Amazon CloudWatch

API Reference

API Version 2010-08-01

Amazon CloudWatch: API Reference

Table of Contents

Welcome

Amazon CloudWatch enables you to publish, monitor, and manage various metrics, as well as configure alarm actions based on data from metrics. This guide provides detailed information about CloudWatch actions, data types, parameters, and errors. For more information about CloudWatch features, see Amazon CloudWatch and the Amazon CloudWatch User Guide.

For information about the metrics that other AWS products send to CloudWatch, see the Amazon CloudWatch Metrics and Dimensions Reference in the *Amazon CloudWatch User Guide*.

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

- Actions (p. 2): An alphabetical list of all CloudWatch actions.
- Data Types (p. 25): An alphabetical list of all CloudWatch data types.
- Common Parameters (p. 38): Parameters that all Query actions can use.
- Common Errors (p. 40): Client and server errors that all actions can return.
- Regions and Endpoints: Supported regions and endpoints for all AWS products.

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

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

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

Actions

The following actions are supported:

- DeleteAlarms (p. 3)
- DescribeAlarmHistory (p. 4)
- DescribeAlarms (p. 6)
- DescribeAlarmsForMetric (p. 8)
- DisableAlarmActions (p. 10)
- EnableAlarmActions (p. 11)
- GetMetricStatistics (p. 12)
- ListMetrics (p. 15)
- PutMetricAlarm (p. 17)
- PutMetricData (p. 21)
- SetAlarmState (p. 24)

DeleteAlarms

Deletes the specified alarms. In the event of an error, no alarms are deleted.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 38).

AlarmNames.member.N

The alarms to be deleted.

Type: array of Strings

Array Members: Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: Yes

Errors

For information about the errors that are common to all actions, see Common Errors (p. 40).

ResourceNotFound

The named resource does not exist.

HTTP Status Code: 404

DescribeAlarmHistory

Retrieves the history for the specified alarm. You can filter the results by date range or item type. If an alarm name is not specified, the histories for all alarms are returned.

Note that Amazon CloudWatch retains the history of an alarm even if you delete the alarm.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 38).

AlarmName

The name of the alarm.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: No

EndDate

The ending date to retrieve alarm history.

Type: Timestamp

Required: No

HistoryItemType

The type of alarm histories to retrieve.

Type: String

Valid Values: `ConfigurationUpdate | StateUpdate | Action`

Required: No

MaxRecords

The maximum number of alarm history records to retrieve.

Type: Integer

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

Required: No

NextToken

The token returned by a previous call to indicate that there is more data available.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1024.

Required: No

StartDate

The starting date to retrieve alarm history.

Type: Timestamp

Required: No

Response Elements

The following elements are returned by the service.

AlarmHistoryItems.member.N

The alarm histories, in JSON format.

Type: array of AlarmHistoryItem (p. 26) objects

NextToken

The token that marks the start of the next batch of returned results.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1024.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 40).

InvalidNextToken

The next token specified is invalid.

HTTP Status Code: 400

DescribeAlarms

Retrieves the specified alarms. If no alarms are specified, all alarms are returned. Alarms can be retrieved by using only a prefix for the alarm name, the alarm state, or a prefix for any action.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 38).

ActionPrefix

The action name prefix.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

AlarmNamePrefix

The alarm name prefix. You cannot specify `AlarmNames` if this parameter is specified.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: No

AlarmNames.member.N

The names of the alarms.

Type: array of Strings

Array Members: Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: No

MaxRecords

The maximum number of alarm descriptions to retrieve.

Type: Integer

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

Required: No

NextToken

The token returned by a previous call to indicate that there is more data available.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1024.

Required: No

StateValue

The state value to be used in matching alarms.

Type: String

Valid Values: `OK | ALARM | INSUFFICIENT_DATA`

Required: No

Response Elements

The following elements are returned by the service.

MetricAlarms.member.N

The information for the specified alarms.

Type: array of MetricAlarm (p. 31) objects

NextToken

The token that marks the start of the next batch of returned results.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1024.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 40).

InvalidNextToken

The next token specified is invalid.

HTTP Status Code: 400

DescribeAlarmsForMetric

Retrieves the alarms for the specified metric. Specify a statistic, period, or unit to filter the results.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 38).

Dimensions.member.N

The dimensions associated with the metric. If the metric has any associated dimensions, you must specify them in order for the call to succeed.

