

---

# Amazon CloudWatch

## API Reference

**API Version 2010-08-01**



## **Amazon CloudWatch: API Reference**

Copyright © 2018 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

## Table of Contents

Welcome .....	1
Actions .....	2
DeleteAlarms .....	3
Request Parameters .....	3
Errors .....	3
See Also .....	3
DeleteDashboards .....	4
Request Parameters .....	4
Errors .....	4
See Also .....	4
DescribeAlarmHistory .....	5
Request Parameters .....	5
Response Elements .....	6
Errors .....	6
See Also .....	6
DescribeAlarms .....	7
Request Parameters .....	7
Response Elements .....	8
Errors .....	8
See Also .....	8
DescribeAlarmsForMetric .....	9
Request Parameters .....	9
Response Elements .....	10
Errors .....	10
See Also .....	10
DisableAlarmActions .....	11
Request Parameters .....	11
Errors .....	11
See Also .....	11
EnableAlarmActions .....	12
Request Parameters .....	12
Errors .....	12
See Also .....	12
GetDashboard .....	13
Request Parameters .....	13
Response Elements .....	13
Errors .....	13
See Also .....	14
GetMetricData .....	15
Request Parameters .....	15
Response Elements .....	16
Errors .....	16
Examples .....	17
See Also .....	24
GetMetricStatistics .....	25
Request Parameters .....	25
Response Elements .....	28
Errors .....	28
See Also .....	28
GetMetricWidgetImage .....	30
Request Parameters .....	30
Response Elements .....	31
Errors .....	31
Example .....	31

See Also .....	32
ListDashboards .....	33
Request Parameters .....	33
Response Elements .....	33
Errors .....	33
See Also .....	34
ListMetrics .....	35
Request Parameters .....	35
Response Elements .....	35
Errors .....	36
See Also .....	36
PutDashboard .....	37
Request Parameters .....	37
Response Elements .....	37
Errors .....	38
Examples .....	38
See Also .....	42
PutMetricAlarm .....	43
Request Parameters .....	43
Errors .....	48
See Also .....	48
PutMetricData .....	49
Request Parameters .....	49
Errors .....	50
Examples .....	50
See Also .....	52
SetAlarmState .....	53
Request Parameters .....	53
Errors .....	53
See Also .....	54
Data Types .....	55
AlarmHistoryItem .....	56
Contents .....	56
See Also .....	56
DashboardEntry .....	58
Contents .....	58
See Also .....	58
DashboardValidationMessage .....	59
Contents .....	59
See Also .....	59
Datapoint .....	60
Contents .....	60
See Also .....	61
Dimension .....	62
Contents .....	62
See Also .....	62
DimensionFilter .....	63
Contents .....	63
See Also .....	63
MessageData .....	64
Contents .....	64
See Also .....	64
Metric .....	65
Contents .....	65
See Also .....	65
MetricAlarm .....	66
Contents .....	66

See Also .....	70
MetricDataQuery .....	71
Contents .....	71
See Also .....	72
MetricDataResult .....	73
Contents .....	73
See Also .....	74
MetricDatum .....	75
Contents .....	75
See Also .....	76
MetricStat .....	78
Contents .....	78
See Also .....	78
StatisticSet .....	79
Contents .....	79
See Also .....	79
Dashboard Body Structure and Syntax .....	80
Overall Structure .....	80
Properties of a Text Widget Object .....	82
Properties of a Metric Widget Object .....	82
Metric Widget: Format for Each Metric in the Array .....	85
Dashboard Widget Object: Rendering Properties Object Format .....	86
Dashboard Widget Object: Annotation Properties .....	87
Dashboard Widget Object: yAxis Properties Format .....	92
GetMetricWidgetImage: Metric Widget Structure and Syntax .....	93
Overall Structure .....	93
Format for Each Metric in the Array of Metrics .....	95
Options Object Format .....	96
Annotation Properties Format .....	99
yAxis Properties Format .....	102
Making API Requests .....	104
Amazon CloudWatch Endpoints .....	104
Query Parameters .....	104
Request Identifiers .....	104
Query API Authentication .....	104
Available Libraries .....	104
Making API Requests Using the POST Method .....	105
Common Parameters .....	107
Common Errors .....	109

