Amazon CloudWatch

API Reference

API Version 2010-08-01

Amazon CloudWatch: API Reference

Table of Contents

Welcome 1
Actions 2
DeleteAlarms 3
Request Parameters 3
Errors 3
See Also 3
DeleteAnomalyDetector 5
Request Parameters 5
Errors 5
See Also 6
DeleteDashboards 7
Request Parameters 7
Errors 7
See Also 7
DeleteInsightRules 8
Request Parameters 8
Response Elements 8
Errors 8
See Also 8
DescribeAlarmHistory 10
Request Parameters 10
Response Elements 11
Errors 11
See Also 11
DescribeAlarms 13
Request Parameters 13
Response Elements 15
Errors 15
See Also 15
DescribeAlarmsForMetric 16
Request Parameters 16
Response Elements 17
Errors 17
See Also 17
DescribeAnomalyDetectors 18
Request Parameters 18
Response Elements 19
Errors 19
Example 19
See Also 21
DescribeInsightRules 22
Request Parameters 22
Response Elements 22
Errors 22
See Also 23
DisableAlarmActions 24
Request Parameters 24
Errors 24
See Also 24
DisableInsightRules 25
Request Parameters 25
Response Elements 25
Errors 25
See Also 25

EnableAlarmActions 27
Request Parameters 27
Errors 27
See Also 27
EnableInsightRules 28
Request Parameters 28
Response Elements 28
Errors 28
See Also 28
GetDashboard 30
Request Parameters 30
Response Elements 30
Errors 30
See Also 31
GetInsightRuleReport 32
Request Parameters 32
Response Elements 34
Errors 34
See Also 35
GetMetricData 36
Request Parameters 36
Response Elements 38
Errors 38
Examples 38
See Also 46
GetMetricStatistics 47
Request Parameters 47
Response Elements 50
Errors 50
See Also 50
GetMetricWidgetImage 52
Request Parameters 52
Response Elements 53
Errors 53
Example 53
See Also 54
ListDashboards 55
Request Parameters 55
Response Elements 55
Errors 55
See Also 56
ListMetrics 57
Request Parameters 57
Response Elements 57
Errors 58
See Also 58
ListTagsForResource 59
Request Parameters 59
Response Elements 59
Errors 59
See Also 60
PutAnomalyDetector 61
Request Parameters 61
Errors 62
Example 62
See Also 62
PutCompositeAlarm 64

Request Parameters 64
Errors 66
Example 67
See Also 67
PutDashboard 68
Request Parameters 68
Response Elements 68
Errors 69
Examples 69
See Also 73
PutInsightRule 74
Request Parameters 74
Errors 75
See Also 75
PutMetricAlarm 76
Request Parameters 76
Errors 82
Examples 82
See Also 85
PutMetricData 86
Request Parameters 86
Errors 87
Examples 87
See Also 89
SetAlarmState 90
Request Parameters 90
Errors 91
Example 91
See Also 91
TagResource 93
Request Parameters 93
Errors 93
See Also 94
UntagResource 95
Request Parameters 95
Errors 95
See Also 96
Data Types 97
AlarmHistoryItem 98
Contents 98
See Also 99
AnomalyDetector 100
Contents 100
See Also 101
AnomalyDetectorConfiguration 102
Contents 102
See Also 102
CompositeAlarm 103
Contents 103
See Also 105
DashboardEntry 106
Contents 106
See Also 106
DashboardValidationMessage 107
Contents 107
See Also 107
Datapoint 108

Contents ... 108
See Also ... 109
Dimension ... 110
Contents ... 110
See Also ... 110
DimensionFilter ... 111
Contents ... 111
See Also ... 111
InsightRule ... 112
Contents ... 112
See Also ... 112
InsightRuleContributor ... 114
Contents ... 114
See Also ... 114
InsightRuleContributorDatapoint ... 115
Contents ... 115
See Also ... 115
InsightRuleMetricDatapoint ... 116
Contents ... 116
See Also ... 117
MessageData ... 118
Contents ... 118
See Also ... 118
Metric ... 119
Contents ... 119
See Also ... 119
MetricAlarm ... 120
Contents ... 120
See Also ... 124
MetricDataQuery ... 125
Contents ... 125
See Also ... 126
MetricDataResult ... 127
Contents ... 127
See Also ... 128
MetricDatum ... 129
Contents ... 129
See Also ... 130
MetricStat ... 132
Contents ... 132
See Also ... 133
PartialFailure ... 134
Contents ... 134
See Also ... 134
Range ... 135
Contents ... 135
See Also ... 135
StatisticSet ... 136
Contents ... 136
See Also ... 136
Tag ... 137
Contents ... 137
See Also ... 137
Dashboard Body Structure and Syntax ... 138
Overall Structure ... 138
Widgets Array Structure ... 140
Properties of a Text Widget Object ... 141

Properties of a Log Widget Object ... 142
Properties of a Metric Widget Object ... 143
Metric Widget: Format for Each Metric in the Array ... 146
Dashboard Widget Object: Rendering Properties Object Format ... 148
Dashboard Widget Object: Annotation Properties ... 150
Dashboard Widget Object: yAxis Properties Format ... 154
GetMetricWidgetImage: Metric Widget Structure and Syntax ... 156
Overall Structure ... 156
Format for Each Metric in the Array of Metrics ... 158
Options Object Format ... 160
Annotation Properties Format ... 162
yAxis Properties Format ... 166
Making API Requests ... 167
Amazon CloudWatch Endpoints ... 167
Query Parameters ... 167
Request Identifiers ... 167
Query API Authentication ... 167
Available Libraries ... 167
Making API Requests Using the POST Method ... 168
Common Parameters ... 170
Common Errors ... 172

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. 97): An alphabetical list of all CloudWatch data types.
- Common Parameters (p. 170): Parameters that all Query actions can use.
- Common Errors (p. 172): 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)
- DeleteAnomalyDetector (p. 5)
- DeleteDashboards (p. 7)
- DeleteInsightRules (p. 8)
- DescribeAlarmHistory (p. 10)
- DescribeAlarms (p. 13)
- DescribeAlarmsForMetric (p. 16)
- DescribeAnomalyDetectors (p. 18)
- DescribeInsightRules (p. 22)
- DisableAlarmActions (p. 24)
- DisableInsightRules (p. 25)
- EnableAlarmActions (p. 27)
- EnableInsightRules (p. 28)
- GetDashboard (p. 30)
- GetInsightRuleReport (p. 32)
- GetMetricData (p. 36)
- GetMetricStatistics (p. 47)
- GetMetricWidgetImage (p. 52)
- ListDashboards (p. 55)
- ListMetrics (p. 57)
- ListTagsForResource (p. 59)
- PutAnomalyDetector (p. 61)
- PutCompositeAlarm (p. 64)
- PutDashboard (p. 68)
- PutInsightRule (p. 74)
- PutMetricAlarm (p. 76)
- PutMetricData (p. 86)
- SetAlarmState (p. 90)
- TagResource (p. 93)
- UntagResource (p. 95)

DeleteAlarms

Deletes the specified alarms. You can delete up to 100 alarms in one operation. However, this total can include no more than one composite alarm. For example, you could delete 99 metric alarms and one composite alarms with one operation, but you can't delete two composite alarms with one operation.

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

> **Note**
> It is possible to create a loop or cycle of composite alarms, where composite alarm A depends on composite alarm B, and composite alarm B also depends on composite alarm A. In this scenario, you can't delete any composite alarm that is part of the cycle because there is always still a composite alarm that depends on that alarm that you want to delete.
> To get out of such a situation, you must break the cycle by changing the rule of one of the composite alarms in the cycle to remove a dependency that creates the cycle. The simplest change to make to break a cycle is to change the `AlarmRule` of one of the alarms to `False`.
> Additionally, the evaluation of composite alarms stops if CloudWatch detects a cycle in the evaluation path.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

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. 172).

ResourceNotFound

The named resource does not exist.

HTTP Status Code: 404

See Also

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

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

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

DeleteAnomalyDetector

Deletes the specified anomaly detection model from your account.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

Dimensions.member.N

The metric dimensions associated with the anomaly detection model to delete.

Type: Array of Dimension (p. 110) objects

Array Members: Maximum number of 10 items.

Required: No

MetricName

The metric name associated with the anomaly detection model to delete.

Type: String

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

Required: Yes

Namespace

The namespace associated with the anomaly detection model to delete.

Type: String

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

Pattern: `[^:].*`

Required: Yes

Stat

The statistic associated with the anomaly detection model to delete.

Type: String

Pattern: `(SampleCount|Average|Sum|Minimum|Maximum|p(\d{1,2}|100)(\.\d{0,2})?|[ou]\d+(\.\d*)?)(_E|_L|_H)?`

Required: Yes

Errors

For information about the errors that are common to all actions, see Common Errors (p. 172).

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

MissingParameter

An input parameter that is required is missing.

HTTP Status Code: 400

ResourceNotFound

The named resource does not exist.

HTTP Status Code: 404

See Also

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

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

DeleteDashboards

Deletes all dashboards that you specify. You can specify up to 100 dashboards to delete. If there is an error during this call, no dashboards are deleted.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

DashboardNames.member.N

The dashboards to be deleted. This parameter is required.

Type: Array of strings

Required: Yes

Errors

For information about the errors that are common to all actions, see Common Errors (p. 172).

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

ResourceNotFound

The specified dashboard does not exist.

HTTP Status Code: 404

See Also

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

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

DeleteInsightRules

Permanently deletes the specified Contributor Insights rules.

If you create a rule, delete it, and then re-create it with the same name, historical data from the first time the rule was created might not be available.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

RuleNames.member.N

An array of the rule names to delete. If you need to find out the names of your rules, use DescribeInsightRules.

Type: Array of strings

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

Pattern: `[\x20-\x7E]+`

Required: Yes

Response Elements

The following element is returned by the service.

Failures.member.N

An array listing the rules that could not be deleted. You cannot delete built-in rules.

Type: Array of PartialFailure (p. 134) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 172).

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

See Also

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

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

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 either all metric alarms or all composite alarms are returned.

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. 170).

AlarmName

The name of the alarm.

Type: String

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

Required: No

AlarmTypes.member.N

Use this parameter to specify whether you want the operation to return metric alarms or composite alarms. If you omit this parameter, only metric alarms are returned.

Type: Array of strings

Valid Values: `CompositeAlarm | MetricAlarm`

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

Required: No

ScanBy

Specified whether to return the newest or oldest alarm history first. Specify `TimestampDescending` to have the newest event history returned first, and specify `TimestampAscending` to have the oldest history returned first.

Type: String

Valid Values: `TimestampDescending | TimestampAscending`

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. 98) objects

NextToken

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

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 172).

InvalidNextToken

The next token specified is invalid.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface

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

DescribeAlarms

Retrieves the specified alarms. You can filter the results by specifying a 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. 170).

ActionPrefix

Use this parameter to filter the results of the operation to only those alarms that use a certain alarm action. For example, you could specify the ARN of an SNS topic to find all alarms that send notifications to that topic.

Type: String

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

Required: No

AlarmNamePrefix

An alarm name prefix. If you specify this parameter, you receive information about all alarms that have names that start with this prefix.

If this parameter is specified, you cannot specify `AlarmNames`.

Type: String

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

Required: No

AlarmNames.member.N

The names of the alarms to retrieve information about.

Type: Array of strings

Array Members: Maximum number of 100 items.

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

Required: No

AlarmTypes.member.N

Use this parameter to specify whether you want the operation to return metric alarms or composite alarms. If you omit this parameter, only metric alarms are returned.

Type: Array of strings

Valid Values: `CompositeAlarm | MetricAlarm`

Required: No

ChildrenOfAlarmName

If you use this parameter and specify the name of a composite alarm, the operation returns information about the "children" alarms of the alarm you specify. These are the metric alarms and composite alarms referenced in the `AlarmRule` field of the composite alarm that you specify in `ChildrenOfAlarmName`. Information about the composite alarm that you name in `ChildrenOfAlarmName` is not returned.

If you specify `ChildrenOfAlarmName`, you cannot specify any other parameters in the request except for `MaxRecords` and `NextToken`. If you do so, you receive a validation error.

> **Note**
> Only the `Alarm Name`, `ARN`, `StateValue` (OK/ALARM/INSUFFICIENT_DATA), and `StateUpdatedTimestamp` information are returned by this operation when you use this parameter. To get complete information about these alarms, perform another `DescribeAlarms` operation and specify the parent alarm names in the `AlarmNames` parameter.

Type: String

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

Required: No

ParentsOfAlarmName

If you use this parameter and specify the name of a metric or composite alarm, the operation returns information about the "parent" alarms of the alarm you specify. These are the composite alarms that have `AlarmRule` parameters that reference the alarm named in `ParentsOfAlarmName`. Information about the alarm that you specify in `ParentsOfAlarmName` is not returned.

If you specify `ParentsOfAlarmName`, you cannot specify any other parameters in the request except for `MaxRecords` and `NextToken`. If you do so, you receive a validation error.

> **Note**
> Only the Alarm Name and ARN are returned by this operation when you use this parameter. To get complete information about these alarms, perform another `DescribeAlarms` operation and specify the parent alarm names in the `AlarmNames` parameter.

Type: String

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

Required: No

StateValue

Specify this parameter to receive information only about alarms that are currently in the state that you specify.

Type: String

Valid Values: `OK | ALARM | INSUFFICIENT_DATA`

Required: No

Response Elements

The following elements are returned by the service.

CompositeAlarms.member.N

The information about any composite alarms returned by the operation.

Type: Array of CompositeAlarm (p. 103) objects

MetricAlarms.member.N

The information about any metric alarms returned by the operation.

Type: Array of MetricAlarm (p. 120) objects

NextToken

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

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 172).

InvalidNextToken

The next token specified is invalid.

HTTP Status Code: 400

See Also

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

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

DescribeAlarmsForMetric

Retrieves the alarms for the specified metric. To filter the results, specify a statistic, period, or unit.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

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. 110) 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 1.

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. 120) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 172).

See Also

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

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

DescribeAnomalyDetectors

Lists the anomaly detection models that you have created in your account. You can list all models in your account or filter the results to only the models that are related to a certain namespace, metric name, or metric dimension.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

Dimensions.member.N

Limits the results to only the anomaly detection models that are associated with the specified metric dimensions. If there are multiple metrics that have these dimensions and have anomaly detection models associated, they're all returned.

Type: Array of Dimension (p. 110) objects

Array Members: Maximum number of 10 items.

Required: No

MaxResults

The maximum number of results to return in one operation. The maximum value that you can specify is 100.

To retrieve the remaining results, make another call with the returned `NextToken` value.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

MetricName

Limits the results to only the anomaly detection models that are associated with the specified metric name. If there are multiple metrics with this name in different namespaces that have anomaly detection models, they're all returned.

Type: String

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

Required: No

Namespace

Limits the results to only the anomaly detection models that are associated with the specified namespace.

Type: String

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

Pattern: `[^:].*`

Required: No

NextToken

Use the token returned by the previous operation to request the next page of results.

Type: String

Required: No

Response Elements

The following elements are returned by the service.

AnomalyDetectors.member.N

The list of anomaly detection models returned by the operation.

Type: Array of AnomalyDetector (p. 100) objects

NextToken

A token that you can use in a subsequent operation to retrieve the next set of results.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 172).

InternalServiceError

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

HTTP Status Code: 500

InvalidNextToken

The next token specified is invalid.

HTTP Status Code: 400

InvalidParameterValue

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

HTTP Status Code: 400

Example

The following example lists all the anomaly detectors for metrics with the name `CPUUtilization`.

Sample Request

```
{
    "MetricName:": "CPUUtilization"
}
```

Sample Response

```
{
    "AnomalyDetectors": [
        {
            "Namespace": "AWS/EC2",
            "MetricName": "CPUUtilization",
            "Dimensions": [
                {
                    "Name": "dimension1",
                    "Value": "value1"
                },
                {
                    "Name": "dimension2",
                    "Value": "value2"
                }
            ],
            "Stat": "Average",
            "Configuration": {
                "ExcludedTimeRanges": [

                ]
            }
        },
        {
            "Namespace": "AWS/EC2",
            "MetricName": "CPUUtilization",
            "Dimensions": [
                {
                    "Name": "dimension1",
                    "Value": "value1"
                }
            ],
            "Stat": "SampleCount",
            "Configuration": {
                "ExcludedTimeRanges": [

                ]
            }
        },
        {
            "Namespace": "APITest1",
            "MetricName": "Metric1",
            "Dimensions": [
                {
                    "Name": "dimension1",
                    "Value": "value1"
                }
            ],
            "Stat": "SampleCount",
            "Configuration": {
                "ExcludedTimeRanges": [

                ]
            }
        },
        {
            "Namespace": "CustomNamespace",
            "MetricName": "CPUUtilization",
            "Dimensions": [
                {
                    "Name": "dimension1",
                    "Value": "value1"
                },
                {
                    "Name": "dimension2",
                    "Value": "value2"
                }
            ],
            "Stat": "Maximum",
            "Configuration": {
                "ExcludedTimeRanges": [

                ]
            }
        }
    ]
}
```

See Also

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

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

DescribeInsightRules

Returns a list of all the Contributor Insights rules in your account. All rules in your account are returned with a single operation.

For more information about Contributor Insights, see Using Contributor Insights to Analyze High-Cardinality Data.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

MaxResults

This parameter is not currently used. Reserved for future use. If it is used in the future, the maximum value might be different.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 500.

Required: No

NextToken

Reserved for future use.

Type: String

Required: No

Response Elements

The following elements are returned by the service.

InsightRules.member.N

The rules returned by the operation.

Type: Array of InsightRule (p. 112) objects

NextToken

Reserved for future use.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 172).

InvalidNextToken

The next token specified is invalid.

HTTP Status Code: 400

See Also

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

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

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. 170).

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. 172).

See Also

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

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

DisableInsightRules

Disables the specified Contributor Insights rules. When rules are disabled, they do not analyze log groups and do not incur costs.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

RuleNames.member.N

An array of the rule names to disable. If you need to find out the names of your rules, use DescribeInsightRules.