Type: array of Dimension (p. 28) objects

Array Members: Maximum number of 10 items.

Required: No

ExtendedStatistic

The percentile statistic for the metric. Specify a value between p0.0 and p100.

Type: String

Pattern: `p(\d{1,2}(\.\d{0,2})?|100)`

Required: No

MetricName

The name of the metric.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: Yes

Namespace

The namespace of the metric.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[^:].*`

Required: Yes

Period

The period, in seconds, over which the statistic is applied.

Type: Integer

Valid Range: Minimum value of 60.

Required: No

Statistic

The statistic for the metric, other than percentiles. For percentile statistics, use `ExtendedStatistics`.

Type: String

Valid Values: `SampleCount | Average | Sum | Minimum | Maximum`

Required: No

Unit

The unit for the metric.

Type: String

Valid Values: `Seconds | Microseconds | Milliseconds | Bytes | Kilobytes | Megabytes | Gigabytes | Terabytes | Bits | Kilobits | Megabits | Gigabits | Terabits | Percent | Count | Bytes/Second | Kilobytes/Second | Megabytes/Second | Gigabytes/Second | Terabytes/Second | Bits/Second | Kilobits/Second | Megabits/Second | Gigabits/Second | Terabits/Second | Count/Second | None`

Required: No

Response Elements

The following element is returned by the service.

MetricAlarms.member.N

The information for each alarm with the specified metric.

Type: array of MetricAlarm (p. 31) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 40).

DisableAlarmActions

Disables the actions for the specified alarms. When an alarm's actions are disabled, the alarm actions do not execute when the alarm state changes.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 38).

AlarmNames.member.N

The names of the alarms.

Type: array of Strings

Array Members: Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: Yes

Errors

For information about the errors that are common to all actions, see Common Errors (p. 40).

EnableAlarmActions

Enables the actions for the specified alarms.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 38).

AlarmNames.member.N

The names of the alarms.

Type: array of Strings

Array Members: Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: Yes

Errors

For information about the errors that are common to all actions, see Common Errors (p. 40).

GetMetricStatistics

Gets statistics for the specified metric.

Amazon CloudWatch retains metric data as follows:

- Data points with a period of 60 seconds (1 minute) are available for 15 days
- Data points with a period of 300 seconds (5 minute) are available for 63 days
- Data points with a period of 3600 seconds (1 hour) are available for 455 days (15 months)

Note that CloudWatch started retaining 5-minute and 1-hour metric data as of 9 July 2016.

The maximum number of data points returned from a single call is 1,440. If you request more than 1,440 data points, Amazon CloudWatch returns an error. To reduce the number of data points, you can narrow the specified time range and make multiple requests across adjacent time ranges, or you can increase the specified period. A period can be as short as one minute (60 seconds). Note that data points are not returned in chronological order.

Amazon CloudWatch aggregates data points based on the length of the period that you specify. For example, if you request statistics with a one-hour period, Amazon CloudWatch aggregates all data points with time stamps that fall within each one-hour period. Therefore, the number of values aggregated by CloudWatch is larger than the number of data points returned.

For a list of metrics and dimensions supported by AWS services, see the Amazon CloudWatch Metrics and Dimensions Reference in the *Amazon CloudWatch User Guide*.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 38).

Dimensions.member.N

The dimensions. CloudWatch treats each unique combination of dimensions as a separate metric. You can't retrieve statistics using combinations of dimensions that were not specially published. You must specify the same dimensions that were used when the metrics were created. For an example, see Dimension Combinations in the *Amazon CloudWatch User Guide*.

Type: array of Dimension (p. 28) objects

Array Members: Maximum number of 10 items.

Required: No

EndTime

The time stamp that determines the last data point to return.

The value specified is exclusive; results will include data points up to the specified time stamp. The time stamp must be in ISO 8601 UTC format (for example, 2016-10-10T23:00:00Z).

Type: Timestamp

Required: Yes

ExtendedStatistics.member.N

The percentile statistics. Specify values between p0.0 and p100.

Type: array of Strings

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

Pattern: `p(\d{1,2}(\.\d{0,2})?|100)`

Required: No

MetricName

The name of the metric, with or without spaces.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: Yes

Namespace

The namespace of the metric, with or without spaces.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[^:].*`

Required: Yes

Period

The granularity, in seconds, of the returned data points. A period can be as short as one minute (60 seconds) and must be a multiple of 60. The default value is 60.