# 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 \(p. 55\)](#): An alphabetical list of all CloudWatch data types.
- [Common Parameters \(p. 107\)](#): Parameters that all Query actions can use.
- [Common Errors \(p. 109\)](#): 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)
- [DeleteDashboards](#) (p. 4)
- [DescribeAlarmHistory](#) (p. 5)
- [DescribeAlarms](#) (p. 7)
- [DescribeAlarmsForMetric](#) (p. 9)
- [DisableAlarmActions](#) (p. 11)
- [EnableAlarmActions](#) (p. 12)
- [GetDashboard](#) (p. 13)
- [GetMetricData](#) (p. 15)
- [GetMetricStatistics](#) (p. 25)
- [GetMetricWidgetImage](#) (p. 30)
- [ListDashboards](#) (p. 33)
- [ListMetrics](#) (p. 35)
- [PutDashboard](#) (p. 37)
- [PutMetricAlarm](#) (p. 43)
- [PutMetricData](#) (p. 49)
- [SetAlarmState](#) (p. 53)

# DeleteAlarms

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

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 107\)](#).

### **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 \(p. 109\)](#).

### **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 V2](#)



# DeleteDashboards

Deletes all dashboards that you specify. You may 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 \(p. 107\)](#).

### **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 \(p. 109\)](#).

### **InternalServerError**

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 V2](#)

# DescribeAlarmHistory

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

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 \(p. 107\)](#).

### **AlarmName**

The name of the alarm.

Type: String

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

Required: No

### **EndDate**

The ending date to retrieve alarm history.

Type: Timestamp

Required: No

### **HistoryItemType**

The type of alarm histories to retrieve.

Type: String

Valid Values: `ConfigurationUpdate` | `StateUpdate` | `Action`

Required: No

### **MaxRecords**

The maximum number of alarm history records to retrieve.

Type: Integer

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

Required: No

### **NextToken**

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

Type: String

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

Required: No

### **StartDate**

The starting date to retrieve alarm history.

Type: Timestamp

Required: No

## Response Elements

The following elements are returned by the service.

### **AlarmHistoryItems.member.N**

The alarm histories, in JSON format.

Type: Array of [AlarmHistoryItem \(p. 56\)](#) objects

### **NextToken**

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

Type: String

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

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 109\)](#).

### **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 V2](#)

# DescribeAlarms

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

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 107\)](#).

### ActionPrefix

The action name prefix.

Type: String

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

Required: No

### AlarmNamePrefix

The alarm name 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.

Type: Array of strings

Array Members: Maximum number of 100 items.

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

Required: No

### MaxRecords

The maximum number of alarm descriptions to retrieve.

Type: Integer

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

Required: No

### NextToken

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

Type: String

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

Required: No

### **StateValue**

The state value to be used in matching alarms.

Type: String

Valid Values: OK | ALARM | INSUFFICIENT\_DATA

Required: No

## Response Elements

The following elements are returned by the service.

### **MetricAlarms.member.N**

The information for the specified alarms.

Type: Array of [MetricAlarm \(p. 66\)](#) objects

### **NextToken**

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

Type: String

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

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 109\)](#).

### **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 V2](#)

# 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. 107).

### **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. 62) 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 \(p. 66\)](#) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 109\)](#).

## See Also

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

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

# 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 \(p. 107\)](#).

### **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 \(p. 109\)](#).

## See Also

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

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)



# EnableAlarmActions

Enables the actions for the specified alarms.

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 107\)](#).

### **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 \(p. 109\)](#).

## See Also

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

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

# 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 \(p. 107\)](#).

### **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 \(p. 80\)](#).

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 \(p. 109\)](#).

### **InternalServerError**

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 V2](#)

# GetMetricData

You can use the `GetMetricData` API to retrieve as many as 100 different metrics in a single request, with a total of as many as 100,800 datapoints. 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.

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#) (p. 107).

### EndTime

The time stamp indicating the latest data to be returned.

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 100 `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. 71) 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

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

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.

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.

#### **MetricDataResults.member.N**

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

Type: Array of [MetricDataResult](#) (p. 73) objects

#### **NextToken**

A token that marks the next batch of returned results.

Type: String

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

## Errors

For information about the errors that are common to all actions, see [Common Errors](#) (p. 109).

