "frank",
        "$1": "127.0.0.1"
      }
    },
    {
      "eventNumber": 2,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:50:35 -0700] \"GET
/apache_pb.gif HTTP/1.0\" 200 4355",
      "extractedValues": {
        "$7": "4355",
        "$6": "200",
        "$4": "10/Oct/2000:13:50:35 -0700",
        "$5": "GET /apache_pb.gif HTTP/1.0",
        "$2": "-",
        "$3": "frank",
        "$1": "127.0.0.1"
      }
    }
  ]
}
```

Test a metric filter pattern that matches successful requests in Apache access.log events

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.TestMetricFilter
{
  "filterPattern": "[..., status_code=200, size]",
  "logEventMessages": [
```

```
    "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET /apache_pb.gif HT
TP/1.0\" 200 1534",
    "127.0.0.1 - frank [10/Oct/2000:13:35:22 -0700] \"GET /apache_pb.gif HT
TP/1.0\" 500 5324",
    "127.0.0.1 - frank [10/Oct/2000:13:50:35 -0700] \"GET /apache_pb.gif HT
TP/1.0\" 200 4355"
  ]
}
```

Sample Response

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


{
  "matches": [
    {
      "eventNumber": 0,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET
/apache_pb.gif HTTP/1.0\" 200 1534",
      "extractedValues": {
        "$status_code": "200",
        "$size": "1534",
        "$4": "10/Oct/2000:13:25:15 -0700",
        "$5": "GET /apache_pb.gif HTTP/1.0",
        "$2": "-",
        "$3": "frank",
        "$1": "127.0.0.1"
      }
    },
    {
      "eventNumber": 2,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:50:35 -0700] \"GET
/apache_pb.gif HTTP/1.0\" 200 4355",
      "extractedValues": {
        "$status_code": "200",
        "$size": "4355",
        "$4": "10/Oct/2000:13:50:35 -0700",
        "$5": "GET /apache_pb.gif HTTP/1.0",
        "$2": "-",
        "$3": "frank",
        "$1": "127.0.0.1"
      }
    }
  ]
}
```

Test a metric filter pattern that matches 4XX response codes for html pages in Apache access.log events

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.TestMetricFilter
{
  "filterPattern": "[..., request=*.html*, status_code=4*,]",
  "logEventMessages": [
    "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET /index.html HTTP/1.0\"
 404 1534",
    "127.0.0.1 - frank [10/Oct/2000:13:35:22 -0700] \"GET /about-us/index.html
 HTTP/1.0\" 200 5324",
    "127.0.0.1 - frank [10/Oct/2000:13:50:35 -0700] \"GET /apache_pb.gif HT
TP/1.0\" 404 4355",
    "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET /products/index.html
 HTTP/1.0\" 400 1534",
  ]
}
```

Sample Response

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


{
  "matches": [
    {
      "eventNumber": 0,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET
/index.html HTTP/1.0\" 404 1534",
      "extractedValues": {
        "$status_code": "404",
        "$request": "GET /index.html HTTP/1.0",
```

```
        "$7": "1534",
        "$4": "10/Oct/2000:13:25:15 -0700",
        "$2": "-",
        "$3": "frank",
        "$1": "127.0.0.1"
      }
    },
    {
      "eventNumber": 3,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET
/products/index.html HTTP/1.0\" 400 1534",
      "extractedValues": {
        "$status_code": "400",
        "$request": "GET /products/index.html HTTP/1.0",
        "$7": "1534",
        "$4": "10/Oct/2000:13:25:15 -0700",
        "$2": "-",
        "$3": "frank",
        "$1": "127.0.0.1"
      }
    }
  ]
}
```

Test a metric filter pattern that matches occurrences of "[ERROR]" in log events

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.TestMetricFilter
{
  "filterPattern": "\"[ERROR]\"",
  "logEventMessages": [
    "02 May 2014 00:34:12,525 [INFO] Starting the application",
    "02 May 2014 00:35:14,245 [DEBUG] Database connection established",
    "02 May 2014 00:34:14,663 [INFO] Executing SQL Query",
   "02 May 2014 00:34:16,142 [ERROR] Unhanded exception: InvalidQueryException",

    "02 May 2014 00:34:16,224 [ERROR] Terminating the application"
```

```
  ]
}
```