If the `StartTime` parameter specifies a time stamp that is greater than 15 days ago, you must specify the period as follows or no data points in that time range is returned:

- Start time between 15 and 63 days ago - Use a multiple of 300 seconds (5 minutes).
- Start time greater than 63 days ago - Use a multiple of 3600 seconds (1 hour).

Type: Integer

Valid Range: Minimum value of 60.

Required: Yes

StartTime

The time stamp that determines the first data point to return. Note that start times are evaluated relative to the time that CloudWatch receives the request.

The value specified is inclusive; results include data points with the specified time stamp. The time stamp must be in ISO 8601 UTC format (for example, 2016-10-03T23:00:00Z).

CloudWatch rounds the specified time stamp as follows:

- Start time less than 15 days ago - Round down to the nearest whole minute. For example, 12:32:34 is rounded down to 12:32:00.
- Start time between 15 and 63 days ago - Round down to the nearest 5-minute clock interval. For example, 12:32:34 is rounded down to 12:30:00.
- Start time greater than 63 days ago - Round down to the nearest 1-hour clock interval. For example, 12:32:34 is rounded down to 12:00:00.

Type: Timestamp

Required: Yes

Statistics.member.N

The metric statistics, other than percentile. For percentile statistics, use `ExtendedStatistic`.

Type: array of Strings

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

Valid Values: `SampleCount | Average | Sum | Minimum | Maximum`

Required: No

Unit

The unit for a given metric. Metrics may be reported in multiple units. Not supplying a unit results in all units being returned. If the metric only ever reports one unit, specifying a unit has no effect.

Type: String

Valid Values: `Seconds | Microseconds | Milliseconds | Bytes | Kilobytes | Megabytes | Gigabytes | Terabytes | Bits | Kilobits | Megabits | Gigabits | Terabits | Percent | Count | Bytes/Second | Kilobytes/Second | Megabytes/Second | Gigabytes/Second | Terabytes/Second | Bits/Second | Kilobits/Second | Megabits/Second | Gigabits/Second | Terabits/Second | Count/Second | None`

Required: No

Response Elements

The following elements are returned by the service.

Datapoints.member.N

The data points for the specified metric.

Type: array of Datapoint (p. 27) objects

Label

A label for the specified metric.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 40).

InternalServiceError

Request processing has failed due to some unknown error, exception, or failure.

HTTP Status Code: 500

InvalidParameterCombination

Parameters that cannot be used together were used together.

HTTP Status Code: 400

InvalidParameterValue

The value of an input parameter is bad or out-of-range.

HTTP Status Code: 400

MissingParameter

An input parameter that is required is missing.

HTTP Status Code: 400

ListMetrics

List the specified metrics. You can use the returned metrics with GetMetricStatistics (p. 12) to obtain statistical data.

Up to 500 results are returned for any one call. To retrieve additional results, use the returned token with subsequent calls.

After you create a metric, allow up to fifteen minutes before the metric appears. Statistics about the metric, however, are available sooner using GetMetricStatistics (p. 12).

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 38).

Dimensions.member.N

The dimensions to filter against.

Type: array of DimensionFilter (p. 29) objects

Array Members: Maximum number of 10 items.

Required: No

MetricName

The name of the metric to filter against.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: No

Namespace

The namespace to filter against.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[^:].*`

Required: No

NextToken

The token returned by a previous call to indicate that there is more data available.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1024.

Required: No

Response Elements

The following elements are returned by the service.

Metrics.member.N

The metrics.

Type: array of Metric (p. 30) objects

NextToken

The token that marks the start of the next batch of returned results.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1024.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 40).

InternalServiceError

Request processing has failed due to some unknown error, exception, or failure.

HTTP Status Code: 500

InvalidParameterValue

The value of an input parameter is bad or out-of-range.

HTTP Status Code: 400

PutMetricAlarm

Creates or updates an alarm and associates it with the specified metric. Optionally, this operation can associate one or more Amazon SNS resources with the alarm.

When this operation creates an alarm, the alarm state is immediately set to `INSUFFICIENT_DATA`. The alarm is evaluated and its state is set appropriately. Any actions associated with the state are then executed.

When you update an existing alarm, its state is left unchanged, but the update completely overwrites the previous configuration of the alarm.