Type: Array of strings

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

Pattern: `[\x20-\x7E]+`

Required: Yes

Response Elements

The following element is returned by the service.

Failures.member.N

An array listing the rules that could not be disabled. You cannot disable built-in rules.

Type: Array of PartialFailure (p. 134) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 172).

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

See Also

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

- AWS Command Line Interface

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

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. 170).

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. 172).

See Also

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

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

EnableInsightRules

Enables the specified Contributor Insights rules. When rules are enabled, they immediately begin analyzing log data.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

RuleNames.member.N

An array of the rule names to enable. If you need to find out the names of your rules, use DescribeInsightRules.

Type: Array of strings

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

Pattern: `[\x20-\x7E]+`

Required: Yes

Response Elements

The following element is returned by the service.

Failures.member.N

An array listing the rules that could not be enabled. You cannot disable or enable built-in rules.

Type: Array of PartialFailure (p. 134) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 172).

InvalidParameterValue

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

HTTP Status Code: 400

LimitExceeded

The operation exceeded one or more limits.

HTTP Status Code: 400

MissingParameter

An input parameter that is required is missing.

HTTP Status Code: 400

See Also

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

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

GetDashboard

Displays the details of the dashboard that you specify.

To copy an existing dashboard, use `GetDashboard`, and then use the data returned within `DashboardBody` as the template for the new dashboard when you call `PutDashboard` to create the copy.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

DashboardName

The name of the dashboard to be described.

Type: String

Required: Yes

Response Elements

The following elements are returned by the service.

DashboardArn

The Amazon Resource Name (ARN) of the dashboard.

Type: String

DashboardBody

The detailed information about the dashboard, including what widgets are included and their location on the dashboard. For more information about the `DashboardBody` syntax, see Dashboard Body Structure and Syntax.

Type: String

DashboardName

The name of the dashboard.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 172).

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

ResourceNotFound

The specified dashboard does not exist.

HTTP Status Code: 404

See Also

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

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

GetInsightRuleReport

This operation returns the time series data collected by a Contributor Insights rule. The data includes the identity and number of contributors to the log group.

You can also optionally return one or more statistics about each data point in the time series. These statistics can include the following:

- `UniqueContributors` -- the number of unique contributors for each data point.
- `MaxContributorValue` -- the value of the top contributor for each data point. The identity of the contributor might change for each data point in the graph.

  If this rule aggregates by COUNT, the top contributor for each data point is the contributor with the most occurrences in that period. If the rule aggregates by SUM, the top contributor is the contributor with the highest sum in the log field specified by the rule's `Value`, during that period.
- `SampleCount` -- the number of data points matched by the rule.
- `Sum` -- the sum of the values from all contributors during the time period represented by that data point.
- `Minimum` -- the minimum value from a single observation during the time period represented by that data point.
- `Maximum` -- the maximum value from a single observation during the time period represented by that data point.
- `Average` -- the average value from all contributors during the time period represented by that data point.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

EndTime

The end time of the data to use in the report. When used in a raw HTTP Query API, it is formatted as `yyyy-MM-dd'T'HH:mm:ss`. For example, `2019-07-01T23:59:59`.

Type: Timestamp

Required: Yes

MaxContributorCount

The maximum number of contributors to include in the report. The range is 1 to 100. If you omit this, the default of 10 is used.

Type: Integer

Required: No

Metrics.member.N

Specifies which metrics to use for aggregation of contributor values for the report. You can specify one or more of the following metrics:

- `UniqueContributors` -- the number of unique contributors for each data point.
- `MaxContributorValue` -- the value of the top contributor for each data point. The identity of the contributor might change for each data point in the graph.

If this rule aggregates by COUNT, the top contributor for each data point is the contributor with the most occurrences in that period. If the rule aggregates by SUM, the top contributor is the contributor with the highest sum in the log field specified by the rule's `Value`, during that period.

- `SampleCount` -- the number of data points matched by the rule.
- `Sum` -- the sum of the values from all contributors during the time period represented by that data point.
- `Minimum` -- the minimum value from a single observation during the time period represented by that data point.
- `Maximum` -- the maximum value from a single observation during the time period represented by that data point.
- `Average` -- the average value from all contributors during the time period represented by that data point.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 32.

Pattern: `[\x20-\x7E]+`

Required: No

OrderBy

Determines what statistic to use to rank the contributors. Valid values are SUM and MAXIMUM.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 32.

Pattern: `[\x20-\x7E]+`

Required: No

Period

The period, in seconds, to use for the statistics in the `InsightRuleMetricDatapoint` results.

Type: Integer

Valid Range: Minimum value of 1.

Required: Yes

RuleName

The name of the rule that you want to see data from.

Type: String

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

Pattern: `[\x20-\x7E]+`

Required: Yes

StartTime

The start time of the data to use in the report. When used in a raw HTTP Query API, it is formatted as `yyyy-MM-dd'T'HH:mm:ss`. For example, `2019-07-01T23:59:59`.

Type: Timestamp

Required: Yes

Response Elements

The following elements are returned by the service.

AggregateValue

The sum of the values from all individual contributors that match the rule.

Type: Double

AggregationStatistic

Specifies whether this rule aggregates contributor data by COUNT or SUM.

Type: String

ApproximateUniqueCount

An approximate count of the unique contributors found by this rule in this time period.

Type: Long

Contributors.member.N

An array of the unique contributors found by this rule in this time period. If the rule contains multiple keys, each combination of values for the keys counts as a unique contributor.

Type: Array of InsightRuleContributor (p. 114) objects

KeyLabels.member.N

An array of the strings used as the keys for this rule. The keys are the dimensions used to classify contributors. If the rule contains more than one key, then each unique combination of values for the keys is counted as a unique contributor.

Type: Array of strings

MetricDatapoints.member.N

A time series of metric data points that matches the time period in the rule request.

Type: Array of InsightRuleMetricDatapoint (p. 116) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 172).

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

ResourceNotFound

The named resource does not exist.

HTTP Status Code: 404

See Also

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

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

GetMetricData

You can use the `GetMetricData` API to retrieve as many as 500 different metrics in a single request, with a total of as many as 100,800 data points. You can also optionally perform math expressions on the values of the returned statistics, to create new time series that represent new insights into your data. For example, using Lambda metrics, you could divide the Errors metric by the Invocations metric to get an error rate time series. For more information about metric math expressions, see Metric Math Syntax and Functions in the *Amazon CloudWatch User Guide*.

Calls to the `GetMetricData` API have a different pricing structure than calls to `GetMetricStatistics`. For more information about pricing, see Amazon CloudWatch Pricing.

Amazon CloudWatch retains metric data as follows:

- Data points with a period of less than 60 seconds are available for 3 hours. These data points are high-resolution metrics and are available only for custom metrics that have been defined with a `StorageResolution` of 1.
- 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).

Data points that are initially published with a shorter period are aggregated together for long-term storage. For example, if you collect data using a period of 1 minute, the data remains available for 15 days with 1-minute resolution. After 15 days, this data is still available, but is aggregated and retrievable only with a resolution of 5 minutes. After 63 days, the data is further aggregated and is available with a resolution of 1 hour.

If you omit `Unit` in your request, all data that was collected with any unit is returned, along with the corresponding units that were specified when the data was reported to CloudWatch. If you specify a unit, the operation returns only data that was collected with that unit specified. If you specify a unit that does not match the data collected, the results of the operation are null. CloudWatch does not perform unit conversions.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

EndTime

The time stamp indicating the latest data to be returned.

The value specified is exclusive; results include data points up to the specified time stamp.

For better performance, specify `StartTime` and `EndTime` values that align with the value of the metric's `Period` and sync up with the beginning and end of an hour. For example, if the `Period` of a metric is 5 minutes, specifying 12:05 or 12:30 as `EndTime` can get a faster response from CloudWatch than setting 12:07 or 12:29 as the `EndTime`.

Type: Timestamp

Required: Yes

MaxDatapoints

The maximum number of data points the request should return before paginating. If you omit this, the default of 100,800 is used.

Type: Integer

Required: No

MetricDataQueries.member.N

The metric queries to be returned. A single `GetMetricData` call can include as many as 500 `MetricDataQuery` structures. Each of these structures can specify either a metric to retrieve, or a math expression to perform on retrieved data.

Type: Array of MetricDataQuery (p. 125) objects

Required: Yes

NextToken

Include this value, if it was returned by the previous call, to get the next set of data points.

Type: String

Required: No

ScanBy

The order in which data points should be returned. `TimestampDescending` returns the newest data first and paginates when the `MaxDatapoints` limit is reached. `TimestampAscending` returns the oldest data first and paginates when the `MaxDatapoints` limit is reached.

Type: String

Valid Values: `TimestampDescending | TimestampAscending`

Required: No

StartTime

The time stamp indicating the earliest data to be returned.

The value specified is inclusive; results include data points with the specified time stamp.

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.

If you set `Period` to 5, 10, or 30, the start time of your request is rounded down to the nearest time that corresponds to even 5-, 10-, or 30-second divisions of a minute. For example, if you make a query at (HH:mm:ss) 01:05:23 for the previous 10-second period, the start time of your request is rounded down and you receive data from 01:05:10 to 01:05:20. If you make a query at 15:07:17 for the previous 5 minutes of data, using a period of 5 seconds, you receive data timestamped between 15:02:15 and 15:07:15.

For better performance, specify `StartTime` and `EndTime` values that align with the value of the metric's `Period` and sync up with the beginning and end of an hour. For example, if the `Period` of a metric is 5 minutes, specifying 12:05 or 12:30 as `StartTime` can get a faster response from CloudWatch than setting 12:07 or 12:29 as the `StartTime`.

Type: Timestamp

Required: Yes

Response Elements

The following elements are returned by the service.

Messages.member.N

Contains a message about this `GetMetricData` operation, if the operation results in such a message. An example of a message that might be returned is `Maximum number of allowed metrics exceeded`. If there is a message, as much of the operation as possible is still executed.

A message appears here only if it is related to the global `GetMetricData` operation. Any message about a specific metric returned by the operation appears in the `MetricDataResult` object returned for that metric.

Type: Array of MessageData (p. 118) objects

MetricDataResults.member.N

The metrics that are returned, including the metric name, namespace, and dimensions.

Type: Array of MetricDataResult (p. 127) objects

NextToken

A token that marks the next batch of returned results.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 172).

InvalidNextToken

The next token specified is invalid.

HTTP Status Code: 400

Examples

Example

The following example requests three separate metrics across two namespaces.

Sample Request

```
{
  "StartTime": 1518867432,
  "EndTime": 1518868032,
  "MetricDataQueries": [
    {
      "Id": "m1",
      "MetricStat": {
        "Metric": {
          "Namespace": "AWS/EC2",
          "MetricName": "NetworkOut",
          "Dimensions": [
            {
              "Name": "AutoScalingGroupName",
              "Value": "my-asg-A"
            }
          ]
        },
        "Period": 300,
        "Stat": "Average",
        "Unit": "Bytes"
      }
    },
    {
      "Id": "m2",
      "MetricStat": {
        "Metric": {
          "Namespace": "AWS/EC2",
          "MetricName": "NetworkOut",
          "Dimensions": [
            {
              "Name": "AutoScalingGroupName",
              "Value": "my-asg-A"
            }
          ]
        },
        "Period": 300,
        "Stat": "SampleCount",
        "Unit": "Count"
      }
    },
    {
      "Id": "m3",
      "MetricStat": {
        "Metric": {
          "Namespace": "AWS/ELB",
          "MetricName": "HealthyHostCount",
          "Dimensions": [
            {
              "Name": "LoadBalancerName",
              "Value": "my-lb-B"
            }
          ]
        },
        "Period": 300,
        "Stat": "Sum",
        "Unit": "None"
      }
    }
  ]
}
```

Sample Response

```
{
  "MetricDataResults": [
    {
      "Id": "m1",
      "StatusCode": "Complete",
      "Label": "AWS/EC2 NetworkOut Average",
      "Timestamps": [
        1518868032,
        1518867732,
        1518867432
      ],
      "Values": [
        15000,
        14000,
        16000
      ]
    },
    {
      "Id": "m2",
      "StatusCode": "Complete",
      "Label": "AWS/EC2 NetworkOut SampleCount",
      "Timestamps": [
        1518868032,
        1518867732,
        1518867432
      ],
      "Values": [
        15,
        14,
        16
      ]
    },
    {
      "Id": "m3",
      "StatusCode": "Complete",
      "Label": "AWS/EC2 HealthyHostCount",
      "Timestamps": [
        1518868032,
        1518867732,
        1518867432
      ],
      "Values": [
        15,
        14,
        16
      ]
    }
  ]
}
```

Example

The following example retrieves the `NetworkOut` metric for two Auto Scaling groups, and uses them in an expression. These two metrics are called m1 and m2, and the expression calculates e1 as the results of m2/m1. The raw values and time stamps of the `NetworkOut` metrics are not returned.

Sample Request

```
{
  "StartTime": 1518867432,
  "EndTime": 1518868032,
  "MetricQueries": [
    {
      "Id": "e1",
      "Expression": "m2 / m1",
      "Label": "my-asg-B / my-asg-A"
    },
    {
      "Id": "m1",
      "MetricStat": {
        "Metric": {
          "Namespace": "AWS/EC2",
          "MetricName": "NetworkOut",
          "Dimensions": [
            {
              "Name": "AutoScalingGroupName",
              "Value": "my-asg-A"
            }
          ]
        },
        "Period": 300,
        "Stat": "SampleCount",
        "Unit": "Bytes"
      },
      "ReturnData": false
    },
    {
      "Id": "m2",
      "MetricStat": {
        "Metric": {
          "Namespace": "AWS/EC2",
          "MetricName": "NetworkOut",
          "Dimensions": [
            {
              "Name": "AutoScalingGroupName",
              "Value": "my-asg-B"
            }
          ]
        },
        "Period": 300,
        "Stat": "SampleCount",
        "Unit": "Bytes"
      },
      "ReturnData": false
    }
  ]
}
```

Sample Response

```
{
  "MetricDataResults": [
    {
      "Id": "m1",
      "StatusCode": "Complete"
    },
    {
      "Id": "m2",
      "StatusCode": "Complete"
    },
    {
      "Id": "e1",
      "StatusCode": "Complete",
      "Label": "my-asg-B / my-asg-A",
      "Timestamps": [
        1518868032,
        1518867732,
        1518867432
      ],
      "Values": [
        100,
        100,
        100
      ]
    }
  ]
}
```

Example

In the following example, two levels of expressions are used, with the result of one expression used as an input to the next expression:

Sample Request

```
{
  "StartTime": 1518867432,
  "EndTime": 1518868032,
  "MetricDataQueries": [
    {
      "Id": "e1",
      "Expression": "e2 + m3",
      "Label": "my-asg-A * my-asg-B + my-asg-C"
    },
    {
      "Id": "e2",
      "Expression": "m1 * m2",
       "Label": "my-asg-A * my-asg-B"
    },
    {
      "Id": "m1",
      "MetricStat": {
        "Metric": {
          "Namespace": "AWS/EC2",
          "MetricName": "NetworkOut",
          "Dimensions": [
            {
              "Name": "AutoScalingGroupName",
              "Value": "my-asg-A"
            }
          ]
        },
        "Period": 300,
        "Stat": "SampleCount",
        "Unit": "Bytes"
      },
      "ReturnData": false
    },
    {
      "Id": "m2",
      "MetricStat": {
        "Metric": {
          "Namespace": "AWS/EC2",
          "MetricName": "NetworkOut",
          "Dimensions": [
            {
              "Name": "AutoScalingGroupName",
              "Value": "my-asg-B"
            }
          ]
        },
        "Period": 300,
        "Stat": "SampleCount",
        "Unit": "Bytes"
      },
      "ReturnData": false
    },
    {
      "Id": "m3",
      "MetricStat": {
        "Metric": {
          "Namespace": "AWS/EC2",
          "MetricName": "NetworkOut",
          "Dimensions": [
            {
              "Name": "AutoScalingGroupName",
              "Value": "my-asg-C"
            }
          ]
        },
        "Period": 300,
        "Stat": "SampleCount",
        "Unit": "Bytes"
      },
      "ReturnData": false
    }
  ]
}
```

Sample Response

```
{
  "MetricDataResults": [
    {
      "Id": "m1",
      "StatusCode": "Complete"
    },
    {
      "Id": "m2",
      "StatusCode": "Complete"
    },
    {
      "Id": "m3",
      "StatusCode": "Complete"
    },
    {
      "Id": "e1",
      "StatusCode": "Complete",
      "Label": "my-asg-A * my-asg-B + my-asg-C",
      "Timestamps": [
        1518868032,
        1518867732,
        1518867432
      ],
      "Values": [
        200,
        200,
        200
      ]
    },
    {
      "Id": "e2",
      "StatusCode": "Complete",
      "Label": "my-asg-A * my-asg-B",
      "Timestamps": [
        1518868032,
        1518867732,
        1518867432
      ],
      "Values": [
        100,
        100,
        100
      ]
    }
  ]
}
```

Example

In the following example, custom metrics are searched and assigned IDs that contain either "error" or "request", even if the original metric names did not contain those words. Then an error rate is calculated using the `METRICS("string")` function on the assigned IDs.

