Cloud Eye

API Reference

Issue	12
Date	2018-09-30

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: https://www.huawei.com

Email: support@huawei.com

Contents

1 Before You Start

1.1 Overview

Welcome to *Cloud Eye API Reference*. Cloud Eye is a multi-dimensional resource monitoring platform. Customers can use Cloud Eye to monitor the utilization of service resources, track the running status of cloud services, configure alarm rules and notifications, and quickly respond to resource changes.

This document describes how to use application programming interfaces (APIs) to perform operations on metrics, alarm rules, and monitoring data, such as querying the metric list and the alarm rule list, creating alarm rules, and deleting alarm rules. For details about all supported operations, see **API Overview**.

If you plan to access Cloud Eye through an API, ensure that you are familiar with Cloud Eye concepts. For details, see "What Is Cloud Eye?".

1.2 API Calling

Cloud Eye supports Representational State Transfer (REST) APIs, allowing you to call APIs using HTTPS. For details about API calling, see **Calling APIs**.

1.3 Endpoints

An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions. For the endpoints of all services, see **Regions and Endpoints**.

1.4 Notes and Constraints

- The number of alarm rules that you can create is determined by your quota. To view or increase the quota, see "Quota Adjustment" in the *Cloud Eye User Guide*.
- For more constraints, see API description.

1.5 Concepts

- Account

  An account is created upon successful registration. The account has full access permissions for all of its cloud services and resources. It can be used to reset user passwords and grant user permissions. The account is a payment entity, which should not be used directly to perform routine management. For security purposes, create Identity and Access Management (IAM) users and grant them permissions for routine management.

- User

  An IAM user is created by an account in IAM to use cloud services. Each IAM user has its own identity credentials (password and access keys).

  API authentication requires information such as the account name, username, and password.

- Region

  A region is a geographic area in which cloud resources are deployed. Availability zones (AZs) in the same region can communicate with each other over an intranet, while AZs in different regions are isolated from each other. Deploying cloud resources in different regions can better suit certain user requirements or comply with local laws or regulations.

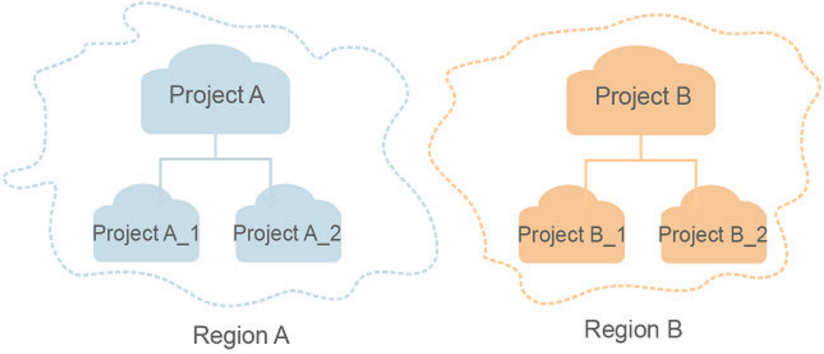
- AZ

  An AZ comprises of one or more physical data centers equipped with independent ventilation, fire, water, and electricity facilities. Computing, network, storage, and other resources in an AZ are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to allow you to build cross-AZ high-availability systems.

- Project

  A project corresponds to a region. Default projects are defined to group and physically isolate resources (including computing, storage, and network resources) across regions. Users can be granted permissions in a default project to access all resources under their accounts in the region associated with the project. If you need more refined access control, create subprojects under a default project and create resources in subprojects. Then you can assign users the permissions required to access only the resources in the specific subprojects.

Figure 1-1 Project isolation model



2 API Overview

Cloud Eye APIs allow you to use all Cloud Eye functions. For example, you can query the metric list and create alarm rules.

Table 2-1 API description

Type	Subtype	API	Description
Cloud Eye API	API version management	**Querying All API Versions**	Query all API versions supported by Cloud Eye.
		Querying a Specified API Version	Query a specified API version supported by Cloud Eye.
	Metric management	**Querying the Metric List**	Query the list of metrics that currently monitored by Cloud Eye.
	Alarm rule management	**Querying the Alarm Rule List**	Query the alarm rule list.
		Querying an Alarm Rule	Query the alarm rule information based on the alarm rule ID.
		Enabling or Disabling an Alarm Rule	Enable or disable an alarm rule based on the alarm rule ID.
		Deleting an Alarm Rule	Delete an alarm rule based on the alarm rule ID.
		Creating an Alarm Rule	Create an alarm rule.
	Monitoring data management	**Querying Monitoring Data**	Query the monitoring data of a specified metric of specified granularity in a specified time range.

Type	Subtype	API	Description
		Adding Monitoring Data	Add one or more pieces of metric monitoring data.
		Querying the Host Configuration	Query the host configuration for a specified event type in a specified period of time. You can specify the dimension of data to be queried.
	Quota management	**Querying Quotas**	Query the alarm rule quota.

3 Calling APIs

3.1 Making an API Request

This section describes the structure of a REST API request, and uses the IAM API for obtaining a user token as an example to demonstrate how to call an API. The obtained token can then be used to authenticate the calling of other APIs.

Request URI

A request URI is in the following format:

{URI-scheme}://{Endpoint}/{resource-path}?{query-string}

Although a request URI is included in the request header, most programming languages or frameworks require the request URI to be transmitted separately.

Table 3-1 URI parameter description

Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from **Regions and Endpoints**.
resource-path	Access path of an API for performing a specified operation. Obtain the path from the URI of an API. For example, the **resource-path** of the API used to obtain a user token is **/v3/auth/tokens**.
query-string	Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of *Parameter name*=*Parameter value*. For example, **?limit=10** indicates that a maximum of 10 data records will be displayed.

NOTE

To simplify the URI display in this document, each API is provided only with a **resource-path** and a request method. The **URI-scheme** of all APIs is **HTTPS**, and the endpoints of all APIs in the same region are identical.

Request Methods

The HTTP protocol defines the following request methods that can be used to send a request to the server.

Table 3-2 HTTP methods

Method	Description
GET	Requests the server to return specified resources.
PUT	Requests the server to update specified resources.
POST	Requests the server to add resources or perform special operations.
DELETE	Requests the server to delete specified resources, for example, an object.
HEAD	Same as GET except that the server must return only the response header.
PATCH	Requests the server to update partial content of a specified resource. If the resource does not exist, a new resource will be created.

For example, in the case of the API used to obtain a user token, the request method is **POST**. The request is as follows:

```
POST https://{{endpoint}}/v3/auth/tokens
```

Request Header

You can also add additional header fields to a request, such as the fields required by a specified URI or HTTP method. For example, to request for the authentication information, add **Content-Type**, which specifies the request body type.

Common request header fields are as follows.

Table 3-3 Common request header fields

Parameter	Description	Mandatory	Example Value
Host	Specifies the server domain name and port number of the resources being requested. The value can be obtained from the URL of the service API. The value is in the format of *Hostname:Port number*. If the port number is not specified, the default port is used. The default port number for **https** is **443**.	No This field is mandatory for AK/SK authentication.	code.test.com or code.test.com:443
Content-Type	Specifies the type (or format) of the message body. The default value **application/json** is recommended. Other values of this field will be provided for specific APIs if any.	Yes	application/json
Content-Length	Specifies the length of the request body. The unit is byte.	No	3495
X-Project-Id	Specifies the project ID. Obtain the project ID by following the instructions in **Obtaining a Project ID**.	No This field is mandatory for requests that use AK/SK authentication in the Dedicated Cloud (DeC) scenario or multi-project scenario.	e9993fc787d94b6c886cbaa340f9c0f4

Parameter	Description	Mandatory	Example Value
X-Auth-Token	Specifies the user token. It is a response to the API for obtaining a user token (This is the only API that does not require authentication). After the request is processed, the value of **X-Subject-Token** in the response header is the token value.	No This field is mandatory for token authentication.	The following is part of an example token: MIIPAgYJKoZIhvcNAQcCo...ggg1BBIINPXsidG9rZ

NOTE

In addition to supporting authentication using tokens, APIs support authentication using AK/SK, which uses SDKs to sign a request. During the signature, the **Authorization** (signature authentication) and **X-Sdk-Date** (time when a request is sent) headers are automatically added in the request.

For more details, see "Authentication Using AK/SK" in **Authentication**.

The API used to obtain a user token does not require authentication. Therefore, only the **Content-Type** field needs to be added to requests for calling the API. An example of such requests is as follows:

```
POST https://{{endpoint}}/v3/auth/tokens
Content-Type: application/json
```

(Optional) Request Body

This part is optional. The body of a request is often sent in a structured format as specified in the **Content-Type** header field. The request body transfers content except the request header.

The request body varies between APIs. Some APIs do not require the request body, such as the APIs requested using the GET and DELETE methods.

In the case of the API used to obtain a user token, the request parameters and parameter description can be obtained from the API request. The following provides an example request with a body included. Replace *username*, *domainname*, ******** (login password), and *xxxxxxxxxxxxxxxxxx* (project name) with the actual values. Obtain a project name from **Regions and Endpoints**.

NOTE

The **scope** parameter specifies where a token takes effect. You can set **scope** to an account or a project under an account. In the following example, the token takes effect only for the resources in a specified project. For more information about this API, see "Obtaining a User Token".

```
POST https://{{endpoint}}/v3/auth/tokens
Content-Type: application/json
```

```
{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "username",
                    "password": "********",
                    "domain": {
                        "name": "domainname"
                    }
                }
            }
        },
        "scope": {
            "project": {
                "name": "xxxxxxxxxxxxxxxxxx"
            }
        }
    }
}
```

If all data required for the API request is available, you can send the request to call the API through **curl**, **Postman**, or coding. In the response to the API used to obtain a user token, **x-subject-token** is the desired user token. This token can then be used to authenticate the calling of other APIs.

3.2 Authentication

Requests for calling an API can be authenticated using either of the following methods:

- Token-based authentication: Requests are authenticated using a token.
- AK/SK-based authentication: Requests are authenticated by encrypting the request body using an AK/SK pair. AK/SK-based authentication is recommended because it is more secure than token-based authentication.

Token-based Authentication

NOTE

The validity period of a token is 24 hours. When using a token for authentication, cache it to prevent frequently calling the IAM API used to obtain a user token.

A token specifies temporary permissions in a computer system. During API authentication using a token, the token is added to requests to get permissions for calling the API.

You can obtain a token by calling the Obtaining User Token API. When you call the API, set **auth.scope** in the request body to **project**.

```
{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "username",
                    "password": "********",
                    "domain": {
                        "name": "domainname"
                    }
                }
            }
        },
        "scope": {
            "project": {
                "name": "xxxxxxxx"
            }
        }
    }
}
```

After a token is obtained, the **X-Auth-Token** header field must be added to requests to specify the token when calling other APIs. For example, if the token is **ABCDEFJ....**, **X-Auth-Token: ABCDEFJ....** can be added to a request as follows:

```
POST https://{{endpoint}}/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

AK/SK-based Authentication

NOTE

AK/SK-based authentication supports API requests with a body not larger than 12 MB. For API requests with a larger body, token-based authentication is recommended.

In AK/SK-based authentication, AK/SK is used to sign requests and the signature is then added to the requests for authentication.

- AK: access key ID, which is a unique identifier used in conjunction with a secret access key to sign requests cryptographically.
- SK: secret access key used in conjunction with an AK to sign requests cryptographically. It identifies a request sender and prevents the request from being modified.

In AK/SK-based authentication, you can use an AK/SK to sign requests based on the signature algorithm or using the signing SDK. For details about how to sign requests and use the signing SDK, see **API Request Signing Guide**.

NOTE

The signing SDK is only used for signing requests and is different from the SDKs provided by services.

3.3 Response

Status Code

After sending a request, you will receive a response, including a status code, response header, and response body.

A status code is a group of digits, ranging from 1xx to 5xx. It indicates the status of a request. For more information, see **Status Codes**.

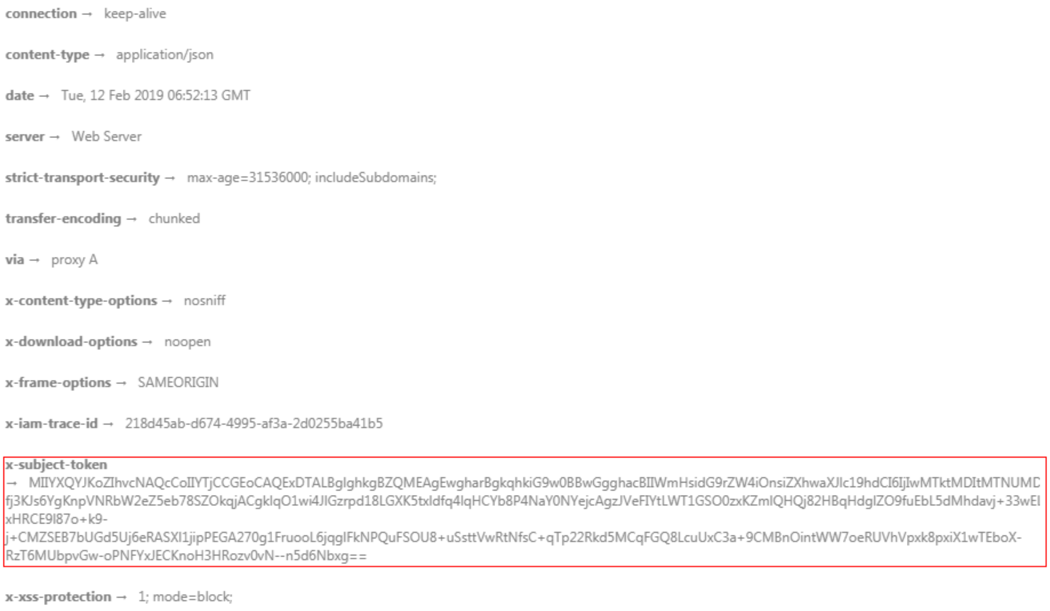
For example, if status code **201** is returned for calling the API used to obtain a user token, the request is successful.

Response Header

Similar to a request, a response also has a header, for example, **Content-Type**.

Figure 3-1 shows the response header fields for the API used to obtain a user token. The **x-subject-token** header field is the desired user token. This token can then be used to authenticate the calling of other APIs.

Figure 3-1 Header fields of the response to the request for obtaining a user token



