API Reference

Internet Monitor

API Version 2021-06-03

Internet Monitor: API Reference

Table of Contents

Welcome

Internet Monitor is a feature of Amazon CloudWatch Network Monitoring that provides visibility into how internet issues impact performance and availability between your applications hosted on AWS and your end users. Internet Monitor uses the connectivity data that AWS captures from its global networking footprint to calculate a baseline of performance and availability for internet traffic. This is the same data that AWS uses to monitor internet uptime and availability. With those measurements as a baseline, Internet Monitor raises awareness for you when there are significant problems for your end users in the different geographic locations where your application runs.

Internet Monitor publishes internet measurements to CloudWatch Logs and CloudWatch Metrics,to easily support using CloudWatch tools with health information for geographies and networks specific to your application. Internet Monitor sends health events to Amazon EventBridge so that you can set up notifications. If an overall issue is caused by the AWS network, you also automatically receive an AWS Health Dashboard notification with the steps that AWS is taking to mitigate the problem.

To use Internet Monitor, you create a *monitor* and associate your application's resources with it - VPCs, NLBs, CloudFront distributions, or WorkSpaces directories - so Internet Monitor can determine where your application's internet traffic is. Internet Monitor then provides internet measurements from AWS that are specific to the locations and ASNs (typically, internet service providers or ISPs) that communicate with your application.

For more information, see Using Internet Monitor in the *Amazon CloudWatch User Guide*.

This document was last published on September 4, 2025.

Actions

The following actions are supported:

- CreateMonitor
- DeleteMonitor
- GetHealthEvent
- GetInternetEvent
- GetMonitor
- GetQueryResults
- GetQueryStatus
- ListHealthEvents
- ListInternetEvents
- ListMonitors
- ListTagsForResource
- StartQuery
- StopQuery
- TagResource
- UntagResource
- UpdateMonitor

CreateMonitor

Creates a monitor in Internet Monitor. A monitor is built based on information from the application resources that you add: VPCs, Network Load Balancers (NLBs), Amazon CloudFront distributions, and Amazon WorkSpaces directories. Internet Monitor then publishes internet measurements from AWS that are specific to the *city-networks*. That is, the locations and ASNs (typically internet service providers or ISPs), where clients access your application. For more information, see Using Internet Monitor in the *Amazon CloudWatch User Guide*.

When you create a monitor, you choose the percentage of traffic that you want to monitor. You can also set a maximum limit for the number of city-networks where client traffic is monitored, that caps the total traffic that Internet Monitor monitors. A city-network maximum is the limit of city-networks, but you only pay for the number of city-networks that are actually monitored. You can update your monitor at any time to change the percentage of traffic to monitor or the city-networks maximum. For more information, see Choosing a city-network maximum value in the *Amazon CloudWatch User Guide*.

Request Syntax

```
POST /v20210603/Monitors HTTP/1.1
Content-type: application/json

{
   "ClientToken": "string",
   "HealthEventsConfig": {
      "AvailabilityLocalHealthEventsConfig": {
         "HealthScoreThreshold": number,
         "MinTrafficImpact": number,
         "Status": "string"
      },
      "AvailabilityScoreThreshold": number,
      "PerformanceLocalHealthEventsConfig": {
         "HealthScoreThreshold": number,
         "MinTrafficImpact": number,
         "Status": "string"
      },
      "PerformanceScoreThreshold": number
   },
   "InternetMeasurementsLogDelivery": {
      "S3Config": {
         "BucketName": "string",
         "BucketPrefix": "string",
         "LogDeliveryStatus": "string"
      }
   },
   "MaxCityNetworksToMonitor": number,
   "MonitorName": "string",
   "Resources": [ "string" ],
   "Tags": {
      "string" : "string"
   },
   "TrafficPercentageToMonitor": number
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

ClientToken

A unique, case-sensitive string of up to 64 ASCII characters that you specify to make an idempotent API request. Don't reuse the same client token for other API requests.

Type: String

Required: No

HealthEventsConfig

Defines the threshold percentages and other configuration information for when Internet Monitor creates a health event. Internet Monitor creates a health event when an internet issue that affects your application end users has a health score percentage that is at or below a specific threshold, and, sometimes, when other criteria are met.

If you don't set a health event threshold, the default value is 95%.

For more information, see Change health event thresholds in the Internet Monitor section of the *CloudWatch User Guide*.

Type: HealthEventsConfig object

Required: No

InternetMeasurementsLogDelivery

Publish internet measurements for Internet Monitor to an Amazon S3 bucket in addition to CloudWatch Logs.

Type: InternetMeasurementsLogDelivery object

Required: No

MaxCityNetworksToMonitor

The maximum number of city-networks to monitor for your resources. A city-network is the location (city) where clients access your application resources from and the ASN or network provider, such as an internet service provider (ISP), that clients access the resources through. Setting this limit can help control billing costs.

To learn more, see Choosing a city-network maximum value in the Internet Monitor section of the *CloudWatch User Guide*.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 500000.

Required: No

MonitorName

The name of the monitor.

Type: String

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

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

Resources

The resources to include in a monitor, which you provide as a set of Amazon Resource Names (ARNs). Resources can be VPCs, NLBs, Amazon CloudFront distributions, or Amazon WorkSpaces directories.

You can add a combination of VPCs and CloudFront distributions, or you can add WorkSpaces directories, or you can add NLBs. You can't add NLBs or WorkSpaces directories together with any other resources.

> **Note**
>
> If you add only Amazon VPC resources, at least one VPC must have an Internet Gateway attached to it, to make sure that it has internet connectivity.

Type: Array of strings

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `arn:.*`

Required: No

Tags

The tags for a monitor. You can add a maximum of 50 tags in Internet Monitor.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

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

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

Required: No

TrafficPercentageToMonitor

The percentage of the internet-facing traffic for your application that you want to monitor with this monitor. If you set a city-networks maximum, that limit overrides the traffic percentage that you set.

To learn more, see Choosing an application traffic percentage to monitor in the Internet Monitor section of the *CloudWatch User Guide*.

Type: Integer

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

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "Arn": "string",
   "Status": "string"
}
```

Response Elements

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

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

Arn

The Amazon Resource Name (ARN) of the monitor.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 512.

Pattern: `arn:.*`

Status

The status of a monitor.

Type: String

Valid Values: `PENDING | ACTIVE | INACTIVE | ERROR`

Errors

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

AccessDeniedException

You don't have sufficient permission to perform this action.

HTTP Status Code: 403

ConflictException

The requested resource is in use.

HTTP Status Code: 409

InternalServerException

An internal error occurred.

HTTP Status Code: 500

LimitExceededException

The request exceeded a service quota.

HTTP Status Code: 403

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 429

ValidationException

Invalid request.

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 v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteMonitor

Deletes a monitor in Internet Monitor.

Request Syntax

```
DELETE /v20210603/Monitors/MonitorName HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

MonitorName

The name of the monitor to delete.

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

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
```

Response Elements

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

Errors

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

AccessDeniedException

You don't have sufficient permission to perform this action.

HTTP Status Code: 403

InternalServerException

An internal error occurred.

HTTP Status Code: 500

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 429

ValidationException

Invalid request.

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 v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetHealthEvent

Gets information that Internet Monitor has created and stored about a health event for a specified monitor. This information includes the impacted locations, and all the information related to the event, by location.

The information returned includes the impact on performance, availability, and round-trip time, information about the network providers (ASNs), the event type, and so on.

Information rolled up at the global traffic level is also returned, including the impact type and total traffic impact.