### 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": 1518867832,
  "MetricDataQueries": [
    {
      "Id": "m1",
      "MetricStat": {
        "Metric": {
          "Namespace": "AWS/EC2",
          "MetricName": "NetworkOut",
          "Dimensions": [
            {
              "Name": "AutoScalingGroupName",
              "Value": "my-asg-A"
            }
          ]
        }
      },
      "Period": 300,
      "Stat": "Average",
      "Unit": "Gigabytes"
    },
    {
      "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": [
        1518867432,
        1518867732,
        1518868032
      ],
      "Values": [
        15000,
        14000,
        16000
      ]
    },
    {
      "Id": "m2",
      "StatusCode": "Complete",
      "Label": "AWS/EC2 NetworkOut SampleCount",
      "Timestamps": [
        1518867432,
        1518867732,
        1518868032
      ],
      "Values": [
        15,
        14,
        16
      ]
    },
    {
      "Id": "m3",
      "StatusCode": "Complete",
      "Label": "AWS/EC2 HealthyHostCount",
      "Timestamps": [
        1518867432,
        1518867732,
        1518868032
      ],
      "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": 1518867832,
  "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": "Gigabytes"
      },
      "ReturnData": false
    },
    {
      "Id": "m2",
      "MetricStat": {
        "Metric": {
          "Namespace": "AWS/EC2",
          "MetricName": "NetworkOut",
          "Dimensions": [
            {
              "Name": "AutoScalingGroupName",
              "Value": "my-asg-B"
            }
          ]
        },
        "Period": 300,
        "Stat": "SampleCount",
        "Unit": "Gigabytes"
      },
      "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": [
    1518867432,
    1518867732,
    1518868032
  ],
  "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": 1518867832,
  "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": "Gigabytes"
    }
  ]
}

```

```

    },
    "ReturnData": false
  },
  {
    "Id": "m2",
    "MetricStat": {
      "Metric": {
        "Namespace": "AWS/EC2",
        "MetricName": "NetworkOut",
        "Dimensions": [
          {
            "Name": "AutoScalingGroupName",
            "Value": "my-asg-B"
          }
        ]
      },
      "Period": 300,
      "Stat": "SampleCount",
      "Unit": "Gigabytes"
    },
    "ReturnData": false
  },
  {
    "Id": "m3",
    "MetricStat": {
      "Metric": {
        "Namespace": "AWS/EC2",
        "MetricName": "NetworkOut",
        "Dimensions": [
          {
            "Name": "AutoScalingGroupName",
            "Value": "my-asg-C"
          }
        ]
      },
      "Period": 300,
      "Stat": "SampleCount",
      "Unit": "Gigabytes"
    },
    "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": [
      1518867432,
      1518867732,
      1518868032
    ],
    "Values": [
      200,
      200,
      200
    ]
  },
  {
    "Id": "e2",
    "StatusCode": "Complete",
    "Label": "my-asg-A * my-asg-B",
    "Timestamps": [
      1518867432,
      1518867732,
      1518868032
    ],
    "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": 1518867832,
  "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": [
        1518867432,
        1518867732,
        1518868032
      ],
      "Values": [
        0.1,
        0.5,
        0.3
      ]
    },
    {
      "Id": "errorRatePercent",
      "Label": "% Error Rate",
      "StatusCode": "Complete",
      "Timestamps": [
        1518867432,
        1518867732,
        1518868032
      ],
      "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 V2](#)

# 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. 107).

### **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 \(p. 62\)](#) 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. 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. 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. Metrics may be reported in multiple units. Not supplying a unit results in all units being returned. If you specify only a unit that the metric does not report, the results of the call are null.

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 \(p. 60\)](#) 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 \(p. 109\)](#).

### **InternalServerError**

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 V2](#)

# 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 \(p. 107\)](#).

### 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 \(p. 93\)](#).

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="http://monitoring.amazonaws.com/doc/2010-08-01/">
  <GetMetricWidgetImageResult>
    <MetricWidgetImage>
      iVBORwOKGgoAAAANSUHEUgAAAlgAAAGQEAYAAAip...
    </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 \(p. 109\)](#).

## 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 V2](#)

# 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 \(p. 107\)](#).

### **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

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

Required: No

## Response Elements

The following elements are returned by the service.

### **DashboardEntries.member.N**

The list of matching dashboards.

Type: Array of [DashboardEntry \(p. 58\)](#) objects

### **NextToken**

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

Type: String

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

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 109\)](#).

### **InternalServerError**

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 V2](#)

## ListMetrics

List the specified metrics. You can use the returned metrics with [GetMetricData \(p. 15\)](#) or [GetMetricStatistics \(p. 25\)](#) 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 \(p. 15\)](#) or [GetMetricStatistics \(p. 25\)](#).

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 107\)](#).