Sample Response

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


{
  "matches": [
    {
      "eventNumber": 3,
      "eventMessage": "02 May 2014 00:34:16,142 [ERROR] Unhanded exception:
InvalidQueryException",
      "extractedValues": {}
    },
    {
      "eventNumber": 4,
     "eventMessage": "02 May 2014 00:34:16,224 [ERROR] Terminating the applic
ation",
      "extractedValues": {}
    }
  ]
}
```

Test a metric filter pattern that matches occurrences of "[ERROR]" and "Exception" in log events

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

Sample Request

```
POST / HTTP/1.1
Host: logs.<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, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.TestMetricFilter
```

```
{
  "filterPattern": "\"[ERROR]\" Exception",
  "logEventMessages": [
    "02 May 2014 00:34:12,525 [INFO] Starting the application",
    "02 May 2014 00:35:14,245 [DEBUG] Database connection established",
    "02 May 2014 00:34:14,663 [INFO] Executing SQL Query",
   "02 May 2014 00:34:16,142 [ERROR] Unhanded exception: InvalidQueryException",

    "02 May 2014 00:34:16,224 [ERROR] Terminating the application"
  ]
}
```

Sample Response

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


{
  "matches": [
    {
      "eventNumber": 3,
      "eventMessage": "02 May 2014 00:34:16,142 [ERROR] Unhanded exception:
InvalidQueryException",
      "extractedValues": {}
    }
  ]
}
```

Data Types

The Amazon CloudWatch Logs API Reference API contains several data types that various actions use. This section describes each data type in detail.

> **Note**
> The order of each element in the response is not guaranteed. Applications should not assume a particular order.

The following data types are supported:

Destination

Description

A cross account destination that is the recipient of subscription log events.

Contents

AccessPolicy
An IAM policy document that governs which AWS accounts can create subscription filters against this destination.

Type: String

Length constraints: Minimum length of 1.

Required: No

Arn
ARN of this destination.

Type: String

Required: No

CreationTime
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC specifying when this destination was created.

Type: Long

Valid range: Minimum value of 0.

Required: No

DestinationName
Name of the destination.

Type: String

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

Pattern: `[^:*]*`

Required: No

RoleArn
A role for impersonation for delivering log events to the target.

Type: String

Length constraints: Minimum length of 1.

Required: No

TargetArn
ARN of the physical target where the log events will be delivered (eg. ARN of a Kinesis stream).

Type: String

Length constraints: Minimum length of 1.

Required: No

ExportTask

Description

Represents an export task.

Contents

Destination
Name of Amazon S3 bucket to which the log data was exported.

Type: String

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

Required: No

DestinationPrefix
Prefix that was used as the start of Amazon S3 key for every object exported.

Type: String

Required: No

ExecutionInfo
Execution info about the export task.

Type: ExportTaskExecutionInfo (p. 89) object

Required: No

From
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC. Events with a timestamp prior to this time are not exported.

Type: Long

Valid range: Minimum value of 0.

Required: No

LogGroupName
The name of the log group from which logs data was exported.

Type: String

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

Pattern: `[\.\-_/#A-Za-z0-9]+`

Required: No

Status
Status of the export task.

Type: ExportTaskStatus (p. 89) object

Required: No

TaskId
Id of the export task.

Type: String

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

Required: No

TaskName
The name of the export task.

Type: String

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

Required: No

To
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC. Events with a timestamp later than this time are not exported.

Type: Long

Valid range: Minimum value of 0.

Required: No

ExportTaskExecutionInfo

Description

Represents the status of an export task.

Contents

CompletionTime
A point in time when the export task got completed.

Type: Long

Valid range: Minimum value of 0.

Required: No

CreationTime
A point in time when the export task got created.

Type: Long

Valid range: Minimum value of 0.