Request Syntax

```
GET /v20210603/Monitors/MonitorName/HealthEvents/EventId?
LinkedAccountId=LinkedAccountId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

EventId

The internally-generated identifier of a health event. Because `EventID` contains the forward slash ("/") character, you must URL-encode the `EventID` field in the request URL.

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

Pattern: `[a-zA-Z0-9/_.-]+`

Required: Yes

LinkedAccountId

The account ID for an account that you've set up cross-account sharing for in Internet Monitor. You configure cross-account sharing by using Amazon CloudWatch Observability Access Manager. For more information, see Internet Monitor cross-account observability in the Internet Monitor User Guide.

Length Constraints: Fixed length of 12.

MonitorName

The name of the monitor.

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

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "CreatedAt": "string",
   "EndedAt": "string",
   "EventArn": "string",
   "EventId": "string",
   "HealthScoreThreshold": number,
   "ImpactedLocations": [
      {
         "ASName": "string",
         "ASNumber": number,
         "CausedBy": {
            "AsPath": [
               {
                  "ASName": "string",
                  "ASNumber": number
               }
            ],
            "NetworkEventType": "string",
            "Networks": [
               {
                  "ASName": "string",
                  "ASNumber": number
               }
            ]
         },
         "City": "string",
         "Country": "string",
         "CountryCode": "string",
         "InternetHealth": {
            "Availability": {
               "ExperienceScore": number,
               "PercentOfClientLocationImpacted": number,
               "PercentOfTotalTrafficImpacted": number
            },
            "Performance": {
               "ExperienceScore": number,
               "PercentOfClientLocationImpacted": number,
               "PercentOfTotalTrafficImpacted": number,
               "RoundTripTime": {
                  "P50": number,
                  "P90": number,
                  "P95": number
               }
            }
         },
         "Ipv4Prefixes": [ "string" ],
         "Latitude": number,
         "Longitude": number,
         "Metro": "string",
         "ServiceLocation": "string",
         "Status": "string",
         "Subdivision": "string",
         "SubdivisionCode": "string"
      }
   ],
   "ImpactType": "string",
   "LastUpdatedAt": "string",
   "PercentOfTotalTrafficImpacted": number,
   "StartedAt": "string",
   "Status": "string"
}
```

Response Elements

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

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

CreatedAt

The time when a health event was created.

Type: Timestamp

EndedAt

The time when a health event was resolved. If the health event is still active, the end time is not set.

Type: Timestamp

EventArn

The Amazon Resource Name (ARN) of the event.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `arn:.*`

EventId

The internally-generated identifier of a health event.

Type: String

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

Pattern: `[a-zA-Z0-9/_.-]+`

HealthScoreThreshold

The threshold percentage for a health score that determines, along with other configuration information, when Internet Monitor creates a health event when there's an internet issue that affects your application end users.

Type: Double

Valid Range: Minimum value of 0. Maximum value of 100.

ImpactedLocations

The locations affected by a health event.

Type: Array of ImpactedLocation objects

ImpactType

The type of impairment of a specific health event.

Type: String

Valid Values: `AVAILABILITY | PERFORMANCE | LOCAL_AVAILABILITY | LOCAL_PERFORMANCE`

LastUpdatedAt

The time when a health event was last updated or recalculated.

Type: Timestamp

PercentOfTotalTrafficImpacted

The impact on total traffic that a health event has, in increased latency or reduced availability. This is the percentage of how much latency has increased or availability has decreased during the event, compared to what is typical for traffic from this client location to the AWS location using this client network.

Type: Double

StartedAt

The time when a health event started.

Type: Timestamp

Status

The status of a health event.

Type: String

Valid Values: `ACTIVE | RESOLVED`

Errors

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

AccessDeniedException

You don't have sufficient permission to perform this action.

HTTP Status Code: 403

InternalServerException

An internal error occurred.

HTTP Status Code: 500

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 429

ValidationException

Invalid request.

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 v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetInternetEvent

Gets information that Internet Monitor has generated about an internet event. Internet Monitor displays information about recent global health events, called internet events, on a global outages map that is available to all AWS customers.

The information returned here includes the impacted location, when the event started and (if the event is over) ended, the type of event (`PERFORMANCE` or `AVAILABILITY`), and the status (`ACTIVE` or `RESOLVED`).

Request Syntax

```
GET /v20210603/InternetEvents/EventId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

EventId

The `EventId` of the internet event to return information for.

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

Pattern: `[a-zA-Z0-9-]+`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "ClientLocation": {
      "ASName": "string",
      "ASNumber": number,
      "City": "string",
      "Country": "string",
      "Latitude": number,
      "Longitude": number,
      "Metro": "string",
      "Subdivision": "string"
   },
   "EndedAt": "string",
   "EventArn": "string",
   "EventId": "string",
   "EventStatus": "string",
   "EventType": "string",
   "StartedAt": "string"
}
```

Response Elements

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

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

ClientLocation

The impacted location, such as a city, where clients access AWS application resources.

Type: ClientLocation object

EndedAt

The time when the internet event ended. If the event hasn't ended yet, this value is empty.

Type: Timestamp

EventArn

The Amazon Resource Name (ARN) of the internet event.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `arn:.*`

EventId

The internally-generated identifier of an internet event.

Type: String

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

Pattern: `[a-zA-Z0-9-]+`

EventStatus

The status of the internet event.

Type: String

Valid Values: `ACTIVE | RESOLVED`

EventType

The type of network impairment.

Type: String

Valid Values: `AVAILABILITY | PERFORMANCE`

StartedAt

The time when the internet event started.

Type: Timestamp

Errors

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

AccessDeniedException

You don't have sufficient permission to perform this action.

HTTP Status Code: 403

InternalServerException

An internal error occurred.

HTTP Status Code: 500

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 429

ValidationException

Invalid request.

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 v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetMonitor

Gets information about a monitor in Internet Monitor based on a monitor name. The information returned includes the Amazon Resource Name (ARN), create time, modified time, resources included in the monitor, and status information.

Request Syntax

```
GET /v20210603/Monitors/MonitorName?LinkedAccountId=LinkedAccountId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

LinkedAccountId

The account ID for an account that you've set up cross-account sharing for in Internet Monitor. You configure cross-account sharing by using Amazon CloudWatch Observability Access Manager. For more information, see Internet Monitor cross-account observability in the Internet Monitor User Guide.

Length Constraints: Fixed length of 12.

MonitorName

The name of the monitor.

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

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json
```

```
{
   "CreatedAt": "string",
   "HealthEventsConfig": {
      "AvailabilityLocalHealthEventsConfig": {
         "HealthScoreThreshold": number,
         "MinTrafficImpact": number,
         "Status": "string"
      },
      "AvailabilityScoreThreshold": number,
      "PerformanceLocalHealthEventsConfig": {
         "HealthScoreThreshold": number,
         "MinTrafficImpact": number,
         "Status": "string"
      },
      "PerformanceScoreThreshold": number
   },
   "InternetMeasurementsLogDelivery": {
      "S3Config": {
         "BucketName": "string",
         "BucketPrefix": "string",
         "LogDeliveryStatus": "string"
      }
   },
   "MaxCityNetworksToMonitor": number,
   "ModifiedAt": "string",
   "MonitorArn": "string",
   "MonitorName": "string",
   "ProcessingStatus": "string",
   "ProcessingStatusInfo": "string",
   "Resources": [ "string" ],
   "Status": "string",
   "Tags": {
      "string" : "string"
   },
   "TrafficPercentageToMonitor": number
}
```

Response Elements

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

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

CreatedAt

The time when the monitor was created.

Type: Timestamp

HealthEventsConfig

The list of health event threshold configurations. The threshold percentage for a health score determines, along with other configuration information, when Internet Monitor creates a health event when there's an internet issue that affects your application end users.