```
connection →  keep-alive

content-type →  application/json

date →  Tue, 12 Feb 2019 06:52:13 GMT

server →  Web Server

strict-transport-security →  max-age=31536000; includeSubdomains;

transfer-encoding →  chunked

via →  proxy A

x-content-type-options →  nosniff

x-download-options →  noopen

x-frame-options →  SAMEORIGIN

x-iam-trace-id →  218d45ab-d674-4995-af3a-2d0255ba41b5

x-subject-token
→  MIIYXQYJKoZIhvcNAQcCoIIYTjCCGEoCAQExDTALBglghkgBZQMEAgEwgharBgkqhkiG9w0BBwGgghacBIIWmHsidG9rZW4iOnsiZXhwaXJlc19hdCI6IjIwMTktMDItMTNUMD
fj3KJs6YgKnpVNRbW2eZ5eb78SZOkqjACgklqO1wi4JIGzrpd18LGXK5txldfq4lqHCYb8P4NaY0NYejcAgzJVeFIYtLWT1GSO0zxKZmIQHQj82HBqHdglZO9fuEbL5dMhdavj+33wEI
xHRCE9I87o+k9-
j+CMZSEB7bUGd5Uj6eRASXl1jipPEGA270g1FruooL6jqglFkNPQuFSOU8+uSsttVwRtNfsC+qTp22Rkd5MCqFGQ8LcuUxC3a+9CMBnOintWW7oeRUVhVpxk8pxiX1wTEboX-
RzT6MUbpvGw-oPNFYxJECKnoH3HRozv0vN--n5d6Nbxg==

x-xss-protection →  1; mode=block;
```

(Optional) Response Body

The body of a response is often returned in structured format as specified in the **Content-Type** header field. The response body transfers content except the response header.

The following is part of the response body for the API used to obtain a user token.

```
{
    "token": {
        "expires_at": "2019-02-13T06:52:13.855000Z",
        "methods": [
            "password"
        ],
        "catalog": [
            {
                "endpoints": [
                    {
                        "region_id": "az-01",
......
```

If an error occurs during API calling, an error code and a message will be displayed. The following shows an error response body.

```
{
    "error_msg": "The format of message is error",
    "error_code": "AS.0001"
}
```

In the response body, **error_code** is an error code, and **error_msg** provides information about the error.

4 Getting Started

Overview

This topic describes how to invoke a number of Cloud Eye APIs to create an alarm rule for the ECS CPU usage.

NOTE

The validity period of a token obtained from IAM is 24 hours. If you want to use a token for authentication, cache it to avoid frequently calling the IAM API.

Creation Procedure

1. **Obtain the user token.**
2. **Query the list of metrics that can be monitored.**
3. **Create an alarm rule.**

Procedure

1. Obtain the user token.

   Send **POST https://***IAM endpoint***/v3/auth/tokens**.

   Add **Content-Type:application/json** to the request headers.

   The request body is as follows:

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "James",
          "password": "**********",
          "domain": {
            "name": "A-Company"
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "XXX",
    "domain": {
     "name": "A-Company"
    }
   }
  }
 }
}
```

You need to specify the following parameters:

- **user.name**: username, which is set based on the obtained token body
- **password**: login password
- **domain.name:**: name of the account to which the user belongs. If the account is used to obtain the token, values of **user.name** of the account and **domain.name** are the same. In this case, enter the **user.name** value. Otherwise, enter the domain name to which the account belongs.
- **project.name**: region

  NOTE

  Obtain **X-Subject-Token** from the response header, that is, the signed token.

2. Query the list of metrics that can be monitored.

   Send **GET https://***Cloud Eye endpoint***/V1.0/{project_id}/metrics**.

   Add the value of **X-Auth-Token** obtained in **1** to the request header.

   After the request is successfully responded, the **metrics** information is returned, such as **"metric_name": "cpu_util"** in the following figure.

```
{
  "metrics": [
    {
      "namespace": "SYS.ECS",
      "dimensions": [
        {
          "name": "instance_id",
          "value": "d9112af5-6913-4f3b-bd0a-3f96711e004d"
        }
      ],
      "metric_name": "cpu_util",
      "unit": "%"
    }
  ],
  "meta_data": {
    "count": 1,
    "marker": "SYS.ECS.cpu_util.instance_id:d9112af5-6913-4f3b-bd0a-3f96711e004d",
    "total": 7
  }
}
```

   If the request fails, an error code and error information are returned. For details, see **Error Codes**.

3. Create an alarm rule.

   Send **POST https://***Cloud Eye endpoint***/V1.0/{project_id}/alarms**.

   Specify the following parameters in the request body:

```
{
  "alarm_name": "alarm-rp0E",  //Alarm rule name (mandatory, string)
  "alarm_description": "",
  "metric": {
    "namespace": "SYS.ECS",  //Namespace (mandatory, string)
    "dimensions": [
      {
        "name": "instance_id",
        "value": "33328f02-3814-422e-b688-bfdba93d4051"
```

```
            }
        ],
        "metric_name": "cpu_util"   //Metric name (mandatory, string)
    },
    "condition": {
        "period": 300,      //Monitoring period (mandatory, integer)
        "filter": "average",     //Data rollup method (mandatory, string)
        "comparison_operator": ">=",    //Operator of the alarm threshold (mandatory, string)
        "value": 80, //Threshold (mandatory, string)
        "unit": "%",  //Data unit (mandatory, string)
        "count": 1
    },
    "alarm_enabled": true,
    "alarm_action_enabled": true,
    "alarm_level": 2,
    "alarm_actions": [
        {
            "type": "notification",
            "notificationList": [ ]
        }
    ],
    "ok_actions": [
        {
            "type": "notification",
            "notificationList": [ ]
        }
    ]
}
```

If the request is responded, the alarm rule ID is returned.

```
{
    "alarm_id":"al1450321795427dR8p5mQBo"
}
```

If the request fails, an error code and error information are returned. For details, see **Error Codes**.

You can query, enable, disable, or delete alarm rules based on the alarm rule ID obtained in **3**.

5 API Description

5.1 API Version Management

5.1.1 Querying All API Versions

Function

This API is used to query all API versions supported by Cloud Eye.

URI

GET /

Request

Example request

```
GET https://{Cloud Eye endpoint}/
```

Response

- Response parameters

Table 5-1 Response parameters

Parameter	Type	Description
versions	Array of objects	Specifies the list of all versions. **Table 5-2** describes the parameters.

Table 5-2 versions field data structure description

Parameter	Type	Description
id	String	Specifies the version ID, for example, v1.
links	Array of objects	Specifies the API URL. For details, see **Table 5-3**.
version	String	Specifies the API version. If the APIs of this version support microversions, set this parameter to the supported maximum microversion. If the microversion is not supported, leave this parameter blank.
status	String	Specifies the version status. Possible values are as follows: **CURRENT**: indicates a primary version. **SUPPORTED**: indicates an old version but is still supported. **DEPRECATED**: indicates a deprecated version which may be deleted later.
updated	String	Specifies the version release time, which must be the UTC time. For example, the release time of v1 is 2014-06-28T12:20:21Z.
min_version	String	If the APIs of this version support microversions, set this parameter to the supported minimum microversion. If not, leave this parameter blank.

Table 5-3 links field data structure description

Parameter	Type	Description
href	String	Specifies the reference address of the current API version.
rel	String	Specifies the relationship between the current API version and the referenced address.

- Example response

```
{
  "versions": [
    {
      "id": "V1.0",
      "links": [
        {
          "href": "https://x.x.x.x/V1.0/",
          "rel": "self"
        }
      ],
      "min_version": "",
      "status": "CURRENT",
      "updated": "2018-09-30T00:00:00Z",
   "version": ""
  }
 ]
}
```

Returned Values

- Normal

  200

- Abnormal

Returned Value	Description
400 Bad Request	Request error
401 Unauthorized	The authentication information is not provided or is incorrect.
403 Forbidden	You are forbidden to access the page requested.
408 Request Timeout	The request timed out.
429 Too Many Requests	Concurrent requests are excessive.
500 Internal Server Error	Failed to complete the request because of an internal service error.
503 Service Unavailable	The service is currently unavailable.

Error Code

For details, see **Error Codes**.

5.1.2 Querying a Specified API Version

Function

This API is used to query a specified API version supported by Cloud Eye.

URI

GET /{api_version}

- Parameter description

Table 5-4 Parameter description

Parameter	Mandatory	Description
api_version	Yes	Specifies the API version.

- Example

```
GET https://{Cloud Eye endpoint}/V1.0
```

Request

None

Response

- Response parameters

Table 5-5 Response parameters

Parameter	Type	Description
version	Objects	Specifies the list of all versions. For details, see **Table 5-6**.

Table 5-6 versions field data structure description

Parameter	Type	Description
id	String	Specifies the version ID, for example, v1.
links	Array of objects	Specifies the API URL. For details, see **Table 5-7**.
version	String	Specifies the API version. If the APIs of this version support microversions, set this parameter to the supported maximum microversion. If the microversion is not supported, leave this parameter blank.
status	String	Specifies the version status. Possible values are as follows: **CURRENT**: indicates a primary version. **SUPPORTED**: indicates an old version but is still supported. **DEPRECATED**: indicates a deprecated version which may be deleted later.
updated	String	Specifies the version release time, which must be the UTC time. For example, the release time of v1 is 2014-06-28T12:20:21Z.
min_version	String	If the APIs of this version support microversions, set this parameter to the supported minimum microversion. If not, leave this parameter blank.

Table 5-7 links field data structure description

Parameter	Type	Description
href	String	Specifies the reference address of the current API version.
rel	String	Specifies the relationship between the current API version and the referenced address.

- Example response

```
{
  "version": {
     "id": "V1.0",
     "links": [
       {
         "href": "https://x.x.x.x/V1.0/",
         "rel": "self"
       }
     ],
     "min_version": "",
     "status": "CURRENT",
     "updated": "2018-09-30T00:00:00Z",
     "version": ""
   }
}
```

Returned Value

- Normal

  200

- Abnormal

Returned Value	Description
400 Bad Request	Request error
401 Unauthorized	The authentication information is not provided or is incorrect.
403 Forbidden	You are forbidden to access the page requested.
408 Request Timeout	The request timed out.
429 Too Many Requests	Concurrent requests are excessive.
500 Internal Server Error	Failed to complete the request because of an internal service error.
503 Service Unavailable	The service is currently unavailable.

Error Code

For details, see **Error Codes**.

5.2 Metric Management

5.2.1 Querying the Metric List

Function

This API is used to query the metric list. You can specify the namespace, metric, dimension, sorting order, start records, and the maximum number of records when using this API to query metrics.