Sample Request

```
{
  "StartTime": 1518867432,
  "EndTime": 1518868032,
  "MetricDataQueries": [
    {
      "Id": "errorRate",
      "Label": "Error Rate",
      "Expression": "errors/requests"
    },
    {
      "Id": "errorRatePercent",
      "Label": "% Error Rate",
      "Expression": "errorRate*100"
    },
    {
      "Id": "requests",
      "Expression": "SUM(METRICS('request'))",
      "ReturnData": false
    },
    {
      "Id": "errors",
      "Expression": "SUM(METRICS('error'))",
      "ReturnData": false
    },
    {
      "Id": "error1",
      "MetricStat": {
        "Metric": {
          "Namespace": "MyService",
          "MetricName": "BadRequests",
          "Dimensions": [
            {
              "Name": "Component",
              "Value": "component-1"
            }
          ]
        },
        "Period": 60,
        "Stat": "Sum"
      },
      "ReturnData": false
    },
    {
      "Id": "error2",
      "MetricStat": {
        "Metric": {
          "Namespace": "MyService",
          "MetricName": "ConnectionErrors",
          "Dimensions": [
            {
              "Name": "Component",
              "Value": "component-1"
            }
          ]
        },
        "Period": 60,
        "Stat": "Sum"
      },
      "ReturnData": false
    },
    {
      "Id": "request1",
      "MetricStat": {
        "Metric": {
          "Namespace": "MyService",
          "MetricName": "InternalRequests",
          "Dimensions": [
            {
              "Name": "Component",
              "Value": "component-1"
            }
          ]
        },
        "Period": 60,
        "Stat": "Sum"
      },
      "ReturnData": false
    },
    {
      "Id": "request2",
      "MetricStat": {
        "Metric": {
          "Namespace": "MyService",
          "MetricName": "ExternalRequests",
          "Dimensions": [
            {
              "Name": "Component",
              "Value": "component-1"
            }
          ]
        },
        "Period": 60,
        "Stat": "Sum"
      },
      "ReturnData": false
    }
  ]
}
```

Sample Response

```
{
  "MetricDataResults": [
    {
      "Id": "errorRate",
      "Label": "Error Rate",
      "StatusCode": "Complete",
      "Timestamps": [
        1518868032,
        1518867732,
        1518867432
      ],
      "Values": [
        0.1,
        0.5,
        0.3
      ]
    },
    {
      "Id": "errorRatePercent",
      "Label": "% Error Rate",
      "StatusCode": "Complete",
      "Timestamps": [
        1518868032,
        1518867732,
        1518867432
      ],
      "Values": [
        10,
        50,
        30
      ]
    }
  ]
}
```

See Also

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

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

GetMetricStatistics

Gets statistics for the specified metric.

The maximum number of data points returned from a single call is 1,440. If you request more than 1,440 data points, 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. Data points are not returned in chronological order.

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, 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.

CloudWatch needs raw data points to calculate percentile statistics. If you publish data using a statistic set instead, you can only retrieve percentile statistics for this data if one of the following conditions is true:

- The SampleCount value of the statistic set is 1.
- The Min and the Max values of the statistic set are equal.

Percentile statistics are not available for metrics when any of the metric values are negative numbers.

Amazon CloudWatch retains metric data as follows:

- Data points with a period of less than 60 seconds are available for 3 hours. These data points are high-resolution metrics and are available only for custom metrics that have been defined with a `StorageResolution` of 1.
- 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).

Data points that are initially published with a shorter period are aggregated together for long-term storage. For example, if you collect data using a period of 1 minute, the data remains available for 15 days with 1-minute resolution. After 15 days, this data is still available, but is aggregated and retrievable only with a resolution of 5 minutes. After 63 days, the data is further aggregated and is available with a resolution of 1 hour.

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

For information about 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. 170).

Dimensions.member.N

The dimensions. If the metric contains multiple dimensions, you must include a value for each dimension. CloudWatch treats each unique combination of dimensions as a separate metric. If a specific combination of dimensions was not published, you can't retrieve statistics for it. 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*. For more information about specifying dimensions, see Publishing Metrics in the *Amazon CloudWatch User Guide*.

Type: Array of Dimension (p. 110) 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 include data points up to the specified time stamp. In a raw HTTP query, 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. When calling `GetMetricStatistics`, you must specify either `Statistics` or `ExtendedStatistics`, but not both. Percentile statistics are not available for metrics when any of the metric values are negative numbers.

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. For metrics with regular resolution, a period can be as short as one minute (60 seconds) and must be a multiple of 60. For high-resolution metrics that are collected at intervals of less than one minute, the period can be 1, 5, 10, 30, 60, or any multiple of 60. High-resolution metrics are those metrics stored by a `PutMetricData` call that includes a `StorageResolution` of 1 second.

If the `StartTime` parameter specifies a time stamp that is greater than 3 hours ago, you must specify the period as follows or no data points in that time range is returned:

- Start time between 3 hours and 15 days ago - Use a multiple of 60 seconds (1 minute).
- 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 1.

Required: Yes

StartTime

The time stamp that determines the first data point to return. 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. In a raw HTTP query, 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.

If you set `Period` to 5, 10, or 30, the start time of your request is rounded down to the nearest time that corresponds to even 5-, 10-, or 30-second divisions of a minute. For example, if you make a query at (HH:mm:ss) 01:05:23 for the previous 10-second period, the start time of your request is rounded down and you receive data from 01:05:10 to 01:05:20. If you make a query at 15:07:17 for the previous 5 minutes of data, using a period of 5 seconds, you receive data timestamped between 15:02:15 and 15:07:15.

Type: Timestamp

Required: Yes

Statistics.member.N

The metric statistics, other than percentile. For percentile statistics, use `ExtendedStatistics`. When calling `GetMetricStatistics`, you must specify either `Statistics` or `ExtendedStatistics`, but not both.

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. If you omit `Unit`, all data that was collected with any unit is returned, along with the corresponding units that were specified when the data was reported to CloudWatch. If you specify a unit, the operation returns only data that was collected with that unit specified. If you specify a unit that does not match the data collected, the results of the operation are null. CloudWatch does not perform unit conversions.

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. 108) 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. 172).

InternalServiceError

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

HTTP Status Code: 500

InvalidParameterCombination

Parameters were used together that cannot be 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

See Also

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

- AWS Command Line Interface

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

GetMetricWidgetImage

You can use the `GetMetricWidgetImage` API to retrieve a snapshot graph of one or more Amazon CloudWatch metrics as a bitmap image. You can then embed this image into your services and products, such as wiki pages, reports, and documents. You could also retrieve images regularly, such as every minute, and create your own custom live dashboard.

The graph you retrieve can include all CloudWatch metric graph features, including metric math and horizontal and vertical annotations.

There is a limit of 20 transactions per second for this API. Each `GetMetricWidgetImage` action has the following limits:

- As many as 100 metrics in the graph.
- Up to 100 KB uncompressed payload.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

MetricWidget

A JSON string that defines the bitmap graph to be retrieved. The string includes the metrics to include in the graph, statistics, annotations, title, axis limits, and so on. You can include only one `MetricWidget` parameter in each `GetMetricWidgetImage` call.

For more information about the syntax of `MetricWidget` see GetMetricWidgetImage: Metric Widget Structure and Syntax.

If any metric on the graph could not load all the requested data points, an orange triangle with an exclamation point appears next to the graph legend.

Type: String

Required: Yes

OutputFormat

The format of the resulting image. Only PNG images are supported.

The default is `png`. If you specify `png`, the API returns an HTTP response with the content-type set to `text/xml`. The image data is in a `MetricWidgetImage` field. For example:

```
<GetMetricWidgetImageResponse xmlns=<URLstring>>

<GetMetricWidgetImageResult>

<MetricWidgetImage>

iVBORw0KGgoAAAANSUhEUgAAAlgAAAGQEAYAAAAip...

</MetricWidgetImage>

</GetMetricWidgetImageResult>

<ResponseMetadata>

<RequestId>6f0d4192-4d42-11e8-82c1-f539a07e0e3b</RequestId>
```

```
</ResponseMetadata>

</GetMetricWidgetImageResponse>
```

The image/png setting is intended only for custom HTTP requests. For most use cases, and all actions using an AWS SDK, you should use png. If you specify image/png, the HTTP response has a content-type set to image/png, and the body of the response is a PNG image.

Type: String

Required: No

Response Elements

The following element is returned by the service.

MetricWidgetImage

The image of the graph, in the output format specified.

Type: Base64-encoded binary data object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 172).

Example

The following is an example of a GetMetricWidgetImage call. This example displays a graph showing an image of the Average statistic for the CPUUtilization metric for two Amazon EC2 instances, with both horizontal and vertical annotations.

```
{
   "OutputFormat":"png",
   "MetricWidget":{
      "width":600,
      "height":395,
      "metrics":[
         [
            "AWS/EC2",
            "CPUUtilization",
            "InstanceId",
            "i-1234567890abcdef0",
            {
               "stat":"Average"
            }
         ],
         [
            "AWS/EC2",
            "CPUUtilization",
            "InstanceId",
            "i-0987654321abcdef0",
            {
               "stat":"Average"
            }
         ]
      ],
      "period":300,
      "start":"-P30D",
      "end":"PT0H",
      "stacked":false,
      "yAxis":{
         "left":{
            "min":0.1,
            "max":1
         },
         "right":{
            "min":0
         }
      },
      "title":"CPU for Two Instances",
      "annotations":{
         "horizontal":[
            {
               "color":"#ff6961",
               "label":"Trouble threshold start",
               "fill":"above",
               "value":0.5
            }
         ],
         "vertical":[
            {
               "visible":true,
               "color":"#9467bd",
               "label":"Bug fix deployed",
               "value":"2018-08-28T15:25:26Z",
               "fill":"after"
            }
         ]
      }
   }
}
```

See Also

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

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

ListDashboards

Returns a list of the dashboards for your account. If you include `DashboardNamePrefix`, only those dashboards with names starting with the prefix are listed. Otherwise, all dashboards in your account are listed.

`ListDashboards` returns up to 1000 results on one page. If there are more than 1000 dashboards, you can call `ListDashboards` again and include the value you received for `NextToken` in the first call, to receive the next 1000 results.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

DashboardNamePrefix

If you specify this parameter, only the dashboards with names starting with the specified string are listed. The maximum length is 255, and valid characters are A-Z, a-z, 0-9, ".", "-", and "_".

Type: String

Required: No

NextToken

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

Type: String

Required: No

Response Elements

The following elements are returned by the service.

DashboardEntries.member.N

The list of matching dashboards.

Type: Array of DashboardEntry (p. 106) objects

NextToken

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

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 172).

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

See Also

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

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

ListMetrics

List the specified metrics. You can use the returned metrics with GetMetricData or GetMetricStatistics 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 GetMetricData or GetMetricStatistics.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

Dimensions.member.N

The dimensions to filter against.

Type: Array of DimensionFilter (p. 111) 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

Required: No

Response Elements

The following elements are returned by the service.

Metrics.member.N

The metrics.

Type: Array of Metric (p. 119) objects

NextToken

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

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 172).

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

See Also

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

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

ListTagsForResource

Displays the tags associated with a CloudWatch resource. Currently, alarms and Contributor Insights rules support tagging.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

ResourceARN

The ARN of the CloudWatch resource that you want to view tags for.

The ARN format of an alarm is `arn:aws:cloudwatch:`*`Region`*`:`*`account-id`*`:alarm:`*`alarm-name`*

The ARN format of a Contributor Insights rule is `arn:aws:cloudwatch:`*`Region`*`:`*`account-id`*`:insight-rule:`*`insight-rule-name`*

For more information about ARN format, see Resource Types Defined by Amazon CloudWatch in the *Amazon Web Services General Reference*.

Type: String

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

Required: Yes

Response Elements

The following element is returned by the service.

Tags.member.N

The list of tag keys and values associated with the resource you specified.

Type: Array of Tag (p. 137) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 172).

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

ResourceNotFound

The named resource does not exist.

HTTP Status Code: 404

See Also

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

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

PutAnomalyDetector

Creates an anomaly detection model for a CloudWatch metric. You can use the model to display a band of expected normal values when the metric is graphed.

For more information, see CloudWatch Anomaly Detection.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

Configuration

The configuration specifies details about how the anomaly detection model is to be trained, including time ranges to exclude when training and updating the model. You can specify as many as 10 time ranges.

The configuration can also include the time zone to use for the metric.

You can in

Type: AnomalyDetectorConfiguration (p. 102) object

Required: No

Dimensions.member.N

The metric dimensions to create the anomaly detection model for.

Type: Array of Dimension (p. 110) objects

Array Members: Maximum number of 10 items.

Required: No

MetricName

The name of the metric to create the anomaly detection model for.

Type: String

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

Required: Yes

Namespace

The namespace of the metric to create the anomaly detection model for.

Type: String

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

Pattern: `[^:].*`

Required: Yes

Stat

The statistic to use for the metric and the anomaly detection model.

Type: String

Pattern: `(SampleCount|Average|Sum|Minimum|Maximum|p(\d{1,2}|100)(\.\d{0,2})?|[ou]\d+(\.\d*)?)(_E|_L|_H)?`

Required: Yes

Errors

For information about the errors that are common to all actions, see Common Errors (p. 172).

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

LimitExceeded

The operation exceeded one or more limits.

HTTP Status Code: 400

MissingParameter

An input parameter that is required is missing.

HTTP Status Code: 400

Example

The following example creates an anomaly detection model for the metric `CPUUtilization`.

Sample Request

```
{
    "Namespace": "AWS/EC2",
    "MetricName:": "CPUUtilization",
    "Stat": "Average",
    "Dimensions": [
        {
            "Name": "dimension1",
            "Value": "value1"
        },
        {
            "Name": "dimension2",
            "Value": "value2"
        }
    ]
}
```

See Also

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

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

PutCompositeAlarm

Creates or updates a *composite alarm*. When you create a composite alarm, you specify a rule expression for the alarm that takes into account the alarm states of other alarms that you have created. The composite alarm goes into ALARM state only if all conditions of the rule are met.

The alarms specified in a composite alarm's rule expression can include metric alarms and other composite alarms.

Using composite alarms can reduce alarm noise. You can create multiple metric alarms, and also create a composite alarm and set up alerts only for the composite alarm. For example, you could create a composite alarm that goes into ALARM state only when more than one of the underlying metric alarms are in ALARM state.

Currently, the only alarm actions that can be taken by composite alarms are notifying SNS topics.

> **Note**
> It is possible to create a loop or cycle of composite alarms, where composite alarm A depends on composite alarm B, and composite alarm B also depends on composite alarm A. In this scenario, you can't delete any composite alarm that is part of the cycle because there is always still a composite alarm that depends on that alarm that you want to delete.
> To get out of such a situation, you must break the cycle by changing the rule of one of the composite alarms in the cycle to remove a dependency that creates the cycle. The simplest change to make to break a cycle is to change the `AlarmRule` of one of the alarms to `False`.
> Additionally, the evaluation of composite alarms stops if CloudWatch detects a cycle in the evaluation path.

When this operation creates an alarm, the alarm state is immediately set to `INSUFFICIENT_DATA`. The alarm is then evaluated and its state is set appropriately. Any actions associated with the new state are then executed. For a composite alarm, this initial time after creation is the only time that the alarm can be in `INSUFFICIENT_DATA` state.

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

ActionsEnabled

Indicates whether actions should be executed during any changes to the alarm state of the composite alarm. The default is `TRUE`.

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:sns:`*`region`*`:`*`account-id`*`:`*`sns-topic-name`*

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 composite alarm.

Type: String

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

Required: No

AlarmName

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

Type: String

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

Required: Yes

AlarmRule

An expression that specifies which other alarms are to be evaluated to determine this composite alarm's state. For each alarm that you reference, you designate a function that specifies whether that alarm needs to be in ALARM state, OK state, or INSUFFICIENT_DATA state. You can use operators (AND, OR and NOT) to combine multiple functions in a single expression. You can use parenthesis to logically group the functions in your expression.

You can use either alarm names or ARNs to reference the other alarms that are to be evaluated.

Functions can include the following:

- `ALARM("`*`alarm-name or alarm-ARN`*`")` is TRUE if the named alarm is in ALARM state.
- `OK("`*`alarm-name or alarm-ARN`*`")` is TRUE if the named alarm is in OK state.
- `INSUFFICIENT_DATA("`*`alarm-name or alarm-ARN`*`")` is TRUE if the named alarm is in INSUFFICIENT_DATA state.
- `TRUE` always evaluates to TRUE.
- `FALSE` always evaluates to FALSE.

TRUE and FALSE are useful for testing a complex `AlarmRule` structure, and for testing your alarm actions.

Alarm names specified in `AlarmRule` can be surrounded with double-quotes ("), but do not have to be.

The following are some examples of `AlarmRule`:

- `ALARM(CPUUtilizationTooHigh) AND ALARM(DiskReadOpsTooHigh)` specifies that the composite alarm goes into ALARM state only if both CPUUtilizationTooHigh and DiskReadOpsTooHigh alarms are in ALARM state.
- `ALARM(CPUUtilizationTooHigh) AND NOT ALARM(DeploymentInProgress)` specifies that the alarm goes to ALARM state if CPUUtilizationTooHigh is in ALARM state and DeploymentInProgress is not in ALARM state. This example reduces alarm noise during a known deployment window.
- `(ALARM(CPUUtilizationTooHigh) OR ALARM(DiskReadOpsTooHigh)) AND OK(NetworkOutTooHigh)` goes into ALARM state if CPUUtilizationTooHigh OR DiskReadOpsTooHigh is in ALARM state, and if NetworkOutTooHigh is in OK state. This provides another example of using a composite alarm to prevent noise. This rule ensures that you are

not notified with an alarm action on high CPU or disk usage if a known network problem is also occurring.

The `AlarmRule` can specify as many as 100 "children" alarms. The `AlarmRule` expression can have as many as 500 elements. Elements are child alarms, TRUE or FALSE statements, and parentheses.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 10240.

Required: Yes

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:sns:`*`region`*`:`*`account-id`*`:`*`sns-topic-name`*

Type: Array of strings

Array Members: Maximum number of 5 items.

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

Required: No

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:sns:`*`region`*`:`*`account-id`*`:`*`sns-topic-name`*

Type: Array of strings

Array Members: Maximum number of 5 items.

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

Required: No

Tags.member.N

A list of key-value pairs to associate with the composite alarm. You can associate as many as 50 tags with an alarm.

Tags can help you organize and categorize your resources. You can also use them to scope user permissions, by granting a user permission to access or change only resources with certain tag values.

Type: Array of Tag (p. 137) objects

Required: No

Errors

For information about the errors that are common to all actions, see Common Errors (p. 172).

LimitExceeded

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

HTTP Status Code: 400

Example

Composite alarm example

The following example creates an alarm that notifies an SNS group when either of two specified metric alarms exceeds its threshold.