For more information, see Change health event thresholds in the Internet Monitor section of the *CloudWatch User Guide*.

Type: HealthEventsConfig object

InternetMeasurementsLogDelivery

Publish internet measurements for Internet Monitor to another location, such as an Amazon S3 bucket. The measurements are also published to Amazon CloudWatch Logs.

Type: InternetMeasurementsLogDelivery object

MaxCityNetworksToMonitor

The maximum number of city-networks to monitor for your resources. A city-network is the location (city) where clients access your application resources from and the ASN or network provider, such as an internet service provider (ISP), that clients access the resources through. This limit can help control billing costs.

To learn more, see Choosing a city-network maximum value in the Internet Monitor section of the *CloudWatch User Guide*.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 500000.

ModifiedAt

The last time that the monitor was modified.

Type: Timestamp

MonitorArn

The Amazon Resource Name (ARN) of the monitor.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 512.

Pattern: `arn:.*`

MonitorName

The name of the monitor.

Type: String

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

Pattern: `[a-zA-Z0-9_.-]+`

ProcessingStatus

The health of the data processing for the monitor.

Type: String

Valid Values: `OK | INACTIVE | COLLECTING_DATA | INSUFFICIENT_DATA | FAULT_SERVICE | FAULT_ACCESS_CLOUDWATCH`

ProcessingStatusInfo

Additional information about the health of the data processing for the monitor.

Type: String

Resources

The resources monitored by the monitor. Resources are listed by their Amazon Resource Names (ARNs).

Type: Array of strings

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `arn:.*`

Status

The status of the monitor.

Type: String

Valid Values: `PENDING | ACTIVE | INACTIVE | ERROR`

Tags

The tags that have been added to monitor.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

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

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

TrafficPercentageToMonitor

The percentage of the internet-facing traffic for your application to monitor with this monitor. If you set a city-networks maximum, that limit overrides the traffic percentage that you set.

To learn more, see Choosing an application traffic percentage to monitor in the Internet Monitor section of the *CloudWatch User Guide*.

Type: Integer

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

Errors

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

AccessDeniedException

You don't have sufficient permission to perform this action.

HTTP Status Code: 403

InternalServerException

An internal error occurred.

HTTP Status Code: 500

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 429

ValidationException

Invalid request.

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 v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetQueryResults

Return the data for a query with the Internet Monitor query interface. Specify the query that you want to return results for by providing a `QueryId` and a monitor name.

For more information about using the query interface, including examples, see Using the Internet Monitor query interface in the Internet Monitor User Guide.

Request Syntax

```
GET /v20210603/Monitors/MonitorName/Queries/QueryId/Results?
MaxResults=MaxResults&NextToken=NextToken HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

MaxResults

The number of query results that you want to return with this call.

Valid Range: Minimum value of 1. Maximum value of 1000.

MonitorName

The name of the monitor to return data for.

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

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

NextToken

The token for the next set of results. You receive this token from a previous call.

QueryId

The ID of the query that you want to return data results for. A `QueryId` is an internally-generated identifier for a specific query.

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "Data": [
      [ "string" ]
   ],
   "Fields": [
      {
         "Name": "string",
         "Type": "string"
      }
   ],
   "NextToken": "string"
}
```

Response Elements

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

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

Data

The data results that the query returns. Data is returned in arrays, aligned with the `Fields` for the query, which creates a repository of Internet Monitor information for your application. Then, you can filter the information in the repository by using `FilterParameters` that you define.

Type: Array of arrays of strings

Fields

The fields that the query returns data for. Fields are name-data type pairs, such as `availability_score-float`.

Type: Array of QueryField objects

NextToken

The token for the next set of results. You receive this token from a previous call.

Type: String

Errors

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

AccessDeniedException

You don't have sufficient permission to perform this action.

HTTP Status Code: 403

InternalServerException

An internal error occurred.

HTTP Status Code: 500

LimitExceededException

The request exceeded a service quota.

HTTP Status Code: 403

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 429

ValidationException

Invalid request.

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 v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetQueryStatus

Returns the current status of a query for the Internet Monitor query interface, for a specified query ID and monitor. When you run a query, check the status to make sure that the query has SUCCEEDED before you review the results.

- QUEUED: The query is scheduled to run.
- RUNNING: The query is in progress but not complete.
- SUCCEEDED: The query completed sucessfully.
- FAILED: The query failed due to an error.
- CANCELED: The query was canceled.

Request Syntax

```
GET /v20210603/Monitors/MonitorName/Queries/QueryId/Status HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

MonitorName

The name of the monitor.

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

Pattern: [a-zA-Z0-9_.-]+

Required: Yes

QueryId

The ID of the query that you want to return the status for. A QueryId is an internally-generated dentifier for a specific query.

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "Status": "string"
}
```

Response Elements

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

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

Status

The current status for a query.

Type: String

Valid Values: `QUEUED | RUNNING | SUCCEEDED | FAILED | CANCELED`

Errors

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

AccessDeniedException

You don't have sufficient permission to perform this action.

HTTP Status Code: 403

InternalServerException

An internal error occurred.

HTTP Status Code: 500

LimitExceededException

The request exceeded a service quota.

HTTP Status Code: 403

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 429

ValidationException

Invalid request.

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 v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListHealthEvents

Lists all health events for a monitor in Internet Monitor. Returns information for health events including the event start and end times, and the status.

Note

Health events that have start times during the time frame that is requested are not included in the list of health events.

Request Syntax

```
GET /v20210603/Monitors/MonitorName/HealthEvents?
EndTime=EndTime&EventStatus=EventStatus&LinkedAccountId=LinkedAccountId&MaxResults=MaxResults&N
 HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

EndTime

The time when a health event ended. If the health event is still ongoing, then the end time is not set.

EventStatus

The status of a health event.

Valid Values: `ACTIVE | RESOLVED`

LinkedAccountId

The account ID for an account that you've set up cross-account sharing for in Internet Monitor. You configure cross-account sharing by using Amazon CloudWatch Observability Access Manager. For more information, see Internet Monitor cross-account observability in the Internet Monitor User Guide.

Length Constraints: Fixed length of 12.

MaxResults

The number of health event objects that you want to return with this call.

Valid Range: Minimum value of 1. Maximum value of 25.

MonitorName

The name of the monitor.

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

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

NextToken

The token for the next set of results. You receive this token from a previous call.

StartTime

The time when a health event started.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "HealthEvents": [
      {
         "CreatedAt": "string",
         "EndedAt": "string",
         "EventArn": "string",
         "EventId": "string",
         "HealthScoreThreshold": number,
         "ImpactedLocations": [
            {
               "ASName": "string",
               "ASNumber": number,
               "CausedBy": { 
                  "AsPath": [ 
                     { 
                        "ASName": "string",
                        "ASNumber": number
                     }
                  ],
                  "NetworkEventType": "string",
                  "Networks": [ 
                     { 
                        "ASName": "string",
                        "ASNumber": number
                     }
                  ]
               },
               "City": "string",
               "Country": "string",
               "CountryCode": "string",
               "InternetHealth": { 
                  "Availability": { 
                     "ExperienceScore": number,
                     "PercentOfClientLocationImpacted": number,
                     "PercentOfTotalTrafficImpacted": number
                  },
                  "Performance": { 
                     "ExperienceScore": number,
                     "PercentOfClientLocationImpacted": number,
                     "PercentOfTotalTrafficImpacted": number,
                     "RoundTripTime": { 
                        "P50": number,
                        "P90": number,
                        "P95": number
                     }
                  }
               },
               "Ipv4Prefixes": [ "string" ],
               "Latitude": number,
               "Longitude": number,
               "Metro": "string",
               "ServiceLocation": "string",
               "Status": "string",
               "Subdivision": "string",
               "SubdivisionCode": "string"
            }
```

```
         ],
         "ImpactType": "string",
         "LastUpdatedAt": "string",
         "PercentOfTotalTrafficImpacted": number,
         "StartedAt": "string",
         "Status": "string"
      }
   ],
   "NextToken": "string"
}
```

Response Elements

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

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

HealthEvents

A list of health events.