URI

GET /V1.0/{project_id}/metrics

- Parameter description

Table 5-8 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 5-9 Query parameter description

Parameter	Mandatory	Type	Description
namespace	No	String	Query the namespace of a service. For example, see **Namespace** for ECS namespace. The value must be in the **service.item** format and can contain 3 to 32 characters. **service** and **item** each must be a string that starts with a letter and contains only uppercase letters, lowercase letters, digits, and underscores (_).
metric_name	No	String	Specifies the metric ID. For example, if the **monitoring metric** of an ECS is CPU usage, **metric_name** is **cpu_util**.

Parameter	Mandatory	Type	Description
dim	No	String	Specifies the dimension. For example, the **dimension** of an ECS is **instance_id**. A maximum of three dimensions are supported, and the dimensions are numbered from 0 in **dim.{i}=key,value** format. The **key** cannot exceed 32 characters and the **value** cannot exceed 256 characters. Single dimension: **dim.0=instance_id, 6f3c6f91-4b24-4e1b-b7d1-a94ac1cb011d** Multiple dimensions: **dim.0=key,value&dim.1=key,value**
start	No	String	Specifies the paging start value. The format is **namespace.metric_name.key:value**. Example: **start=SYS.ECS.cpu_util.instance_id:d9112af5-6913-4f3b-bd0a-3f96711e004d**.
limit	No	Integer	The value ranges from **1** to **1000**, and is **1000** by default. This parameter is used to limit the number of query results.
order	No	String	Specifies the result sorting method, which is sorted by timestamp. The default value is **desc**. ● **asc**: The query results are displayed in the ascending order. ● **desc**: The query results are displayed in the descending order.

- Example request

  Request example 1: Query the list of all metrics that can be monitored.

```
GET https://{Cloud Eye endpoint}/V1.0/{project_id}/metrics
```

Request example 2: Query the CPU usage of the ECS whose ID is **6f3c6f91-4b24-4e1b-b7d1-a94ac1cb011d**. Retain 10 records in descending order by timestamp.

```
GET https://{Cloud Eye endpoint}/V1.0/{project_id}/metrics?
namespace=SYS.ECS&metric_name=cpu_util&dim.0=instance_id,6f3c6f91-4b24-4e1b-b7d1-
a94ac1cb011d&limit=10&order=desc
```

Request

None

Response

- Response parameters

Table 5-10 Response parameters

Parameter	Type	Description
metrics	Array of objects	Specifies the list of metric objects. For details, see **Table 5-11**.
meta_data	Object	Specifies the metadata of query results, including the pagination information. For details, see **Table 5-13**.

Table 5-11 metrics field data structure description

Parameter	Type	Description
namespace	String	Specifies the metric namespace.
dimensions	Array of objects	Specifies the list of metric dimensions. For details, see **Table 5-12**.
metric_name	String	Specifies the metric name, such as **cpu_util**.
unit	String	Specifies the metric unit.

Table 5-12 dimensions field data structure description

Parameter	Type	Description
name	String	Specifies the dimension. For example, the ECS dimension is **instance_id**. For details, see the **key** field in **Dimension**.
value	String	Specifies the dimension value, for example, an ECS ID. The value is a string of 1 to 256 characters.

Table 5-13 meta_data field data structure description

Parameter	Type	Description
count	Integer	Specifies the number of returned results.
marker	String	Specifies the pagination marker. For example, you have queried 10 records this time and the tenth record is about **cpu_util**. In your next query, if **start** is set to **cpu_util**, you can start your query from the next metric of **cpu_util**.
total	Integer	Specifies the total number of metrics.

- Example response

```
{
    "metrics": [
        {
            "namespace": "SYS.ECS",
            "dimensions": [
                {
                    "name": "instance_id",
                    "value": "d9112af5-6913-4f3b-bd0a-3f96711e004d"
                }
            ],
            "metric_name": "cpu_util",
            "unit": "%"
        }
    ],
    "meta_data": {
        "count": 1,
        "marker": "SYS.ECS.cpu_util.instance_id:d9112af5-6913-4f3b-bd0a-3f96711e004d",
        "total": 7
    }
}
```

Returned Values

- Normal

  200

- Abnormal

Returned Values	Description
400 Bad Request	Request error.
401 Unauthorized	The authentication information is not provided or is incorrect.
403 Forbidden	You are forbidden to access the page requested.
408 Request Timeout	The request timed out.
429 Too Many Requests	Concurrent requests are excessive.
500 Internal Server Error	Failed to complete the request because of an internal service error.

Returned Values	Description
503 Service Unavailable	The service is currently unavailable.

Error Code

For details, see **Error Codes**.

5.3 Alarm Rule Management

5.3.1 Querying the Alarm Rule List

Function

This API is used to query the alarm rule list. You can specify the paging parameters to limit the number of query results displayed on a page. You can also set the sorting order of query results.

URI

GET /V1.0/{project_id}/alarms

- Parameter description

Table 5-14 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 5-15 Parameter description

Parameter	Type	Description
alarms	Array of objects	Specifies the alarm rule list.

Table 5-16 Query parameter description

Parameter	Mandatory	Type	Description
start	No	String	Specifies the first queried alarm to be displayed on a page. The value is **alarm_id**.
limit	No	Integer	The value ranges from **1** to **100**, and is **100** by default. This parameter is used to limit the number of query results.
order	No	String	Specifies the result sorting method, which is sorted by timestamp. The default value is **desc**. ● **asc**: The query results are displayed in the ascending order. ● **desc**: The query results are displayed in the descending order.

- Example

  Request example 1: Query the current alarm rule list.

  ```
  GET https://{Cloud Eye endpoint}/V1.0/{project_id}/alarms
  ```

  Request example 2: Query the alarm rule list. Start by setting **alarm_id** to **al1441967036681YkazZ0deN** and retain 10 records in the descending order of time stamps.

  ```
  GET https://{Cloud Eye endpoint}/V1.0/{project_id}/alarms?
  start=al1441967036681YkazZ0deN&limit=10&order=desc
  ```

Request

None

Response

- Response parameters

Table 5-17 Response parameters

Parameter	Type	Description
metric_alarms	Array of objects	Specifies the list of alarm objects. For details, see **Table 5-18**.

Parameter	Type	Description
meta_data	Object	Specifies the metadata of query results, including the pagination information. For details, see **Table 5-24**.

Table 5-18 metric_alarms field data structure description

Parameter	Type	Description
alarm_name	String	Specifies the alarm rule name.
alarm_description	String	Provides supplementary information about the alarm rule.
metric	Object	Specifies the alarm metric. For details, see **Table 5-19**.
condition	Object	Specifies the alarm triggering condition. For details, see **Table 5-23**.
alarm_enabled	Boolean	Specifies whether to enable the alarm rule.
alarm_level	Integer	Specifies the alarm severity. Possible values are **1**, **2** (default), **3** and **4**, which indicates critical, major, minor, and informational, respectively.
alarm_action_enabled	Boolean	Specifies whether to enable the action to be triggered by an alarm.
alarm_actions	Array of objects	Specifies the action triggered by an alarm. For details, see **Table 5-21**.
ok_actions	Array of objects	Specifies the action to be triggered after the alarm is cleared. For details, see **Table 5-22**.
alarm_id	String	Specifies the alarm rule ID.
update_time	long	Specifies the time when the alarm status changed. The value is a UNIX timestamp and the unit is ms.
alarm_state	String	Specifies the alarm status. The value can be: ● **ok**: The alarm status is normal. ● **alarm**: An alarm is generated. ● **insufficient_data**: The required data is insufficient.

Table 5-19 metric field data structure description

Parameter	Type	Description
namespace	String	Query the namespace of a service. For example, see **Namespace** for ECS namespace.
dimensions	Array of objects	Specifies the list of metric dimensions. For details, see **Table 5-20**.
metric_name	String	Specifies the metric ID. For example, if the **monitoring metric** of an ECS is CPU usage, **metric_name** is **cpu_util**.

Table 5-20 dimensions field data structure description

Parameter	Type	Description
name	String	Specifies the dimension. For example, the ECS dimension is **instance_id**, which is listed in the **key** column in **Dimension**.
value	String	Specifies the dimension value, for example, an ECS ID. The value is a string of 1 to 256 characters.

Table 5-21 alarm_actions field data structure description

Parameter	Type	Description
type	String	Specifies the alarm notification type. ● **notification**: indicates that a notification will be sent to the user. ● **autoscaling**: indicates that a scaling action will be triggered.
notificationList	Array of strings	Specifies the list of objects to be notified if the alarm status changes. **NOTE** The IDs in the list are character strings.

Table 5-22 ok_actions field data structure description

Parameter	Type	Description
type	String	Specifies the notification type when an alarm is triggered. ● **notification**: indicates that a notification will be sent to the user. ● **autoscaling**: indicates that a scaling action will be triggered.
notificationList	Array of strings	Specifies the ID list of objects to be notified if the alarm status changes. **NOTE** The IDs in the list are character strings.

Table 5-23 condition field data structure description

Parameter	Type	Description
period	Integer	Specifies the interval (seconds) for checking whether the configured alarm rules are met.
filter	String	Specifies the data rollup method. The following methods are supported: ● **average**: Cloud Eye calculates the average value of metric data within a rollup period. ● **max**: Cloud Eye calculates the maximum value of metric data within a rollup period. ● **min**: Cloud Eye calculates the minimum value of metric data within a rollup period. ● **sum**: Cloud Eye calculates the sum of metric data within a rollup period. ● **variance**: Cloud Eye calculates the variance value of metric data within a rollup period.
comparison_operator	String	Specifies the operator of alarm thresholds. Possible values are **>**, **=**, **<**, **≥**, and **≤**.
value	Double	Specifies the alarm threshold. The value ranges from **0** to **Number. MAX_VALUE (1.7976931348623157e+108)**. For detailed thresholds, see the value range of each metric in the appendix. For example, you can set ECS **cpu_util** in **Metrics** to **80**.
unit	String	Specifies the data unit. The value contains a maximum of 32 characters.

Parameter	Type	Description
count	Integer	Specifies the number of consecutive occurrence times that the alarm policy was met. The value ranges from **1** to **5**.

Table 5-24 meta_data field data structure description

Parameter	Type	Description
count	Integer	Specifies the number of returned results.
marker	String	Specifies the pagination marker. For example, you have queried 10 records this time and **alarm_id** of the tenth record is **1441967036681YkazZ0deN**. In your next query, if **start** is set to **al1441967036681YkazZ0deN**, you can start your query from the next alarm rule ID of **al1441967036681YkazZ0deN**.
total	Integer	Specifies the total number of query results.

- Example response

```
{
    "metric_alarms": [
        {
            "alarm_name": "alarm-ttttttt",
            "alarm_description": "",
            "metric": {
                "namespace": "SYS.ECS",
                "dimensions": [
                    {
                        "name": "instance_id",
                        "value": "07814c0e-59a1-4fcd-a6fb-56f2f6923046"
                    }
                ],
                "metric_name": "cpu_util"
            },
            "condition": {
                "period": 300,
                "filter": "average",
                "comparison_operator": ">=",
                "value": 0,
                "unit": "%",
                "count": 3
            },
            "alarm_enabled": true,
            "alarm_level": 2,
            "alarm_action_enabled": false,
            "alarm_id": "al15330507498596W7vmlGKL",
            "update_time": 1533050749992,
            "alarm_state": "alarm"
        },
        {
            "alarm_name": "alarm-m5rwxxxxxxx",
            "alarm_description": "",
            "metric": {
                "namespace": "SYS.ECS",
                "dimensions": [
                    {
                        "name": "instance_id",
                        "value": "30f3858d-4377-4514-9081-be5bdbf1392e"
                    }
                ],
                "metric_name": "network_incoming_bytes_aggregate_rate"
            },
            "condition": {
                "period": 300,
                "filter": "average",
                "comparison_operator": ">=",
                "value": 12,
                "unit": "B/s",
                "count": 3
            },
            "alarm_enabled": true,
            "alarm_level": 2,
            "alarm_action_enabled": true,
            "alarm_actions": [
                {
                    "type": "notification",
                    "notificationList": [
                        "urn:smn:region:68438a86d98e427e907e0097b7e35d48:test0315"
                    ]
                }
            ],
            "ok_actions": [
                {
                    "type": "notification",
                    "notificationList": [
                        "urn:smn:region:68438a86d98e427e907e0097b7e35d48:test0315"
                    ]
                }
            ],
            "alarm_id": "al1533031226533nKJexAlbq",
            "update_time": 1533204036276,
            "alarm_state": "ok"
        }
    ],
    "meta_data": {
        "count": 2,
        "marker": "al1533031226533nKJexAlbq",
        "total": 389
    }
}
```

Returned Values

- Normal

  200

- Abnormal

Returned Value	Description
400 Bad Request	Request error.
401 Unauthorized	The authentication information is not provided or is incorrect.
403 Forbidden	You are forbidden to access the page requested.
408 Request Timeout	The request timed out.
429 Too Many Requests	Concurrent requests are excessive.

