

---

# Amazon CloudWatch

## API Reference

API Version 2010-08-01



## Amazon CloudWatch: API Reference

Copyright © 2014 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

The following are trademarks of Amazon Web Services, Inc.: Amazon, Amazon Web Services Design, AWS, Amazon CloudFront, Cloudfront, CloudTrail, Amazon DevPay, DynamoDB, ElastiCache, Amazon EC2, Amazon Elastic Compute Cloud, Amazon Glacier, Kinesis, Kindle, Kindle Fire, AWS Marketplace Design, Mechanical Turk, Amazon Redshift, Amazon Route 53, Amazon S3, Amazon VPC. In addition, Amazon.com graphics, logos, page headers, button icons, scripts, and service names are trademarks, or trade dress of Amazon in the U.S. and/or other countries. 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 .....	3
DeleteAlarms .....	4
Description .....	4
Request Parameters .....	4
Errors .....	4
DescribeAlarmHistory .....	5
Description .....	5
Request Parameters .....	5
Response Elements .....	6
Errors .....	6
DescribeAlarms .....	7
Description .....	7
Request Parameters .....	7
Response Elements .....	8
Errors .....	8
DescribeAlarmsForMetric .....	9
Description .....	9
Request Parameters .....	9
Response Elements .....	10
DisableAlarmActions .....	11
Description .....	11
Request Parameters .....	11
EnableAlarmActions .....	12
Description .....	12
Request Parameters .....	12
GetMetricStatistics .....	13
Description .....	13
Request Parameters .....	13
Response Elements .....	15
Errors .....	15
ListMetrics .....	16
Description .....	16
Request Parameters .....	16
Response Elements .....	16
Errors .....	17
PutMetricAlarm .....	18
Description .....	18
Request Parameters .....	18
Errors .....	20
PutMetricData .....	21
Description .....	21
Request Parameters .....	21
Errors .....	21
Examples .....	22
SetAlarmState .....	24
Description .....	24
Request Parameters .....	24
Errors .....	24
Data Types .....	26
AlarmHistoryItem .....	26
Description .....	26
Contents .....	27
Datapoint .....	27
Description .....	27

Contents .....	27
DescribeAlarmHistoryResult .....	28
Description .....	28
Contents .....	28
DescribeAlarmsForMetricResult .....	29
Description .....	29
Contents .....	29
DescribeAlarmsResult .....	29
Description .....	29
Contents .....	29
Dimension .....	30
Description .....	30
Contents .....	30
DimensionFilter .....	30
Description .....	30
Contents .....	30
GetMetricStatisticsResult .....	31
Description .....	31
Contents .....	31
ListMetricsResult .....	31
Description .....	31
Contents .....	31
Metric .....	31
Description .....	31
Contents .....	32
MetricAlarm .....	32
Description .....	32
Contents .....	32
MetricDatum .....	35
Description .....	35
Contents .....	35
StatisticSet .....	37
Description .....	37
Contents .....	37
Common Parameters .....	38
.....	38
Common Errors .....	40
.....	40

# Welcome

---

This is the *Amazon CloudWatch API Reference*. This guide provides detailed information about Amazon CloudWatch actions, data types, parameters, and errors. For detailed information about Amazon CloudWatch features and their associated API calls, go to the [Amazon CloudWatch Developer Guide](#).

Amazon CloudWatch is a web service that enables you to publish, monitor, and manage various metrics, as well as configure alarm actions based on data from metrics. For more information about this product go to <http://aws.amazon.com/cloudwatch>.

For information about the namespace, metric names, and dimensions that other Amazon Web Services products use to send metrics to CloudWatch, go to [Amazon CloudWatch Metrics, Namespaces, and Dimensions Reference](#) in the *Amazon CloudWatch Developer Guide*.