Type: Array of HealthEvent objects

NextToken

The token for the next set of results. You receive this token from a previous call.

Type: String

Errors

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

AccessDeniedException

You don't have sufficient permission to perform this action.

HTTP Status Code: 403

InternalServerException

An internal error occurred.

HTTP Status Code: 500

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 429

ValidationException

Invalid request.

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 v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListInternetEvents

Lists internet events that cause performance or availability issues for client locations. Internet Monitor displays information about recent global health events, called internet events, on a global outages map that is available to all AWS customers.

You can constrain the list of internet events returned by providing a start time and end time to define a total time frame for events you want to list. Both start time and end time specify the time when an event started. End time is optional. If you don't include it, the default end time is the current time.

You can also limit the events returned to a specific status (`ACTIVE` or `RESOLVED`) or type (`PERFORMANCE` or `AVAILABILITY`).

Request Syntax

```
GET /v20210603/InternetEvents?
EndTime=EndTime&EventStatus=EventStatus&EventType=EventType&InternetEventMaxResults=MaxResults&
 HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

EndTime

The end time of the time window that you want to get a list of internet events for.

EventStatus

The status of an internet event.

EventType

The type of network impairment.

MaxResults

The number of query results that you want to return with this call.

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

NextToken

The token for the next set of results. You receive this token from a previous call.

StartTime

The start time of the time window that you want to get a list of internet events for.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "InternetEvents": [
      {
         "ClientLocation": {
            "ASName": "string",
            "ASNumber": number,
            "City": "string",
            "Country": "string",
            "Latitude": number,
            "Longitude": number,
            "Metro": "string",
            "Subdivision": "string"
         },
         "EndedAt": "string",
         "EventArn": "string",
         "EventId": "string",
         "EventStatus": "string",
         "EventType": "string",
         "StartedAt": "string"
      }
   ],
   "NextToken": "string"
}
```

Response Elements

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

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

InternetEvents

A set of internet events returned for the list operation.

Type: Array of InternetEventSummary objects

NextToken

The token for the next set of results. You receive this token from a previous call.

Type: String

Errors

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

AccessDeniedException

You don't have sufficient permission to perform this action.

HTTP Status Code: 403

InternalServerException

An internal error occurred.

HTTP Status Code: 500

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 429

ValidationException

Invalid request.

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 v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListMonitors

Lists all of your monitors for Internet Monitor and their statuses, along with the Amazon Resource Name (ARN) and name of each monitor.

Request Syntax

```
GET /v20210603/Monitors?
IncludeLinkedAccounts=IncludeLinkedAccounts&MaxResults=MaxResults&MonitorStatus=MonitorStatus&N
 HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

IncludeLinkedAccounts

A boolean option that you can set to TRUE to include monitors for linked accounts in a list of monitors, when you've set up cross-account sharing in Internet Monitor. You configure cross-account sharing by using Amazon CloudWatch Observability Access Manager. For more information, see Internet Monitor cross-account observability in the Internet Monitor User Guide.

MaxResults

The number of monitor objects that you want to return with this call.

Valid Range: Minimum value of 1. Maximum value of 25.

MonitorStatus

The status of a monitor. This includes the status of the data processing for the monitor and the status of the monitor itself.

For information about the statuses for a monitor, see Monitor.

NextToken

The token for the next set of results. You receive this token from a previous call.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "Monitors": [
      {
         "MonitorArn": "string",
         "MonitorName": "string",
         "ProcessingStatus": "string",
         "Status": "string"
      }
   ],
   "NextToken": "string"
}
```

Response Elements

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

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

Monitors

A list of monitors.

Type: Array of Monitor objects

NextToken

The token for the next set of results. You receive this token from a previous call.

Type: String

Errors

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

AccessDeniedException

You don't have sufficient permission to perform this action.

HTTP Status Code: 403

InternalServerException

An internal error occurred.

HTTP Status Code: 500

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 429

ValidationException

Invalid request.

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 v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListTagsForResource

Lists the tags for a resource. Tags are supported only for monitors in Internet Monitor.

Request Syntax

```
GET /tags/ResourceArn HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

ResourceArn

The Amazon Resource Name (ARN) for a resource.

Length Constraints: Minimum length of 20. Maximum length of 512.

Pattern: `arn:.*`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "Tags": {
      "string" : "string"
   }
}
```

Response Elements

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

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

Tags

Tags for a resource.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

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

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

Errors

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

AccessDeniedException

You don't have sufficient permission to perform this action.

HTTP Status Code: 403

BadRequestException

A bad request was received.

HTTP Status Code: 400

InternalServerErrorException

There was an internal server error.

HTTP Status Code: 500

NotFoundException

The request specifies something that doesn't exist.

HTTP Status Code: 404

TooManyRequestsException

There were too many requests.

HTTP Status Code: 429

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 v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

StartQuery

Start a query to return data for a specific query type for the Internet Monitor query interface. Specify a time period for the data that you want returned by using StartTime and EndTime. You filter the query results to return by providing parameters that you specify with FilterParameters.

For more information about using the query interface, including examples, see Using the Internet Monitor query interface in the Internet Monitor User Guide.

Request Syntax

```
POST /v20210603/Monitors/MonitorName/Queries HTTP/1.1
Content-type: application/json

{
   "EndTime": "string",
   "FilterParameters": [
      {
         "Field": "string",
         "Operator": "string",
         "Values": [ "string" ]
      }
   ],
   "LinkedAccountId": "string",
   "QueryType": "string",
   "StartTime": "string"
}
```

URI Request Parameters

The request uses the following URI parameters.

MonitorName

The name of the monitor to query.

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

Pattern: [a-zA-Z0-9_.-]+

Required: Yes

Request Body

The request accepts the following data in JSON format.

EndTime

The timestamp that is the end of the period that you want to retrieve data for with your query.

Type: Timestamp

Required: Yes

FilterParameters

The `FilterParameters` field that you use with Internet Monitor queries is a string the defines how you want a query to be filtered. The filter parameters that you can specify depend on the query type, since each query type returns a different set of Internet Monitor data.

For more information about specifying filter parameters, see Using the Internet Monitor query interface in the Internet Monitor User Guide.

Type: Array of FilterParameter objects

Required: No

LinkedAccountId

The account ID for an account that you've set up cross-account sharing for in Internet Monitor. You configure cross-account sharing by using Amazon CloudWatch Observability Access Manager. For more information, see Internet Monitor cross-account observability in the Internet Monitor User Guide.

Type: String

Length Constraints: Fixed length of 12.

Required: No

QueryType

The type of query to run. The following are the three types of queries that you can run using the Internet Monitor query interface:

- `MEASUREMENTS`: Provides availability score, performance score, total traffic, and round-trip times, at 5 minute intervals.