Returned Value	Description
500 Internal Server Error	Failed to complete the request because of an internal service error.
503 Service Unavailable	The service is currently unavailable.

Error Code

For details, see **Error Codes**.

5.3.2 Querying an Alarm Rule

Function

This API is used to query an alarm rule based on the alarm rule ID.

URI

GET /V1.0/{project_id}/alarms/{alarm_id}

- Parameter description

Table 5-25 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
alarm_id	Yes	Specifies the alarm rule ID.

- Example

```
GET https://{Cloud Eye endpoint}/V1.0/{project_id}/alarms/al1441967036681YkazZ0deN
```

Request

None

Response

- Response parameters

Parameter	Type	Description
metric_alarms	Array of objects	Specifies the list of alarm objects. For details, see **Table 5-26**.

Table 5-26 metric_alarms field data structure description

Parameter	Type	Description
alarm_name	String	Specifies the alarm rule name.
alarm_description	String	Provides supplementary information about the alarm rule.
metric	Array of objects	Specifies the alarm metric. For details, see **Table 5-27**.
condition	Array of objects	Specifies the alarm triggering condition. For details, see **Table 5-31**.
alarm_enabled	Boolean	Specifies whether to enable the alarm rule.
alarm_level	Integer	Specifies the alarm severity. Possible values are **1**, **2**, **3** and **4**, which indicates critical, major, minor, and informational, respectively.
alarm_action_enabled	Boolean	Specifies whether to enable the action to be triggered by an alarm.
alarm_actions	Array of objects	Specifies the action triggered by an alarm. For details, see **Table 5-29**.
ok_actions	Array of objects	Specifies the action to be triggered after the alarm is cleared. For details, see **Table 5-30**.
alarm_id	String	Specifies the alarm rule ID.
update_time	long	Specifies the time when the alarm status changed. The value is a UNIX timestamp and the unit is ms.
alarm_state	String	Specifies the alarm status. The value can be: ● **ok**: The alarm status is normal. ● **alarm**: An alarm is generated. ● **insufficient_data**: The required data is insufficient.

Table 5-27 metric field data structure description

Parameter	Type	Description
namespace	String	Query the namespace of a service. For example, see **Namespace** for ECS namespace.
dimensions	Array of objects	Specifies the list of metric dimensions. For details, see **Table 5-28**.

Parameter	Type	Description
metric_name	String	Specifies the metric ID. For example, if the **monitoring metric** of an ECS is CPU usage, **metric_name** is **cpu_util**.

Table 5-28 dimensions field data structure description

Parameter	Type	Description
name	String	Specifies the dimension. For example, the ECS dimension is **instance_id**, which is listed in the **key** column in **Dimension**.
value	String	Specifies the dimension value, for example, an ECS ID. The value is a string of 1 to 256 characters.

Table 5-29 alarm_actions field data structure description

Parameter	Type	Description
type	String	Specifies the alarm notification type. - **notification**: indicates that a notification will be sent to the user. - **autoscaling**: indicates that a scaling action will be triggered.
notificationList	Array of strings	Specifies the list of objects to be notified if the alarm status changes. **NOTE** The IDs in the list are character strings.

Table 5-30 ok_actions field data structure description

Parameter	Type	Description
type	String	Specifies the notification type when an alarm is triggered. - **notification**: indicates that a notification will be sent to the user. - **autoscaling**: indicates that a scaling action will be triggered.

Parameter	Type	Description
notificationList	Array of strings	Specifies the list of objects to be notified if the alarm status changes. **NOTE** The IDs in the list are character strings.

Table 5-31 condition field data structure description

Parameter	Type	Description
period	Integer	Specifies the interval (seconds) for checking whether the configured alarm rules are met.
filter	String	Specifies the data rollup method. The following methods are supported: ● **average**: Cloud Eye calculates the average value of metric data within a rollup period. ● **max**: Cloud Eye calculates the maximum value of metric data within a rollup period. ● **min**: Cloud Eye calculates the minimum value of metric data within a rollup period. ● **sum**: Cloud Eye calculates the sum of metric data within a rollup period. ● **variance**: Cloud Eye calculates the variance value of metric data within a rollup period.
comparison_operator	String	Specifies the operator of alarm thresholds. Possible values are **>**, **=**, **<**, **≥**, and **≤**.
value	Double	Specifies the alarm threshold. The value ranges from **0** to **Number. MAX_VALUE (1.7976931348623157e+108)**. For detailed thresholds, see the value range of each metric in the appendix. For example, you can set ECS **cpu_util** in **Metrics** to **80**.
unit	String	Specifies the data unit. The value contains a maximum of 32 characters.
count	Integer	Specifies the number of consecutive occurrence times that the alarm policy was met. The value ranges from **1** to **5**.

- Example response

```
{
"metric_alarms":
  [
  {
   "alarm_name":"alarm-ipwx",
   "alarm_description":"",
   "metric":
     {
      "namespace":"SYS.ELB",
      "dimensions":
      [
       {
        "name":"lb_instance_id",
        "value":"44d06d10-bce0-4237-86b9-7b4d1e7d5621"
       }
      ],
      "metric_name":"m8_out_Bps"
      },
    "condition":
     {
      "period":300,
      "filter":"sum",
      "comparison_operator":">=",
      "value":0,
      "unit":"",
      "count":1
      },
    "alarm_enabled":true,
    "alarm_level": 2,
    "alarm_action_enabled":true,
    "alarm_actions":
     [
      {
       "type":"notification",
       "notificationList":["urn:smn:region:68438a86d98e427e907e0097b7e35d48:sd"]
      }
     ],
    "ok_actions":
     [
      {
       "type":"notification",
       "notificationList":["urn:smn:region:68438a86d98e427e907e0097b7e35d48:sd"]
      }
     ],
    "alarm_id":"al1498096535573r8DNy7Gyk",
    "update_time":1498100100000,
    "alarm_state":"alarm"
   }
  ]
}
```

Returned Values

- Normal

  200

- Abnormal

Returned Value	Description
400 Bad Request	Request error.
401 Unauthorized	The authentication information is not provided or is incorrect.
403 Forbidden	You are forbidden to access the page requested.
408 Request Timeout	The request timed out.
429 Too Many Requests	Concurrent requests are excessive.

Returned Value	Description
500 Internal Server Error	Failed to complete the request because of an internal service error.
503 Service Unavailable	The service is currently unavailable.

Error Code

For details, see **Error Codes**.

5.3.3 Enabling or Disabling an Alarm Rule

Function

This API is used to enable or disable an alarm rule.

URI

PUT /V1.0/{project_id}/alarms/{alarm_id}/action

- Parameter description

Table 5-32 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
alarm_id	Yes	Specifies the alarm rule ID.

- Example

```
PUT https://{Cloud Eye endpoint}/V1.0/{project_id}/alarms/al1441967036681YkazZ0deN/action
```

Request

- Request parameters

Table 5-33 Request parameters

Parameter	Mandatory	Type	Description
alarm_enabled	Yes	Boolean	Specifies whether the alarm rule is enabled. - **true**: indicates that the alarm rule is enabled. - **false**: indicates that the alarm rule is disabled.

- Example request

```
{
    "alarm_enabled":true
}
```

Response

The response has no message body.

Returned Values

- Normal

  204

- Abnormal

Returned Value	Description
400 Bad Request	Request error.
401 Unauthorized	The authentication information is not provided or is incorrect.
403 Forbidden	You are forbidden to access the page requested.
408 Request Timeout	The request timed out.
429 Too Many Requests	Concurrent requests are excessive.
500 Internal Server Error	Failed to complete the request because of an internal service error.
503 Service Unavailable	The service is currently unavailable.

Error Code

For details, see **Error Codes**.

5.3.4 Deleting an Alarm Rule

Function

This API is used to delete an alarm rule.

URI

DELETE /V1.0/{project_id}/alarms/{alarm_id}

- Parameter description

Table 5-34 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
alarm_id	Yes	Specifies the alarm rule ID.

- Example

```
DELETE https://{Cloud Eye endpoint}/V1.0/{project_id}/alarms/al1441967036681YkazZ0deN
```

Request

The request has no message body.

Response

The response has no message body.

Returned Values

- Normal

  204

- Abnormal

Returned Value	Description
400 Bad Request	Request error.
401 Unauthorized	The authentication information is not provided or is incorrect.
403 Forbidden	You are forbidden to access the page requested.
408 Request Timeout	The request timed out.
429 Too Many Requests	Concurrent requests are excessive.

Returned Value	Description
500 Internal Server Error	Failed to complete the request because of an internal service error.
503 Service Unavailable	The service is currently unavailable.

Error Code

For details, see **Error Codes**.

5.3.5 Creating an Alarm Rule

Function

This API is used to create an alarm rule.

URI

POST /V1.0/{project_id}/alarms

- Parameter description

Table 5-35 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

- Example

```
POST https://{Cloud Eye endpoint}/V1.0/{project_id}/alarms
```

Request

- Request parameters

Table 5-36 Request parameters

Parameter	Mandatory	Type	Description
alarm_name	Yes	String	Specifies the alarm rule name. The value can contain 1 to 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Mandatory	Type	Description
alarm_description	No	String	Provides supplementary information about the alarm rule. Enter 0 to 256 characters.
metric	Yes	Object	Specifies the alarm metric. For details, see **Table 5-37**.
condition	Yes	Object	Specifies the alarm triggering condition. For details, see **Table 5-42**.
alarm_enabled	No	Boolean	Specifies whether to enable the alarm. The default value is **true**.
alarm_action_enabled	No	Boolean	Specifies whether to enable the action to be triggered by an alarm. The default value is **true**. **NOTE** If you set **alarm_action_enabled** to **true**, you must specify either **alarm_actions** or **ok_actions**. (You do not need to configure the deprecated parameter **insufficientdata_actions**.) If **alarm_actions** and **ok_actions** coexist, their **notificationList** values must be the same. (You do not need to configure the deprecated parameter **insufficientdata_actions**.)
alarm_level	No	Integer	Specifies the alarm severity. Possible values are **1**, **2**, **3** and **4**, which indicates critical, major, minor, and informational, respectively.

alarm_actions	No	Arrays of objects	Specifies the action triggered by an alarm.
			An example structure is as follows:
			{
			"type": "notification","notificationList" : ["urn:smn:region: 68438a86d98e427e907e0097b7e35d47:sd"]
			}
			For details, see **Table 5-39**.

Parameter	Mandatory	Type	Description
ok_actions	No	Arrays of objects	Specifies the action to be triggered after the alarm is cleared. Its structure is: { "type": "notification","notificationList" : ["urn:smn:region: 68438a86d98e427e907e0097b7e35d47:sd"] } For details, see **Table 5-40**.

Table 5-37 metric field data structure description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Specifies the namespace of a service. For example, see **Namespace** for ECS namespace. The value must be in the **service.item** format and can contain 3 to 32 characters. **service** and **item** each must be a string that starts with a letter and contains only letters, digits, and underscores (_).
dimensions	Yes	Arrays of objects	Specifies the list of metric dimensions. For details, see **Table 5-38**.
metric_name	Yes	String	Specifies the metric name. The value can be a string of 1 to 64 characters, must start with a letter, and contain only letters, digits, and underscores (_). For details, see the metric name queried in **Querying the Metric List**.
resource_group_id	No	String	Specifies the resource group ID selected during the alarm rule creation, for example, **rg1603786526428bWbVmk4rP**.

Table 5-38 dimensions field data structure description

Parameter	Mandatory	Type	Description
name	No	String	Specifies the dimension. For example, the ECS dimension is **instance_id**, which is listed in the **key** column in **Dimension**. The value can be a string of 1 to 32 characters, must start with a letter, and contain only letters, digits, underscores (_), and hyphens (-).
value	No	String	The value can be a string of 1 to 256 characters, must start with a letter or a digit, and contain only letters, digits, underscores (_), and hyphens (-).

Table 5-39 alarm_actions field data structure description

Parameter	Mandatory	Type	Description
type	Yes	String	Specifies the alarm notification type. ● **notification**: indicates that a notification will be sent to the user. ● **autoscaling**: indicates that a scaling action will be triggered.