If you are an AWS Identity and Access Management (IAM) user, you must have Amazon EC2 permissions for some operations:

- `ec2:DescribeInstanceStatus` and `ec2:DescribeInstances` for all alarms on EC2 instance status metrics
- `ec2:StopInstances` for alarms with stop actions
- `ec2:TerminateInstances` for alarms with terminate actions
- `ec2:DescribeInstanceRecoveryAttribute` and `ec2:RecoverInstances` for alarms with recover actions

If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm, but the stop or terminate actions won't be performed. However, if you are later granted the required permissions, the alarm actions that you created earlier will be performed.

If you are using an IAM role (for example, an Amazon EC2 instance profile), you cannot stop or terminate the instance using alarm actions. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.

If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot stop or terminate an Amazon EC2 instance using alarm actions.

Note that you must create at least one stop, terminate, or reboot alarm using the Amazon EC2 or CloudWatch console to create the **EC2ActionsAccess** IAM role. After this IAM role is created, you can create stop, terminate, or reboot alarms using a command-line interface or an API.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 38).

ActionsEnabled

Indicates whether actions should be executed during any changes to the alarm state.

Type: Boolean

Required: No

AlarmActions.member.N

The actions to execute when this alarm transitions to the `ALARM` state from any other state. Each action is specified as an Amazon Resource Name (ARN).

Valid Values: arn:aws:automate:*region*:ec2:stop | arn:aws:automate:*region*:ec2:terminate | arn:aws:automate:*region*:ec2:recover

Valid Values (for use with IAM roles): arn:aws:swf:us-east-1:{*customer-account*}:action/actions/AWS_EC2.InstanceId.Stop/1.0 | arn:aws:swf:us-east-1:{*customer-account*}:action/actions/AWS_EC2.InstanceId.Terminate/1.0 | arn:aws:swf:us-east-1:{*customer-account*}:action/actions/AWS_EC2.InstanceId.Reboot/1.0

Type: array of Strings

Array Members: Maximum number of 5 items.

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

AlarmDescription

The description for the alarm.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1024.

Required: No

AlarmName

The name for the alarm. This name must be unique within the AWS account.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: Yes

ComparisonOperator

The arithmetic operation to use when comparing the specified statistic and threshold. The specified statistic value is used as the first operand.

Type: String

Valid Values: `GreaterThanOrEqualToThreshold | GreaterThanThreshold | LessThanThreshold | LessThanOrEqualToThreshold`

Required: Yes

Dimensions.member.N

The dimensions for the metric associated with the alarm.

Type: array of Dimension (p. 28) objects

Array Members: Maximum number of 10 items.

Required: No

EvaluationPeriods

The number of periods over which data is compared to the specified threshold.

Type: Integer

Valid Range: Minimum value of 1.

Required: Yes

ExtendedStatistic

The percentile statistic for the metric associated with the alarm. Specify a value between p0.0 and p100.

Type: String

Pattern: `p(\d{1,2}(\.\d{0,2})?|100)`

Required: No

InsufficientDataActions.member.N

The actions to execute when this alarm transitions to the `INSUFFICIENT_DATA` state from any other state. Each action is specified as an Amazon Resource Name (ARN).

Valid Values: arn:aws:automate:*region*:ec2:stop | arn:aws:automate:*region*:ec2:terminate | arn:aws:automate:*region*:ec2:recover

Valid Values (for use with IAM roles): arn:aws:swf:us-east-1:{*customer-account*}:action/actions/AWS_EC2.InstanceId.Stop/1.0 | arn:aws:swf:us-east-1:{*customer-account*}:action/actions/AWS_EC2.InstanceId.Terminate/1.0 | arn:aws:swf:us-east-1:{*customer-account*}:action/actions/AWS_EC2.InstanceId.Reboot/1.0

Type: array of Strings

Array Members: Maximum number of 5 items.

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

MetricName

The name for the metric associated with the alarm.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: Yes

Namespace

The namespace for the metric associated with the alarm.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[^:].*`

Required: Yes

OKActions.member.N

The actions to execute when this alarm transitions to an `OK` state from any other state. Each action is specified as an Amazon Resource Name (ARN).