- TOP_LOCATIONS: Provides availability score, performance score, total traffic, and time to first byte (TTFB) information, for the top location and ASN combinations that you're monitoring, by traffic volume.
- TOP_LOCATION_DETAILS: Provides TTFB for Amazon CloudFront, your current configuration, and the best performing EC2 configuration, at 1 hour intervals.
- OVERALL_TRAFFIC_SUGGESTIONS: Provides TTFB, using a 30-day weighted average, for all traffic in each AWS location that is monitored.
- OVERALL_TRAFFIC_SUGGESTIONS_DETAILS: Provides TTFB, using a 30-day weighted average, for each top location, for a proposed AWS location. Must provide an AWS location to search.
- ROUTING_SUGGESTIONS: Provides the predicted average round-trip time (RTT) from an IP prefix toward an AWS location for a DNS resolver. The RTT is calculated at one hour intervals, over a one hour period.

For lists of the fields returned with each query type and more information about how each type of query is performed, see Using the Internet Monitor query interface in the Internet Monitor User Guide.

Type: String

Valid Values: MEASUREMENTS | TOP_LOCATIONS | TOP_LOCATION_DETAILS | OVERALL_TRAFFIC_SUGGESTIONS | OVERALL_TRAFFIC_SUGGESTIONS_DETAILS | ROUTING_SUGGESTIONS

Required: Yes

StartTime

The timestamp that is the beginning of the period that you want to retrieve data for with your query.

Type: Timestamp

Required: Yes

Response Syntax

```
HTTP/1.1 200
Content-type: application/json
```

```
{
   "QueryId": "string"
}
```

Response Elements

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

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

QueryId

The internally-generated identifier of a specific query.

Type: String

Errors

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

AccessDeniedException

You don't have sufficient permission to perform this action.

HTTP Status Code: 403

InternalServerException

An internal error occurred.

HTTP Status Code: 500

LimitExceededException

The request exceeded a service quota.

HTTP Status Code: 403

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 429

ValidationException

Invalid request.

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 v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

StopQuery

Stop a query that is progress for a specific monitor.

Request Syntax

```
DELETE /v20210603/Monitors/MonitorName/Queries/QueryId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

MonitorName

The name of the monitor.

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

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

QueryId

The ID of the query that you want to stop. A `QueryId` is an internally-generated identifier for a specific query.

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
```

Response Elements

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

Errors

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

AccessDeniedException

You don't have sufficient permission to perform this action.

HTTP Status Code: 403

InternalServerException

An internal error occurred.

HTTP Status Code: 500

LimitExceededException

The request exceeded a service quota.

HTTP Status Code: 403

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 429

ValidationException

Invalid request.

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 v2

- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

TagResource

Adds a tag to a resource. Tags are supported only for monitors in Internet Monitor. You can add a maximum of 50 tags in Internet Monitor.

A minimum of one tag is required for this call. It returns an error if you use the `TagResource` request with 0 tags.

Request Syntax

```
POST /tags/ResourceArn HTTP/1.1
Content-type: application/json

{
   "Tags": {
      "string" : "string"
   }
}
```

URI Request Parameters

The request uses the following URI parameters.

ResourceArn

The Amazon Resource Name (ARN) for a tag that you add to a resource. Tags are supported only for monitors in Internet Monitor.

Length Constraints: Minimum length of 20. Maximum length of 512.

Pattern: `arn:.*`

Required: Yes

Request Body

The request accepts the following data in JSON format.

Tags

Tags that you add to a resource. You can add a maximum of 50 tags in Internet Monitor.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

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

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

Required: Yes

Response Syntax

```
HTTP/1.1 204
```

Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

Errors

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

AccessDeniedException

You don't have sufficient permission to perform this action.

HTTP Status Code: 403

BadRequestException

A bad request was received.

HTTP Status Code: 400

InternalServerErrorException

There was an internal server error.

HTTP Status Code: 500

NotFoundException

The request specifies something that doesn't exist.

HTTP Status Code: 404

TooManyRequestsException

There were too many requests.

HTTP Status Code: 429

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 v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UntagResource

Removes a tag from a resource.

Request Syntax

```
DELETE /tags/ResourceArn?tagKeys=TagKeys HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

ResourceArn

The Amazon Resource Name (ARN) for a tag you remove a resource from.

Length Constraints: Minimum length of 20. Maximum length of 512.

Pattern: `arn:.*`

Required: Yes

TagKeys

Tag keys that you remove from a resource.

Array Members: Minimum number of 0 items. Maximum number of 200 items.

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

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 204
```

Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

Errors

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

AccessDeniedException

You don't have sufficient permission to perform this action.

HTTP Status Code: 403

BadRequestException

A bad request was received.

HTTP Status Code: 400

InternalServerErrorException

There was an internal server error.

HTTP Status Code: 500

NotFoundException

The request specifies something that doesn't exist.

HTTP Status Code: 404

TooManyRequestsException

There were too many requests.

HTTP Status Code: 429

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 v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateMonitor

Updates a monitor. You can update a monitor to change the percentage of traffic to monitor or the maximum number of city-networks (locations and ASNs), to add or remove resources, or to change the status of the monitor. Note that you can't change the name of a monitor.

The city-network maximum that you choose is the limit, but you only pay for the number of city-networks that are actually monitored. For more information, see Choosing a city-network maximum value in the *Amazon CloudWatch User Guide*.

Request Syntax

```
PATCH /v20210603/Monitors/MonitorName HTTP/1.1
Content-type: application/json

{
   "ClientToken": "string",
   "HealthEventsConfig": {
      "AvailabilityLocalHealthEventsConfig": {
         "HealthScoreThreshold": number,
         "MinTrafficImpact": number,
         "Status": "string"
      },
      "AvailabilityScoreThreshold": number,
      "PerformanceLocalHealthEventsConfig": {
         "HealthScoreThreshold": number,
         "MinTrafficImpact": number,
         "Status": "string"
      },
      "PerformanceScoreThreshold": number
   },
   "InternetMeasurementsLogDelivery": {
      "S3Config": {
         "BucketName": "string",
         "BucketPrefix": "string",
         "LogDeliveryStatus": "string"
      }
   },
   "MaxCityNetworksToMonitor": number,
   "ResourcesToAdd": [ "string" ],
   "ResourcesToRemove": [ "string" ],
   "Status": "string",
   "TrafficPercentageToMonitor": number
}
```

URI Request Parameters

The request uses the following URI parameters.

MonitorName

The name of the monitor.

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

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

Request Body

The request accepts the following data in JSON format.

ClientToken

A unique, case-sensitive string of up to 64 ASCII characters that you specify to make an idempotent API request. You should not reuse the same client token for other API requests.

Type: String

Required: No

HealthEventsConfig

The list of health score thresholds. A threshold percentage for health scores, along with other configuration information, determines when Internet Monitor creates a health event when there's an internet issue that affects your application end users.

For more information, see Change health event thresholds in the Internet Monitor section of the *CloudWatch User Guide*.

Type: HealthEventsConfig object

Required: No

InternetMeasurementsLogDelivery

Publish internet measurements for Internet Monitor to another location, such as an Amazon S3 bucket. The measurements are also published to Amazon CloudWatch Logs.

Type: InternetMeasurementsLogDelivery object

Required: No

MaxCityNetworksToMonitor

The maximum number of city-networks to monitor for your application. A city-network is the location (city) where clients access your application resources from and the ASN or network provider, such as an internet service provider (ISP), that clients access the resources through. Setting this limit can help control billing costs.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 500000.

Required: No

ResourcesToAdd

The resources to include in a monitor, which you provide as a set of Amazon Resource Names (ARNs). Resources can be VPCs, NLBs, Amazon CloudFront distributions, or Amazon WorkSpaces directories.

You can add a combination of VPCs and CloudFront distributions, or you can add WorkSpaces directories, or you can add NLBs. You can't add NLBs or WorkSpaces directories together with any other resources.