Parameter	Mandatory	Type	Description
notificationList	Yes	Arrays of strings	Specifies the list of objects to be notified if the alarm status changes. You can configure a maximum of 5 object IDs. You can obtain the **topicUrn** value from SMN in the following format: urn:smn:([a-z]|[A-Z]|[0-9]|\-){1,32}:([a-z]|[A-Z]|[0-9]){32}:([a-z]|[A-Z]|[0-9]|\-|_){1,256}. If **type** is set to **notification**, the value of **notificationList** cannot be empty. If **type** is set to **autoscaling**, the value of **notificationList** must be **[]**. **NOTE** ● To make the AS alarm rules take effect, you must bind scaling policies. For details, see the *Auto Scaling API Reference*. ● If you set **alarm_action_enabled** to **true**, you must specify either **alarm_actions** or **ok_actions**. (You do not need to configure the deprecated parameter **insufficientdata_actions**.) ● If **alarm_actions** and **ok_actions** coexist, their **notificationList** values must be the same. (You do not need to configure the deprecated parameter **insufficientdata_actions**.) ● The IDs in the list are character strings.

Table 5-40 ok_actions field data structure description

Parameter	Mandatory	Type	Description
type	Yes	String	Specifies the notification type when an alarm is triggered. ● **notification**: indicates that a notification will be sent to the user. ● **autoscaling**: indicates that a scaling action will be triggered.

Parameter	Mandatory	Type	Description
notificationList	Yes	Arrays of objects	Specifies the list of objects to be notified if the alarm status changes. You can configure a maximum of 5 object IDs. You can obtain the **topicUrn** value from SMN in the following format: urn:smn:([a-z]|[A-Z]|[0-9]|\-){1,32}:([a-z]|[A-Z]|[0-9]){32}:([a-z]|[A-Z]|[0-9]|\-|_){1,256}. **NOTE** If you set **alarm_action_enabled** to **true**, you must specify either **alarm_actions** or **ok_actions**. (You do not need to configure the deprecated parameter **insufficientdata_actions**.) If **alarm_actions** and **ok_actions** coexist, their **notificationList** values must be the same. (You do not need to configure the deprecated parameter **insufficientdata_actions**.)

Table 5-41 insufficientdata_actions field data structure description

Parameter	Mandatory	Type	Description
type	Yes	String	Specifies the notification type when an alarm is triggered. ● **notification**: indicates that a notification will be sent to the user. ● **autoscaling**: indicates that a scaling action will be triggered.
notificationList	Yes	Arrays of objects	Specifies the list of objects to be notified if the alarm status changes. You can add up to 5 objects. You can obtain the **topicUrn** value from SMN in the following format: urn:smn:([a-z]|[A-Z]|[0-9]|\-){1,32}:([a-z]|[A-Z]|[0-9]){32}:([a-z]|[A-Z]|[0-9]|\-|_){1,256}. **NOTE** ● If you set **alarm_action_enabled** to **true**, you must specify either **alarm_actions** or **ok_actions**. (You do not need to configure the deprecated parameter **insufficientdata_actions**.) ● If **alarm_actions** and **ok_actions** coexist, their **notificationList** values must be the same. (You do not need to configure the deprecated parameter **insufficientdata_actions**.) ● The IDs in the list are character strings.

Table 5-42 condition field data structure description

Parameter	Mandatory	Type	Description
period	Yes	Integer	Specifies the period during which Cloud Eye determines whether to trigger an alarm. Unit: second Possible values are **1**, **300**, **1200**, **3600**, **14400**, and **86400**. **NOTE** ● If you set **period** to **1**, Cloud Eye uses raw data to determine whether to trigger an alarm.
filter	Yes	String	Specifies the data rollup method. Possible values are **max**, **min**, **average**, **sum**, or **variance**.
comparison_operator	Yes	String	Specifies the operator of alarm thresholds. Possible values are **>**, **=**, **<**, **≥**, and **≤**.
value	Yes	Double	Specifies the alarm threshold. The value ranges from **0** to **Number. MAX_VALUE (1.7976931348623157e+108)**. For detailed thresholds, see the value range of each metric in the appendix. For example, you can set ECS **cpu_util** in **Metrics**to **80**.
unit	No	String	Specifies the data unit. The value contains a maximum of 32 characters.
count	Yes	Integer	Specifies the number of consecutive occurrence times that the alarm policy was met. The value ranges from **1** to **5**.

- Example request

```
{
    "alarm_name": "alarm-rp0E",
    "alarm_description": "",
    "metric": {
        "namespace": "SYS.ECS",
        "dimensions": [
            {
                "name": "instance_id",
                "value": "33328f02-3814-422e-b688-bfdba93d4051"
            }
        ],
        "metric_name": "network_outgoing_bytes_rate_inband"
    },
    "condition": {
        "period": 300,
        "filter": "average",
        "comparison_operator": ">=",
        "value": 6,
        "unit": "B/s",
        "count": 1
   },
    "alarm_enabled": true,
    "alarm_action_enabled": true,
    "alarm_level": 2,
    "alarm_actions": [
        {
            "type": "notification",
            "notificationList": ["urn:smn:region:68438a86d98e427e907e0097b7e35d48:sd"]
        }
    ],
    "ok_actions": [
        {
            "type": "notification",
            "notificationList": ["urn:smn:region:68438a86d98e427e907e0097b7e35d48:sd"]
        }
    ],
    "insufficientdata_actions": [
        {
            "type": "notification",
            "notificationList": ["urn:smn:region:68438a86d98e427e907e0097b7e35d48:sd"]
        }
    ]
}
```

Response

- Response parameters

Table 5-43 Response parameters

Parameter	Type	Description
alarm_id	String	Specifies the alarm rule ID.

- Example response

```
{
    "alarm_id":"al1450321795427dR8p5mQBo"
}
```

Returned Values

- Normal

  201

- Abnormal

Returned Values	Description
400 Bad Request	Request error.
401 Unauthorized	The authentication information is not provided or is incorrect.
403 Forbidden	You are forbidden to access the page requested.
408 Request Timeout	The request timed out.
429 Too Many Requests	Concurrent requests are excessive.
500 Internal Server Error	Failed to complete the request because of an internal service error.

Returned Values	Description
503 Service Unavailable	The service is currently unavailable.

Error Code

For details, see **Error Codes**.

5.4 Monitoring Data Management

5.4.1 Querying Monitoring Data

Function

This API is used to query the monitoring data at a specified granularity for a specified metric in a specified period of time. You can specify the dimension of data to be queried.

URI

GET /V1.0/{project_id}/metric-data?namespace={namespace}&metric_name={metric_name}&dim.{i}=key,value&from={from}&to={to}&period={period}&filter={filter}

- Parameter description

Table 5-44 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 5-45 Query parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Specifies the namespace of a service. For example, see **Namespace** for ECS namespace. The value must be in the **service.item** format and can contain 3 to 32 characters. **service** and **item** each must be a string that starts with a letter and contains only uppercase letters, lowercase letters, digits, and underscores (_).
metric_name	Yes	String	Specifies the metric name. You can obtain the metric names of existing alarm rules by referring to **Querying the Metric List**.
from	Yes	String	Specifies the start time of the query. The value is a UNIX timestamp and the unit is ms. Set the value of **from** to at least one period earlier than the current time. Rollup aggregates the raw data generated within a period to the start time of the period. Therefore, if values of **from** and **to** are within a period, the query result will be empty due to the rollup failure. Take the 5-minute period as an example. If it is 10:35 now, the raw data generated between 10:30 and 10:35 will be aggregated to 10:30. Therefore, in this example, if the value of **period** is 5 minutes, the value of **from** should be 10:30 or earlier. **NOTE** Cloud Eye rounds up the value of **from** based on the level of granularity required to perform the rollup.

Parameter	Mandatory	Type	Description
to	Yes	String	Specifies the end time of the query. The value is a UNIX timestamp and the unit is ms. The value of parameter **from** must be earlier than that of parameter **to**.
period	Yes	Integer	Specifies how often Cloud Eye aggregates data. Possible values are: ● **1**: Cloud Eye performs no aggregation and displays raw data. ● **300**: Cloud Eye aggregates data every 5 minutes. ● **1200**: Cloud Eye aggregates data every 20 minutes. ● **3600**: Cloud Eye aggregates data every 1 hour. ● **14400**: Cloud Eye aggregates data every 4 hours. ● **86400**: Cloud Eye aggregates data every 24 hours.

Parameter	Mandatory	Type	Description
filter	Yes	String	Specifies the data rollup method. The following methods are supported: ● **average**: Cloud Eye calculates the average value of metric data within a rollup period. ● **max**: Cloud Eye calculates the maximum value of metric data within a rollup period. ● **min**: Cloud Eye calculates the minimum value of metric data within a rollup period. ● **sum**: Cloud Eye calculates the sum of metric data within a rollup period. ● **variance**: Cloud Eye calculates the variance value of metric data within a rollup period. **NOTE** Rollup uses a rollup method to aggregate raw data generated within a specific period. Take the 5-minute period as an example. If it is 10:35 now, the raw data generated between 10:30 and 10:35 will be aggregated to 10:30.
dim	Yes	String	Currently, a maximum of three metric dimensions are supported, and the dimensions are numbered from 0 in the **dim.{i}=key,value** format. The **key** cannot exceed 32 characters and the **value** cannot exceed 256 characters. The following dimensions are only examples. For details about dimensions of each service, see the description of each service, for example, instance_id of the ECS in **Dimension**. Single dimension: **dim.0=instance_id,i-12345** Multiple dimensions: **dim.0=instance_id,i-12345&dim.1=instance_name,i-1234**

- Example:

Request example 1: View the CPU usage of ECS whose ID is **6f3c6f91-4b24-4e1b-b7d1-a94ac1cb011d** from 2019-04-30 20:00:00 to 2019-04-30 22:00:00. The monitoring interval is 20 minutes.

```
GET https://{Cloud Eye endpoint}/V1.0/{project_id}/metric-data?
namespace=SYS.ECS&metric_name=cpu_util&dim.0=instance_id,6f3c6f91-4b24-4e1b-b7d1-
a94ac1cb011d&from=1556625600000&to=1556632800000&period=1200&filter=min
```

Request

None

Response

- Response parameters

Table 5-46 Response parameters

Parameter	Type	Description
datapoints	Array of objects	Specifies the metric data list. For details, see Table 5-47. Since Cloud Eye rounds up the value of **from** based on the level of granularity for data query, **datapoints** may contain more data points than expected.
metric_name	String	Specifies the metric ID. For example, if the monitoring metric of an ECS is CPU usage, **metric_name** is **cpu_util**.

Table 5-47 datapoints field data structure description

Parameter	Type	Description
average	double	Specifies the metric value. The value of this parameter is the same as that of parameter **filter**. The value can be **max/min/average/sum/variance**.
timestamp	long	Specifies the time when the metric is collected. It is a UNIX timestamp in milliseconds.
unit	String	Specifies the metric unit.

- Example response

```
{
    "datapoints": [
        {
            "average": 0,
            "timestamp": 1442341200000,
            "unit": "Count"
        }
    ],
    "metric_name": "cpu_util"
}
```

Returned Values

- Normal

  200

- Abnormal

Returned Values	Description
400 Bad Request	Request error.
401 Unauthorized	The authentication information is not provided or is incorrect.
403 Forbidden	You are forbidden to access the page requested.
408 Request Timeout	The request timed out.
429 Too Many Requests	Concurrent requests are excessive.
500 Internal Server Error	Failed to complete the request because of an internal service error.
503 Service Unavailable	The service is currently unavailable.

Error Code

For details, see **Error Codes**.

5.4.2 Adding Monitoring Data

Function

This API is used to add one or more pieces of custom metric monitoring data to solve the problem that the system metrics cannot meet specific service requirements.

URI

POST /V1.0/{project_id}/metric-data

- Parameter description

Table 5-48 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request

NOTICE

1. The size of a POST request cannot exceed 512 KB. Otherwise, the request will be denied.
2. The period for sending POST requests must be shorter than the minimum aggregation period. Otherwise, the aggregated data will be noncontinuous. For example, if the aggregation period is 5 minutes and the POST request sending period is 7 minutes, the data will be aggregated every 10 minutes, rather than 5 minutes.
3. Timestamp (collect_time) in the POST request body value must be within the period that starts from three days before the current time to 10 minutes after the current time. If it is not in this range, you are not allowed to insert the metric data.

- Request parameters

Table 5-49 Parameter description

Parameter	Type	Mandatory	Description
Array elements	Array of objects	Yes	Specifies whether to add one or more pieces of custom metric monitoring data.

Table 5-50 Array elements

Parameter	Mandatory	Type	Description
metric	Yes	Object	Specifies the metric data. For details, see **Table 5-51**.
ttl	Yes	Integer	Specifies the data validity period. The unit is second. The value range is 0–604,800 seconds. If the validity period expires, the data will be automatically deleted.