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

- [Actions](#): An alphabetical list of all Amazon CloudWatch actions.
- [Data Types](#): An alphabetical list of all Amazon CloudWatch data types.
- [Common Parameters](#): Parameters that all Query actions can use.
- [Common Errors](#): Client and server errors that all actions can return.
- [Regions and Endpoints](#): Itemized regions and endpoints for all AWS products.
- [WSDL Location](http://monitoring.amazonaws.com/doc/2010-08-01/CloudWatch.wsdl): <http://monitoring.amazonaws.com/doc/2010-08-01/CloudWatch.wsdl>

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

- [AWS SDK for Java Documentation](#)
- [AWS SDK for .NET Documentation](#)
- [AWS SDK for PHP Documentation](#)
- [AWS SDK for Ruby Documentation](#)

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

- [AWS Java Developer Center](#)
- [AWS PHP Developer Center](#)
- [AWS Python Developer Center](#)
- [AWS Ruby Developer Center](#)

- [AWS Windows and .NET Developer Center](#)

This document was last updated on September 16, 2014.

# Actions

---

The following actions are supported:

- [DeleteAlarms](#) (p. 4)
- [DescribeAlarmHistory](#) (p. 5)
- [DescribeAlarms](#) (p. 7)
- [DescribeAlarmsForMetric](#) (p. 9)
- [DisableAlarmActions](#) (p. 11)
- [EnableAlarmActions](#) (p. 12)
- [GetMetricStatistics](#) (p. 13)
- [ListMetrics](#) (p. 16)
- [PutMetricAlarm](#) (p. 18)
- [PutMetricData](#) (p. 21)
- [SetAlarmState](#) (p. 24)

# DeleteAlarms

## Description

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

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 38\)](#).

### **AlarmNames.member.N**

A list of alarms to be deleted.

Type: String list

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

Length constraints: Minimum of 0 item(s) in the list. Maximum of 100 item(s) in the list.

Required: Yes

## Errors

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

### **ResourceNotFound**

The named resource does not exist.

HTTP Status Code: 404



# DescribeAlarmHistory

## Description

Retrieves history for the specified alarm. Filter alarms by date range or item type. If an alarm name is not specified, Amazon CloudWatch returns histories for all of the owner's alarms.

### Note

Amazon CloudWatch retains the history of an alarm for two weeks, whether or not you delete the alarm.

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 38\)](#).

### AlarmName

The name of the alarm.

Type: String

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

Required: No

### EndDate

The ending date to retrieve alarm history.

Type: DateTime

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

Required: No

### NextToken

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

Type: String

Required: No

### StartDate

The starting date to retrieve alarm history.

Type: DateTime

Required: No

## Response Elements

The following elements are returned in a structure named `DescribeAlarmHistoryResult`.

### **AlarmHistoryItems**

A list of alarm histories in JSON format.

Type: [AlarmHistoryItem \(p. 26\)](#) list

### **NextToken**

A string that marks the start of the next batch of returned results.

Type: String

## Errors

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

### **InvalidNextToken**

The next token specified is invalid.

HTTP Status Code: 400

# DescribeAlarms

## Description

Retrieves alarms with the specified names. If no name is specified, all alarms for the user 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 common parameters that all actions use, see [Common Parameters \(p. 38\)](#).

### ActionPrefix

The action name prefix.

Type: String

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

Required: No

### AlarmNamePrefix

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

Type: String

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

Required: No

### AlarmNames.member.N

A list of alarm names to retrieve information for.

Type: String list

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

Length constraints: Minimum of 0 item(s) in the list. Maximum of 100 item(s) in the list.

Required: No

### MaxRecords

The maximum number of alarm descriptions to retrieve.

Type: Integer

Required: No

### NextToken

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

Type: String

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 in a structure named `DescribeAlarmsResult`.

### **MetricAlarms**

A list of information for the specified alarms.

Type: [MetricAlarm \(p. 32\)](#) list

### **NextToken**

A string that marks the start of the next batch of returned results.

Type: String

## Errors

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

### **InvalidNextToken**

The next token specified is invalid.