Valid Values: arn:aws:automate:*region*:ec2:stop | arn:aws:automate:*region*:ec2:terminate | arn:aws:automate:*region*:ec2:recover

Valid Values (for use with IAM roles): arn:aws:swf:us-east-1:{*customer-account*}:action/actions/AWS_EC2.InstanceId.Stop/1.0 | arn:aws:swf:us-east-1:{*customer-account*}:action/actions/AWS_EC2.InstanceId.Terminate/1.0 | arn:aws:swf:us-east-1:{*customer-account*}:action/actions/AWS_EC2.InstanceId.Reboot/1.0

Type: array of Strings

Array Members: Maximum number of 5 items.

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

Period

The period, in seconds, over which the specified statistic is applied.

Type: Integer

Valid Range: Minimum value of 60.

Required: Yes

Statistic

The statistic for the metric associated with the alarm, other than percentile. For percentile statistics, use `ExtendedStatistic`.

Type: String

Valid Values: `SampleCount | Average | Sum | Minimum | Maximum`

Required: No

Threshold

The value against which the specified statistic is compared.

Type: Double

Required: Yes

Unit

The unit of measure for the statistic. For example, the units for the Amazon EC2 NetworkIn metric are Bytes because NetworkIn tracks the number of bytes that an instance receives on all network interfaces. You can also specify a unit when you create a custom metric. Units help provide conceptual meaning to your data. Metric data points that specify a unit of measure, such as Percent, are aggregated separately.

If you specify a unit, you must use a unit that is appropriate for the metric. Otherwise, the Amazon CloudWatch alarm can get stuck in the `INSUFFICIENT DATA` state.

Type: String

Valid Values: `Seconds | Microseconds | Milliseconds | Bytes | Kilobytes | Megabytes | Gigabytes | Terabytes | Bits | Kilobits | Megabits | Gigabits | Terabits | Percent | Count | Bytes/Second | Kilobytes/Second | Megabytes/Second | Gigabytes/Second | Terabytes/Second | Bits/Second | Kilobits/Second | Megabits/Second | Gigabits/Second | Terabits/Second | Count/Second | None`

Required: No

Errors

For information about the errors that are common to all actions, see Common Errors (p. 40).

LimitExceeded

The quota for alarms for this customer has already been reached.

HTTP Status Code: 400

PutMetricData

Publishes metric data points to Amazon CloudWatch. Amazon CloudWatch associates the data points with the specified metric. If the specified metric does not exist, Amazon CloudWatch creates the metric. When Amazon CloudWatch creates a metric, it can take up to fifteen minutes for the metric to appear in calls to ListMetrics (p. 15).

Each `PutMetricData` request is limited to 8 KB in size for HTTP GET requests and is limited to 40 KB in size for HTTP POST requests.

Although the `Value` parameter accepts numbers of type `Double`, Amazon CloudWatch rejects values that are either too small or too large. Values must be in the range of 8.515920e-109 to 1.174271e+108 (Base 10) or 2e-360 to 2e360 (Base 2). In addition, special values (e.g., NaN, +Infinity, -Infinity) are not supported.

Data points with time stamps from 24 hours ago or longer can take at least 48 hours to become available for GetMetricStatistics (p. 12) from the time they are submitted.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 38).

MetricData.member.N

The data for the metric.

Type: array of MetricDatum (p. 34) objects

Required: Yes

Namespace

The namespace for the metric data.

You cannot specify a namespace that begins with "AWS/". Namespaces that begin with "AWS/" are reserved for use by Amazon Web Services products.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[^:].*`

Required: Yes

Errors

For information about the errors that are common to all actions, see Common Errors (p. 40).

InternalServiceError

Request processing has failed due to some unknown error, exception, or failure.

HTTP Status Code: 500

InvalidParameterCombination

Parameters that cannot be used together were used together.

HTTP Status Code: 400

InvalidParameterValue

The value of an input parameter is bad or out-of-range.

HTTP Status Code: 400

MissingParameter

An input parameter that is required is missing.