Parameter	Mandatory	Type	Description
collect_time	Yes	long	Specifies the time when the data was collected. The time is UNIX timestamp (ms) format. **NOTE** Since there is a latency between the client and the server, the data timestamp to be inserted should be within the period that starts from three days before the current time plus 20s to 10 minutes after the current time minus 20s. In this way, the timestamp will be inserted to the database without being affected by the latency.
value	Yes	double	Specifies the monitoring metric data to be added. The value can be an integer or a floating point number.
unit	No	String	Specifies the data unit. Enter a maximum of 32 characters.
type	No	String	Specifies the enumerated type. Valid value: ● int ● float

Table 5-51 metric field data structure description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Specifies the customized namespace. For example, see **Namespace** for the customized ECS namespace. The value must be in the **service.item** format and can contain 3 to 32 characters. **service** and **item** each must be a string that starts with a letter and contains only uppercase letters, lowercase letters, digits, and underscores (_). In addition, **service** cannot start with **SYS** and **AGT**, and **namespace** cannot be **SERVICE.BMS** because this namespace has been used by the system. This parameter can be left blank when **alarm_type** is set to **(EVENT.SYS|EVENT.CUSTOM)**.
dimensions	Yes	Array of objects	Specifies the metric dimension. A maximum of three dimensions are supported. For details, see **Table 5-52**.
metric_name	Yes	String	Specifies the metric ID. For example, if the **monitoring metric** of an ECS is CPU usage, **metric_name** is **cpu_util**.

Table 5-52 dimensions field data structure description

Parameter	Mandatory	Type	Description
name	No	String	Specifies the dimension. For example, the ECS dimension is **instance_id**, which is listed in the **key** column in **Dimension**. The value can be a string of 1 to 32 characters and must start with a letter and contain only uppercase letters, lowercase letters, digits, underscores (_), and hyphens (-).

Parameter	Mandatory	Type	Description
value	No	String	Specifies the dimension value, for example, an ECS ID. The value can be a string of 1 to 256 characters and must start with a letter or a digit and contain only uppercase letters, lowercase letters, digits, underscores (_), and hyphens (-).

- Example request

```
[
    {
        "metric": {
            "namespace": "MINE.APP",
            "dimensions": [
                {
                    "name": "instance_id",
                    "value": "33328f02-3814-422e-b688-bfdba93d4050"
                }
            ],
            "metric_name": "cpu_util"
        },
        "ttl": 172800,
        "collect_time": 1463598260000,
        "type": "int",
        "value": 60,
        "unit": "%"
    },
    {
        "metric": {
            "namespace": "MINE.APP",
            "dimensions": [
                {
                    "name": "instance_id",
                    "value": "33328f02-3814-422e-b688-bfdba93d4050"
                }
            ],
            "metric_name": "cpu_util"
        },
        "ttl": 172800,
        "collect_time": 1463598270000,
        "type": "int",
        "value": 70,
        "unit": "%"
    }
]
```

Response

The response has no message body.

Returned Values

- Normal

  201
- Abnormal

Returned Value	Description
400 Bad Request	Request error.
401 Unauthorized	The authentication information is not provided or is incorrect.
403 Forbidden	You are forbidden to access the page requested.
408 Request Timeout	The request timed out.
429 Too Many Requests	Concurrent requests are excessive.
500 Internal Server Error	Failed to complete the request because of an internal service error.
503 Service Unavailable	The service is currently unavailable.

Error Code

For details, see **Error Codes**.

5.4.3 Querying the Host Configuration

Function

This API is used to query the host configuration for a specified event type in a specified period of time. You can specify the dimension of data to be queried.

NOTICE

This API is provided for SAP Monitor in the HANA scenario to query the host configuration. In other scenarios, the host configuration cannot be queried with this API.

URI

GET /V1.0/{project_id}/event-data

- Parameter description

Table 5-53 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

- Parameters that are used to query the host configuration

Parameter	Mandatory	Type	Description
namespace	Yes	String	Query the namespace of a service. For example, see **Namespace** for ECS namespace. The value must be in the **service.item** format and can contain 3 to 32 characters. **service** and **item** each must be a string that starts with a letter and contains only uppercase letters, lowercase letters, digits, and underscores (_).
type	Yes	String	Specifies the event type. It can contain only letters, underscores (_), and hyphens (-). It must start with a letter and cannot exceed 64 characters, for example, **instance_host_info**.
from	Yes	String	Specifies the start time of the query. The value is a UNIX timestamp and the unit is ms.
to	Yes	String	Specifies the end time of the query. The value is a UNIX timestamp and the unit is ms. The value of parameter **from** must be earlier than that of parameter **to**.
dim	Yes	String	Specifies the dimension. For example, the ECS **dimension** is **instance_id**. Specifies the dimension. A maximum of three dimensions are supported, and the dimensions are numbered from 0 in **dim.{i}=key,value** format. The **key** cannot exceed 32 characters and the **value** cannot exceed 256 characters. Example: **dim.0=instance_id,i-12345**

- Example: Query the configuration information about the ECS whose **ID** is **33328f02-3814-422e-b688-bfdba93d4051** and **type** is **instance_host_info**.

```
GET https://{Cloud Eye endpoint}/V1.0/{project_id}/event-data?namespace=SYS.ECS&dim.
0=instance_id,33328f02-3814-422e-b688-
bfdba93d4051&type=instance_host_info&from=1450234543422&to=1450320943422
```

Request

None

Response

- Response parameters

Table 5-54 Response parameters

Parameter	Type	Description
datapoints	Array of objects	Specifies the configuration list. If the corresponding configuration information does not exist, **datapoints** is an empty array and is **[]**. For details, see **Table 5-55**.

Table 5-55 datapoints field data structure description

Parameter	Type	Description
type	String	Specifies the event type, for example, **instance_host_info**.
timestamp	long	Specifies the time when the event is reported. It is a UNIX timestamp and the unit is ms.
value	String	Specifies the host configuration information.

- Example response

```
{
    "datapoints": [
        {
            "type": "instance_host_info",
            "timestamp": 1450231200000,
            "value": "xxx"
        },
        {
            "type": "instance_host_info",
            "timestamp": 1450231800000,
            "value": "xxx"
        }
    ]
}
```

Returned Values

- Normal

  200

- Abnormal

Returned Values	Description
400 Bad Request	Request error.
401 Unauthorized	The authentication information is not provided or is incorrect.
403 Forbidden	You are forbidden to access the page requested.
408 Request Timeout	The request timed out.
429 Too Many Requests	Concurrent requests are excessive.
500 Internal Server Error	Failed to complete the request because of an internal service error.
503 Service Unavailable	The service is currently unavailable.

Error Code

For details, see **Error Codes**.

5.5 Quota Management

5.5.1 Querying Quotas

Function

This API is used to query a resource quota and the used amount. The current resource refers to alarm rules only.

URI

GET /V1.0/{project_id}/quotas

- Parameter description

Table 5-56 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

- Example: Query the alarm rule quota.

```
GET https://{Cloud Eye endpoint}/V1.0/{project_id}/quotas
```

Request

None

Response

- Response parameters

Table 5-57 Response parameters

Parameter	Type	Description
quotas	Object	Specifies the quota list.

Table 5-58 Response parameters

Parameter	Type	Description
resources	Array of objects	Specifies the resource quota list.

Table 5-59 Response parameters

Parameter	Type	Description
type	String	Specifies the quota type. **alarm** indicates the alarm rule.
used	Integer	Specifies the used amount of the quota.
unit	String	Specifies the quota unit.
quota	Integer	Specifies the total amount of the quota.

- Example response

```
{
"quotas":
   {
   "resources": [
         {
            "unit":"",
            "type":"alarm",
            "quota":1000,
            "used":10
         }
      ]
   }
}
```

Returned Values

- Normal

  200

- Abnormal

Returned Value	Description
400 Bad Request	Request error.
401 Unauthorized	The authentication information is not provided or is incorrect.
403 Forbidden	You are forbidden to access the page requested.
408 Request Timeout	The request timed out.
429 Too Many Requests	Concurrent requests are excessive.
500 Internal Server Error	Failed to complete the request because of an internal service error.
503 Service Unavailable	The service is currently unavailable.

Error Code

For details, see **Error Codes**.

6 Common Parameters

6.1 Status Codes

- Normal

Returned Value	Description
200 OK	The results of GET and PUT operations are returned as expected.
201 Created	The results of the POST operation are returned as expected.
202 Accepted	The request has been accepted for processing.
204 No Content	The results of the DELETE operation are returned as expected.

- Abnormal

Returned Value	Description
400 Bad Request	The server failed to process the request.
401 Unauthorized	You must enter a username and password to access the requested page.
403 Forbidden	You are forbidden to access the requested page.
404 Not Found	The server cannot find the requested page.
405 Method Not Allowed	You are not allowed to use the method specified in the request.
406 Not Acceptable	The response generated by the server cannot be accepted by the client.

Returned Value	Description
407 Proxy Authentication Required	You must use the proxy server for authentication so that the request can be processed.
408 Request Timeout	The request timed out.
409 Conflict	The request could not be processed due to a conflict.
500 Internal Server Error	Failed to complete the request because of a service error.
501 Not Implemented	Failed to complete the request because the server does not support the requested function.
502 Bad Gateway	Failed to complete the request because the request is invalid.
503 Service Unavailable	Failed to complete the request. The service is unavailable.
504 Gateway Timeout	A gateway timeout error occurred.

6.2 Error Codes

Function

If an error occurs during API calling, the system returns error information. This section describes the error codes contained in the error information for Cloud Eye APIs.

Example Response

```
{
    "code": 400,
    "element": "Bad Request",
    "message": "The system received a request which cannot be recognized",
    "details": {
        "details": "Some content in message body is not correct",
        "code": "ces.0014"
    }
}
```

Glossary

Glossary	Description
Cloud Eye	Cloud Eye
Built-in metric	Each service has its own built-in metrics and dimensions. For example, an ECS (SYS.ECS) supports **cpu_util**.

Glossary	Description
Metric	A metric consists of the namespace, dimension (optional), and metric name. A metric name solely does not identify any object.

Error Code Description

Module	HTTP Status Code	Error Code	Error Code Description	Error Message	Measure
Cloud Eye	500	ces. 0007	Internal service error	Internal service error.	Contact technical support.
API	400	ces. 0001	The request content cannot be empty.	The content must be specified.	Specify the request content.
	400	ces. 0003	The project ID is left blank or is incorrect.	The tenant ID is left blank or incorrect.	Add or use the correct tenant ID.
	400	ces. 0004	The API version is not specified.	The API version must be specified.	Specify the API version in the request URL.
	400	ces. 0005	The API version is incorrect.	The API version is incorrect.	Use the correct API version.
	400	ces. 0006	The paging address is incorrect.	The paging address is incorrect.	Use the correct paging information.
	403	ces. 0009	System metrics cannot be added.	Adding SYS metric is not allowed	Use correct rights to add metrics.
	403	ces. 0010	System metrics cannot be deleted.	Deleting SYS metric is not allowed	Use correct rights to delete metrics.
	400	ces. 0011	The request is invalid.	The request is invalid.	Check the request.

Module	HTTP Status Code	Error Code	Error Code Description	Error Message	Measure
	400	ces. 0013	The URL parameter is invalid or does not exist.	The URL parameter is invalid or does not exist.	Check the URL parameter.
	400	ces. 0014	Some content in the message body is correct.	Some content in message body is not correct.	Check the request body parameters.
	401	ces. 0015	Authentication fails or invalid authentication information is not provided.	Authentication fails or the authentication information is not provided.	Check whether the user name or password (or AK or SK) for obtaining the token is correct.
	404	ces. 0016	The requested resource does not exist.	The requested resource does not exist.	Check whether the requested resource exists.
	403	ces. 0017	The authentication information is incorrect or the service invoker does not have sufficient rights.	The authentication information is incorrect or the service invoker does not have sufficient rights.	Check whether the user name or password (or AK or SK) or the user rights for obtaining the token are correct.
Cassandra	500	ces. 0008	Database error	Database error.	Contact technical support.
Kafka	500	ces. 0012	The message queue is abnormal or is not ready.	The message queue is abnormal or is not ready.	Contact technical support.
Zookeeper	500	ces. 0021	Internal locking error	Internal locking error	Contact technical support.

Module	HTTP Status Code	Error Code	Error Code Description	Error Message	Measure
Blueflood	500	ces. 0019	The metric processing engine is abnormal.	The metric processing engine is abnormal.	Contact technical support.
Alarm	400	ces. 0002	The alarm ID cannot be left blank.	The alarm ID must be specified.	Specify the alarm ID.
	403	ces. 0018	The number of alarm rules created exceeds the quota.	The number of alarms exceeds the quota	Apply for a higher alarm quota.
	400	ces. 0028	The metric and notification type do not match when an alarm rule is created.	The metric does not support the alarm action type.	Modify the metric or notification type according to the parameter description to make them match.