HTTP Status Code: 400

# DescribeAlarmsForMetric

## Description

Retrieves all alarms for a single metric. Specify a statistic, period, or unit to filter the set of alarms further.

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 38\)](#).

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

The list of dimensions associated with the metric.

Type: [Dimension \(p. 30\)](#) list

Length constraints: Minimum of 0 item(s) in the list. Maximum of 10 item(s) in the list.

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.

Required: Yes

### **Period**

The period in seconds over which the statistic is applied.

Type: Integer

Required: No

### **Statistic**

The statistic for the metric.

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` | `Mega-`  
`bytes` | `Gigabytes` | `Terabytes` | `Bits` | `Kilobits` | `Megabits` | `Gigabits` |  
`Terabits` | `Percent` | `Count` | `Bytes/Second` | `Kilobytes/Second` | `Mega-`  
`bytes/Second` | `Gigabytes/Second` | `Terabytes/Second` | `Bits/Second` | `Kilob-`

its/Second | Megabits/Second | Gigabits/Second | Terabits/Second |  
Count/Second | None

Required: No

## Response Elements

The following element is returned in a structure named `DescribeAlarmsForMetricResult`.

### **MetricAlarms**

A list of information for each alarm with the specified metric.

Type: [MetricAlarm](#) (p. 32) list

# DisableAlarmActions

## Description

Disables actions for the specified alarms. When an alarm's actions are disabled the alarm's state may change, but none of the alarm's actions will execute.

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 38\)](#).

### **AlarmNames.member.N**

The names of the alarms to disable actions for.

Type: String list

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

Length constraints: Minimum of 0 item(s) in the list. Maximum of 100 item(s) in the list.

Required: Yes

# EnableAlarmActions

## Description

Enables actions for the specified alarms.

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 38\)](#).

### **AlarmNames.member.N**

The names of the alarms to enable actions for.

Type: String list

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

Length constraints: Minimum of 0 item(s) in the list. Maximum of 100 item(s) in the list.

Required: Yes



# GetMetricStatistics

## Description

Gets statistics for the specified metric.

The maximum number of data points that can be queried is 50,850, whereas the maximum number of data points returned from a single `GetMetricStatistics` request is 1,440. If you make a request that generates more than 1,440 data points, Amazon CloudWatch returns an error. In such a case, you can alter the request by narrowing the specified time range or increasing the specified period. Alternatively, you can make multiple requests across adjacent time ranges. `GetMetricStatistics` does not return the data in chronological order.

Amazon CloudWatch aggregates data points based on the length of the `period` that you specify. For example, if you request statistics with a one-minute granularity, Amazon CloudWatch aggregates data points with time stamps that fall within the same one-minute period. In such a case, the data points queried can greatly outnumber the data points returned.

The following examples show various statistics allowed by the data point query maximum of 50,850 when you call `GetMetricStatistics` on Amazon EC2 instances with detailed (one-minute) monitoring enabled:

- Statistics for up to 400 instances for a span of one hour
- Statistics for up to 35 instances over a span of 24 hours
- Statistics for up to 2 instances over a span of 2 weeks

For information about the namespace, metric names, and dimensions that other Amazon Web Services products use to send metrics to CloudWatch, go to [Amazon CloudWatch Metrics, Namespaces, and Dimensions Reference](#) in the *Amazon CloudWatch Developer Guide*.

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 38\)](#).

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

A list of dimensions describing qualities of the metric.

Type: [Dimension \(p. 30\)](#) list

Length constraints: Minimum of 0 item(s) in the list. Maximum of 10 item(s) in the list.

Required: No

### **EndTime**

The time stamp to use for determining the last datapoint to return. The value specified is exclusive; results will include datapoints up to the time stamp specified. The time stamp must be in ISO 8601 UTC format (e.g., 2014-09-03T23:00:00Z).