HTTP Status Code: 400

Examples

Example

The following example puts data for a single metric containing one dimension:

Sample Request

```
https://monitoring.&api-domain;/doc/2010-08-01/
?Action=PutMetricData
&Version=2010-08-01
&Namespace=TestNamespace
&MetricData.member.1.MetricName=buffers
&MetricData.member.1.Unit=Bytes
&MetricData.member.1.Value=231434333
&MetricData.member.1.Dimensions.member.1.Name=InstanceType
&MetricData.member.1.Dimensions.member.1.Value=m1.small
&AUTHPARAMS
```

Example

The following example puts data for a single metric containing two dimensions:

Sample Request

```
https://monitoring.&api-domain;/doc/2010-08-01/
?Action=PutMetricData
&Version=2010-08-01
&Namespace=TestNamespace
&MetricData.member.1.MetricName=buffers
&MetricData.member.1.Unit=Bytes
&MetricData.member.1.Value=231434333
&MetricData.member.1.Dimensions.member.1.Name=InstanceID
&MetricData.member.1.Dimensions.member.1.Value=i-aaba32d4
&MetricData.member.1.Dimensions.member.2.Name=InstanceType
&MetricData.member.1.Dimensions.member.2.Value=m1.small
&AUTHPARAMS
```

Example

The following example puts data for two metrics, each with two dimensions:

Sample Request

```
https://monitoring.&api-domain;/doc/2010-08-01/
?Action=PutMetricData
&Version=2010-08-01
&Namespace=TestNamespace
&MetricData.member.1.MetricName=buffers
&MetricData.member.1.Unit=Bytes
&MetricData.member.1.Value=231434333
&MetricData.member.1.Dimensions.member.1.Name=InstanceID
&MetricData.member.1.Dimensions.member.1.Value=i-aaba32d4
&MetricData.member.1.Dimensions.member.2.Name=InstanceType
&MetricData.member.1.Dimensions.member.2.Value=m1.small
&MetricData.member.2.MetricName=latency
```

```
&MetricData.member.2.Unit=Milliseconds
&MetricData.member.2.Value=23
&MetricData.member.2.Dimensions.member.1.Name=InstanceID
&MetricData.member.2.Dimensions.member.1.Value=i-aaba32d4
&MetricData.member.2.Dimensions.member.2.Name=InstanceType
&MetricData.member.2.Dimensions.member.2.Value=m1.small
&AUTHPARAMS
```

SetAlarmState

Temporarily sets the state of an alarm for testing purposes. When the updated state differs from the previous value, the action configured for the appropriate state is invoked. For example, if your alarm is configured to send an Amazon SNS message when an alarm is triggered, temporarily changing the alarm state to `ALARM` sends an Amazon SNS message. The alarm returns to its actual state (often within seconds). Because the alarm state change happens very quickly, it is typically only visible in the alarm's **History** tab in the Amazon CloudWatch console or through DescribeAlarmHistory (p. 4).

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 38).

AlarmName

The name for the alarm. This name must be unique within the AWS account. The maximum length is 255 characters.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: Yes

StateReason

The reason that this alarm is set to this specific state, in text format.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1023.

Required: Yes

StateReasonData

The reason that this alarm is set to this specific state, in JSON format.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 4000.

Required: No

StateValue

The value of the state.

Type: String

Valid Values: `OK | ALARM | INSUFFICIENT_DATA`

Required: Yes

Errors

For information about the errors that are common to all actions, see Common Errors (p. 40).

InvalidFormat

Data was not syntactically valid JSON.

HTTP Status Code: 400

ResourceNotFound

The named resource does not exist.

HTTP Status Code: 404

Data Types

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

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

The following data types are supported:

AlarmHistoryItem

Represents the history of a specific alarm.

Contents

AlarmName

The descriptive name for the alarm.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: No

HistoryData

Data about the alarm, in JSON format.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 4095.

Required: No

HistoryItemType

The type of alarm history item.

Type: String

Valid Values: `ConfigurationUpdate | StateUpdate | Action`

Required: No

HistorySummary

A summary of the alarm history, in text format.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: No

Timestamp

The time stamp for the alarm history item.

Type: Timestamp

Required: No

Datapoint

Encapsulates the statistical data that Amazon CloudWatch computes from metric data.

Contents

Average

The average of the metric values that correspond to the data point.

Type: Double

Required: No

ExtendedStatistics , ExtendedStatistics.entry.N.key (key), ExtendedStatisticsentry.N.value (value)

The percentile statistic for the data point.

Type: String to Double map

Required: No

Maximum

The maximum metric value for the data point.

Type: Double

Required: No

Minimum

The minimum metric value for the data point.

Type: Double

Required: No

SampleCount

The number of metric values that contributed to the aggregate value of this data point.

Type: Double

Required: No

Sum

The sum of the metric values for the data point.

Type: Double

Required: No

Timestamp

The time stamp used for the data point.

Type: Timestamp

Required: No

Unit

The standard unit for the data point.