### **Dimensions.member.N**

The dimensions to filter against.

Type: Array of [DimensionFilter \(p. 63\)](#) objects

Array Members: Maximum number of 10 items.

Required: No

### **MetricName**

The name of the metric to filter against.

Type: String

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

Required: No

### **Namespace**

The namespace to filter against.

Type: String

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

Pattern: [ ^ : ] . \*

Required: No

### **NextToken**

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

Type: String

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

Required: No

## Response Elements

The following elements are returned by the service.



**Metrics.member.N**

The metrics.

Type: Array of [Metric \(p. 65\)](#) objects

**NextToken**

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

Type: String

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

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 109\)](#).

**InternalServerError**

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 V2](#)

# 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.

There is no limit to the number of dashboards in your account. 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 \(p. 107\)](#).

### **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 \(p. 80\)](#).

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 may not render.

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

Type: Array of [DashboardValidationMessage \(p. 59\)](#) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 109\)](#).

### InternalServerError

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 V2](#)

# PutMetricAlarm

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

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

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

If you are an IAM user, you must have Amazon EC2 permissions for some 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
- `ec2:DescribeInstanceRecoveryAttribute` and `ec2:RecoverInstances` for alarms with recover actions

If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm, but the stop or terminate actions 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 about service-linked roles, see [AWS service-linked role](#).

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#) (p. 107).

### **ActionsEnabled**

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

Type: Boolean

Required: No

### **AlarmActions.member.N**

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

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



|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 the AWS account.

Type: String

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

Required: Yes

#### **ComparisonOperator**

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

Type: String

Valid Values: GreaterThanOrEqualToThreshold | GreaterThanThreshold |  
LessThanThreshold | LessThanOrEqualToThreshold

Required: Yes

#### **DatapointsToAlarm**

The number of datapoints 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 associated with the alarm.

Type: Array of [Dimension](#) (p. 62) 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 which 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 associated with the alarm. Specify a value between p0.0 and p100. When you call `PutMetricAlarm`, 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: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.

Type: String

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

Required: Yes

#### **Namespace**

The namespace for the metric associated with the alarm.

Type: String

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

Pattern: [ ^ : ] . \*

Required: Yes

#### **OKActions.member.N**

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

Valid Values: `arn:aws:automate:region:ec2:stop` |  
`arn:aws:automate:region:ec2:terminate` | `arn:aws:automate:region:ec2:recover`  
| `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 period, in seconds, over which the specified statistic is applied. Valid values are 10, 30, and any multiple of 60.

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 may often lapse into `INSUFFICIENT_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: Yes

### Statistic

The statistic for the metric associated with the alarm, other than percentile. For percentile statistics, use `ExtendedStatistic`. When you call `PutMetricAlarm`, you must specify either `Statistic` or `ExtendedStatistic`, but not both.

Type: String

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

Required: No

### Threshold

The value against which the specified statistic is compared.

Type: Double

Required: Yes

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

Type: String

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

Required: No

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 109\)](#).

### **LimitExceeded**

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

HTTP Status Code: 400

## See Also

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

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

## 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](#) (p. 35).

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 8.515920e-109 to 1.174271e+108 (Base 10) or 2e-360 to 2e360 (Base 2). In addition, special values (for example, NaN, +Infinity, -Infinity) are not supported.

You can use up to 10 dimensions per metric to further clarify what data the metric collects. 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](#) (p. 15) or [GetMetricStatistics](#) (p. 25) from the time they are submitted.

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. 107).

### **MetricData.member.N**

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

Type: Array of [MetricDatum](#) (p. 75) objects

Required: Yes

### **Namespace**

The namespace for the metric data.

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

Type: String

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

Pattern: [ ^ : ] . \*

Required: Yes

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 109\)](#).

### 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 V2](#)

# 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. The alarm returns to its actual state (often within seconds). Because the alarm state change happens quickly, it is typically only visible in the alarm's **History** tab in the Amazon CloudWatch console or through [DescribeAlarmHistory](#) (p. 5).

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#) (p. 107).

### **AlarmName**

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

Type: String

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

Required: Yes

### **StateReason**

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

Type: String

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

Required: Yes

### **StateReasonData**

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

Type: String

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

Required: No

### **StateValue**

The value of the state.

Type: String

Valid Values: `OK` | `ALARM` | `INSUFFICIENT_DATA`

Required: Yes

## Errors

For information about the errors that are common to all actions, see [Common Errors](#) (p. 109).