Type: `DateTime`

Required: Yes

### **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.

Required: Yes

**Period**

The granularity, in seconds, of the returned datapoints. `Period` must be at least 60 seconds and must be a multiple of 60. The default value is 60.

Type: Integer

Required: Yes

**StartTime**

The time stamp to use for determining the first datapoint to return. The value specified is inclusive; results include datapoints with the time stamp specified. The time stamp must be in ISO 8601 UTC format (e.g., 2014-09-03T23:00:00Z).

**Note**

The specified start time is rounded down to the nearest value. Datapoints are returned for start times up to two weeks in the past. Specified start times that are more than two weeks in the past will not return datapoints for metrics that are older than two weeks.

Data that is timestamped 24 hours or more in the past may take in excess of 48 hours to become available from submission time using `GetMetricStatistics`.

Type: DateTime

Required: Yes

**Statistics.member.N**

The metric statistics to return. For information about specific statistics returned by `GetMetricStatistics`, see [Statistics](#) in the *Amazon CloudWatch Developer Guide*.

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

Type: String list

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

Length constraints: Minimum of 1 item(s) in the list. Maximum of 5 item(s) in the list.

Required: Yes

**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 elements are returned in a structure named `GetMetricStatisticsResult`.

### **Datapoints**

The datapoints for the specified metric.

Type: [Datapoint \(p. 27\)](#) list

### **Label**

A label describing the specified metric.

Type: String

## Errors

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

### **InternalServerError**

Indicates that the request processing has failed due to some unknown error, exception, or failure.

HTTP Status Code: 500

### **InvalidParameterCombination**

Parameters that must not be used together were used together.

HTTP Status Code: 400

### **InvalidParameterValue**

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

HTTP Status Code: 400

### **MissingRequiredParameter**

An input parameter that is mandatory for processing the request is not supplied.

HTTP Status Code: 400

# ListMetrics

## Description

Returns a list of valid metrics stored for the AWS account owner. Returned metrics can be used with [GetMetricStatistics](#) (p. 13) to obtain statistical data for a given metric.

**Note**

Up to 500 results are returned for any one call. To retrieve further results, use returned `NextToken` values with subsequent `ListMetrics` operations.

**Note**

If you create a metric with the [PutMetricData](#) (p. 21) action, allow up to fifteen minutes for the metric to appear in calls to the `ListMetrics` action. Statistics about the metric, however, are available sooner using [GetMetricStatistics](#) (p. 13).

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 38).

**Dimensions.member.N**

A list of dimensions to filter against.

Type: [DimensionFilter](#) (p. 30) list

Length constraints: Minimum of 0 item(s) in the list. Maximum of 10 item(s) in the list.

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.

Required: No

**NextToken**

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

Type: String

Required: No

## Response Elements

The following elements are returned in a structure named `ListMetricsResult`.

**Metrics**

A list of metrics used to generate statistics for an AWS account.

Type: [Metric \(p. 31\)](#) list

**NextToken**

A string that marks the start of the next batch of returned results.

Type: String

## Errors

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

**InternalServerError**

Indicates that the request processing has failed due to some unknown error, exception, or failure.

HTTP Status Code: 500

**InvalidParameterValue**

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

HTTP Status Code: 400

# PutMetricAlarm

## Description

Creates or updates an alarm and associates it with the specified Amazon CloudWatch metric. Optionally, this operation can associate one or more Amazon Simple Notification Service 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 `StateValue` is set appropriately. Any actions associated with the `StateValue` is then executed.

### Note

When updating an existing alarm, its `StateValue` is left unchanged.

### Note

If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions: `ec2:DescribeInstanceStatus`, `ec2:DescribeInstances`, `ec2:StopInstances`, and `ec2:TerminateInstances` in order for the alarm action to be performed. If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm but the stop or terminate actions won't be performed on the Amazon EC2 instance. However, if you are later granted permission to use the associated Amazon EC2 APIs, the alarm actions you created earlier will be performed. For more information about IAM permissions, see [Permissions and Policies](#) in .