Sample Request

```
{
    "AlarmDescription": "The host is experiencing problems",
    "AlarmRule": "ALARM(CPUUtilizationTooHigh) OR ALARM(DiskReadOpsTooHigh)",
    "AlarmName": "overall-health-alarm",
    "AlarmActions": [
        "arn:aws:sns:us-west-1:123456789012:my_sns_topic"
    ]
}
```

See Also

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

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

PutDashboard

Creates a dashboard if it does not already exist, or updates an existing dashboard. If you update a dashboard, the entire contents are replaced with what you specify here.

All dashboards in your account are global, not region-specific.

A simple way to create a dashboard using `PutDashboard` is to copy an existing dashboard. To copy an existing dashboard using the console, you can load the dashboard and then use the View/edit source command in the Actions menu to display the JSON block for that dashboard. Another way to copy a dashboard is to use `GetDashboard`, and then use the data returned within `DashboardBody` as the template for the new dashboard when you call `PutDashboard`.

When you create a dashboard with `PutDashboard`, a good practice is to add a text widget at the top of the dashboard with a message that the dashboard was created by script and should not be changed in the console. This message could also point console users to the location of the `DashboardBody` script or the CloudFormation template used to create the dashboard.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

DashboardBody

The detailed information about the dashboard in JSON format, including the widgets to include and their location on the dashboard. This parameter is required.

For more information about the syntax, see Dashboard Body Structure and Syntax.

Type: String

Required: Yes

DashboardName

The name of the dashboard. If a dashboard with this name already exists, this call modifies that dashboard, replacing its current contents. Otherwise, a new dashboard is created. The maximum length is 255, and valid characters are A-Z, a-z, 0-9, "-", and "_". This parameter is required.

Type: String

Required: Yes

Response Elements

The following element is returned by the service.

DashboardValidationMessages.member.N

If the input for `PutDashboard` was correct and the dashboard was successfully created or modified, this result is empty.

If this result includes only warning messages, then the input was valid enough for the dashboard to be created or modified, but some elements of the dashboard might not render.

If this result includes error messages, the input was not valid and the operation failed.

Type: Array of DashboardValidationMessage (p. 107) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 172).

InternalServiceError

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

HTTP Status Code: 500

InvalidParameterInput

Some part of the dashboard data is invalid.

HTTP Status Code: 400

Examples

Example

The following example creates a dashboard with just one text widget.

```


{
"DashboardName":"Dashboard with only one text widget"
"DashboardBody":
{
   "widgets":[
      {
         "type":"text",
         "x":0,
         "y":7,
         "width":3,
         "height":3,
         "properties":{
            "markdown":"Hello world"
         }
      }
   ]
}
}
```

Example

The following example modifies an existing dashboard to include one metric widget and one text widget.

```
{
    "DashboardName": "Two-Widget Dashboard",
    "DashboardBody": {
        "widgets": [
            {
                "type": "metric",
                "x": 0,
                "y": 0,
                "width": 12,
                "height": 6,
                "properties": {
                    "metrics": [
                        [
                            "AWS/EC2",
                            "CPUUtilization",
                            "InstanceId",
                            "i-012345"
                        ]
                    ],
                    "period": 300,
                    "stat": "Average",
                    "region": "us-east-1",
                    "title": "EC2 Instance CPU"
                }
            },
            {
                "type": "text",
                "x": 0,
                "y": 7,
                "width": 3,
                "height": 3,
                "properties": {
                    "markdown": "Hello world"
                }
            }
        ]
    }
}
```

Example

The following example creates a dashboard with two metric widgets, side by side.

```
{
    "DashboardName": "Two-metric-widget Dashboard",
    "DashboardBody": {
        "widgets": [
            {
                "type": "metric",
                "x": 0,
                "y": 0,
                "width": 12,
                "height": 6,
                "properties": {
                    "metrics": [
                        [
                            "AWS/EC2",
                            "CPUUtilization",
                            "InstanceId",
                            "i-012345"
                        ]
                    ],
                    "period": 300,
                    "stat": "Average",
                    "region": "us-east-1",
                    "title": "EC2 Instance CPU"
                }
            },
            {
                "type": "metric",
                "x": 12,
                "y": 0,
                "width": 12,
                "height": 6,
                "properties": {
                    "metrics": [
                        [
                            "AWS/S3",
                            "BucketSizeBytes",
                            "BucketName",
                            "MyBucketName"
                        ]
                    ],
                    "period": 86400,
                    "stat": "Maximum",
                    "region": "us-east-1",
                    "title": "MyBucketName bytes"
                }
            }
        ]
    }
}
```

Example

The following example creates a dashboard with one widget at the top that shows the `DiskReadBytes` metric for three EC2 instances on one graph, and a separate widget below that, with an alarm.

```
{
    "DashboardName": "Dashboard with a three-metric graph and an alarm",
    "DashboardBody": {
        "widgets": [
            {
                "type": "metric",
                "x": 0,
                "y": 0,
                "width": 12,
                "height": 6,
                "properties": {
                    "metrics": [
                        [
                            "AWS/EC2",
                            "DiskReadBytes",
                            "InstanceId",
                            "i-xyz"
                        ],
                        [
                            ".",
                            ".",
                            ".",
                            "i-abc"
                        ],
                        [
                            ".",
                            ".",
                            ".",
                            "i-123"
                        ]
                    ],
                    "period": 300,
                    "stat": "Average",
                    "region": "us-east-1",
                    "title": "EC2 Instance CPU"
                }
            },
            {
                "type": "metric",
                "x": 0,
                "y": 7,
                "width": 12,
                "height": 12,
                "properties": {
                    "annotations": {
                        "alarms": [
                            "arn:aws:cloudwatch:us-east-1:123456789012:alarm:myalarm"
                        ]
                    },
                    "period": 60,
                    "title": "MyAlarm"
                }
            }
        ]
    }
}
```

Example

The following example creates a dashboard with one metric widget and one metric math widget.

```
{
   "DashboardName":" One metric math widget and One metric widget",
   "DashboardBody":{
      "widgets":[
         {
            "type":"metric",
            "x":0,
            "y":0,
            "width":6,
            "height":6,
            "properties":{
               "metrics":[
                  [
                     "AWS/EC2",
                     "CPUUtilization",
                     "InstanceId",
                     "i-012345"
                  ]
               ],
               "region":"us-east-1",
               "stat":"Average",
               "period":300,
               "title":"EC2 Instance CPU"
            }
         },
         {
            "type":"metric",
            "x":6,
            "y":0,
            "width":6,
            "height":6,
            "properties":{
               "metrics":[
                  [
                     {
                        "expression":"SUM(METRICS())",
                        "label":"Expression1",
                        "id":"e1",
                        "visible":true
                     }
                  ],
                  [
                     "AWS/EC2",
                     "CPUUtilization",
                     "InstanceId",
                     "i-xyz",
                     {
                        "id":"m1",
                        "visible":true
                     }
                  ],
                  [
                     "...",
                     "i-abc",
                     {
                        "id":"m2",
                        "visible":true
                     }
                  ],
                  [
                     "...",
                     "i-123",
                     {
                        "id":"m3",
                        "visible":true
                     }
                  ],
                  [
                     "...",
                     "i-456",
                     {
                        "id":"m4",
                        "visible":true
                     }
                  ]
               ],
               "region":"us-east-1",
               "stat":"Average",
               "period":300,
               "title":"Sum of CPUUtilization of four Instances"
            }
         }
      ]
   }
}
```

See Also

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

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

PutInsightRule

Creates a Contributor Insights rule. Rules evaluate log events in a CloudWatch Logs log group, enabling you to find contributor data for the log events in that log group. For more information, see Using Contributor Insights to Analyze High-Cardinality Data.

If you create a rule, delete it, and then re-create it with the same name, historical data from the first time the rule was created might not be available.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

RuleDefinition

The definition of the rule, as a JSON object. For details on the valid syntax, see Contributor Insights Rule Syntax.

Type: String

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

Pattern: `[\x00-\x7F]+`

Required: Yes

RuleName

A unique name for the rule.

Type: String

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

Pattern: `[\x20-\x7E]+`

Required: Yes

RuleState

The state of the rule. Valid values are ENABLED and DISABLED.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 32.

Pattern: `[\x20-\x7E]+`

Required: No

Tags.member.N

A list of key-value pairs to associate with the Contributor Insights rule. You can associate as many as 50 tags with a rule.

Tags can help you organize and categorize your resources. You can also use them to scope user permissions, by granting a user permission to access or change only the resources that have certain tag values.

To be able to associate tags with a rule, you must have the `cloudwatch:TagResource` permission in addition to the `cloudwatch:PutInsightRule` permission.

If you are using this operation to update an existing Contributor Insights rule, any tags you specify in this parameter are ignored. To change the tags of an existing rule, use TagResource.

Type: Array of Tag (p. 137) objects

Required: No

Errors

For information about the errors that are common to all actions, see Common Errors (p. 172).

InvalidParameterValue

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

HTTP Status Code: 400

LimitExceeded

The operation exceeded one or more limits.

HTTP Status Code: 400

MissingParameter

An input parameter that is required is missing.

HTTP Status Code: 400

See Also

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

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

PutMetricAlarm

Creates or updates an alarm and associates it with the specified metric, metric math expression, or anomaly detection model.

Alarms based on anomaly detection models cannot have Auto Scaling actions.

When this operation creates an alarm, the alarm state is immediately set to `INSUFFICIENT_DATA`. The alarm is then evaluated and its state is set appropriately. Any actions associated with the new 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 IAM user, you must have Amazon EC2 permissions for some alarm operations:

- `iam:CreateServiceLinkedRole` for all alarms with EC2 actions
- `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
- No specific permissions are needed 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 are not performed. However, if you are later granted the required permissions, the alarm actions that you created earlier are performed.

If you are using an IAM role (for example, an 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 AWS STS, you cannot stop or terminate an EC2 instance using alarm actions.

The first time you create an alarm in the AWS Management Console, the CLI, or by using the PutMetricAlarm API, CloudWatch creates the necessary service-linked role for you. The service-linked role is called `AWSServiceRoleForCloudWatchEvents`. For more information, see AWS service-linked role.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

ActionsEnabled

Indicates whether actions should be executed during any changes to the alarm state. The default is `TRUE`.

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` | `arn:aws:automate:`*`region`*`:ec2:reboot` | `arn:aws:sns:`*`region`*`:`*`account-id`*`:`*`sns-topic-name`* | `arn:aws:autoscaling:`*`region`*`:`*`account-id`*`:scalingPolicy:`*`policy-id`*`:autoScalingGroupName/`*`group-friendly-name`*`:policyName/`*`policy-friendly-name`*

Valid Values (for use with IAM roles): `arn:aws:swf:`*`region`*`:`*`account-id`*`:action/actions/AWS_EC2.InstanceId.Stop/1.0` | `arn:aws:swf:`*`region`*`:`*`account-id`*`:action/actions/AWS_EC2.InstanceId.Terminate/1.0` | `arn:aws:swf:`*`region`*`:`*`account-id`*`: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 your 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.

The values `LessThanLowerOrGreaterThanUpperThreshold`, `LessThanLowerThreshold`, and `GreaterThanUpperThreshold` are used only for alarms based on anomaly detection models.

Type: String

Valid Values: `GreaterThanOrEqualToThreshold | GreaterThanThreshold | LessThanThreshold | LessThanOrEqualToThreshold | LessThanLowerOrGreaterThanUpperThreshold | LessThanLowerThreshold | GreaterThanUpperThreshold`

Required: Yes

DatapointsToAlarm

The number of data points that must be breaching to trigger the alarm. This is used only if you are setting an "M out of N" alarm. In that case, this value is the M. For more information, see Evaluating an Alarm in the *Amazon CloudWatch User Guide*.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

Dimensions.member.N

The dimensions for the metric specified in `MetricName`.

Type: Array of Dimension (p. 110) objects

Array Members: Maximum number of 10 items.

Required: No

EvaluateLowSampleCountPercentile

Used only for alarms based on percentiles. If you specify `ignore`, the alarm state does not change during periods with too few data points to be statistically significant. If you specify `evaluate` or omit this parameter, the alarm is always evaluated and possibly changes state no matter how many data points are available. For more information, see Percentile-Based CloudWatch Alarms and Low Data Samples.

Valid Values: `evaluate | ignore`

Type: String

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

Required: No

EvaluationPeriods

The number of periods over which data is compared to the specified threshold. If you are setting an alarm that requires that a number of consecutive data points be breaching to trigger the alarm, this value specifies that number. If you are setting an "M out of N" alarm, this value is the N.

An alarm's total current evaluation period can be no longer than one day, so this number multiplied by `Period` cannot be more than 86,400 seconds.

Type: Integer

Valid Range: Minimum value of 1.

Required: Yes

ExtendedStatistic

The percentile statistic for the metric specified in `MetricName`. Specify a value between p0.0 and p100. When you call `PutMetricAlarm` and specify a `MetricName`, you must specify either `Statistic` or `ExtendedStatistic,` but not both.

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`

| arn:aws:automate:*region*:ec2:reboot | arn:aws:sns:*region*:*account-id*:*sns-topic-name* | arn:aws:autoscaling:*region*:*account-id*:scalingPolicy:*policy-id*:autoScalingGroupName/*group-friendly-name*:policyName/*policy-friendly-name*

Valid Values (for use with IAM roles): >arn:aws:swf:*region*:*account-id*:action/actions/AWS_EC2.InstanceId.Stop/1.0 | arn:aws:swf:*region*:*account-id*:action/actions/AWS_EC2.InstanceId.Terminate/1.0 | arn:aws:swf:*region*:*account-id*: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. For each PutMetricAlarm operation, you must specify either MetricName or a Metrics array.

If you are creating an alarm based on a math expression, you cannot specify this parameter, or any of the Dimensions, Period, Namespace, Statistic, or ExtendedStatistic parameters. Instead, you specify all this information in the Metrics array.

Type: String

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

Required: No

Metrics.member.N

An array of MetricDataQuery structures that enable you to create an alarm based on the result of a metric math expression. For each PutMetricAlarm operation, you must specify either MetricName or a Metrics array.

Each item in the Metrics array either retrieves a metric or performs a math expression.

One item in the Metrics array is the expression that the alarm watches. You designate this expression by setting ReturnValue to true for this object in the array. For more information, see MetricDataQuery.

If you use the Metrics parameter, you cannot include the MetricName, Dimensions, Period, Namespace, Statistic, or ExtendedStatistic parameters of PutMetricAlarm in the same operation. Instead, you retrieve the metrics you are using in your math expression as part of the Metrics array.

Type: Array of MetricDataQuery (p. 125) objects

Required: No

Namespace

The namespace for the metric associated specified in MetricName.

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 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` | `arn:aws:automate:`*`region`*`:ec2:reboot` | `arn:aws:sns:`*`region`*`:`*`account-id`*`:`*`sns-topic-name`* | `arn:aws:autoscaling:`*`region`*`:`*`account-id`*`:scalingPolicy:`*`policy-id`*`:autoScalingGroupName/`*`group-friendly-name`*`:policyName/`*`policy-friendly-name`*

Valid Values (for use with IAM roles): `arn:aws:swf:`*`region`*`:`*`account-id`*`:action/actions/AWS_EC2.InstanceId.Stop/1.0` | `arn:aws:swf:`*`region`*`:`*`account-id`*`:action/actions/AWS_EC2.InstanceId.Terminate/1.0` | `arn:aws:swf:`*`region`*`:`*`account-id`*`: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 length, in seconds, used each time the metric specified in `MetricName` is evaluated. Valid values are 10, 30, and any multiple of 60.

`Period` is required for alarms based on static thresholds. If you are creating an alarm based on a metric math expression, you specify the period for each metric within the objects in the `Metrics` array.

Be sure to specify 10 or 30 only for metrics that are stored by a `PutMetricData` call with a `StorageResolution` of 1. If you specify a period of 10 or 30 for a metric that does not have sub-minute resolution, the alarm still attempts to gather data at the period rate that you specify. In this case, it does not receive data for the attempts that do not correspond to a one-minute data resolution, and the alarm might often lapse into INSUFFICENT_DATA status. Specifying 10 or 30 also sets this alarm as a high-resolution alarm, which has a higher charge than other alarms. For more information about pricing, see Amazon CloudWatch Pricing.

An alarm's total current evaluation period can be no longer than one day, so `Period` multiplied by `EvaluationPeriods` cannot be more than 86,400 seconds.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

Statistic

The statistic for the metric specified in `MetricName`, other than percentile. For percentile statistics, use `ExtendedStatistic`. When you call `PutMetricAlarm` and specify a `MetricName`, you must specify either `Statistic` or `ExtendedStatistic,` but not both.

Type: String

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

Required: No

Tags.member.N

A list of key-value pairs to associate with the alarm. You can associate as many as 50 tags with an alarm.

Tags can help you organize and categorize your resources. You can also use them to scope user permissions by granting a user permission to access or change only resources with certain tag values.

Type: Array of Tag (p. 137) objects

Required: No

Threshold

The value against which the specified statistic is compared.

This parameter is required for alarms based on static thresholds, but should not be used for alarms based on anomaly detection models.

Type: Double

Required: No

ThresholdMetricId

If this is an alarm based on an anomaly detection model, make this value match the ID of the `ANOMALY_DETECTION_BAND` function.

For an example of how to use this parameter, see the **Anomaly Detection Model Alarm** example on this page.

If your alarm uses this parameter, it cannot have Auto Scaling actions.

Type: String

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

Required: No

TreatMissingData

Sets how this alarm is to handle missing data points. If `TreatMissingData` is omitted, the default behavior of `missing` is used. For more information, see Configuring How CloudWatch Alarms Treats Missing Data.

Valid Values: `breaching | notBreaching | ignore | missing`

Type: String

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

Required: No

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 don't specify `Unit`, CloudWatch retrieves all unit types that have been published for the metric and attempts to evaluate the alarm. Usually, metrics are published with only one unit, so the alarm works as intended.

However, if the metric is published with multiple types of units and you don't specify a unit, the alarm's behavior is not defined and it behaves predictably.

We recommend omitting `Unit` so that you don't inadvertently specify an incorrect unit that is not published for this metric. Doing so causes the alarm to be 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. 172).

LimitExceeded

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

HTTP Status Code: 400

Examples

Static Threshold Alarm

The following example creates an alarm that notifies an SNS group when the CPUUtilization of a certain instance goes over 40% for three out of four periods.