> **Note**
>
> If you add only Amazon Virtual Private Clouds resources, at least one VPC must have an Internet Gateway attached to it, to make sure that it has internet connectivity.

Type: Array of strings

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `arn:.*`

Required: No

ResourcesToRemove

The resources to remove from a monitor, which you provide as a set of Amazon Resource Names (ARNs).

Type: Array of strings

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `arn:.*`

Required: No

Status

The status for a monitor. The accepted values for `Status` with the `UpdateMonitor` API call are the following: `ACTIVE` and `INACTIVE`. The following values are *not* accepted: `PENDING`, and `ERROR`.

Type: String

Valid Values: `PENDING | ACTIVE | INACTIVE | ERROR`

Required: No

TrafficPercentageToMonitor

The percentage of the internet-facing traffic for your application that you want to monitor with this monitor. If you set a city-networks maximum, that limit overrides the traffic percentage that you set.

To learn more, see Choosing an application traffic percentage to monitor in the Internet Monitor section of the *CloudWatch User Guide*.

Type: Integer

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

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "MonitorArn": "string",
   "Status": "string"
}
```

Response Elements

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

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

MonitorArn

The Amazon Resource Name (ARN) of the monitor.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 512.

Pattern: `arn:.*`

Status

The status of a monitor.

Type: String

Valid Values: `PENDING | ACTIVE | INACTIVE | ERROR`

Errors

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

AccessDeniedException

You don't have sufficient permission to perform this action.

HTTP Status Code: 403

InternalServerException

An internal error occurred.

HTTP Status Code: 500

LimitExceededException

The request exceeded a service quota.

HTTP Status Code: 403

ResourceNotFoundException

The request specifies a resource that doesn't exist.

HTTP Status Code: 404

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 429

ValidationException

Invalid request.

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 v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3

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

Data Types

The Amazon CloudWatch Internet Monitor 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:

- AvailabilityMeasurement
- ClientLocation
- FilterParameter
- HealthEvent
- HealthEventsConfig
- ImpactedLocation
- InternetEventSummary
- InternetHealth
- InternetMeasurementsLogDelivery
- LocalHealthEventsConfig
- Monitor
- Network
- NetworkImpairment
- PerformanceMeasurement
- QueryField
- RoundTripTime
- S3Config

AvailabilityMeasurement

Internet Monitor calculates measurements about the availability for your application's internet traffic between client locations and AWS. AWS has substantial historical data about internet performance and availability between AWS services and different network providers and geographies. By applying statistical analysis to the data, Internet Monitor can detect when the performance and availability for your application has dropped, compared to an estimated baseline that's already calculated. To make it easier to see those drops, we report that information to you in the form of health scores: a performance score and an availability score.

Availability in Internet Monitor represents the estimated percentage of traffic that is not seeing an availability drop. For example, an availability score of 99% for an end user and service location pair is equivalent to 1% of the traffic experiencing an availability drop for that pair.

For more information, see How Internet Monitor calculates performance and availability scores in the Internet Monitor section of the *Amazon CloudWatch User Guide*.

Contents

ExperienceScore

Experience scores, or health scores are calculated for different geographic and network provider combinations (that is, different granularities) and also summed into global scores. If you view performance or availability scores without filtering for any specific geography or service provider, Internet Monitor provides global health scores.

The Internet Monitor chapter in the *CloudWatch User Guide* includes detailed information about how Internet Monitor calculates health scores, including performance and availability scores, and when it creates and resolves health events. For more information, see How AWS calculates performance and availability scores in the Internet Monitor section of the *CloudWatch User Guide*.

Type: Double

Required: No

PercentOfClientLocationImpacted

The percentage of impact caused by a health event for client location traffic globally.

For information about how Internet Monitor calculates impact, see Inside Internet Monitor in the Internet Monitor section of the Amazon CloudWatch User Guide.

Type: Double

Required: No

PercentOfTotalTrafficImpacted

The impact on total traffic that a health event has, in increased latency or reduced availability. This is the percentage of how much latency has increased or availability has decreased during the event, compared to what is typical for traffic from this client location to the AWS location using this client network.

For information about how Internet Monitor calculates impact, see How Internet Monitor works in the Internet Monitor section of the Amazon CloudWatch User Guide.

Type: Double

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ClientLocation

The impacted location, such as a city, that AWS clients access application resources from.

Contents

ASName

The name of the internet service provider (ISP) or network (ASN).

Type: String

Required: Yes

ASNumber

The Autonomous System Number (ASN) of the network at an impacted location.

Type: Long

Required: Yes

City

The name of the city where the internet event is located.

Type: String

Required: Yes

Country

The name of the country where the internet event is located.

Type: String

Required: Yes

Latitude

The latitude where the internet event is located.

Type: Double

Required: Yes

Longitude

The longitude where the internet event is located.

Type: Double

Required: Yes

Metro

The metro area where the health event is located.

Metro indicates a metropolitan region in the United States, such as the region around New York City. In non-US countries, this is a second-level subdivision. For example, in the United Kingdom, it could be a county, a London borough, a unitary authority, council area, and so on.

Type: String

Required: No

Subdivision

The subdivision location where the health event is located. The subdivision usually maps to states in most countries (including the United States). For United Kingdom, it maps to a country (England, Scotland, Wales) or province (Northern Ireland).

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 Java V2
- AWS SDK for Ruby V3

FilterParameter

A filter that you use with the results of a Internet Monitor query that you created and ran. The query sets up a repository of data that is a subset of your application's Internet Monitor data. `FilterParameter` is a string that defines how you want to filter the repository of data to return a set of results, based on your criteria.

The filter parameters that you can specify depend on the query type that you used to create the repository, since each query type returns a different set of Internet Monitor data.

For each filter, you specify a field (such as `city`), an operator (such as `not_equals`, and a value or array of values (such as `["Seattle", "Redmond"]`). Separate values in the array with commas.

For more information about specifying filter parameters, see Using the Internet Monitor query interface in the Internet Monitor User Guide.

Contents

Field

A data field that you want to filter, to further scope your application's Internet Monitor data in a repository that you created by running a query. A field might be `city`, for example. The field must be one of the fields that was returned by the specific query that you used to create the repository.

Type: String

Required: No

Operator

The operator to use with the filter field and a value, such as `not_equals`.

Type: String

Valid Values: `EQUALS | NOT_EQUALS`

Required: No

Values

One or more values to be used, together with the specified operator, to filter data for a query. For example, you could specify an array of values such as `["Seattle", "Redmond"]`. Values in the array are separated by commas.

Type: Array of strings

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 Java V2
- AWS SDK for Ruby V3

HealthEvent

Information about a health event created in a monitor in Internet Monitor.

Contents

EventArn

The Amazon Resource Name (ARN) of the event.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `arn:.*`

Required: Yes

EventId

The internally-generated identifier of a specific network traffic impairment health event.

Type: String

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

Pattern: `[a-zA-Z0-9/_.-]+`

Required: Yes

ImpactedLocations

The locations impacted by the health event.

Type: Array of ImpactedLocation objects

Required: Yes

ImpactType

The type of impairment for a health event.

Type: String

Valid Values: `AVAILABILITY | PERFORMANCE | LOCAL_AVAILABILITY | LOCAL_PERFORMANCE`

Required: Yes

LastUpdatedAt

When the health event was last updated.

Type: Timestamp

Required: Yes

StartedAt

When a health event started.

Type: Timestamp

Required: Yes

Status

The status of a health event.

Type: String

Valid Values: `ACTIVE | RESOLVED`

Required: Yes

CreatedAt

When the health event was created.

Type: Timestamp

Required: No

EndedAt

The time when a health event ended. If the health event is still active, then the end time is not set.

Type: Timestamp

Required: No

HealthScoreThreshold

The value of the threshold percentage for performance or availability that was configured when Internet Monitor created the health event.