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

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

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 38).

### ActionsEnabled

Indicates whether or not actions should be executed during any changes to the alarm's state.

Type: Boolean

Required: No

### AlarmActions.member.N

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

Type: String list

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

Length constraints: Minimum of 0 item(s) in the list. Maximum of 5 item(s) in the list.

Required: No

### AlarmDescription

The description for the alarm.

Type: String

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

Required: No

**AlarmName**

The descriptive name for the alarm. This name must be unique within the user's AWS account

Type: String

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

Required: Yes

**ComparisonOperator**

The arithmetic operation to use when comparing the specified `Statistic` and `Threshold`. The specified `Statistic` value is used as the first operand.

Type: String

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

Required: Yes

**Dimensions.member.N**

The dimensions for the alarm's associated metric.

Type: [Dimension \(p. 30\)](#) list

Length constraints: Minimum of 0 item(s) in the list. Maximum of 10 item(s) in the list.

Required: No

**EvaluationPeriods**

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

Type: Integer

Required: Yes

**InsufficientDataActions.member.N**

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

Type: String list

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

Length constraints: Minimum of 0 item(s) in the list. Maximum of 5 item(s) in the list.

Required: No

**MetricName**

The name for the alarm's associated metric.

Type: String

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

Required: Yes

**Namespace**

The namespace for the alarm's associated metric.

Type: String

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

Required: Yes

**OKActions.member.N**

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

Type: String list

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

Length constraints: Minimum of 0 item(s) in the list. Maximum of 5 item(s) in the list.

Required: No

**Period**

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

Type: Integer

Required: Yes

**Statistic**

The statistic to apply to the alarm's associated metric.

Type: String

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

Required: Yes

**Threshold**

The value against which the specified statistic is compared.

Type: Double

Required: Yes

**Unit**

The unit for the alarm's associated metric.

Type: String

Valid Values: `Seconds` | `Microseconds` | `Milliseconds` | `Bytes` | `Kilobytes` | `Mega-`  
`bytes` | `Gigabytes` | `Terabytes` | `Bits` | `Kilobits` | `Megabits` | `Gigabits` |  
`Terabits` | `Percent` | `Count` | `Bytes/Second` | `Kilobytes/Second` | `Mega-`  
`bytes/Second` | `Gigabytes/Second` | `Terabytes/Second` | `Bits/Second` | `Kilob-`  
`its/Second` | `Megabits/Second` | `Gigabits/Second` | `Terabits/Second` |  
`Count/Second` | `None`

Required: No

## Errors

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

**LimitExceeded**

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

HTTP Status Code: 400



# PutMetricData

## Description

Publishes metric data points to Amazon CloudWatch. Amazon CloudWatch associates the data points with the specified metric. If the specified metric does not exist, Amazon CloudWatch creates the metric. When Amazon CloudWatch creates a metric, it can take up to fifteen minutes for the metric to appear in calls to the [ListMetrics](#) (p. 16) action.

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

### Important

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

Data that is timestamped 24 hours or more in the past may take in excess of 48 hours to become available from submission time using `GetMetricStatistics`.

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 38).

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

A list of data describing the metric.

Type: [MetricDatum](#) (p. 35) list

Required: Yes

### **Namespace**

The namespace for the metric data.

#### **Note**

You cannot specify a namespace that begins with "AWS/". Namespaces that begin with "AWS/" are reserved for other Amazon Web Services products that send metrics to Amazon CloudWatch.

Type: String

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

Required: Yes

## Errors

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

### **InternalServerError**

Indicates that the request processing has failed due to some unknown error, exception, or failure.

HTTP Status Code: 500

### **InvalidParameterCombination**

Parameters that must not be used together were used together.

HTTP Status Code: 400

**InvalidParameterValue**

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

HTTP Status Code: 400

**MissingRequiredParameter**

An input parameter that is mandatory for processing the request is not supplied.