### **InvalidFormat**

Data was not syntactically valid JSON.

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 V2](#)

# 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. 56)
- [DashboardEntry](#) (p. 58)
- [DashboardValidationMessage](#) (p. 59)
- [Datapoint](#) (p. 60)
- [Dimension](#) (p. 62)
- [DimensionFilter](#) (p. 63)
- [MessageData](#) (p. 64)
- [Metric](#) (p. 65)
- [MetricAlarm](#) (p. 66)
- [MetricDataQuery](#) (p. 71)
- [MetricDataResult](#) (p. 73)
- [MetricDatum](#) (p. 75)
- [MetricStat](#) (p. 78)
- [StatisticSet](#) (p. 79)

# AlarmHistoryItem

Represents the history of a specific alarm.

## Contents

### AlarmName

The descriptive name for the alarm.

Type: String

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

Required: No

### HistoryData

Data about the alarm, in JSON format.

Type: String

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

Required: No

### HistoryItemType

The type of alarm history item.

Type: String

Valid Values: `ConfigurationUpdate` | `StateUpdate` | `Action`

Required: No

### HistorySummary

A summary of the alarm history, in text format.

Type: String

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

Required: No

### Timestamp

The time stamp for the alarm history item.

Type: Timestamp

Required: No

## See Also

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

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)

- [AWS SDK for Java](#)
- [AWS SDK for Ruby V2](#)

# 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 V2](#)

# 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 V2](#)



# 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 V2](#)

# 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 V2](#)

# 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 V2](#)

# 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 V2](#)

# Metric

Represents a specific metric.

## Contents

### **Dimensions.member.N**

The dimensions for the metric.

Type: Array of [Dimension \(p. 62\)](#) 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 V2](#)

# MetricAlarm

Represents an alarm.

## Contents

### **ActionsEnabled**

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

Type: Boolean

Required: No

### **AlarmActions.member.N**

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

Type: Array of strings

Array Members: Maximum number of 5 items.

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

Required: No

### **AlarmArn**

The Amazon Resource Name (ARN) of the alarm.

Type: String

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

Required: No

### **AlarmConfigurationUpdatedTimestamp**

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

Type: Timestamp

Required: No

### **AlarmDescription**

The description of the alarm.

Type: String

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

Required: No

### **AlarmName**

The name of the alarm.

Type: String

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

Required: No

### **ComparisonOperator**

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

Type: String

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

Required: No

### **DatapointsToAlarm**

The number of datapoints 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 \(p. 62\)](#) 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.

Type: String

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

Required: No

**Namespace**

The namespace of the metric associated with the alarm.

Type: String

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

Pattern: [ ^ : ] . \*

Required: No

**OKActions.member.N**

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

Type: Array of strings

Array Members: Maximum number of 5 items.

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

Required: No

**Period**

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

Type: Integer

Valid Range: Minimum value of 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

**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 V2](#)

# MetricDataQuery

This structure 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 100 `MetricDataQuery` structures.

## Contents

### Expression

The math expression to be performed on the returned data, if this structure is performing a math expression. For more information about metric math expressions, see [Metric Math Syntax and Functions](#) in the *Amazon CloudWatch User Guide*.

Within one `MetricDataQuery` structure, 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 structure 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 structure is performing a data retrieval and not performing a math expression on the returned data.

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

Type: [MetricStat \(p. 78\)](#) object

Required: No

### ReturnData

Indicates whether to return the time stamps 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.

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 V2](#)

# MetricDataResult

A `GetMetricData` call returns an array of `MetricDataResult` structures. Each of these structures includes the data points for that metric, along with the time stamps 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. 64) 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 time stamps for the data points, formatted in Unix timestamp format. The number of time stamps 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 time stamps and the time stamp 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 V2](#)

# 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 \(p. 62\)](#) 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 \(p. 79\)](#) 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

The unit of the metric.

Type: String

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

Required: No

### Value

The value for the metric.

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

Type: Double

Required: No

### 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 8.515920e-109 to 1.174271e+108 (Base 10) or 2e-360 to 2e360 (Base 2). In addition, special values (for example, NaN, +Infinity, -Infinity) are not supported.

Type: 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 V2](#)



# 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 \(p. 65\)](#) object

Required: Yes

### Period

The period, in seconds, to use when retrieving the metric.

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

The unit to use for the returned data points.