Required: No

ExportTaskStatus

Description

Represents the status of an export task.

Contents

Code
Status code of the export task.

Type: String

Valid Values: `CANCELLED | COMPLETED | FAILED | PENDING | PENDING_CANCEL | RUNNING`

Required: No

Message
Status message related to the `code`.

Type: String

Required: No

FilteredLogEvent

Description

Represents a matched event from a `FilterLogEvents` request.

Contents

EventId
A unique identifier for this event.

Type: String

Required: No

IngestionTime
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

Type: Long

Valid range: Minimum value of 0.

Required: No

LogStreamName
The name of the log stream this event belongs to.

Type: String

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

Pattern: `[^:*]*`

Required: No

Message
The data contained in the log event.

Type: String

Length constraints: Minimum length of 1.

Required: No

Timestamp
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

Type: Long

Valid range: Minimum value of 0.

Required: No

InputLogEvent

Description

A log event is a record of some activity that was recorded by the application or resource being monitored. The log event record that Amazon CloudWatch Logs understands contains two properties: the timestamp of when the event occurred, and the raw event message.

Contents

Message
Type: String

Length constraints: Minimum length of 1.

Required: Yes

Timestamp
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

Type: Long

Valid range: Minimum value of 0.

Required: Yes

LogGroup

Description

No action documentation available.

Contents

Arn
Type: String

Required: No

CreationTime
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

Type: Long

Valid range: Minimum value of 0.

Required: No

LogGroupName
Type: String

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

Pattern: `[\.\-_/#A-Za-z0-9]+`

Required: No

MetricFilterCount
The number of metric filters associated with the log group.

Type: Number

Required: No

RetentionInDays
Specifies the number of days you want to retain log events in the specified log group. Possible values are: 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 3653.

Type: Number

Required: No

StoredBytes
Type: Long

Valid range: Minimum value of 0.

Required: No

LogStream

Description

A log stream is sequence of log events from a single emitter of logs.

Contents

Arn
Type: String

Required: No

CreationTime
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

Type: Long

Valid range: Minimum value of 0.

Required: No

FirstEventTimestamp
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

Type: Long

Valid range: Minimum value of 0.

Required: No

LastEventTimestamp
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

Type: Long

Valid range: Minimum value of 0.

Required: No

LastIngestionTime
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

Type: Long

Valid range: Minimum value of 0.

Required: No

LogStreamName
Type: String

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

Pattern: `[^:*]*`

Required: No

StoredBytes
Type: Long

Valid range: Minimum value of 0.

Required: No

UploadSequenceToken
A string token used for making PutLogEvents requests. A `sequenceToken` can only be used once, and PutLogEvents requests must include the `sequenceToken` obtained from the response of the previous request.

Type: String

Length constraints: Minimum length of 1.

Required: No

MetricFilter

Description

Metric filters can be used to express how Amazon CloudWatch Logs would extract metric observations from ingested log events and transform them to metric data in a CloudWatch metric.

Contents

CreationTime
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

Type: Long

Valid range: Minimum value of 0.

Required: No

FilterName
A name for a metric or subscription filter.

Type: String

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

Pattern: `[^:*]*`

Required: No

FilterPattern
A symbolic description of how Amazon CloudWatch Logs should interpret the data in each log event. For example, a log event may contain timestamps, IP addresses, strings, and so on. You use the filter pattern to specify what to look for in the log event message.

Type: String

Length constraints: Minimum length of 0. Maximum length of 512.

Required: No

MetricTransformations
Type: array of MetricTransformation (p. 95) objects

Length constraints: Minimum of 1 item(s) in the list. Maximum of 1 item(s) in the list.

Required: No

MetricFilterMatchRecord

Description

No action documentation available.

Contents

EventMessage
Type: String

Length constraints: Minimum length of 1.

Required: No

EventNumber
Type: Long

Required: No

ExtractedValues
Type: String to String map

Required: No

MetricTransformation

Description

No action documentation available.

Contents

MetricName
The name of the CloudWatch metric to which the monitored log information should be published. For example, you may publish to a metric called ErrorCount.