HTTP Status Code: 400

## Examples

### Sample Request

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

```
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
```

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

```
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
```

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

```
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
```

## Sample Response

```
<PutMetricDataResponse xmlns="http://monitoring.&api-domain;/doc/2010-08-01/">
  <ResponseMetadata>
    <RequestId>e16fc4d3-9a04-11e0-9362-093a1cae5385</RequestId>
  </ResponseMetadata>
</PutMetricDataResponse>
```

# SetAlarmState

## Description

Temporarily sets the state of an alarm. When the updated `StateValue` differs from the previous value, the action configured for the appropriate state is invoked. This is not a permanent change. The next periodic alarm check (in about a minute) will set the alarm to its actual state.

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 38\)](#).

### AlarmName

The descriptive name for the alarm. This name must be unique within the user's 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 human-readable 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 machine-readable JSON format)

Type: String

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

Required: No

### StateValue

The value of the state.

Type: String

Valid Values: OK | ALARM | INSUFFICIENT\_DATA

Required: Yes

## Errors

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

### InvalidFormat

Data was not syntactically valid JSON.

HTTP Status Code: 400

**ResourceNotFound**

The named resource does not exist.

HTTP Status Code: 404

# Data Types

---

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

**Note**

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

The following data types are supported:

- [AlarmHistoryItem](#) (p. 26)
- [Datapoint](#) (p. 27)
- [DescribeAlarmHistoryResult](#) (p. 28)
- [DescribeAlarmsForMetricResult](#) (p. 29)
- [DescribeAlarmsResult](#) (p. 29)
- [Dimension](#) (p. 30)
- [DimensionFilter](#) (p. 30)
- [GetMetricStatisticsResult](#) (p. 31)
- [ListMetricsResult](#) (p. 31)
- [Metric](#) (p. 31)
- [MetricAlarm](#) (p. 32)
- [MetricDatum](#) (p. 35)
- [StatisticSet](#) (p. 37)

## AlarmHistoryItem

### Description

The `AlarmHistoryItem` data type contains descriptive information about the history of a specific alarm. If you call [DescribeAlarmHistory](#) (p. 5), Amazon CloudWatch returns this data type as part of the [DescribeAlarmHistoryResult](#) (p. 28) data type.

## Contents

### **AlarmName**

The descriptive name for the alarm.

Type: String

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

Required: No

### **HistoryData**

Machine-readable 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 human-readable summary of the alarm history.

Type: String

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

Required: No

### **Timestamp**

The time stamp for the alarm history item.

Type: DateTime

Required: No

## Datapoint

### Description

The `Datapoint` data type encapsulates the statistical data that Amazon CloudWatch computes from metric data.

### Contents

#### **Average**

The average of metric values that correspond to the datapoint.

Type: Double

Required: No

**Maximum**

The maximum of the metric value used for the datapoint.

Type: Double

Required: No

**Minimum**

The minimum metric value used for the datapoint.

Type: Double

Required: No

**SampleCount**

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

Type: Double

Required: No

**Sum**

The sum of metric values used for the datapoint.

Type: Double

Required: No

**Timestamp**

The time stamp used for the datapoint.

Type: DateTime

Required: No

**Unit**

The standard unit used for the datapoint.

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

## DescribeAlarmHistoryResult

### Description

The output for the [DescribeAlarmHistory](#) (p. 5) action.

### Contents

**AlarmHistoryItems**

A list of alarm histories in JSON format.



Type: [AlarmHistoryItem](#) (p. 26) list

Required: No

**NextToken**

A string that marks the start of the next batch of returned results.

Type: String

Required: No

## DescribeAlarmsForMetricResult

### Description

The output for the [DescribeAlarmsForMetric](#) (p. 9) action.

### Contents

**MetricAlarms**

A list of information for each alarm with the specified metric.

Type: [MetricAlarm](#) (p. 32) list

Required: No

## DescribeAlarmsResult

### Description

The output for the [DescribeAlarms](#) (p. 7) action.