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 V2](#)

# 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 V2](#)

# Dashboard Body Structure and Syntax

## Contents

- [Overall Structure \(p. 80\)](#)
- [Properties of a Text Widget Object \(p. 82\)](#)
- [Properties of a Metric Widget Object \(p. 82\)](#)
- [Metric Widget: Format for Each Metric in the Array \(p. 85\)](#)

## Overall Structure

A DashboardBody is a string in JSON format. It is an array of between 0 and 100 widget objects.

```
{
  "widgets": [ {widget1},{widget2}, ... {widgetN} ]
}
```

An example of this structure with one metric widget and one text widget:

```
{
  "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"
      }
    }
  ]
}
```

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.

Each widget of any type can have the following properties.

#### **type**

The type of widget.

Valid Values: `metric` | `text`

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 \(p. 82\)](#) or [Properties of a Text Widget Object \(p. 82\)](#).

Type: Object

Required: Yes

## Properties of a Text Widget Object

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

### **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"
      }
    }
  ]
}
About
```

## Properties of a Metric Widget Object

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

### **metrics**

To include one or more metrics (without alarms) in the widget, specify a `metrics` array. One `metrics` array can include 0–100 metrics. 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 \(p. 85\)](#).

Type: Array of arrays

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

### 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 \(p. 87\)](#). Use this parameter only for metric widgets.

Type: Array

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.

### title

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

Type: String

Required: No

### 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. Use this parameter only for metric widgets.

Type: String

Required: Yes

### 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

### view

Specify `timeSeries` to display this metric as a graph, or `singleValue` to display it as a number. Use this parameter only for metric widgets.

Valid Values: `timeSeries` | `singleValue`

Type: String

Required: No

### 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

### 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 \(p. 92\)](#).

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",
    "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 metric in the `metrics` array has the following format:

```
[Namespace, MetricName, Dimension1Name, Dimension1Value, Dimension2Name, Dimension2Value...  
  {Rendering Properties Object}]
```

### 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

### 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

## 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 \(p. 86\)](#).

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" ]
```

## 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, and multiple horizontal and 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 minimum and maximum values for the Y-axis of the graph. 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 `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

# GetMetricWidgetImage: Metric Widget Structure and Syntax

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

## Contents

- [Overall Structure \(p. 93\)](#)
- [Format for Each Metric in the Array of Metrics \(p. 95\)](#)
- [Annotation Properties Format \(p. 99\)](#)
- [yAxis Properties Format \(p. 102\)](#)

## 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 \(p. 95\)](#).

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 \(p. 99\)](#).

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

#### **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

#### **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 \(p. 102\)](#).

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 \(p. 96\)](#).

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": "#ddddd",
        "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 \(p. 107\)](#).

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

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

## Request Identifiers

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

## Query API Authentication

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

## Available Libraries

AWS provides libraries, sample code, tutorials, and other resources for software developers who prefer to build applications using language-specific APIs instead of the command-line tools and Query API.

These libraries provide basic functions (not included in the APIs), such as request authentication, request retries, and error handling so that it is easier to get started. Libraries and resources are available for the following languages and platforms:

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

For libraries and sample code in all languages, see [Sample Code & Libraries](#).

## 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
<i>host</i>	The Amazon CloudWatch endpoint. For example, <code>monitoring.us-west-1.amazonaws.com</code>
<i>x-amz-date</i>	<p>You must provide the time stamp in either the HTTP Date header or the AWS <i>x-amz-date</i> header. Some HTTP client libraries don't let you set the Date header. When an <i>x-amz-date</i> header is present, the system ignores any Date header during the request authentication.</p> <p>The <i>x-amz-date</i> header must be specified in ISO 8601 basic format. For example: <code>20130315T092054Z</code></p>
<i>Authorization</i>	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 <a href="#">Signature Version 4 Signing Process</a> in the <i>Amazon Web Services General Reference</i> .
<i>x-amz-target</i>	<p>Specifies the CloudWatch operation:</p> <p><code>GraniteServiceVersion20100801.<i>API_Name</i></code></p> <p>For example, for <code>GetMetricData</code> the target value is the following:</p> <p><code>GraniteServiceVersion20100801.GetMetricData</code></p>
<i>Content-Type</i>	Specifies the input format. The valid value is <code>application/json</code>
<i>Accept</i>	Specifies the response format. The valid value is <code>application/json</code>

Header name	Header value
<i>Content-length</i>	Size of the payload in bytes.
<i>Content-Encoding</i>	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'THHMMSS'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