Type: String

Length constraints: Minimum length of 0. Maximum length of 255.

Pattern: `[^:*$]*`

Required: Yes

MetricNamespace
The destination namespace of the new CloudWatch metric.

Type: String

Length constraints: Minimum length of 0. Maximum length of 255.

Pattern: `[^:*$]*`

Required: Yes

MetricValue
What to publish to the metric. For example, if you're counting the occurrences of a particular term like "Error", the value will be "1" for each occurrence. If you're counting the bytes transferred the published value will be the value in the log event.

Type: String

Length constraints: Minimum length of 0. Maximum length of 100.

Required: Yes

OutputLogEvent

Description

No action documentation available.

Contents

IngestionTime
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

Type: Long

Valid range: Minimum value of 0.

Required: No

Message
Type: String

Length constraints: Minimum length of 1.

Required: No

Timestamp
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

Type: Long

Valid range: Minimum value of 0.

Required: No

RejectedLogEventsInfo

Description

No action documentation available.

Contents

ExpiredLogEventEndIndex
Type: Number

Required: No

TooNewLogEventStartIndex
Type: Number

Required: No

TooOldLogEventEndIndex
Type: Number

Required: No

SearchedLogStream

Description

An object indicating the search status of a log stream in a `FilterLogEvents` request.

Contents

LogStreamName
The name of the log stream.

Type: String

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

Pattern: `[^:*]*`

Required: No

SearchedCompletely
Indicates whether all the events in this log stream were searched or more data exists to search by paginating further.

Type: Boolean

Required: No

SubscriptionFilter

Description

No action documentation available.

Contents

CreationTime
A point in time expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

Type: Long

Valid range: Minimum value of 0.

Required: No

DestinationArn
Type: String

Length constraints: Minimum length of 1.

Required: No

FilterName
A name for a metric or subscription filter.

Type: String

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

Pattern: `[^:*]*`

Required: No

FilterPattern
A symbolic description of how Amazon CloudWatch Logs should interpret the data in each log event. For example, a log event may contain timestamps, IP addresses, strings, and so on. You use the filter pattern to specify what to look for in the log event message.

Type: String

Length constraints: Minimum length of 0. Maximum length of 512.

Required: No

LogGroupName
Type: String

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

Pattern: `[\.\-_/#A-Za-z0-9]+`

Required: No

RoleArn

Type: String

Length constraints: Minimum length of 1.

Required: No

Common Parameters

This section lists the request parameters that all actions use. Any action-specific parameters are listed in the topic for the action.

Action
The action to be performed.

Default: None

Type: string

Required: Yes

AuthParams
The parameters that are required to authenticate a Conditional request. Contains:

- AWSAccessKeyID
- SignatureVersion
- Timestamp
- Signature

Default: None

Required: Conditional

AWSAccessKeyId
The access key ID that corresponds to the secret access key that you used to sign the request.

Default: None

Type: string

Required: Yes

Expires
The date and time when the request signature expires, expressed in the format YYYY-MM-DDThh:mm:ssZ, as specified in the ISO 8601 standard.

Condition: Requests must include either *Timestamp* or *Expires*, but not both.

Default: None

Type: string

Required: Conditional

SecurityToken
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**.

Default: None

Type: string

Required: No

Signature
The digital signature that you created for the request. For information about generating a signature, go to the service's developer documentation.

Default: None

Type: string

Required: Yes

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

Default: None

Type: string

Valid Values: `HmacSHA256 | HmacSHA1`

Required: Yes

SignatureVersion
The signature version you use to sign the request. Set this to the value that is recommended for your service.

Default: None

Type: string

Required: Yes

Timestamp
The date and time when the request was signed, expressed in the format YYYY-MM-DDThh:mm:ssZ, as specified in the ISO 8601 standard.

Condition: Requests must include either *Timestamp* or *Expires*, but not both.

Default: None

Type: string

Required: Conditional

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

Default: None

Type: string

Required: Yes

Common Parameters for Signature V4 Signing

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