### Contents

**MetricAlarms**

A list of information for the specified alarms.

Type: [MetricAlarm](#) (p. 32) list

Required: No

**NextToken**

A string that marks the start of the next batch of returned results.

Type: String

Required: No

## Dimension

### Description

The `Dimension` data type further expands on the identity of a metric using a Name, Value pair.

For examples that use one or more dimensions, see [PutMetricData \(p. 21\)](#).

### Contents

#### Name

The name of the dimension.

Type: String

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

Required: Yes

#### Value

The value representing the dimension measurement

Type: String

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

Required: Yes

## DimensionFilter

### Description

The `DimensionFilter` data type is used to filter [ListMetrics \(p. 16\)](#) results.

### 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.

#### Note

Specifying a `Name` without specifying a `Value` returns all values associated with that `Name`.

Type: String

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

Required: No

# GetMetricStatisticsResult

## Description

The output for the [GetMetricStatistics](#) (p. 13) action.

## Contents

### Datapoints

The datapoints for the specified metric.

Type: [Datapoint](#) (p. 27) list

Required: No

### Label

A label describing the specified metric.

Type: String

Required: No

# ListMetricsResult

## Description

The output for the [ListMetrics](#) (p. 16) action.

## Contents

### Metrics

A list of metrics used to generate statistics for an AWS account.

Type: [Metric](#) (p. 31) list

Required: No

### NextToken

A string that marks the start of the next batch of returned results.

Type: String

Required: No

# Metric

## Description

The `Metric` data type contains information about a specific metric. If you call [ListMetrics](#) (p. 16), Amazon CloudWatch returns information contained by this data type.

The example in the Examples section publishes two metrics named buffers and latency. Both metrics are in the examples namespace. Both metrics have two dimensions, InstanceID and InstanceType.

## Contents

### Dimensions

A list of dimensions associated with the metric.

Type: [Dimension \(p. 30\)](#) list

Length constraints: Minimum of 0 item(s) in the list. Maximum of 10 item(s) in the list.

Required: No

### MetricName

The name of the metric.

Type: String

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

Required: No

### Namespace

The namespace of the metric.

Type: String

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

Required: No

## MetricAlarm

### Description

The [MetricAlarm \(p. 32\)](#) data type represents an alarm. You can use [PutMetricAlarm \(p. 18\)](#) to create or update an alarm.

## Contents

### ActionsEnabled

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

Type: Boolean

Required: No

### AlarmActions

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

Type: String list

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

Length constraints: Minimum of 0 item(s) in the list. Maximum of 5 item(s) in the list.

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: DateTime

Required: No

**AlarmDescription**

The description for the alarm.

Type: String

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

Required: No

**AlarmName**

The name of the alarm.

Type: String

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

Required: No

**ComparisonOperator**

The arithmetic operation to use when comparing the specified `Statistic` and `Threshold`. The specified `Statistic` value is used as the first operand.

Type: String

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

Required: No

**Dimensions**

The list of dimensions associated with the alarm's associated metric.

Type: [Dimension \(p. 30\)](#) list

Length constraints: Minimum of 0 item(s) in the list. Maximum of 10 item(s) in the list.

Required: No

**EvaluationPeriods**

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

Type: Integer

Required: No

**InsufficientDataActions**

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

**Important**

The current WSDL lists this attribute as `UnknownActions`.

Type: String list

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

Length constraints: Minimum of 0 item(s) in the list. Maximum of 5 item(s) in the list.

Required: No

**MetricName**

The name of the alarm's metric.

Type: String

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

Required: No

**Namespace**

The namespace of alarm's associated metric.

Type: String

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

Required: No

**OKActions**

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

Type: String list

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

Length constraints: Minimum of 0 item(s) in the list. Maximum of 5 item(s) in the list.

Required: No

**Period**

The period in seconds over which the statistic is applied.