Type: Double

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

PercentOfTotalTrafficImpacted

The impact on total traffic that a health event has, in increased latency or reduced availability. This is the percentage of how much latency has increased or availability has decreased during the event, compared to what is typical for traffic from this client location to the AWS location using this client network.

Type: Double

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

HealthEventsConfig

A complex type with the configuration information that determines the threshold and other conditions for when Internet Monitor creates a health event for an overall performance or availability issue, across an application's geographies.

Defines the percentages, for overall performance scores and availability scores for an application, that are the thresholds for when Internet Monitor creates a health event. You can override the defaults to set a custom threshold for overall performance or availability scores, or both.

You can also set thresholds for local health scores,, where Internet Monitor creates a health event when scores cross a threshold for one or more city-networks, in addition to creating an event when an overall score crosses a threshold.

If you don't set a health event threshold, the default value is 95%.

For local thresholds, you also set a minimum percentage of overall traffic that is impacted by an issue before Internet Monitor creates an event. In addition, you can disable local thresholds, for performance scores, availability scores, or both.

For more information, see Change health event thresholds in the Internet Monitor section of the *CloudWatch User Guide*.

Contents

AvailabilityLocalHealthEventsConfig

The configuration that determines the threshold and other conditions for when Internet Monitor creates a health event for a local availability issue.

Type: LocalHealthEventsConfig object

Required: No

AvailabilityScoreThreshold

The health event threshold percentage set for availability scores.

Type: Double

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

PerformanceLocalHealthEventsConfig

The configuration that determines the threshold and other conditions for when Internet Monitor creates a health event for a local performance issue.

Type: LocalHealthEventsConfig object

Required: No

PerformanceScoreThreshold

The health event threshold percentage set for performance scores.

Type: Double

Valid Range: Minimum value of 0. Maximum value of 100.

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 Java V2
- AWS SDK for Ruby V3

ImpactedLocation

Information about a location impacted by a health event in Internet Monitor.

Geographic regions are hierarchically categorized into country, subdivision, metro and city geographic granularities. The geographic region is identified based on the IP address used at the client locations.

Contents

ASName

The name of the internet service provider (ISP) or network (ASN).

Type: String

Required: Yes

ASNumber

The Autonomous System Number (ASN) of the network at an impacted location.

Type: Long

Required: Yes

Country

The name of the country where the health event is located.

Type: String

Required: Yes

Status

The status of the health event at an impacted location.

Type: String

Valid Values: `ACTIVE | RESOLVED`

Required: Yes

CausedBy

The cause of the impairment. There are two types of network impairments: AWS network issues or internet issues. Internet issues are typically a problem with a network provider, like an internet service provider (ISP).

Type: NetworkImpairment object

Required: No

City

The name of the city where the health event is located.

Type: String

Required: No

CountryCode

The country code where the health event is located. The ISO 3166-2 codes for the country is provided, when available.

Type: String

Required: No

InternetHealth

The calculated health at a specific location.

Type: InternetHealth object

Required: No

Ipv4Prefixes

The IPv4 prefixes at the client location that was impacted by the health event.

Type: Array of strings

Required: No

Latitude

The latitude where the health event is located.

Type: Double

Required: No

Longitude

The longitude where the health event is located.

Type: Double

Required: No

Metro

The metro area where the health event is located.

Metro indicates a metropolitan region in the United States, such as the region around New York City. In non-US countries, this is a second-level subdivision. For example, in the United Kingdom, it could be a county, a London borough, a unitary authority, council area, and so on.

Type: String

Required: No

ServiceLocation

The service location where the health event is located.

Type: String

Required: No

Subdivision

The subdivision location where the health event is located. The subdivision usually maps to states in most countries (including the United States). For United Kingdom, it maps to a country (England, Scotland, Wales) or province (Northern Ireland).

Type: String

Required: No

SubdivisionCode

The subdivision code where the health event is located. The ISO 3166-2 codes for country subdivisions is provided, when available.

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 Java V2
- AWS SDK for Ruby V3

InternetEventSummary

A summary of information about an internet event in Internet Monitor. Internet events are issues that cause performance degradation or availability problems for impacted AWS client locations. Internet Monitor displays information about recent global health events, called internet events, on a global outages map that is available to all AWS customers.

Contents

ClientLocation

The impacted location, such as a city, that AWS clients access application resources from.

Type: ClientLocation object

Required: Yes

EventArn

The Amazon Resource Name (ARN) of the internet event.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `arn:.*`

Required: Yes

EventId

The internally-generated identifier of an internet event.

Type: String

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

Pattern: `[a-zA-Z0-9-]+`

Required: Yes

EventStatus

The status of an internet event.

Type: String

Valid Values: `ACTIVE | RESOLVED`

Required: Yes

EventType

The type of network impairment.

Type: String

Valid Values: `AVAILABILITY | PERFORMANCE`

Required: Yes

StartedAt

The time when an internet event started.

Type: Timestamp

Required: Yes

EndedAt

The time when an internet event ended. If the event hasn't ended yet, this value is empty.

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 Java V2
- AWS SDK for Ruby V3

InternetHealth

Internet health includes measurements calculated by Internet Monitor about the performance and availability for your application on the internet. AWS has substantial historical data about internet performance and availability between AWS services and different network providers and geographies. By applying statistical analysis to the data, Internet Monitor can detect when the performance and availability for your application has dropped, compared to an estimated baseline that's already calculated. To make it easier to see those drops, Internet Monitor reports the information to you in the form of health scores: a performance score and an availability score.

Contents

Availability

Availability in Internet Monitor represents the estimated percentage of traffic that is not seeing an availability drop. For example, an availability score of 99% for an end user and service location pair is equivalent to 1% of the traffic experiencing an availability drop for that pair.

For more information, see How Internet Monitor calculates performance and availability scores in the Internet Monitor section of the *CloudWatch User Guide*.

Type: AvailabilityMeasurement object

Required: No

Performance

Performance in Internet Monitor represents the estimated percentage of traffic that is not seeing a performance drop. For example, a performance score of 99% for an end user and service location pair is equivalent to 1% of the traffic experiencing a performance drop for that pair.

For more information, see How Internet Monitor calculates performance and availability scores in the Internet Monitor section of the *CloudWatch User Guide*.

Type: PerformanceMeasurement object

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 Java V2
- AWS SDK for Ruby V3

InternetMeasurementsLogDelivery

Publish internet measurements to an Amazon S3 bucket in addition to CloudWatch Logs.

Contents

S3Config

The configuration information for publishing Internet Monitor internet measurements to Amazon S3. The configuration includes the bucket name and (optionally) prefix for the S3 bucket to store the measurements, and the delivery status. The delivery status is `ENABLED` or `DISABLED`, depending on whether you choose to deliver internet measurements to S3 logs.

Type: S3Config object

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 Java V2
- AWS SDK for Ruby V3

LocalHealthEventsConfig

A complex type with the configuration information that determines the threshold and other conditions for when Internet Monitor creates a health event for a local performance or availability issue, when scores cross a threshold for one or more city-networks.

Defines the percentages, for performance scores or availability scores, that are the local thresholds for when Internet Monitor creates a health event. Also defines whether a local threshold is enabled or disabled, and the minimum percentage of overall traffic that must be impacted by an issue before Internet Monitor creates an event when a threshold is crossed for a local health score.

If you don't set a local health event threshold, the default value is 60%.

For more information, see Change health event thresholds in the Internet Monitor section of the *CloudWatch User Guide*.

Contents

HealthScoreThreshold

The health event threshold percentage set for a local health score.

Type: Double