6.3 Obtaining a Project ID

A project ID is required for some URLs when an API is called. Therefore, you need to obtain a project ID in advance. The steps are as follows:

1. Obtain the token.

   For details, see **Token-based Authentication**.

2. Obtain a project ID.

   The API for obtaining the project ID is **GET https://iam.eu-west-0.myhuaweicloud.com/v3/projects**.

   Add **X-Auth-Token** to the request header and set its value to the token obtained in the preceding step.

   The following is an example response. The value of **id** is the project ID to be obtained.

```
{
    "links": {},
    "projects": [
        {
            "is_domain": ,
            "description": "",
            "links": {},
            "enabled": true,
            "id": "",   // Project ID
            "parent_id": "",
            "domain_id": "",
            "name": ""
        },
        ...
    ]
}
```

A Appendix

A.1 ECS Monitoring Metrics

Function

This section describes metrics reported by ECS to Cloud Eye as well as their namespaces and dimensions. You can use APIs provided by Cloud Eye to query the metrics of the monitored object and alarms generated for ECS.

Namespace

SYS.ECS

Metrics

Metric	Name	Description	Value Range	Remarks
cpu_util	CPU Usage	This metric is used to show CPU usages (%) of monitored objects.	0% to 100%	ECS monitored **NOTE** The metrics collected using VMTools are accurate.
mem_util	Memory Usage	This metric is used to show memory usages (%) of monitored objects.	0% to 100%	ECS monitored **NOTE** This metric is unavailable if the image has no VMTools installed.
disk_util_inband	Disks Usage	This metric is used to show disk usages (%) of monitored objects.	0% to 100%	ECS monitored **NOTE** This metric is unavailable if the image has no VMTools installed.

Metric	Name	Description	Value Range	Remarks
disk_read_bytes_rate	Disk Read Bandwidth	This metric is used to show the number of bytes read from the monitored object per second (byte/s).	≥ 0	ECS monitored
disk_write_bytes_rate	Disk Write Bandwidth	This metric is used to show the number of bytes written to the monitored object per second (byte/s).	≥ 0	ECS monitored
disk_read_requests_rate	Disk Read IOPS	This metric is used to show the number of read requests sent to the monitored object per second (requests/second).	≥ 0	ECS monitored
disk_write_requests_rate	Disk Write IOPS	This metric is used to show the number of write requests sent to the monitored object per second (requests/second).	≥ 0	ECS monitored
network_incoming_bytes_rate_inband	Inband Incoming Rate	This metric is used to show the number of incoming bytes received by the monitored object per second (byte/s).	≥ 0	ECS monitored
network_outgoing_bytes_rate_inband	Inband Outgoing Rate	This metric is used to show the number of outgoing bytes sent by the monitored object per second (byte/s).	≥ 0	ECS monitored

Metric	Name	Description	Value Range	Remarks
network_incoming_bytes_aggregate_rate	Outband Incoming Rate	This metric is used to show the number of incoming bytes received by the monitored object per second (byte/s) at the virtualization layer.	≥ 0	ECS monitored **NOTE** This metric is unavailable if SR-IOV is enabled.
network_outgoing_bytes_aggregate_rate	Outband Outgoing Rate	This metric is used to show the number of outgoing bytes sent by the monitored object per second (byte/s) at the virtualization layer.	≥ 0	ECS monitored **NOTE** This metric is unavailable if SR-IOV is enabled.

Metric	Name	Description	Value Range	Remarks
inst_sys_status_error	System Status Check Failed	This metric is used to monitor the cloud platform on which ECSs run. The system periodically checks the system status and returns check results using value **0** or **1**. ● **0**: The system is running properly. All check items are normal. ● **1**: The system is not running properly. One or more check items are abnormal. When the power source of the physical host fails or the hardware/software becomes faulty, the check result is **1**.	0 or 1	ECS monitored

NOTE

The image based on which the target ECS is created must have VMTools installed. Otherwise, the **Memory Usage** and **Disk Usage** metrics are unavailable. For details about how to install the VMTools, visit **https://github.com/UVP-Tools/UVP-Tools/**.

Dimension

Key	Value
instance_id	Specifies the ECS ID.

A.2 AS Metrics

Function

This section describes metrics reported by AS to Cloud Eye as well as their namespaces and dimensions. You can use APIs provided by Cloud Eye to view the AS metrics and the alarms generated by Cloud Eye for AS.

Namespace

SYS.AS

Metrics

Metric	Name	Description	Value Range	Remarks
cpu_util	CPU Usage	Average CPU usage of all instances in a monitored object	≥ 0%	The monitored object is an AS group.
mem_util	Memory Usage	Average memory usage of all instances in a monitored object	≥ 0%	The monitored object is an AS group. **NOTE** This metric is unavailable if the image has no VMTools installed.
network_incoming_bytes_rate_inband	Inband Incoming Rate	Average number of incoming bytes per second on all instances in a monitored object	≥ 0	The monitored object is an AS group.
network_outgoing_bytes_rate_inband	Inband Outgoing Rate	Average number of outgoing bytes per second on all instances in a monitored object	≥ 0	The monitored object is an AS group.

Metric	Name	Description	Value Range	Remarks
instance_num	Number of Instances	Number of available instances in a monitored object	≥ 0	The monitored object is an AS group. Available ECSs are **INSERVICE** instances in an AS group.
disk_read_bytes_rate	Disks Read Rate	Number of bytes read from all instances in a monitored object per second	≥ 0	The monitored object is an AS group.
disk_write_bytes_rate	Disks Write Rate	Number of bytes written to all instances in a monitored object per second	≥ 0	The monitored object is an AS group.
disk_read_requests_rate	Disk Read Requests	Number of read requests sent to all instances in a monitored object per second	≥ 0	The monitored object is an AS group.
disk_write_requests_rate	Disks Write Requests	Number of write requests sent to all instances in a monitored object per second	≥ 0	The monitored object is an AS group.

NOTE

For details about whether your OS supports the **Memory Usage**, **Inband Outgoing Rate**, and **Inband Incoming Rate** metrics, see *Elastic Cloud Server User Guide*.

Dimension

Key	Value
AutoScalingGroup	AS group ID

A.3 EVS Metrics

Function

This section describes metrics reported by EVS to Cloud Eye as well as their namespaces and dimensions. You can use APIs provided by Cloud Eye to query the metrics of the monitored object and alarms generated for EVS.

Namespace

SYS.EVS

Metrics

Metric	Name	Description	Value Range	Monitored Object
disk_device_read_bytes_rate	Disk Read Rate	Number of bytes read from the monitored disk per second Unit: byte/s	≥ 0 bytes/s	EVS disk
disk_device_write_bytes_rate	Disk Write Rate	Number of bytes written to the monitored disk per second Unit: byte/s	≥ 0 bytes/s	EVS disk
disk_device_read_requests_rate	Disk Read Request Rate	Number of read requests sent to the monitored disk per second Unit: Request/s	≥ 0 Requests/s	EVS disk
disk_device_write_requests_rate	Disk Write Request Rate	Number of write requests sent to the monitored disk per second Unit: Request/s	≥ 0 Requests/s	EVS disk

Dimension

Key	Value
disk_name	ECS ID-disk name, for example, 6f3c6f91-4b24-4e1b-b7d1-a94ac1cb011d-sda (sda is the disk name)

A.4 SFS Metrics

Function

This topic describes metrics reported by Scalable File Service (SFS) to Cloud Eye as well as their namespaces and dimensions. You can use APIs provided by Cloud Eye to query the metrics of the monitored object and alarms generated for SFS.

Namespace

SYS.SFS

Metrics

Metric	Name	Description	Value Range	Monitored Object
read_bandwidth	Read Bandwidth	Read bandwidth of the file system within a monitoring period Unit: byte/s	≥ 0 bytes/s	File sharing
write_bandwidth	Write Bandwidth	Write bandwidth of the file system within a monitoring period Unit: byte/s	≥ 0 bytes/s	File sharing
rw_bandwidth	Read Write Bandwidth	Read and write bandwidth of the file system within a monitoring period Unit: byte/s	≥ 0 bytes/s	File sharing

Dimension

Key	Value
share_id	File sharing

A.5 EIP and Bandwidth Metrics

Function

This section describes the namespace, list, and dimensions of EIP and Bandwidth metrics on Cloud Eye. You can use APIs provided by Cloud Eye to query the metrics of the monitored object and alarms generated for EIP and Bandwidth.

Namespace

SYS.VPC

Metrics

Metric	Name	Description	Value Range	Monitored Object
upstream_bandwidth	Upstream Bandwidth	Outbound network rate of the monitored object	≥ 0 bits/s	Bandwidth or EIP
downstream_bandwidth	Downstream Bandwidth	Inbound network rate of the monitored object	≥ 0 bits/s	Bandwidth or EIP

Dimension

Key	Value
publicip_id	EIP ID
bandwidth_id	Bandwidth ID

A.6 Monitoring Metrics

Overview

This section describes the metrics that can be monitored by Cloud Eye and their namespaces and dimensions. You can use APIs provided by Cloud Eye to query the metrics of a monitored object and generated alarms.

Namespace

SYS.ELB

Metrics

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m1_cps	Concurrent Connections	Load balancing at Layer 4: total number of TCP and UDP connections from the monitored object to backend servers Load balancing at Layer 7: total number of TCP connections from the clients to the monitored object Unit: Count	≥ 0	Shared load balancer, shared load balancer listener, or classic load balancer	1 minute
m2_act_conn	Active Connections	Number of TCP and UDP connections in the **ESTABLISHED** state between the monitored object and backend servers You can run the following command to view the connections (both Windows and Linux servers): `netstat -an` Unit: Count	≥ 0		
m3_inact_conn	Inactive Connections	Number of TCP connections between the monitored object and backend servers except those in the **ESTABLISHED** state You can run the following command to view the connections (both Windows and Linux servers): `netstat -an` Unit: Count	≥ 0		

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m4_ncps	New Connections	Number of TCP and UDP connections established between clients and the monitored object per second Unit: Count	≥ 0/second		
m5_in_pps	Incoming Packets	Number of packets received by the monitored object per second Unit: Packet/s	≥ 0/second		
m6_out_pps	Outgoing Packets	Number of packets sent from the monitored object per second Unit: Packet/s	≥ 0/second		
m7_in_Bps	Inbound Rate	Traffic used for accessing the monitored object from the Internet Unit: byte/s	≥ 0 bytes/s		
m8_out_Bps	Outbound Rate	Traffic used by the monitored object to access the Internet Unit: byte/s	≥ 0 bytes/s		
m9_abnormal_servers	Unhealthy Servers	Number of unhealthy backend servers associated with the monitored object Unit: Count	≥ 0	Shared load balancer or classic load balancer	1 minute
ma_normal_servers	Healthy Servers	Number of healthy backend servers associated with the monitored object Unit: Count	≥ 0		

a: If a service has multiple dimensions, you must specify all dimensions when you use APIs to query the metrics.

- Example of querying a single metric from both dimensions: dim. 0=lbaas_instance_id,223e9eed-2b02-4ed2-a126-7e806a6fee1f&dim. 1=lbaas_listener_id,3baa7335-8886-4867-8481-7cbba967a917
- Example of querying metrics in batches from both dimensions:

```
"dimensions": [
{
"name": "lbaas_instance_id",
"value": "223e9eed-2b02-4ed2-a126-7e806a6fee1f"
}
{
"name": "lbaas_listener_id",
"value": "3baa7335-8886-4867-8481-7cbba967a917"
}
],
```

Dimensions

Key	Value
lb_instance_id	Specifies the ID of the classic load balancer.
lbaas_instance_id	Specifies the ID of the shared load balancer.
lbaas_listener_id	Specifies the ID of the shared load balancer listener.

A.7 RDS Metrics

Function

This section describes metrics reported by Cloud Eye to the Relational Database Service (RDS) as well as their namespaces and dimensions. You can use APIs provided by Cloud Eye to query the metrics of the monitored object and alarms generated for RDS.