Type: Integer

Required: No

**StateReason**

A human-readable explanation for the alarm's state.

Type: String

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

Required: No

**StateReasonData**

An explanation for the alarm's state in machine-readable 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's state.

Type: DateTime

Required: No

**StateValue**

The state value for the alarm.

Type: String

Valid Values: OK | ALARM | INSUFFICIENT\_DATA

Required: No

**Statistic**

The statistic to apply to the alarm's associated metric.

Type: String

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

Required: No

**Threshold**

The value against which the specified statistic is compared.

Type: Double

Required: No

**Unit**

The unit of the alarm's associated 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

## MetricDatum

### Description

The `MetricDatum` data type encapsulates the information sent with [PutMetricData \(p. 21\)](#) to either create a new metric or add new values to be aggregated into an existing metric.

### Contents

**Dimensions**

A list of dimensions associated with the metric. Note, when using the `Dimensions` value in a query, you need to append `.member.N` to it (e.g., `Dimensions.member.N`).

Type: [Dimension \(p. 30\)](#) list

Length constraints: Minimum of 0 item(s) in the list. Maximum of 10 item(s) in the list.

Required: No

**MetricName**

The name of the metric.

Type: String

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

Required: Yes

**StatisticValues**

A set of statistical values describing the metric.

Type: [StatisticSet \(p. 37\)](#)

Required: No

**Timestamp**

The time stamp used for the metric in ISO 8601 Universal Coordinated Time (UTC) format. If not specified, the default value is set to the time the metric data was received.

Type: DateTime

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.

**Important**

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

Type: Double

Required: No



# StatisticSet

## Description

The `StatisticSet` data type describes the `StatisticValues` component of [MetricDatum \(p. 35\)](#), and 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

# Common Parameters

---

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

**Action**

The action to be performed.

Default: None

Type: string

Required: Yes

**AuthParams**

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

- `AWSSessionToken`
- `AWSSessionTokenDuration`
- `AWSSessionTokenExpiration`
- `AWSSessionTokenSecretKey`
- `AWSSessionTokenSecretKeyID`
- `SignatureVersion`
- `Timestamp`
- `Signature`

Default: None

Required: Conditional

**AWSSessionToken**

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

Default: None

Type: string

Required: Yes

**Expires**

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

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

Default: None

Type: string

Required: Conditional

**SecurityToken**

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

Default: None

Type: string

Required: No

**Signature**

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

Default: None

Type: string

Required: Yes

**SignatureMethod**

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

Default: None

Type: string

Valid Values: HmacSHA256 | HmacSHA1

Required: Yes

**SignatureVersion**

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

Default: None

Type: string

Required: Yes

**Timestamp**

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

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

Default: None

Type: string

Required: Conditional

**Version**

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

Default: None

Type: string

Required: Yes

# Common Errors

---

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

**IncompleteSignature**

The request signature does not conform to AWS standards.

HTTP Status Code: 400

**InternalFailure**

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

HTTP Status Code: 500

**InvalidAction**

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

HTTP Status Code: 400

**InvalidClientTokenId**

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

HTTP Status Code: 403

**InvalidParameterCombination**

Parameters that must not be used together were used together.

HTTP Status Code: 400

**InvalidParameterValue**

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

HTTP Status Code: 400

**InvalidQueryParameter**

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

HTTP Status Code: 400

**MalformedQueryString**

The query string contains a syntax error.

HTTP Status Code: 404

**MissingAction**

The request is missing an action or a required parameter.

HTTP Status Code: 400

**MissingAuthenticationToken**

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

HTTP Status Code: 403

**MissingParameter**

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

**OptInRequired**

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

HTTP Status Code: 403

**RequestExpired**

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

HTTP Status Code: 400

**ServiceUnavailable**

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

HTTP Status Code: 503

**Throttling**

The request was denied due to request throttling.

HTTP Status Code: 400

**ValidationError**

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

HTTP Status Code: 400