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

MinTrafficImpact

The minimum percentage of overall traffic for an application that must be impacted by an issue before Internet Monitor creates an event when a threshold is crossed for a local health score.

If you don't set a minimum traffic impact threshold, the default value is 0.1%.

Type: Double

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Status

The status of whether Internet Monitor creates a health event based on a threshold percentage set for a local health score. The status can be `ENABLED` or `DISABLED`.

Type: String

Valid Values: `ENABLED | DISABLED`

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 Java V2
- AWS SDK for Ruby V3

Monitor

The description of and information about a monitor in Internet Monitor.

Contents

MonitorArn

The Amazon Resource Name (ARN) of the monitor.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 512.

Pattern: `arn:.*`

Required: Yes

MonitorName

The name of the monitor.

Type: String

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

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

Status

The status of a monitor.

Type: String

Valid Values: `PENDING | ACTIVE | INACTIVE | ERROR`

Required: Yes

ProcessingStatus

The health of data processing for the monitor.

Type: String

Valid Values: `OK | INACTIVE | COLLECTING_DATA | INSUFFICIENT_DATA | FAULT_SERVICE | FAULT_ACCESS_CLOUDWATCH`

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 Java V2
- AWS SDK for Ruby V3

Network

An internet service provider (ISP) or network (ASN) in Internet Monitor.

Contents

ASName

The name of the internet service provider (ISP) or network (ASN).

Type: String

Required: Yes

ASNumber

The Autonomous System Number (ASN) of the internet provider or network.

Type: Long

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 Java V2
- AWS SDK for Ruby V3

NetworkImpairment

Information about the network impairment for a specific network measured by Internet Monitor.

Contents

AsPath

The combination of the Autonomous System Number (ASN) of the network and the name of the network.

Type: Array of Network objects

Required: Yes

NetworkEventType

The type of network impairment.

Type: String

Valid Values: `AWS | Internet`

Required: Yes

Networks

The networks that could be impacted by a network impairment event.

Type: Array of Network objects

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 Java V2
- AWS SDK for Ruby V3

PerformanceMeasurement

Internet Monitor calculates measurements about the performance for your application's internet traffic between client locations and AWS. AWS has substantial historical data about internet performance and availability between AWS services and different network providers and geographies. By applying statistical analysis to the data, Internet Monitor can detect when the performance and availability for your application has dropped, compared to an estimated baseline that's already calculated. To make it easier to see those drops, we report that information to you in the form of health scores: a performance score and an availability score.

Performance in Internet Monitor represents the estimated percentage of traffic that is not seeing a performance drop. For example, a performance score of 99% for an end user and service location pair is equivalent to 1% of the traffic experiencing a performance drop for that pair.

For more information, see How Internet Monitor calculates performance and availability scores in the Internet Monitor section of the *CloudWatch User Guide*.

Contents

ExperienceScore

Experience scores, or health scores, are calculated for different geographic and network provider combinations (that is, different granularities) and also totaled into global scores. If you view performance or availability scores without filtering for any specific geography or service provider, Internet Monitor provides global health scores.

The Internet Monitor chapter in the CloudWatch User Guide includes detailed information about how Internet Monitor calculates health scores, including performance and availability scores, and when it creates and resolves health events. For more information, see How AWS calculates performance and availability scores in the Internet Monitor section of the *CloudWatch User Guide*.

Type: Double

Required: No

PercentOfClientLocationImpacted

How much performance impact was caused by a health event at a client location. For performance, this is the percentage of how much latency increased during the event compared to typical performance for traffic, from this client location to an AWS location, using a specific client network.

For more information, see When AWS creates and resolves health events in the Internet Monitor section of the *CloudWatch User Guide*.

Type: Double

Required: No

PercentOfTotalTrafficImpacted

The impact on total traffic that a health event has, in increased latency or reduced availability. This is the percentage of how much latency has increased or availability has decreased during the event, compared to what is typical for traffic from this client location to the AWS location using this client network.

For more information, see When AWS creates and resolves health events in the Internet Monitor section of the *CloudWatch User Guide*.

Type: Double

Required: No

RoundTripTime

This is the percentage of how much round-trip time increased during the event compared to typical round-trip time for your application for traffic.

For more information, see When AWS creates and resolves health events in the Internet Monitor section of the *CloudWatch User Guide*.

Type: RoundTripTime object

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 Java V2
- AWS SDK for Ruby V3

QueryField

Defines a field to query for your application's Internet Monitor data. You create a data repository by running a query of a specific type. Each `QueryType` includes a specific set of fields and datatypes to retrieve data for.

Contents

Name

The name of a field to query your application's Internet Monitor data for, such as `availability_score`.

Type: String

Required: No

Type

The data type for a query field, which must correspond to the field you're defining for `QueryField`. For example, if the query field name is `availability_score`, the data type is `float`.

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 Java V2
- AWS SDK for Ruby V3

RoundTripTime

Round-trip time (RTT) is how long it takes for a request from the user to return a response to the user. Internet Monitor calculates RTT at different percentiles: p50, p90, and p95.

Contents

P50

RTT at the 50th percentile (p50).

Type: Double

Required: No

P90

RTT at the 90th percentile (p90).

Type: Double

Required: No

P95

RTT at the 95th percentile (p95).

Type: Double

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

S3Config

The configuration for publishing Internet Monitor internet measurements to Amazon S3. The configuration includes the bucket name and (optionally) prefix for the S3 bucket to store the measurements, and the delivery status. The delivery status is `ENABLED` or `DISABLED`, depending on whether you choose to deliver internet measurements to S3 logs.

Contents

BucketName

The Amazon S3 bucket name.

Type: String

Length Constraints: Minimum length of 3.

Required: No

BucketPrefix

The Amazon S3 bucket prefix.

Type: String

Required: No

LogDeliveryStatus

The status of publishing Internet Monitor internet measurements to an Amazon S3 bucket.

Type: String

Valid Values: `ENABLED | DISABLED`

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 Java V2
- AWS SDK for Ruby V3

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 Signing AWS API requests in the *IAM User Guide*.

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 Create a signed AWS API request in the *IAM User Guide*.

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

Type: string

Required: Conditional

X-Amz-Date

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

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see Elements of an AWS API request signature in the *IAM User Guide*.

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 STS, see AWS services that work with IAM in the *IAM User Guide*.

Condition: If you're using temporary security credentials from AWS STS, 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 Create a signed AWS API request in the *IAM User Guide*.

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: 403

ExpiredTokenException

The security token included in the request is expired

HTTP Status Code: 403

IncompleteSignature

The request signature does not conform to AWS standards.

HTTP Status Code: 403

InternalFailure

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

HTTP Status Code: 500

MalformedHttpRequestException

Problems with the request at the HTTP level, e.g. we can't decompress the body according to the decompression algorithm specified by the content-encoding.

HTTP Status Code: 400

NotAuthorized

You do not have permission to perform this action.

HTTP Status Code: 401

OptInRequired

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

HTTP Status Code: 403

RequestAbortedException

Convenient exception that can be used when a request is aborted before a reply is sent back (e.g. client closed connection).

HTTP Status Code: 400

RequestEntityTooLargeException

Problems with the request at the HTTP level. The request entity is too large.

HTTP Status Code: 413

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

RequestTimeoutException

Problems with the request at the HTTP level. Reading the Request timed out.

HTTP Status Code: 408

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

UnrecognizedClientException

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

HTTP Status Code: 403

UnknownOperationException

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

HTTP Status Code: 404

ValidationError

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

HTTP Status Code: 400