Sample Request

```
{
    "Namespace": "AWS/EC2",
    "MetricName": "CPUUtilization",
    "Dimensions": [
        {
            "Name": "InstanceId",
            "Value": "i-1234567890abcdef0"
        }
    ],
    "AlarmActions": [
        "arn:aws:sns:us-west-1:123456789012:my_sns_topic"
    ],
    "ComparisonOperator": "GreaterThanThreshold",
    "DatapointsToAlarm": 3,
    "EvaluationPeriods": 4,
    "Period": 60,
    "Statistic": "Average",
    "Threshold": 40,
    "AlarmDescription": "CPU Utilization of i-1234567890abcdef0 with 40% as threshold",
    "AlarmName": "Instance i-1234567890abcdef0 CPU Utilization"
}
```

Metric Math Function Alarm

The following example retrieves three metrics that each track a different type of connection error to a custom service. These error counts are first summed in one expression, then divided by total connection attempts in another expression. The alarm goes to the ALARM state if the error rate is over 3% for two consecutive periods, and notifies two different SNS groups.

Sample Request

```
{
    "AlarmActions": [
        "arn:aws:sns:us-west-1:123456789012:my_sns_topic",
        "arn:aws:sns:us-west-1:123456789012:my_other_sns_topic"
    ],
    "ComparisonOperator": "GreaterThanThreshold",
    "EvaluationPeriods": 3,
    "Threshold": 40,
    "AlarmDescription": "MyService Aggregate Connection Error Rate (Alarm at 3%)",
    "AlarmName": "MyService Connection Error Rate",
    "Metrics": [
        {
            "MetricStat": {
                "Metric": {
                    "MetricName": "ConnectionsFailed",
                    "Namespace": "MyService"
                },
                "Period": 60,
                "Stat": "Sum"
            },
            "Id": "m1",
            "ReturnData": "False"
        },
        {
            "MetricStat": {
                "Metric": {
                    "MetricName": "ConnectionsDropped",
                    "Namespace": "MyService"
                },
                "Period": 60,
                "Stat": "Sum"
            },
            "Id": "m2",
            "ReturnData": "False"
        },
        {
            "MetricStat": {
                "Metric": {
                    "MetricName": "RequestsThrottled",
                    "Namespace": "MyService"
                },
                "Period": 60,
                "Stat": "Sum"
            },
            "Id": "m3",
            "ReturnData": "False"
        },
        {
            "MetricStat": {
                "Metric": {
                    "MetricName": "ConnectionAttempts",
                    "Namespace": "MyService"
                },
                "Period": 60,
                "Stat": "Sum"
            },
            "Id": "m4",
            "ReturnData": "False"
        },
        {
            "Id": "error_total",
            "Expression": "m1+m2+m3",
            "ReturnData": "False"
        },
        {
            "Id": "error_rate",
            "Expression": "(error_total/m4)*100",
            "ReturnData": "True",
            "Label": "Total Connection Error Rate"
        }
    ]
}
```

Anomaly Detection Model Alarm

The following example sets an alarm on an anomaly detection model. The Id of `m1` is assigned to the `CPUUtilization` metric of an instance. `t1` is the anomaly detection model function for that metric, and uses 3 standard deviations to set the width of the band. The setting of `ThresholdMetricId` is `t1` and the `ComparisonOperator` is `LessThanLowerOrGreaterThanUpperThreshold`, specifying that the alarm goes to alarm state when the metric value is outside the anomaly model band in either direction for two consecutive evaluation periods.

Sample Request

```
{
    "AlarmActions": [
        "arn:aws:sns:us-west-1:123456789012:my_sns_topic",
        "arn:aws:sns:us-west-1:123456789012:my_other_sns_topic"
    ],
    "AlarmName": "MyAlarmName",
    "AlarmDescription": "This alarm uses an anomaly detection model",
    "Metrics": [
        {
            "Id": "m1",
            "ReturnData": true,
            "MetricStat": {
                "Metric": {
                    "MetricName": "CPUUtilization",
                    "Namespace": "AWS/EC2",
                    "Dimensions": [
                        {
                            "Name": "instanceId",
                            "Value": "i-1234567890abcdef0"
                        }
                    ]
                },
                "Stat": "Average",
                "Period": 60
            }
        },
        {
            "Id": "t1",
            "Expression": "ANOMALY_DETECTION_BAND(m1, 3)"
        }
    ],
    "EvaluationPeriods": 2,
    "ThresholdMetricId": "t1",
    "ComparisonOperator": "LessThanLowerOrGreaterThanUpperThreshold"
}
```

See Also

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

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

PutMetricData

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

You can publish either individual data points in the `Value` field, or arrays of values and the number of times each value occurred during the period by using the `Values` and `Counts` fields in the `MetricDatum` structure. Using the `Values` and `Counts` method enables you to publish up to 150 values per metric with one `PutMetricData` request, and supports retrieving percentile statistics on this data.

Each `PutMetricData` request is limited to 40 KB in size for HTTP POST requests. You can send a payload compressed by gzip. Each request is also limited to no more than 20 different metrics.

Although the `Value` parameter accepts numbers of type `Double`, CloudWatch rejects values that are either too small or too large. Values must be in the range of -2^360 to 2^360. In addition, special values (for example, NaN, +Infinity, -Infinity) are not supported.

You can use up to 10 dimensions per metric to further clarify what data the metric collects. Each dimension consists of a Name and Value pair. For more information about specifying dimensions, see Publishing Metrics in the *Amazon CloudWatch User Guide*.

Data points with time stamps from 24 hours ago or longer can take at least 48 hours to become available for GetMetricData or GetMetricStatistics from the time they are submitted. Data points with time stamps between 3 and 24 hours ago can take as much as 2 hours to become available for for GetMetricData or GetMetricStatistics.

CloudWatch needs raw data points to calculate percentile statistics. If you publish data using a statistic set instead, you can only retrieve percentile statistics for this data if one of the following conditions is true:

- The `SampleCount` value of the statistic set is 1 and `Min`, `Max`, and `Sum` are all equal.
- The `Min` and `Max` are equal, and `Sum` is equal to `Min` multiplied by `SampleCount`.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

MetricData.member.N

The data for the metric. The array can include no more than 20 metrics per call.

Type: Array of MetricDatum (p. 129) objects

Required: Yes

Namespace

The namespace for the metric data.

To avoid conflicts with AWS service namespaces, you should not specify a namespace that begins with `AWS/`

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. 172).

InternalServiceError

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

HTTP Status Code: 500

InvalidParameterCombination

Parameters were used together that cannot be 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
```

Example

The following example puts data for a high-resolution metric:

Sample Request

```
https://monitoring.&api-domain;/doc/2010-08-01/
?Action=PutMetricData
&Version=2010-08-01
&Namespace=HighResolutionMetric
&MetricData.member.1.MetricName=HighResdata
&MetricData.member.1.Unit=Bytes
&MetricData.member.1.Value=542868
&MetricData.member.1.StorageResolution=1
&AUTHPARAMS
```

Example

The following example puts multiple values for each of two metrics, using `Values` and `Counts` arrays:

Sample Request

```
https://monitoring.&api-domain;/doc/2010-08-01/
```

```
?Action=PutMetricData
&Version=2010-08-01
&Namespace=TestNamespace
&MetricData.member.1.MetricName=Reads
&MetricData.member.1.Unit=Count
&MetricData.member.1.Values.member.1=5
&MetricData.member.1.Values.member.2=8
&MetricData.member.1.Values.member.3=10
&MetricData.member.1.Values.member.4=9
&MetricData.member.1.Counts.member.1=1
&MetricData.member.1.Counts.member.2=5
&MetricData.member.1.Counts.member.3=6
&MetricData.member.1.Counts.member.4=5
&MetricData.member.1.Dimensions.member.1.Name=InstanceID
&MetricData.member.1.Dimensions.member.1.Value=i-aaba32d4
&MetricData.member.2.MetricName=Writes
&MetricData.member.2.Unit=Count
&MetricData.member.2.Values.member.1=2
&MetricData.member.2.Values.member.2=3
&MetricData.member.2.Values.member.3=0
&MetricData.member.2.Counts.member.1=2
&MetricData.member.2.Counts.member.2=2
&MetricData.member.2.Counts.member.3=1
&MetricData.member.2.Dimensions.member.1.Name=InstanceID
&MetricData.member.2.Dimensions.member.1.Value=i-aaba32d4
&AUTHPARAMS
```

See Also

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

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

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 SNS message.

Metric alarms returns to their actual state quickly, often within seconds. Because the metric alarm state change happens quickly, it is typically only visible in the alarm's **History** tab in the Amazon CloudWatch console or through DescribeAlarmHistory.

If you use `SetAlarmState` on a composite alarm, the composite alarm is not guaranteed to return to its actual state. It returns to its actual state only once any of its children alarms change state. It is also reevaluated if you update its configuration.

If an alarm triggers EC2 Auto Scaling policies or application Auto Scaling policies, you must include information in the `StateReasonData` parameter to enable the policy to take the correct action.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

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.

For SNS or EC2 alarm actions, this is just informational. But for EC2 Auto Scaling or application Auto Scaling alarm actions, the Auto Scaling policy uses the information in this field to take the correct action.

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. 172).

InvalidFormat

Data was not syntactically valid JSON.

HTTP Status Code: 400

ResourceNotFound

The named resource does not exist.

HTTP Status Code: 404

Example

The following example sets the alarm state to ALARM, and provides information inside of `StateReasonData` so that Auto Scaling actions can be performed correctly according to your Auto Scaling policies.