Type: String

Valid Values: `Seconds | Microseconds | Milliseconds | Bytes | Kilobytes | Megabytes | Gigabytes | Terabytes | Bits | Kilobits | Megabits | Gigabits | Terabits | Percent | Count | Bytes/Second | Kilobytes/Second | Megabytes/Second | Gigabytes/Second | Terabytes/Second | Bits/Second | Kilobits/Second | Megabits/Second | Gigabits/Second | Terabits/Second | Count/Second | None`

Required: No

Dimension

Expands the identity of a metric.

Contents

Name

The name of the dimension.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: Yes

Value

The value representing the dimension measurement.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: Yes

DimensionFilter

Represents filters for a dimension.

Contents

Name

The dimension name to be matched.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: Yes

Value

The value of the dimension to be matched.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: No

Metric

Represents a specific metric.

Contents

Dimensions.member.N

The dimensions for the metric.

Type: array of Dimension (p. 28) objects

Array Members: Maximum number of 10 items.

Required: No

MetricName

The name of the metric.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: No

Namespace

The namespace of the metric.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[^:].*`

Required: No

MetricAlarm

Represents an alarm.

Contents

ActionsEnabled

Indicates whether actions should be executed during any changes to the alarm state.

Type: Boolean

Required: No

AlarmActions.member.N

The actions to execute when this alarm transitions to the `ALARM` state from any other state. Each action is specified as an Amazon Resource Name (ARN).

Type: array of Strings

Array Members: Maximum number of 5 items.

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

AlarmArn

The Amazon Resource Name (ARN) of the alarm.

Type: String

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

Required: No

AlarmConfigurationUpdatedTimestamp

The time stamp of the last update to the alarm configuration.

Type: Timestamp

Required: No

AlarmDescription

The description of the alarm.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1024.

Required: No

AlarmName

The name of the alarm.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: No

ComparisonOperator

The arithmetic operation to use when comparing the specified statistic and threshold. The specified statistic value is used as the first operand.

Type: String

Valid Values: `GreaterThanOrEqualToThreshold | GreaterThanThreshold | LessThanThreshold | LessThanOrEqualToThreshold`

Required: No

Dimensions.member.N

The dimensions for the metric associated with the alarm.

Type: array of Dimension (p. 28) objects

Array Members: Maximum number of 10 items.

Required: No

EvaluationPeriods

The number of periods over which data is compared to the specified threshold.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

ExtendedStatistic

The percentile statistic for the metric associated with the alarm. Specify a value between p0.0 and p100.

Type: String

Pattern: `p(\d{1,2}(\.\d{0,2})?|100)`

Required: No

InsufficientDataActions.member.N

The actions to execute when this alarm transitions to the `INSUFFICIENT_DATA` state from any other state. Each action is specified as an Amazon Resource Name (ARN).

Type: array of Strings

Array Members: Maximum number of 5 items.

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

MetricName

The name of the metric associated with the alarm.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: No

Namespace

The namespace of the metric associated with the alarm.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[^:].*`

Required: No

OKActions.member.N

The actions to execute when this alarm transitions to the `OK` state from any other state. Each action is specified as an Amazon Resource Name (ARN).

Type: array of Strings

Array Members: Maximum number of 5 items.

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

Period

The period, in seconds, over which the statistic is applied.

Type: Integer

Valid Range: Minimum value of 60.

Required: No

StateReason

An explanation for the alarm state, in text format.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1023.

Required: No

StateReasonData

An explanation for the alarm state, in JSON format.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 4000.

Required: No

StateUpdatedTimestamp

The time stamp of the last update to the alarm state.

Type: Timestamp

Required: No

StateValue

The state value for the alarm.

Type: String

Valid Values: `OK | ALARM | INSUFFICIENT_DATA`

Required: No

Statistic

The statistic for the metric associated with the alarm, other than percentile. For percentile statistics, use `ExtendedStatistic`.

Type: String

Valid Values: `SampleCount | Average | Sum | Minimum | Maximum`

Required: No

Threshold

The value to compare with the specified statistic.

Type: Double

Required: No

Unit

The unit of the metric associated with the alarm.