```
{
    "AlarmName": "ExampleAlarmName",
    "StateValue": "ALARM",
    "StateReason": "Testing Alarm State",
    "StateReasonData": {
        "Version": "1.0",
        "QueryDate": "2018-10-31T14:32:52.031+0000",
        "StartDate": "2018-10-31T14:31:00.000+0000",
        "Statistic": "Average",
        "Period": 60,
        "RecentDatapoints": [
            100
        ],
        "Threshold": 50
    }
}
```

See Also

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

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

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

TagResource

Assigns one or more tags (key-value pairs) to the specified CloudWatch resource. Currently, the only CloudWatch resources that can be tagged are alarms and Contributor Insights rules.

Tags can help you organize and categorize your resources. You can also use them to scope user permissions by granting a user permission to access or change only resources with certain tag values.

Tags don't have any semantic meaning to AWS and are interpreted strictly as strings of characters.

You can use the `TagResource` action with an alarm that already has tags. If you specify a new tag key for the alarm, this tag is appended to the list of tags associated with the alarm. If you specify a tag key that is already associated with the alarm, the new tag value that you specify replaces the previous value for that tag.

You can associate as many as 50 tags with a CloudWatch resource.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

ResourceARN

The ARN of the CloudWatch resource that you're adding tags to.

The ARN format of an alarm is `arn:aws:cloudwatch:`*`Region`*`:`*`account-id`*`:alarm:`*`alarm-name`*

The ARN format of a Contributor Insights rule is `arn:aws:cloudwatch:`*`Region`*`:`*`account-id`*`:insight-rule:`*`insight-rule-name`*

For more information about ARN format, see Resource Types Defined by Amazon CloudWatch in the *Amazon Web Services General Reference*.

Type: String

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

Required: Yes

Tags.member.N

The list of key-value pairs to associate with the alarm.

Type: Array of Tag (p. 137) objects

Required: Yes

Errors

For information about the errors that are common to all actions, see Common Errors (p. 172).

ConcurrentModification

More than one process tried to modify a resource at the same time.

HTTP Status Code: 429

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

ResourceNotFound

The named resource does not exist.

HTTP Status Code: 404

See Also

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

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

UntagResource

Removes one or more tags from the specified resource.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 170).

ResourceARN

The ARN of the CloudWatch resource that you're removing tags from.

The ARN format of an alarm is `arn:aws:cloudwatch:`*`Region`*`:`*`account-id`*`:alarm:`*`alarm-name`*

The ARN format of a Contributor Insights rule is `arn:aws:cloudwatch:`*`Region`*`:`*`account-id`*`:insight-rule:`*`insight-rule-name`*

For more information about ARN format, see Resource Types Defined by Amazon CloudWatch in the *Amazon Web Services General Reference*.

Type: String

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

Required: Yes

TagKeys.member.N

The list of tag keys to remove from the resource.

Type: Array of strings

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

Required: Yes

Errors

For information about the errors that are common to all actions, see Common Errors (p. 172).

ConcurrentModification

More than one process tried to modify a resource at the same time.

HTTP Status Code: 429

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

ResourceNotFound

The named resource does not exist.

HTTP Status Code: 404

See Also

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

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

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 (p. 98)
- AnomalyDetector (p. 100)
- AnomalyDetectorConfiguration (p. 102)
- CompositeAlarm (p. 103)
- DashboardEntry (p. 106)
- DashboardValidationMessage (p. 107)
- Datapoint (p. 108)
- Dimension (p. 110)
- DimensionFilter (p. 111)
- InsightRule (p. 112)
- InsightRuleContributor (p. 114)
- InsightRuleContributorDatapoint (p. 115)
- InsightRuleMetricDatapoint (p. 116)
- MessageData (p. 118)
- Metric (p. 119)
- MetricAlarm (p. 120)
- MetricDataQuery (p. 125)
- MetricDataResult (p. 127)
- MetricDatum (p. 129)
- MetricStat (p. 132)
- PartialFailure (p. 134)
- Range (p. 135)
- StatisticSet (p. 136)
- Tag (p. 137)

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

AlarmType

The type of alarm, either metric alarm or composite alarm.

Type: String

Valid Values: `CompositeAlarm | MetricAlarm`

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

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3

AnomalyDetector

An anomaly detection model associated with a particular CloudWatch metric and statistic. You can use the model to display a band of expected normal values when the metric is graphed.

Contents

Configuration

The configuration specifies details about how the anomaly detection model is to be trained, including time ranges to exclude from use for training the model, and the time zone to use for the metric.

Type: AnomalyDetectorConfiguration (p. 102) object

Required: No

Dimensions.member.N

The metric dimensions associated with the anomaly detection model.

Type: Array of Dimension (p. 110) objects

Array Members: Maximum number of 10 items.

Required: No

MetricName

The name of the metric associated with the anomaly detection model.

Type: String

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

Required: No

Namespace

The namespace of the metric associated with the anomaly detection model.

Type: String

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

Pattern: `[^:].*`

Required: No

Stat

The statistic associated with the anomaly detection model.

Type: String

Pattern: `(SampleCount|Average|Sum|Minimum|Maximum|p(\d{1,2}|100)(\.\d{0,2})?|[ou]\d+(\.\d*)?)(_E|_L|_H)?`

Required: No

StateValue

The current status of the anomaly detector's training. The possible values are `TRAINED | PENDING_TRAINING | TRAINED_INSUFFICIENT_DATA`

Type: String

Valid Values: `PENDING_TRAINING | TRAINED_INSUFFICIENT_DATA | TRAINED`

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3

AnomalyDetectorConfiguration

The configuration specifies details about how the anomaly detection model is to be trained, including time ranges to exclude from use for training the model and the time zone to use for the metric.

Contents

ExcludedTimeRanges.member.N

An array of time ranges to exclude from use when the anomaly detection model is trained. Use this to make sure that events that could cause unusual values for the metric, such as deployments, aren't used when CloudWatch creates the model.

Type: Array of Range (p. 135) objects

Required: No

MetricTimezone

The time zone to use for the metric. This is useful to enable the model to automatically account for daylight savings time changes if the metric is sensitive to such time changes.

To specify a time zone, use the name of the time zone as specified in the standard tz database. For more information, see tz database.

Type: String

Length Constraints: Maximum length of 50.

Pattern: `.*`

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3

CompositeAlarm

The details about a composite 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

AlarmRule

The rule that this alarm uses to evaluate its alarm state.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 10240.

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

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

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

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3

DashboardEntry

Represents a specific dashboard.

Contents

DashboardArn

The Amazon Resource Name (ARN) of the dashboard.

Type: String

Required: No

DashboardName

The name of the dashboard.

Type: String

Required: No

LastModified

The time stamp of when the dashboard was last modified, either by an API call or through the console. This number is expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC.

Type: Timestamp

Required: No

Size

The size of the dashboard, in bytes.

Type: Long

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3

DashboardValidationMessage

An error or warning for the operation.

Contents

DataPath

The data path related to the message.

Type: String

Required: No

Message

A message describing the error or warning.

Type: String

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3

Datapoint

Encapsulates the statistical data that 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), ExtendedStatistics.entry.N.value (value)

The percentile statistic for the data point.

Type: String to double map

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

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

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3

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

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3

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

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3

InsightRule

This structure contains the definition for a Contributor Insights rule.

Contents

Definition

The definition of the rule, as a JSON object. The definition contains the keywords used to define contributors, the value to aggregate on if this rule returns a sum instead of a count, and the filters. For details on the valid syntax, see Contributor Insights Rule Syntax.

Type: String

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

Pattern: `[\x00-\x7F]+`

Required: Yes

Name

The name of the rule.

Type: String

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

Pattern: `[\x20-\x7E]+`

Required: Yes

Schema

For rules that you create, this is always `{"Name": "CloudWatchLogRule", "Version": 1}`. For built-in rules, this is `{"Name": "ServiceLogRule", "Version": 1}`

Type: String

Required: Yes

State

Indicates whether the rule is enabled or disabled.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 32.

Pattern: `[\x20-\x7E]+`

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Go

- AWS SDK for Java
- AWS SDK for Ruby V3

InsightRuleContributor

One of the unique contributors found by a Contributor Insights rule. If the rule contains multiple keys, then a unique contributor is a unique combination of values from all the keys in the rule.

If the rule contains a single key, then each unique contributor is each unique value for this key.

For more information, see GetInsightRuleReport.

Contents

ApproximateAggregateValue

An approximation of the aggregate value that comes from this contributor.

Type: Double

Required: Yes

Datapoints.member.N

An array of the data points where this contributor is present. Only the data points when this contributor appeared are included in the array.

Type: Array of InsightRuleContributorDatapoint (p. 115) objects

Required: Yes

Keys.member.N

One of the log entry field keywords that is used to define contributors for this rule.

Type: Array of strings

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3

InsightRuleContributorDatapoint

One data point related to one contributor.

For more information, see GetInsightRuleReport and InsightRuleContributor.

Contents

ApproximateValue

The approximate value that this contributor added during this timestamp.

Type: Double

Required: Yes

Timestamp

The timestamp of the data point.

Type: Timestamp

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3

InsightRuleMetricDatapoint

One data point from the metric time series returned in a Contributor Insights rule report.

For more information, see GetInsightRuleReport.

Contents

Average

The average value from all contributors during the time period represented by that data point.

This statistic is returned only if you included it in the `Metrics` array in your request.

Type: Double

Required: No

MaxContributorValue

The maximum value provided by one contributor during this timestamp. Each timestamp is evaluated separately, so the identity of the max contributor could be different for each timestamp.

This statistic is returned only if you included it in the `Metrics` array in your request.

Type: Double

Required: No

Maximum

The maximum value from a single occurence from a single contributor during the time period represented by that data point.

This statistic is returned only if you included it in the `Metrics` array in your request.

Type: Double

Required: No

Minimum

The minimum value from a single contributor during the time period represented by that data point.

This statistic is returned only if you included it in the `Metrics` array in your request.

Type: Double

Required: No

SampleCount

The number of occurrences that matched the rule during this data point.

This statistic is returned only if you included it in the `Metrics` array in your request.

Type: Double

Required: No

Sum

The sum of the values from all contributors during the time period represented by that data point.

This statistic is returned only if you included it in the `Metrics` array in your request.

Type: Double

Required: No

Timestamp

The timestamp of the data point.

Type: Timestamp

Required: Yes

UniqueContributors

The number of unique contributors who published data during this timestamp.

This statistic is returned only if you included it in the `Metrics` array in your request.

Type: Double

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3

MessageData

A message returned by the `GetMetricData`API, including a code and a description.

Contents

Code

The error code or status code associated with the message.

Type: String

Required: No

Value

The message text.

Type: String

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3

Metric

Represents a specific metric.

Contents

Dimensions.member.N

The dimensions for the metric.

Type: Array of Dimension (p. 110) objects

Array Members: Maximum number of 10 items.

Required: No

MetricName

The name of the metric. This is a required field.

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

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3

MetricAlarm

The details about a metric 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 | LessThanLowerOrGreaterThanUpperThreshold | LessThanLowerThreshold | GreaterThanUpperThreshold`

Required: No

DatapointsToAlarm

The number of data points that must be breaching to trigger the alarm.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

Dimensions.member.N

The dimensions for the metric associated with the alarm.

Type: Array of Dimension (p. 110) objects

Array Members: Maximum number of 10 items.

Required: No

EvaluateLowSampleCountPercentile

Used only for alarms based on percentiles. If `ignore`, the alarm state does not change during periods with too few data points to be statistically significant. If `evaluate` or this parameter is not used, the alarm is always evaluated and possibly changes state no matter how many data points are available.

Type: String

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

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, if this is an alarm based on a single metric.

Type: String

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

Required: No

Metrics.member.N

An array of MetricDataQuery structures, used in an alarm based on a metric math expression. Each structure either retrieves a metric or performs a math expression. One item in the Metrics array is the math expression that the alarm watches. This expression by designated by having `ReturnValue` set to true.

Type: Array of MetricDataQuery (p. 125) objects

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 1.

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

ThresholdMetricId

In an alarm based on an anomaly detection model, this is the ID of the `ANOMALY_DETECTION_BAND` function used as the threshold for the alarm.

Type: String

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

Required: No

TreatMissingData

Sets how this alarm is to handle missing data points. If this parameter is omitted, the default behavior of `missing` is used.

Type: String

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

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

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3

MetricDataQuery

This structure is used in both `GetMetricData` and `PutMetricAlarm`. The supported use of this structure is different for those two operations.

When used in `GetMetricData`, it indicates the metric data to return, and whether this call is just retrieving a batch set of data for one metric, or is performing a math expression on metric data. A single `GetMetricData` call can include up to 500 `MetricDataQuery` structures.

When used in `PutMetricAlarm`, it enables you to create an alarm based on a metric math expression. Each `MetricDataQuery` in the array specifies either a metric to retrieve, or a math expression to be performed on retrieved metrics. A single `PutMetricAlarm` call can include up to 20 `MetricDataQuery` structures in the array. The 20 structures can include as many as 10 structures that contain a `MetricStat` parameter to retrieve a metric, and as many as 10 structures that contain the `Expression` parameter to perform a math expression. Of those `Expression` structures, one must have `True` as the value for `ReturnData`. The result of this expression is the value the alarm watches.

Any expression used in a `PutMetricAlarm` operation must return a single time series. For more information, see Metric Math Syntax and Functions in the *Amazon CloudWatch User Guide*.

Some of the parameters of this structure also have different uses whether you are using this structure in a `GetMetricData` operation or a `PutMetricAlarm` operation. These differences are explained in the following parameter list.

Contents

Expression

The math expression to be performed on the returned data, if this object is performing a math expression. This expression can use the `Id` of the other metrics to refer to those metrics, and can also use the `Id` of other expressions to use the result of those expressions. For more information about metric math expressions, see Metric Math Syntax and Functions in the *Amazon CloudWatch User Guide*.

Within each MetricDataQuery object, you must specify either `Expression` or `MetricStat` but not both.

Type: String

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

Required: No

Id

A short name used to tie this object to the results in the response. This name must be unique within a single call to `GetMetricData`. If you are performing math expressions on this set of data, this name represents that data and can serve as a variable in the mathematical expression. The valid characters are letters, numbers, and underscore. The first character must be a lowercase letter.

Type: String

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

Required: Yes

Label

A human-readable label for this metric or expression. This is especially useful if this is an expression, so that you know what the value represents. If the metric or expression is shown in a CloudWatch dashboard widget, the label is shown. If Label is omitted, CloudWatch generates a default.

Type: String

Required: No

MetricStat

The metric to be returned, along with statistics, period, and units. Use this parameter only if this object is retrieving a metric and not performing a math expression on returned data.

Within one MetricDataQuery object, you must specify either `Expression` or `MetricStat` but not both.

Type: MetricStat (p. 132) object

Required: No

Period

The granularity, in seconds, of the returned data points. For metrics with regular resolution, a period can be as short as one minute (60 seconds) and must be a multiple of 60. For high-resolution metrics that are collected at intervals of less than one minute, the period can be 1, 5, 10, 30, 60, or any multiple of 60. High-resolution metrics are those metrics stored by a `PutMetricData` operation that includes a `StorageResolution of 1 second`.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

ReturnData

When used in `GetMetricData`, this option indicates whether to return the timestamps and raw data values of this metric. If you are performing this call just to do math expressions and do not also need the raw data returned, you can specify `False`. If you omit this, the default of `True` is used.

When used in `PutMetricAlarm`, specify `True` for the one expression result to use as the alarm. For all other metrics and expressions in the same `PutMetricAlarm` operation, specify `ReturnData` as False.

Type: Boolean

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3

MetricDataResult

A GetMetricData call returns an array of MetricDataResult structures. Each of these structures includes the data points for that metric, along with the timestamps of those data points and other identifying information.

Contents

Id

The short name you specified to represent this metric.

Type: String

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

Required: No

Label

The human-readable label associated with the data.

Type: String

Required: No

Messages.member.N

A list of messages with additional information about the data returned.

Type: Array of MessageData (p. 118) objects

Required: No

StatusCode

The status of the returned data. Complete indicates that all data points in the requested time range were returned. PartialData means that an incomplete set of data points were returned. You can use the NextToken value that was returned and repeat your request to get more data points. NextToken is not returned if you are performing a math expression. InternalError indicates that an error occurred. Retry your request using NextToken, if present.

Type: String

Valid Values: Complete | InternalError | PartialData

Required: No

Timestamps.member.N

The timestamps for the data points, formatted in Unix timestamp format. The number of timestamps always matches the number of values and the value for Timestamps[x] is Values[x].

Type: Array of timestamps

Required: No

Values.member.N

The data points for the metric corresponding to Timestamps. The number of values always matches the number of timestamps and the timestamp for Values[x] is Timestamps[x].

Type: Array of doubles

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3

MetricDatum

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

Contents

Counts.member.N

Array of numbers that is used along with the `Values` array. Each number in the `Count` array is the number of times the corresponding value in the `Values` array occurred during the period.

If you omit the `Counts` array, the default of 1 is used as the value for each count. If you include a `Counts` array, it must include the same amount of values as the `Values` array.

Type: Array of doubles

Required: No

Dimensions.member.N

The dimensions associated with the metric.

Type: Array of Dimension (p. 110) 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. 136) object

Required: No

StorageResolution

Valid values are 1 and 60. Setting this to 1 specifies this metric as a high-resolution metric, so that CloudWatch stores the metric with sub-minute resolution down to one second. Setting this to 60 specifies this metric as a regular-resolution metric, which CloudWatch stores at 1-minute resolution. Currently, high resolution is available only for custom metrics. For more information about high-resolution metrics, see High-Resolution Metrics in the *Amazon CloudWatch User Guide*.

This field is optional, if you do not specify it the default of 60 is used.

Type: Integer

Valid Range: Minimum value of 1.

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

When you are using a `Put` operation, this defines what unit you want to use when storing the metric.

In a `Get` operation, this displays the unit that is used 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

Value

The value for the metric.

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

Type: Double

Required: No

Values.member.N

Array of numbers representing the values for the metric during the period. Each unique value is listed just once in this array, and the corresponding number in the `Counts` array specifies the number of times that value occurred during the period. You can include up to 150 unique values in each `PutMetricData` action that specifies a `Values` array.

Although the `Values` array accepts numbers of type `Double`, CloudWatch rejects values that are either too small or too large. Values must be in the range of -2^360 to 2^360. In addition, special values (for example, NaN, +Infinity, -Infinity) are not supported.

Type: Array of doubles

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3

MetricStat

This structure defines the metric to be returned, along with the statistics, period, and units.

Contents

Metric

The metric to return, including the metric name, namespace, and dimensions.

Type: Metric (p. 119) object

Required: Yes

Period

The granularity, in seconds, of the returned data points. For metrics with regular resolution, a period can be as short as one minute (60 seconds) and must be a multiple of 60. For high-resolution metrics that are collected at intervals of less than one minute, the period can be 1, 5, 10, 30, 60, or any multiple of 60. High-resolution metrics are those metrics stored by a `PutMetricData` call that includes a `StorageResolution` of 1 second.

If the `StartTime` parameter specifies a time stamp that is greater than 3 hours ago, you must specify the period as follows or no data points in that time range is returned:

- Start time between 3 hours and 15 days ago - Use a multiple of 60 seconds (1 minute).
- 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 1.

Required: Yes

Stat

The statistic to return. It can include any CloudWatch statistic or extended statistic.

Type: String

Required: Yes

Unit

When you are using a `Put` operation, this defines what unit you want to use when storing the metric.

In a `Get` operation, if you omit `Unit` then all data that was collected with any unit is returned, along with the corresponding units that were specified when the data was reported to CloudWatch. If you specify a unit, the operation returns only data that was collected with that unit specified. If you specify a unit that does not match the data collected, the results of the operation are null. CloudWatch does not perform unit conversions.

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

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3

PartialFailure

This array is empty if the API operation was successful for all the rules specified in the request. If the operation could not process one of the rules, the following data is returned for each of those rules.

Contents

ExceptionType

The type of error.

Type: String

Required: No

FailureCode

The code of the error.

Type: String

Required: No

FailureDescription

A description of the error.

Type: String

Required: No

FailureResource

The specified rule that could not be deleted.

Type: String

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3

Range

Specifies one range of days or times to exclude from use for training an anomaly detection model.

Contents

EndTime

The end time of the range to exclude. The format is `yyyy-MM-dd'T'HH:mm:ss`. For example, `2019-07-01T23:59:59`.

Type: Timestamp

Required: Yes

StartTime

The start time of the range to exclude. The format is `yyyy-MM-dd'T'HH:mm:ss`. For example, `2019-07-01T23:59:59`.

Type: Timestamp

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3

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

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3

Tag

A key-value pair associated with a CloudWatch resource.

Contents

Key

A string that you can use to assign a value. The combination of tag keys and values can help you organize and categorize your resources.

Type: String

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

Required: Yes

Value

The value for the specified tag key.

Type: String

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

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3

Dashboard Body Structure and Syntax

Contents

- Overall Structure (p. 138)
- Widgets Array Structure (p. 140)
- Properties of a Text Widget Object (p. 141)
- Properties of a Log Widget Object (p. 142)
- Properties of a Metric Widget Object (p. 143)
- Metric Widget: Format for Each Metric in the Array (p. 146)

Overall Structure

A `DashboardBody` is a string in JSON format. It can include an array of between 0 and 100 widget objects, as well as a few other parameters.

The following is an example of this structure with one metric widget and one text widget, a time range starting six hours before the current time, and each graph's period setting always being obeyed.

```
{
   "start": "-PT6H",
   "periodOverride": "inherit",
   "widgets": [
      {
         "type":"metric",
         "x":0,
         "y":0,
         "width":12,
         "height":6,
         "properties":{
            "metrics":[
               [
                  "AWS/EC2",
                  "CPUUtilization",
                  "InstanceId",
                  "i-012345"
               ]
            ],
            "period":300,
            "stat":"Average",
            "region":"us-east-1",
            "title":"EC2 Instance CPU",
            "liveData": false,
            "legend": {
                "position": "right"
              }
         }
      },
      {
         "type":"text",
         "x":0,
         "y":7,
         "width":3,
         "height":3,
         "properties":{
            "markdown":"Hello world"
         }
      }
   ]
}
```

The next example has two widgets. The first includes two metrics and a math expression that sums their total. The second widget is a search expression that displays the `CPUUtilization` for all EC2 instances in the Region.

```
{
   "start": "-PT9H",
   "periodOverride": "inherit",
   "widgets": [
      {
         "type":"metric",
         "x":0,
         "y":0,
         "width":12,
         "height":6,
         "properties":{
            "metrics":[
               [ "AWS/EC2", "DiskReadBytes", "InstanceId", "i-123",{ "id": "m1" } ],
               [ ".", ".", ".", "i-abc", { "id": "m2" } ],
               [ { "expression": "SUM(METRICS())", "label": "Sum of DiskReadbytes", "id":
 "e3" } ]
            ],
            "view": "timeSeries",
            "stacked": false,
            "period":300,
            "stat":"Average",
            "region":"us-east-1",
            "title":"EC2 Instance CPU"
         }
      },
      {
         "type":"metric",
         "x":0,
         "y":0,
         "width":18,
         "height":9,
         "properties":{
            "metrics":[
               [ { "expression": "SEARCH('{AWS/EC2,InstanceId} MetricName=\"CPUUtilization
\"', 'Average', 300)", "id": "e1" } ]
            ],
            "view": "timeSeries",
            "stacked": false,
            "region":"us-east-1",
            "title":"EC2 Instance CPU"
         }
      }
   ]
}
```

The rest of this section includes examples illustrating each part of the `DashboardBody` syntax. For more examples showing the entire command syntax, see PutDashboard in the Amazon CloudWatch API Reference.

The top level of the JSON object can include the following properties.

widgets

The list of widgets in the dashboard. For more information, see Widgets Array Structure (p. 140).

Required: No

end

The end of the time range to use for each widget on the dashboard when the dashboard loads. If you specify a value for `end`, you must also specify a value for `start`. For each of these values, specify an absolute time in the ISO 8601 format. For example, `2018-12-17T06:00:00.000Z`.

Type: String

Required: No

start

The start of the time range to use for each widget on the dashboard.

You can specify `start` without specifying `end` to specify a relative time range that ends with the current time. In this case, the value of `start` must begin with `-PT` if you specify a time range in minutes or hours, and must begin with `-P` if you specify a time range in days, weeks, or months. You can then use M, H, D, W and M as abbreviations for minutes, hours, days, weeks and months. For example, `-PT5M` shows the last 5 minutes, `-PT8H` shows the last 8 hours, and `-P3M` shows the last three months.

You can also use `start` along with an `end` field, to specify an absolute time range. When specifying an absolute time range, use the ISO 8601 format. For example, `2018-12-17T06:00:00.000Z`.

If you omit `start`, the dashboard shows the default time range when it loads.

Type: String

Required: No

periodOverride

Use this field to specify the period for the graphs when the dashboard loads. Specifying `auto` causes the period of all graphs on the dashboard to automatically adapt to the time range of the dashboard. Specifying `inherit` ensures that the period set for each graph is always obeyed.

Valid Values: auto | inherit

Type: String

Required: No

Widgets Array Structure

Each widget of any type can have the following properties.

type

The type of widget.

Valid Values: `metric` | `text` | `log`

Type: String

Required: Yes

x

The horizontal position of the widget on the 24-column dashboard grid. The default is the next available position.

Valid Values: 0–23

Type: Integer

Required: Yes, if `y` is specified. Otherwise not required.

y

The vertical position of the widget on the 24-column dashboard grid. The default is the next available position.

Valid Values: Any integer, 0 or higher.

Type: Integer

Required: Yes, if `x` is specified. Otherwise not required.

width

The width of the widget in grid units (in a 24-column grid). The default is 6.

Valid Values: 1–24

Type: Integer

Required: No

height

The height of the widget in grid units. The default is 6.

Valid Values: 1–1000

Type: Integer

Required: No

properties

The detailed properties of the widget, which differ depending on the widget type. For more information about the format of `properties`, see Properties of a Metric Widget Object (p. 143) or Properties of a Text Widget Object (p. 141).

Type: Object

Required: Yes

Properties of a Text Widget Object

A widget of type `text` must have exactly one `properties` field: `markdown`.

For more information about the style of markdown supported in CloudWatch text widgets, see Using Markdown in the Console.

markdown

The text to be displayed by the widget. Use this parameter only for text widgets.

Type: String

Required: Yes (when the widget `type` is `text`).