Type: String

Valid Values: `Seconds | Microseconds | Milliseconds | Bytes | Kilobytes | Megabytes | Gigabytes | Terabytes | Bits | Kilobits | Megabits | Gigabits | Terabits | Percent | Count | Bytes/Second | Kilobytes/Second | Megabytes/Second | Gigabytes/Second | Terabytes/Second | Bits/Second | Kilobits/Second | Megabits/Second | Gigabits/Second | Terabits/Second | Count/Second | None`

Required: No

MetricDatum

Encapsulates the information sent to either create a metric or add new values to be aggregated into an existing metric.

Contents

Dimensions.member.N

The dimensions associated with the metric.

Type: array of Dimension (p. 28) objects

Array Members: Maximum number of 10 items.

Required: No

MetricName

The name of the metric.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: Yes

StatisticValues

The statistical values for the metric.

Type: StatisticSet (p. 35) object

Required: No

Timestamp

The time the metric data was received, expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

Type: Timestamp

Required: No

Unit

The unit of the metric.

Type: String

Valid Values: `Seconds | Microseconds | Milliseconds | Bytes | Kilobytes | Megabytes | Gigabytes | Terabytes | Bits | Kilobits | Megabits | Gigabits | Terabits | Percent | Count | Bytes/Second | Kilobytes/Second | Megabytes/Second | Gigabytes/Second | Terabytes/Second | Bits/Second | Kilobits/Second | Megabits/Second | Gigabits/Second | Terabits/Second | Count/Second | None`

Required: No

Value

The value for the metric.

Although the parameter accepts numbers of type Double, Amazon CloudWatch rejects values that are either too small or too large. Values must be in the range of 8.515920e-109 to 1.174271e+108 (Base 10) or 2e-360 to 2e360 (Base 2). In addition, special values (for example, NaN, +Infinity, -Infinity) are not supported.

Type: Double

Required: No

StatisticSet

Represents a set of statistics that describes a specific metric.

Contents

Maximum

The maximum value of the sample set.

Type: Double

Required: Yes

Minimum

The minimum value of the sample set.

Type: Double

Required: Yes

SampleCount

The number of samples used for the statistic set.

Type: Double

Required: Yes

Sum

The sum of values for the sample set.

Type: Double

Required: Yes

Making API Requests

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

Amazon CloudWatch Endpoints

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

Query Parameters

Each query request must include some common parameters to handle authentication and selection of an action. For more information, see Common Parameters (p. 38).

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

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

Request Identifiers

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

Query API Authentication

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

Available Libraries

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

- Android
- iOS
- Java
- PHP
- Python
- Ruby
- Windows and .NET

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

Common Parameters

The following table lists the parameters that all actions use for signing Signature Version 4 requests. Any action-specific parameters are listed in the topic for that action. To view sample requests, see Examples of Signed Signature Version 4 Requests or Signature Version 4 Test Suite in the *Amazon Web Services General Reference*.

Action

The action to be performed.

Type: string

Required: Yes

Version

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

Type: string

Required: Yes

X-Amz-Algorithm

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

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

Type: string

Valid Values: `AWS4-HMAC-SHA256`

Required: Conditional

X-Amz-Credential

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

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

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

Type: string

Required: Conditional

X-Amz-Date

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

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

Type: string

Required: Conditional

X-Amz-Security-Token

The temporary security token that was obtained through a call to AWS Security Token Service. For a list of services that support AWS Security Token Service, go to Using Temporary Security Credentials to Access AWS in *Using Temporary Security Credentials*.

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

Type: string

Required: Conditional

X-Amz-Signature

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

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

Type: string

Required: Conditional

X-Amz-SignedHeaders

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

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

Type: string

Required: Conditional

Common Errors

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

IncompleteSignature
The request signature does not conform to AWS standards.

HTTP Status Code: 400

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

HTTP Status Code: 500

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

HTTP Status Code: 400

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

HTTP Status Code: 403

InvalidParameterCombination
Parameters that must not be used together were used together.

HTTP Status Code: 400

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

HTTP Status Code: 400

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

HTTP Status Code: 400

MalformedQueryString
The query string contains a syntax error.

HTTP Status Code: 404

MissingAction
The request is missing an action or a required parameter.

HTTP Status Code: 400

MissingAuthenticationToken

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

HTTP Status Code: 403

MissingParameter

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

OptInRequired

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

HTTP Status Code: 403

RequestExpired

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

HTTP Status Code: 400

ServiceUnavailable

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

HTTP Status Code: 503

Throttling

The request was denied due to request throttling.

HTTP Status Code: 400

ValidationError

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

HTTP Status Code: 400