```
{
   "widgets":[
      {
         "type":"text",
         "x":0,
         "y":7,
         "width":3,
         "height":3,
         "properties":{
            "markdown":"Hello world"
         }
      }
   ]
}
```

Properties of a Log Widget Object

A widget of type `log` represents the results of a CloudWatch Logs Insights query. For more information, see Analyzing Log Data with CloudWatch Logs Insights.

A `log` widget can include the following fields in its `properties` field.

region

The Region of the logs query.

Type: String

Required: Yes

title

The title text to be displayed by the widget.

Type: String

Required: No

query

Contains the CloudWatch Logs Insights query function.

Type: String

Required: Yes (when the widget `type` is `log`).

The `query` string starts with the names of the log groups that are to be queried. You must pre-pend each log group name with `SOURCE`. Separate multiple log groups with a pipe character (|).

Add another pipe character after the list of log groups, and then specify the query syntax. Separate each line in the query syntax with `\n|`

For example, the following line represents a query of two log groups, `service_log1` and `service_log2`. The query displays canaries that have faults.

```
"query": "SOURCE 'service_log1' | SOURCE 'service_log2' |filter Fault > 0\n| fields
 Fault.Message\n| stats count(*) by Canary.Name, Fault.Message"
```

view

Specifies how the query results are displayed. Specify `table` to view the results as a table. Specify `timeSeries` to display this metric as a line graph, or `bar` to display it as a bar graph.

If you omit this parameter, the results are displayed as a table.

Type: String

Required: No (when the widget `type` is `log`).

```
{
    "widgets": [
        {
            "type": "log",
            "x": 12,
            "y": 24,
            "width": 12,
            "height": 6,
            "properties": {
                "region": "us-east-1",
                "title": "Errors (Application Log)",
                "query": "SOURCE 'application1.log' | SOURCE 'application2.log' | filter
 @message like \"[ERROR]\"\n| parse \"Error for [*] [*] due to: *\" canaryName1, canaryId1,
 cause1\n| parse \"Executor canary [*] *\" canaryName2, cause2\n| fields coalesce(cause1,
 cause2) as cause\n| fields coalesce(canaryName1, canaryName2) as canaryName\n| fields
 ispresent(cause) as isP\n| filter isP\n| stats count() as errCount by canaryName,
 substr(cause, 0, 130)\n| sort errCount DESC",
                "view": "table"
            }
        }
    ]
}
```

Properties of a Metric Widget Object

A widget of type `metric` can have the following fields within `properties`:

accountId

Specifies the AWS account ID where all metrics in this widget will come from. This is useful for cross-account dashboards that include widgets from multiple accounts. For more information, see Cross-Account Cross-Region CloudWatch Console.

If you omit this, the current account is used as the default. Use this parameter only for metric widgets.

You can also use an `accountId` field within each metric in the array of `metrics` to create a single widget which includes metrics from multiple accounts.

Type: string

Required: No

annotations

To include an alarm or annotation in the widget, specify an `annotations` array. For more information about the format, see Dashboard Widget Object: Annotation Properties (p. 150). Use this parameter only for metric widgets.

Type: Object

Required: An alarm annotation is required only when the widget `type` is `metric` and `metrics` is not specified. A horizontal or vertical annotation is not required.

liveData

Specify `true` to display *live data* in the widget. Live data is data published within the last minute that has not been fully aggregated. For more information, see Use Live Data.

Type: Boolean

Required: No

legend

Specify `legend` to determine where the legend for the lines on the graph is displayed. The `legend` field contains another field called `position`. Possible values for `position` are `right`, `bottom`, and `hidden`.

For example, the following causes the legend to appear on the right in the graph.

```
"legend": {
     "position": "right"
}
```

Type: Object

Required: No

metrics

Specify a `metrics` array to include one or more metrics (without alarms), math expressions, or search expressions. One `metrics` array can include 0–100 metrics and expressions. Use this parameter only for metric widgets. For more information about the format of `metrics`, see Metric Widget: Format for Each Metric in the Array (p. 146).

Type: Array of arrays

Required: Yes, when the widget `type` is `metric` and `annotations` is not specified..

period

The default period, in seconds, for all metrics in this widget. The period is the length of time represented by one data point on the graph. This default can be overridden within each metric definition. Use this parameter only for metric widgets. The default is 300.

Valid Values: Any multiple of 60, with 60 as the minimum.

Type: Integer

Required: No

region

The region of the metric.

Type: String

Required: Yes

stacked

Specify `true` to display the graph as a stacked line, or `false` to display as separate lines. This parameter is ignored if `view` is `singleValue`. Use this parameter only for metric widgets.

Type: Boolean

Required: No

stat

The default statistic to be displayed for each metric in the array. This default can be overridden within the definition of each individual metric in the `metrics` array. Use this parameter only for metric widgets.

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

Type: String that is a valid CloudWatch statistic.

Required: No

title

The title to be displayed for the graph or number. Use this parameter only for metric widgets.

Type: String

Required: No

view

Specify `timeSeries` to display this metric as a graph, or `singleValue` to display it as a number.

Valid Values: `timeSeries` | `singleValue`

Type: String

Required: No

yAxis

Limits for the minimums and maximums of the y-axis, if this is a graph. This applies to every metric being graphed, unless specific metrics override it. For more information about the format, see Dashboard Widget Object: yAxis Properties Format (p. 154).

Type: YAxes object

Required: No

```
{
   "type":"metric",
   "x":0,
   "y":0,
   "width":12,
   "height":6,
   "properties":{
      "metrics":[
         [
            "AWS/EC2",
            "CPUUtilization",
            "InstanceId",
            "i-012345"
         ],
         [
            "AWS/EC2",
            "NetworkIn",
            "InstanceId",
            "i-012345",
            {
               "yAxis":"right",
               "label":"NetworkIn",
               "period":3600,
               "stat":"Maximum"
            }
         ]
      ],
      "period":300,
      "stat":"Average",
      "region":"us-east-1",
      "title":"EC2 Instance CPU",
      "stacked":true,
      "view":"timeSeries",
      "liveData":false,
      "yAxis":{
         "left":{
            "min":0,
            "max":100
         },
         "right":{
            "min":50
         }
      },
      "annotations":{
         "horizontal":[
            {
               "visible":true,
               "color":"#9467bd",
               "label":"Critical range",
               "value":20,
               "fill":"above",
               "yAxis":"right"
            }
         ]
      }
   }
}
```

Metric Widget: Format for Each Metric in the Array

Each item in the `metrics` array is either a single metric or a math expression or search expression. Each single metric in the `metrics` array has the following format:

```
[Namespace, MetricName, [{DimensionName,DimensionValue}...] [Rendering Properties Object] ]
```

Each expression in the `metrics` array has the following format:

```
[ {"expression" : "Expression", ["label" : "label] , ["id" : Id] }]
```

accountId

Specifies the AWS account ID where this metric comes from. This enables you to create a widget that contains metrics from multiple accounts on a cross-account dashboard. For more information, see Cross-Account Cross-Region CloudWatch Console.

If you omit this, the current account is used as the default. Use this parameter only for metric widgets.

Type: string

Required: No

Namespace

The AWS namespace containing the metric. If you have multiple entries in the `metrics` array, for each one after the first you may specify only `"."` to use the same namespace as the previous metric in the array.

Type: String

Required: Yes

MetricName

The name of the CloudWatch metric. If you have multiple entries in the `metrics` array, for each one after the first you may specify only `"."` to use the same metric name as the previous metric in the array.

Type: String

Required: Yes, for a single metric

Expression

The math expression or search expression, if this is an expression instead of a single metric.

In a search expression using double-quotes for an exact match, each double-quote mark must be escaped with a backslash.

For more information, see Using Metric Math or Using Search Expressions in Graphs in the Amazon CloudWatch User Guide.

Type: String

Required: Yes, for an expression

DimensionName

The name of a dimension to further refine what data is shown. If you have multiple entries in the `metrics` array, for each one after the first you may specify only `"."` to use the same dimension name as in the corresponding dimension specified in the previous metric in the array. You may specify 0 dimensions for a metric, or up to as many dimensions as the metric support.

Type: String

Required: No

DimensionValue

The value to use for that dimension for the metric. Required if there is a corresponding dimension name.

Type: String

Required: No

Id

The Id of this time series. This Id can be used as part of a math expression. The Id must start with a lowercase letter.

Type: String

Required: No

Label

The label to display in the graph to represent this time series.

Type: String

Required: No

region

The region of the metric. Use this parameter only for metric widgets. If you omit this, the current Region is used as the default.

Type: String

Required: No

Rendering Properties Object

Specifies rendering properties to be used for this particular metric, overriding the values specified for the overall widget. For more information about the format, see Dashboard Widget Object: Rendering Properties Object Format (p. 148).

Type: Metric Render Properties Object

Required: No

```
// The simplest example, a metric with no dimensions
        [ "AWS/EC2", "CPUUtilization" ]

 // A metric with a single dimension
        [ "AWS/EC2", "CPUUtilization", "InstanceId", "i-012345" ]

 // A metric with a single dimension and rendering properties
        [ "AWS/EC2", "DiskReadBytes", "InstanceId", "i-xyz", { "yAxis": "right"} ]

 // The following example graphs the DiskReadBytes metric for three instances.
        [ "AWS/EC2", "DiskReadBytes", "InstanceId", "i-xyz" ],
        [ ".", ".", ".", "i-abc" ],
        [ ".", ".", ".", "i-123" ]

 // The following example includes two metrics and a math expression to sum them.
        [ "AWS/EC2", "DiskReadBytes", "InstanceId", "i-123",{ "id": "m1" } ],
        [ ".", ".", ".", "i-abc", { "id": "m2" } ],
        [ { "expression": "SUM(METRICS())", "label": "Sum of DiskReadbytes", "id": "e3" } ]

  // The following example is a search expression showing the EC2 CPUUtilization for each
 instance in the Region.
        [ { "expression": "SEARCH('{AWS/EC2,InstanceId} MetricName=\"CPUUtilization\"',
 'Average', 300)", "id": "e1" } ],
```

Dashboard Widget Object: Rendering Properties Object Format

Each metric in the `metrics` array can optionally have custom rendering properties that override the default rendering properties specified in the `yAxis` parameter of the `widget` object. This section describes the format for those per-metric custom rendering properties.

color

The six-digit HTML hex color code to be used for this metric.

Type: String

Required: No

label

The label to display for this metric in the graph legend. If this is not specified, the metric is given an autogenerated label that distinguishes it from the other metrics in the widget.

Type: String

Required: No

period

The period for this metric, in seconds. The period is the length of time represented by one data point on the graph.

Valid Values: A multiple of 60, with a minimum of 60.

Type: Integer

Required: No

stat

The statistic for this metric, if it is to be different than the statistic used for the other metrics in the array.

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

Type: String that is a valid CloudWatch statistic.

Required: No

visible

Set this to `true` to have the metric appear in the graph, or `false` to have it be hidden. The default is `true`.

Type: Boolean

Required: No

yAxis

Where on the graph to display the y-axis for this metric. The default is `left`.

Valid Values: `left` | `right`

Type: String

Required: No

```
// The third metric has its own rendering properties, overriding those of the rest of the
widget.
       [ "AWS/EC2", "DiskReadBytes", "InstanceId", "i-xyz" ],
       [ ".", ".", ".", "i-abc" ],
       [ ".", ".", ".", "i-123", { "label":"Instance i-123", "yAxis": "right"}  ]
```

Dashboard Widget Object: Annotation Properties

Annotations include alarms, horizontal annotations, and vertical annotations. A single metric widget can have up to one alarm, or it can have one or more horizontal or vertical annotations. A single widget can't have both an alarm and horizontal or vertical annotations.

Alarm Annotations

If you specify an alarm annotation, you cannot also specify a `metrics` array in the same widget.

alarms

The Amazon Resource Name (ARN) of the alarm.

Type: Array of strings. There can be 0–1 strings in the array.

Required: Only if no metrics are listed.

```

"annotations": {
   "alarms": [ "arn1"]
}

```

Horizontal Annotations

horizontal

An array of horizontal annotations. Horizontal annotations have several options for fill shading, including shading above the annotation line, shading below the annotation line, and "band" shading that appears between two linked annotation lines as part of a single band annotation. Each horizontal annotation in the array that is a single annotation, instead of a band annotation, has the following format:

```
{value, label, color, fill, yAxis, visible}
```

Each horizontal annotation that is a band annotation has the following format:

```
[{value, label, color, yAxis, visible}, {value, label}]
```

value

The metric value in the graph where the horizontal annotation line is to appear. On a band shading annotation, the two values for Value define the upper and lower edges of the band.

On a graph with horizontal annotations, the graph is scaled so that all visible horizontal annotations appear on the graph.

Type: Float

Required: Yes

label

A string that appears on the graph next to the annotation.

Type: String

Required: No

color

The six-digit HTML hex color code to be used for the annotation. This color is used for both the annotation line and the fill shading.

Type: String

Required: No

fill

How to use fill shading with the annotation. Valid values are `above` for shading above the annotation, `below` for shading below the annotation, and `none` for no shading. If `fill` is omitted, there is no shading.

The exception is an annotation with band shading. These annotations always have shading between the two values, and any value for `fill` is ignored.

Type: String

Required: No

visible

Set this to `true` to have the annotation appear in the graph, or `false` to have it be hidden. The default is `true`.

Type: Boolean

Required: No

yAxis

If the graph includes multiple metrics, specifies whether the numbers in `Value` refer to the metric associated with the left Y-axis or the right Y-axis, . Valid values are `right` and `left`.

Type: String

Required: No

```
// A single horizontal annotation with fill shading above the annotation line, based on the
 metric associated with the right Y-axis

"annotations": {
     "horizontal": [
         {
              "visible":true,
              "color":"#9467bd",
              "label":"Critical range",
              "value":20,
              "fill":"above",
              "yAxis":"right"
         }
    ]
}
```

```
// A band annotation. Each value has a label, but other parameters for the band are
 specified only with the first number

"annotations": {
    "horizontal": [
        [
            {
                "label": "Band top",
                "value": 200,
                "color": "#9467bd",
                "visible": true,
                "yAxis": "right"
            },
            {
                "value": 95.5,
                "label": "Band bottom"
            }
        ]
    ]
}

// Three annotations on a graph. The first one is a band annotation. The final one is
 hidden.

"annotations": {
    "horizontal": [
        [
            {
                "label": "Band top",
                "value": 200,
                "color": "#9467bd",
                "visible": true,
                "yAxis": "right"
            },
            {
                "value": 95.5,
                "label": "Band bottom"
            }
        ],
        {
            "visible": true,
            "color": "#9467bd",
            "label": "Label for this annotation",
            "value": 20,
            "fill": "below",
            "yAxis": "right"
        },
        {
            "visible": false,
            "color": "#aaa",
            "label": "Hidden annotation",
            "value": 150
        }
    ]
}
```

Vertical Annotations

vertical

An array of vertical annotations. For each vertical annotation, you can choose to have fill shading before the annotation, after it, or between two vertical lines that are linked as a single band annotation. Each vertical annotation in the array that is a single annotation, instead of a band annotation, has the following format:

```
{value, label, color, fill, visible}
```

Each vertical annotation that is a band annotation has the following format:

```
[{value, label, color, visible}, {value, label}]
```

value

The date and time in the graph where the vertical annotation line is to appear. On a band shading annotation, the two values for Value define the beginning and ending edges of the band.

On a graph with vertical annotations, the graph is scaled so that all visible vertical annotations appear on the graph.

This is defined as a string in ISO 8601 format. For more information, see ISO 8601.

Type: String

Required: Yes

label

A string that appears on the graph next to the annotation.

Type: String

Required: No

color

The six-digit HTML hex color code to be used for the annotation. This color is used for both the annotation line and the fill shading.

Type: String

Required: No

fill

How to use fill shading with the annotation. Valid values are `before` for shading before the annotation, `after` for shading after the annotation, and `none` for no shading. If `fill` is omitted, there is no shading.

The exception is an annotation with band shading. These annotations always have shading between the two values, and any value for `fill` is ignored.

Type: String

Required: No

visible

Set this to `true` to have the annotation appear in the graph, or `false` to have it be hidden. The default is `true`.

Type: Boolean

Required: No

```
// A single vertical annotation with fill shading after the annotation line
```

```
"annotations": {
    "vertical": [
        {
            "visible": true,
            "color": "#9467bd",
            "label": "Bug fix deployed",
            "value": "2018-08-28T15:25:26Z",
            "fill": "after"
        }
    ]
}


// A band vertical annotation. Each annotation line has a label, but other parameters for
 the band are specified only with the first value

"annotations": {
    "vertical": [
        [
            {
                "label": "Band start",
                "value": "2018-08-27T15:25:26Z",
                "color": "#9467bd",
                "visible": true
            },
            {
                "value": "2018-08-28T15:25:26Z",
                "label": "Band end"
            }
        ]
    ]
}
```

Dashboard Widget Object: yAxis Properties Format

Defines the settings for the Y-axis of the graph. The settings include the maximum and minimum, a label for the axis, and whether the axis shows the units. Set this within the `widget` object to affect all metrics in the widget. To override the widget settings for a particular metric, set it for the metric in the `metrics` array.

```
{
  left: {
    min: 0,
    max: 100
  },
  right: {
    min: 0
  }
}
```

left

Optional settings for the left Y-axis.

Type: YAxis object

Required: No

right

Optional settings for the right Y-axis.

Type: YAxis object

Required: No

Each of the `left` and `right` objects can include the following parameters:

label

A label for this Y-axis

Type: String

Required: No

min

The minimum value for this Y-axis

Type: Float

Required: No

max

The maximum value for this Y-axis

Type: Float

Required: No

showUnits

Determines whether the units are shown for the metric associated with this axis. The default is true.

Type: Boolean

Required: No

GetMetricWidgetImage: Metric Widget Structure and Syntax

`MetricWidget` is an input parameter for the https://docs.aws.amazon.com/AmazonCloudWatch/latest/APIReference/API_GetMetricWidgetImage.html API. It is a string in JSON format.

Contents

- Overall Structure (p. 156)
- Format for Each Metric in the Array of Metrics (p. 158)
- Annotation Properties Format (p. 162)
- yAxis Properties Format (p. 166)

Overall Structure

The `MetricWidget` string can include the following parameters:

metrics

The metrics to include in the graph, as a `metrics` array. This can include both raw metric and metric math expressions. One `metrics` array can include 1–100 metrics and expressions. For more information about the format of `metrics`, see Format for Each Metric in the Array of Metrics (p. 158).

Type: Array of arrays

Required: Yes.

annotations

The horizontal and vertical annotations to add to the graph, as annotations arrays. For more information about the format, see Annotation Properties Format (p. 162).

Required: No

end

The date and time for the end of the metrics shown in the graph. This can be expressed as either an absolute value, such as **2018-04-25T12:00:00.000Z** or a relative value such as **-PID**.

If you don't specify `end`, the default of `-PT0H` (the current time) is used.

Type: String

Required: No

height

The height of the widget in pixels. The default is 400.

Valid Values: 1–2000

Type: Integer

Required: No, but you should set this if you also set a value for `width`.

legend

Specifies the location and visibility of the graph legend. `legend` contains one field, `position`. The value of `position` can be `bottom`, `right`, or `hidden`. The default is `bottom`.

Type: String

Required: No

liveData

Specify `true` to display *live data* in the widget. Live data is data published within the last minute that has not been fully aggregated. For more information, see Use Live Data.

Type: Boolean

Required: No

period

The default period, in seconds, for all metrics in this widget. This default can be overridden within each metric definition. The default is 300.

Valid Values: 1, 5, 10, 30, 60, and any multiple of 60. 1, 5, 10, and 30 are only for high-resolution metrics.

Type: Integer

Required: No

region

This parameter is optional. If you include it, it must specify the local Region.

Type: String

Required: No

stacked

Specify `true` to display the graph as a stacked line, or `false` to display as separate lines. The default is false.

Type: Boolean

Required: No

start

The date and time for the start of the metrics shown in the graph. This can be expressed as either an absolute value, such as **2018-04-25T12:00:00.000Z** or a relative value such as **-PID**.

If you don't specify `start`, the default of `-PT3H` (three hours ago) is used.

Type: String

Required: No

stat

The default statistic to be displayed for each metric in the array. This default can be overridden within the definition of each individual metric in the `metrics` array.

If you omit this, the default of `Average` is used.

Valid Values: `SampleCount` | `Average` | `Sum` | `Minimum` | `Maximum` | `p`*`??`*

Type: String that is a valid CloudWatch statistic.

Required: No

timezone

The time zone to use for displaying the times in the graph. The format is + or - followed by four digits. The first two digits indicate the number of hours ahead or behind of UTC, and the final two digits are the number of minutes. For example, `+0130` indicates a time 1 hour and 30 minutes ahead of UTC. The default is `+0000`.

Type: String

Required: No

title

The title to be displayed for the graph.

Type: String

Required: No

view

The display format. Specify `timeSeries` to display this metric as a graph. The default is `timeSeries`.

Valid Value: `timeSeries`

Type: String

Required: No

width

The width of the widget in pixels. The default is 600.

Valid Values: 1–2000

Type: Integer

Required: No, but you should set this if you also set a value for `width`.

yAxis

Limits for the minimums and maximums of the y-axis. This applies to every metric being graphed, unless specific metrics override it. For more information about the format, see yAxis Properties Format (p. 166).

Type: YAxis object

Required: No

Format for Each Metric in the Array of Metrics

Each item in the `metrics` array is a CloudWatch metric to display in the graph, or to use as part of a math expression displayed in the graph. For more information about math expressions, see Use Metric Math in the Amazon CloudWatch User Guide.

Each metric in the array has the following format:

```
[Namespace, MetricName, Dimension1Name, Dimension1Value, Dimension2Name, Dimension2Value...
```

```
            {Options Object}]
```

Namespace

The AWS namespace containing the metric. To use the same namespace as the previous metric in the array, you may specify `"."` for each entry after the first.

Type: String

Required: Yes

MetricName

The name of the CloudWatch metric. To use the same name as the previous metric in the array, you may specify `"."` for each entry after the first.

Type: String

Required: Yes

DimensionName

The name of a dimension to further refine what data is shown. To use the same dimension name as the previous metric in the array, you may specify `"."` for each entry after the first. You may specify zero dimensions for a metric, or up to as many dimensions as the metric supports.

Type: String

Required: No

DimensionValue

The value to use for that dimension for the metric. Required if there is a corresponding dimension name.

Type: String

Required: No, unless there is a corresponding dimension name.

Options Object

Specifies either custom rendering properties to be used for the specified CloudWatch metric, or a math expression to display on the graph. For more information about the format, see Options Object Format (p. 160).

Type: Options Object

Required: No

Examples

```
// The simplest example, a metric with no dimensions
        [ "AWS/EC2", "CPUUtilization" ]

 // A metric with a single dimension
        [ "AWS/EC2", "CPUUtilization", "InstanceId", "i-01234567890123456" ]

 // A metric with a single dimension and rendering properties
        [ "AWS/EC2", "DiskReadBytes", "InstanceId", "i-01234567890123456", { yAxis:
 "right"} ]

 // The following example graphs the DiskReadBytes metric for three instances.
        [ "AWS/EC2", "DiskReadBytes", "InstanceId", "i-01234567890123456" ],
        [ ".", ".", ".", "i-abc" ],
        [ ".", ".", ".", "i-123" ]
```

Options Object Format

Specifies either custom rendering properties to be used for the specified CloudWatch metric, or a math expression to display on the graph.

If this object is specified as part of a CloudWatch metric in the `metrics` array, it sets custom rendering properties for this metric and overrides the defaults used for the whole graph.

You can also specify this object to add a math expression to the graph. In this case, the other settings in this object specify the display options for the result of the math expression.

This section describes the format of these options objects.

color

The six-digit HTML hex color code to be used for this metric or expression.

Type: String

Required: No

expression

A math expression to display. For more information about supported math expression functions and format, see Metric Math Syntax and Functions in the Amazon CloudWatch User Guide.

Type: String that is a valid CloudWatch metric math expression.

Required: Yes if this is an expression.

label

The label to display for this metric or expression in the graph legend. If this is not specified, the metric is given an auto-generated label that distinguishes it from the other metrics in the widget.

Type: String

Required: No

id

An identifier for this metric or expression, which must be unique within this widget. The id can be used as a variable to represent this metric or expression within math expressions. Valid characters are letters, numbers, and underscore. The first character must be a lowercase letter.

Type: String

Required: No

period

The period for this metric, in seconds. If specified, this overrides the default period used for other metrics in this graph. This parameter is not applicable for math expressions.

Valid Values: 1, 5, 10, 30, 60, and any multiple of 60. 1, 5, 10, and 30 are only for high-resolution metrics.

Type: Integer

Required: No

stat

The statistic to be displayed for this metric, if it is to be different than the statistic used for the other metrics in the graph. This parameter is not applicable for math expressions.

Valid Values: `SampleCount` | `Average` | `Sum` | `Minimum` | `Maximum` | `p`*`??`*

Type: String that is a valid CloudWatch statistic.

Required: No

visible

Specifies whether this metric or expression is shown on the graph. The default is `true`.

Setting `visible` to `false` is useful if you want to hide the raw metrics that are used in math expressions, and show only the expression results on the graph.

Type: Boolean

Required: No

yAxis

Where on the graph to display the y-axis for this metric or expression. The default is `left`.

Valid Values: `left` | `right`

Type: String

Required: No

Example

In the following example, CloudWatch retrieves a custom `apiLatency` metric. At the top, the p50 statistic is specified to show the median value. Next, for the same metric on the same instance (this is specified by the four fields that are just periods), the Average value is graphed. Next is an options object with a math expression, showing the halfway value of the two metrics. Finally, another expression shows the rate of change.

To show only the results of the two expressions on the graph and hide the raw metrics, you could change the first two instances of `visible` to `false`.

```
{
        "metrics": [
            [
                "MyNamespace",
                "apiLatency",
                "InstanceId",
                "i-0987654321abcdef0",
                {
                    "id": "m1",
                    "stat": "p50",
                    "label": "Median value",
                    "visible": true,
                    "color": "#dddddd",
                    "yAxis": "left",
                    "period": 300
                }
            ],
            [
                ".",
                ".",
                ".",
                ".",
                {
                    "id": "m2",
                    "stat": "Average",
                    "label": "Average value",
                    "visible": true,
                    "color": "#cccccc",
                    "yAxis": "left",
                    "period": 300
                }
            ],
            [
                {
                    "expression": "(m1+m2)/2",
                    "id": "e1",
                    "label": "Half way between average and median",
                    "visible": true,
                    "color": "#000000",
                    "yAxis": "left"
                }
            ],
            [
                {
                    "expression": "RATE(e1)",
                    "yAxis": "right",
                    "label": "rate of change of the half way point"
                }
            ]
        ]
    }
```

Annotation Properties Format

A single graph can have multiple horizontal and vertical annotations. All horizontal annotations are specified in one `horizontal` field, and all vertical annotations are specified in one `vertical` field.

horizontal

An array of horizontal annotations. Horizontal annotations have several options for fill shading, including shading above the annotation line, shading below the annotation line, and "band" shading that appears between two linked annotation lines as part of a single annotation. Each horizontal annotation in the array that does not have band shading has the following format:

```
{value, label, color, fill, yAxis, visible}
```

Each horizontal annotation that does have band shading has the following format:

```
[{value, label, color, yAxis, visible}, {value, label}]
```

vertical

An array of vertical annotations. Vertical annotations have several options for fill shading, including shading before the annotation line, shading after the annotation line, and "band" shading that appears between two linked annotation lines as part of a single band annotation. Each vertical annotation in the array that does not have band shading has the following format:

```
{value, label, color, fill, visible}
```

Each vertical annotation that does have band shading has the following format:

```
[{value, label, color, visible}, {value, label}]
```

The `horizontal` array can include the following fields.

value

The metric value in the graph where the horizontal annotation line is to appear. On a band shading annotation, the two values for `Value` define the upper and lower edges of the band.

On a graph with horizontal annotations, the graph is scaled so that all visible horizontal annotations appear on the graph.

Type: Float

Required: Yes, if horizontal annotations are used.

label

A string that appears on the graph next to the annotation.

Type: String

Required: No

color

The six-digit HTML hex color code to be used for the annotation. This color is used for both the annotation line and the fill shading.

Type: String

Required: No

fill

How to use fill shading with the annotation. Valid values are `above` for shading above the annotation, `below` for shading below the annotation, and `none` for no shading. If `fill` is omitted, there is no shading.

The exception is an annotation with band shading. These annotations always have shading between the two values, and any value for `fill` is ignored.

Type: String

Required: No

visible

Set this to `true` to have the annotation appear in the graph, or `false` to have it be hidden. The default is `true`.

Type: Boolean

Required: No

yAxis

If the graph includes multiple metrics, specifies whether the numbers in `Value` refer to the metric associated with the left Y-axis or the right Y-axis. Valid values are `right` and `left`.

Type: String

Required: No

The `vertical` array can include the following fields.

value

The time stamp where the vertical annotation line is to appear. This must be specified as an absolute time stamp, such as `2018-08-28T15:25:26Z`. On a band shading annotation, the two values for `Value` define the beginning and ending edges of the band.

Type: String

Required: Yes, if vertical annotations are used.

label

A descriptive string that appears on the graph next to the annotation.

Type: String

Required: No

color

The six-digit HTML hex color code to be used for the annotation. This color is used for both the annotation line and the fill shading.

Type: String

Required: No

fill

How to use fill shading with the annotation. Valid values are `before` for shading before the annotation, `after` for shading after the annotation, and `none` for no shading. If `fill` is omitted, there is no shading.

The exception is an annotation with band shading. These annotations always have shading between the two values, and any value for `fill` is ignored.

Type: String

Required: No

visible

Set this to `true` to have the annotation appear in the graph, or `false` to have it be hidden. The default is `true`.

Type: Boolean

Required: No

Examples

```
// A single horizontal annotation with fill shading above the annotation line, based on the
 metric associated with the right Y-axis

"annotations": {
     "horizontal": [
         {
              "visible":true,
              "color":"#9467bd",
              "label":"Critical range",
              "value":20,
              "fill":"above",
              "yAxis":"right"
         }
    ]
}

// A horizontal band annotation. Each value has a label, but other parameters for the band
 need to be specified only with the first number

"annotations": {
    "horizontal": [
        [
             {
                  "label":"Band top",
                  "value":200,
                  "color":"#9467bd",
                  "visible":true,
                  "yAxis":"right"
              },
              {
                  "value":95.5,
                  "label":"Band bottom"
              }
         ]
     ]
}

// A single vertical annotation with fill shading after the annotation line

"annotations": {
    "vertical": [
        {
            "visible": true,
            "color": "#9467bd",
            "label": "Bug fix deployed",
            "value": "2018-08-28T15:25:26Z",
            "fill": "after"
        }
    ]
}


// A vertical band annotation. Each annotation line has a label, but other parameters for
 the band are specified only with the first value

"annotations": {
    "vertical": [
        [
            {
                "label": "Band start",
                "value": "2018-08-27T15:25:26Z",
                "color": "#9467bd",
                "visible": true
            },
            {
                "value": "2018-08-28T15:25:26Z",
                "label": "Band end"
            }
        ]
    ]
}
```

yAxis Properties Format

Defines the minimum and maximum values for the Y-axis of the graph. Set this within the `MetricWidget` object to affect all metrics in the widget. To override the widget settings for a particular metric, set it in the options object for that metric in the `metrics` array.

left

Optional `min` and `max` settings for the left Y-axis.

Type: YAxis object

Required: No

right

Optional `min` and `max` settings for the right Y-axis.

Type: YAxis object

Required: No

Each of the `left` and `right` objects can include the following parameters:

min

The minimum value for this Y-axis.

Type: Float

Required: No

max

The maximum value for this Y-axis.

Type: Float

Required: No

Example

```
{
  left: {
    min: 0,
    max: 100
  },
  right: {
    min: 0
  }
}
```

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. 170).

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

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

Request Identifiers

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

Query API Authentication

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

Available Libraries

AWS provides libraries, sample code, tutorials, and other resources for software developers who prefer to build applications using language-specific APIs instead of the command-line tools and Query API.

These libraries provide basic functions (not included in the APIs), such as request authentication, request retries, and error handling so that it is easier to get started. Libraries and resources are available for the following languages and platforms:

- AWS Mobile SDK for Android
- AWS SDK for Go
- AWS Mobile SDK for iOS
- AWS SDK for Java 2.x
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for JavaScript in Node.js
- AWS SDK for .NET
- AWS SDK for PHP
- AWS SDK for Python (Boto)
- AWS SDK for Ruby

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

Making API Requests Using the POST Method

If you don't use one of the AWS SDKs, you can make CloudWatch API requests over HTTP using the POST request method. The POST method requires you to specify the operation in the header of the request and provide the data for the operation in JSON format in the body of the request.

Header name	Header value
host	The Amazon CloudWatch endpoint. For example, `monitoring.us-west-1.amazonaws.com`
x-amz-date	You must provide the time stamp in either the HTTP Date header or the AWS *x-amz-date* header. Some HTTP client libraries don't let you set the Date header. When an *x-amz-date* header is present, the system ignores any Date header during the request authentication. The *x-amz-date* header must be specified in ISO 8601 basic format. For example: `20130315T092054Z`
Authorization	The set of authorization parameters that AWS uses to ensure the validity and authenticity of the request. For more information about constructing this header, see Signature Version 4 Signing Process in the *Amazon Web Services General Reference*.

x-amz-target	Specifies the CloudWatch operation:
	`GranitServiceVersion20100801.API_Name`
	For example, for GetMetricData the target value is the following:
	`GraniteServiceVersion20100801.GetMetricData`

Content-Type	Specifies the input format. The valid value is `application/json`
Accept	Specifies the response format. The valid value is `application/json`

Header name	Header value
Content-length	Size of the payload in bytes.
Content-Encoding	Specifies the encoding format of the input and output. The valid value is `amz-1.0`

The following is an example header for an HTTP request to return metric data in JSON format:

```
POST / HTTP/1.1
host: monitoring.us-east-1.amazonaws.com
x-amz-target: GraniteServiceVersion20100801.GetMetricData
x-amz-date: 20180112T092034Z
Authorization: AWS4-HMAC-SHA256 Credential=REDACTEDREDACTED/20180411/us-east-1/monitoring/
aws4_request, SignedHeaders=content-encoding;content-length;content-type;host;x-amz-date;x-
amz-target, Signature=e945ed75cb91f88f138445fba02d3af93d96bfd8491e5d03588ae1b65188ff1d
Content-Type: application/json
Accept: application/json
Content-Encoding: amz-1.0
Content-Length: 45
Connection: keep-alive
```

Common Parameters

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

Action

The action to be performed.

Type: string

Required: Yes

Version

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

Type: string

Required: Yes

X-Amz-Algorithm

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

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

Type: string

Valid Values: `AWS4-HMAC-SHA256`

Required: Conditional

X-Amz-Credential

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

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

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

Type: string

Required: Conditional

X-Amz-Date

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

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

Type: string

Required: Conditional

X-Amz-Security-Token

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

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

Type: string

Required: Conditional

X-Amz-Signature

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

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

Type: string

Required: Conditional

X-Amz-SignedHeaders

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

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

Type: string

Required: Conditional

Common Errors

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

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

IncompleteSignature

The request signature does not conform to AWS standards.

HTTP Status Code: 400

InternalFailure

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

HTTP Status Code: 500

InvalidAction

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

HTTP Status Code: 400

InvalidClientTokenId

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

HTTP Status Code: 403

InvalidParameterCombination

Parameters that must not be used together were used together.

HTTP Status Code: 400

InvalidParameterValue

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

HTTP Status Code: 400

InvalidQueryParameter

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

HTTP Status Code: 400

MalformedQueryString

The query string contains a syntax error.

HTTP Status Code: 404

MissingAction

The request is missing an action or a required parameter.

HTTP Status Code: 400

MissingAuthenticationToken

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

HTTP Status Code: 403

MissingParameter

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

OptInRequired

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

HTTP Status Code: 403

RequestExpired

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

HTTP Status Code: 400

ServiceUnavailable

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

HTTP Status Code: 503

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 400

ValidationError

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

HTTP Status Code: 400