Namespace

SYS.RDS

Metrics

Table A-1 MySQL metrics

Metric	Name	Description	Value Range	Remarks
rds001_cpu_usage	CPU Usage	CPU usage of the monitored object Unit: Ratio	0-1	Monitored object: ECS Monitored instance type: MySQL instance

Metric	Name	Description	Value Range	Remarks
rds002_mem_usage	Memory Usage	Memory usage of the monitored object Unit: Ratio	0-1	Monitored object: ECS Monitored instance type: MySQL instance
rds003_iops	IOPS	Average rate at which I/O requests are processed during a specified period Unit: Count/s	≥ 0 counts/s	Monitored object: ECS Monitored instance type: MySQL instance
rds004_bytes_in	Network Input Throughput	Rate at which all incoming traffic passes through the network adapter Unit: byte/s	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: MySQL instance
rds005_bytes_out	Network Output Throughput	Rate at which all outgoing traffic passes through the network adapter Unit: byte/s	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: MySQL instance
rds006_conn_count	Total Connections	Total number of connection attempts to the MySQL server Unit: Count	≥ 0 counts	Monitored object: database Monitored instance type: MySQL instance
rds007_conn_active_count	Current Active Connections	Number of current active connections Unit: Count	≥ 0 counts	Monitored object: database Monitored instance type: MySQL instance
rds008_qps	QPS	Rate at which SQL statement queries (including the stored procedure) are executed Unit: Query/s	≥ 0 queries/s	Monitored object: database Monitored instance type: MySQL instance
rds009_tps	TPS	Rate at which transactions (including those submitted and rolled back) are executed Unit: Transaction/s	≥ 0 transactions/s	Monitored object: database Monitored instance type: MySQL instance

Metric	Name	Description	Value Range	Remarks
rds010_innodb_buf_usage	Buffer Pool Usage	Ratio of dirty data from all data in the InnoDB buffer Unit: Ratio	0-1	Monitored object: database Monitored instance type: MySQL instance
rds011_innodb_buf_hit	Buffer Pool Hit Rate	Ratio of read hits to read requests in the InnoDB buffer Unit: Ratio	0-1	Monitored object: database Monitored instance type: MySQL instance
rds012_innodb_buf_dirty	Buffer Pool Dirty Block Rate	Ratio of used pages to total data in the InnoDB buffer Unit: Ratio	0-1	Monitored object: database Monitored instance type: MySQL instance
rds013_innodb_reads	InnoDB Read Throughput	Average rate at which data is read by the InnoDB buffer Unit: byte/s	≥ 0 bytes/s	Monitored object: database Monitored instance type: MySQL instance
rds014_innodb_writes	InnoDB Write Throughput	Average rate at which data is written by the InnoDB buffer Unit: byte/s	≥ 0 bytes/s	Monitored object: database Monitored instance type: MySQL instance
rds015_innodb_read_count	InnoDB File Read Frequency	Average rate at which the InnoDB reads files Unit: Count/s	≥ 0 counts/s	Monitored object: database Monitored instance type: MySQL instance
rds016_innodb_write_count	InnoDB File Write Frequency	Average rate at which the InnoDB writes data Unit: Count/s	≥ 0 counts/s	Monitored object: database Monitored instance type: MySQL instance
rds017_innodb_log_write_req_count	InnoDB Log Write Request Frequency	Average rate at which log write requests are received Unit: Count/s	≥ 0 counts/s	Monitored object: database Monitored instance type: MySQL instance
rds018_innodb_log_write_count	InnoDB Log Physical Write Frequency	Average rate at which log write requests are received Unit: Count/s	≥ 0 counts/s	Monitored object: database Monitored instance type: MySQL instance

Metric	Name	Description	Value Range	Remarks
rds019_innodb_log_fsync_count	InnoDB Log fsync() Write Frequency	Average rate at which fsync() write requests on log files are received Unit: Count/s	≥ 0 counts/s	Monitored object: database Monitored instance type: MySQL instance
rds020_temp_tbl_count	Temporary Tables Qty	Number of temporary tables automatically created on hard disks when MySQL statements are executed Unit: Table	≥ 0 tables	Monitored object: database Monitored instance type: MySQL instance
rds021_myisam_buf_usage	Key Buffer Usage	MyISAM key buffer usage ratio Unit: Ratio	0-1	Monitored object: database Monitored instance type: MySQL instance
rds022_myisam_buf_write_hit	Key Buffer Write Hit Rate	MyISAM Key buffer write hit ratio of the monitored object Unit: Ratio	0-1	Monitored object: database Monitored instance type: MySQL instance
rds023_myisam_buf_read_hit	Key Buffer Read Hit Rate	MyISAM Key buffer read hit ratio of the monitored object Unit: Ratio	0-1	Monitored object: database Monitored instance type: MySQL instance
rds024_myisam_disk_write_count	MyISAM Disk Write Frequency	Rate at which indexes are written into disks Unit: Count/s	≥ 0 counts/s	Monitored object: database Monitored instance type: MySQL instance
rds025_myisam_disk_read_count	MyISAM Disk Read Frequency	Rate at which indexes are read from disks Unit: Count/s	≥ 0 counts/s	Monitored object: database Monitored instance type: MySQL instance
rds026_myisam_buf_write_count	MyISAM Buffer Pool Write Frequency	Rate at which requests on writing indexes into the MyISAM buffer pool are received Unit: Count/s	≥ 0 counts/s	Monitored object: database Monitored instance type: MySQL instance

Metric	Name	Description	Value Range	Remarks
rds027_myisam_buf_read_count	MyISAM Buffer Pool Read Frequency	Rate at which requests on reading indexes from the MyISAM buffer pool are received Unit: Count/s	≥ 0 counts/s	Monitored object: database Monitored instance type: MySQL instance
rds028_comdml_del_count	Delete Operations per Second	Average rate at which Delete statements are executed Unit: Count/s	≥ 0 executions/s	Monitored object: database Monitored instance type: MySQL instance
rds029_comdml_ins_count	Insert Operations per Second	Average rate at which Insert statements are executed Unit: Execution/s	≥ 0 executions/s	Monitored object: database Monitored instance type: MySQL instance
rds030_comdml_ins_sel_count	Insert_Select Operations per Second	Average rate at which Insert-Select statements are executed Unit: Execution/s	≥ 0 executions/s	Monitored object: database Monitored instance type: MySQL instance
rds031_comdml_rep_count	Replace Operations per Second	Average rate at which Replace statements are executed Unit: Execution/s	≥ 0 executions/s	Monitored object: database Monitored instance type: MySQL instance
rds032_comdml_rep_sel_count	Replace_Selection Operations per Second	Average rate at which Replace_Selection statements are executed Unit: Execution/s	≥ 0 executions/s	Monitored object: database Monitored instance type: MySQL instance
rds033_comdml_sel_count	Select Operations per Second	Average rate at which Select statements are executed Unit: Execution/s	≥ 0 executions/s	Monitored object: database Monitored instance type: MySQL instance
rds034_comdml_upd_count	Update Operations per Second	Average rate at which Update statements are executed Unit: Execution/s	≥ 0 executions/s	Monitored object: database Monitored instance type: MySQL instance
rds035_innodb_del_row_count	Row Delete Speed	Average rate at which rows are deleted from the InnoDB table Unit: Row/s	≥ 0 rows/s	Monitored object: database Monitored instance type: MySQL instance

Metric	Name	Description	Value Range	Remarks
rds036_innodb_ins_row_count	Row Insert Speed	Average rate at which rows are inserted into the InnoDB table Unit: Row/s	≥ 0 rows/s	Monitored object: database Monitored instance type: MySQL instance
rds037_innodb_read_row_count	Row Read Speed	Average rate at which rows are read from the InnoDB table Unit: Row/s	≥ 0 rows/s	Monitored object: database Monitored instance type: MySQL instance
rds038_innodb_upd_row_count	Row Update Speed	Average rate at which rows are updated in the InnoDB table Unit: Row/s	≥ 0 rows/s	Monitored object: database Monitored instance type: MySQL instance
rds039_disk_usage	Disk Utilization	Disk usage of the monitored object Unit: Ratio	0-1	Monitored object: ECS Monitored instance type: MySQL instance
rds047_disk_total_size	Total Disk Size	Total disk size of the monitored object Unit: GB	40-2000 GB	Monitored object: ECS Monitored instance type: MySQL instance
rds048_disk_used_size	Storage Space Used	Amount of used storage space size of the monitored object Unit: GB	0-2000 GB	Monitored object: ECS Monitored instance type: MySQL instance
rds049_disk_read_throughput	Disk Read Throughput	Rate at which data is read from a disk Unit: byte/s	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: MySQL instance
rds050_disk_write_throughput	Disk Write Throughput	Rate at which data is written to a disk Unit: byte/s	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: MySQL instance
rds051_avg_disk_sec_per_read	Average Time per Disk Read	Time required for reading 1 KB disk data Unit: Second	> 0s	Monitored object: ECS Monitored instance type: MySQL instance
rds052_avg_disk_sec_per_write	Average Time per Disk Write	Time required for writing 1 KB data to a disk Unit: Second	> 0s	Monitored object: ECS Monitored instance type: MySQL instance

Metric	Name	Description	Value Range	Remarks
rds053_avg_disk_queue_length	Average Disk Queue Length	Number of processes waiting to be written to the monitored object	≥ 0	Monitored object: ECS Monitored instance type: MySQL instance

Table A-2 PostgreSQL metrics

Metric	Name	Description	Value Range	Remarks
rds001_cpu_usage	CPU Usage	CPU usage of the monitored object Unit: Ratio	0-1	Monitored object: ECS Monitored instance type: PostgreSQL instance
rds002_mem_usage	Memory Usage	Memory usage of the monitored object Unit: Ratio	0-1	Monitored object: ECS Monitored instance type: PostgreSQL instance
rds003_iops	IOPS	Average rate at which I/O requests are processed during a specified period Unit: Count/s	≥ 0 counts/s	Monitored object: ECS Monitored instance type: PostgreSQL instance
rds004_bytes_in	Network Input Throughput	Rate at which all incoming traffic passes through the network adapter Unit: byte/s	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: PostgreSQL instance
rds005_bytes_out	Network Output Throughput	Rate at which all outgoing traffic passes through the network adapter Unit: byte/s	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: PostgreSQL instance
rds039_disk_usage	Disk Utilization	Disk usage of the monitored object Unit: Ratio	0-1	Monitored object: ECS Monitored instance type: PostgreSQL instance

Metric	Name	Description	Value Range	Remarks
rds040_transaction_logs_usage	Transaction Logs Usage	Disk capacity used by transaction logs Unit: MB	≥ 0 MB	Monitored object: database Monitored instance type: PostgreSQL instance
rds041_replication_slot_usage	Replication Slot Usage	Disk capacity used by replication slot files Unit: MB	≥ 0 MB	Monitored object: database Monitored instance type: PostgreSQL instance
rds042_database_connections	Database Connections	Number of database connections in use Unit: Count	≥ 0 counts	Monitored object: database Monitored instance type: PostgreSQL instance
rds043_maximum_used_transaction_ids	Maximum Used Transaction IDs	Maximum transaction IDs that have been used Unit: Count	≥ 0 counts	Monitored object: database Monitored instance type: PostgreSQL instance
rds044_transaction_logs_generations	Transaction Logs Generations	Size of transaction logs generated per second Unit: MB/s	≥ 0 MB/s	Monitored object: database Monitored instance type: PostgreSQL instance
rds045_oldest_replication_slot_lag	Oldest Replication Slot Lag	Lagging size of the most lagging replica in terms of WAL data received Unit: MB	≥ 0 MB	Monitored object: database Monitored instance type: PostgreSQL instance
rds046_replication_lag	Replication Lag	Replication lag delay Unit: ms	≥ 0 ms	Monitored object: database Monitored instance type: PostgreSQL instance
rds047_disk_total_size	Total Disk Size	Total disk size of the monitored object Unit: GB	40-2000 GB	Monitored object: ECS Monitored instance type: PostgreSQL instance

Metric	Name	Description	Value Range	Remarks
rds048_disk_used_size	Storage Space Used	Amount of used storage space size of the monitored object Unit: GB	0-2000 GB	Monitored object: ECS Monitored instance type: PostgreSQL instance
rds049_disk_read_throughput	Disk Read Throughput	Rate at which data is read from a disk Unit: byte/s	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: PostgreSQL instance
rds050_disk_write_throughput	Disk Write Throughput	Rate at which data is written to a disk Unit: byte/s	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: PostgreSQL instance
rds051_avg_disk_sec_per_read	Average Time per Disk Read	Time required for reading 1 KB disk data Unit: Second	> 0s	Monitored object: ECS Monitored instance type: PostgreSQL instance
rds052_avg_disk_sec_per_write	Average Time per Disk Write	Time required for writing 1 KB data to a disk Unit: Second	> 0s	Monitored object: ECS Monitored instance type: PostgreSQL instance
rds053_avg_disk_queue_length	Average Disk Queue Length	Number of processes waiting to be written to the monitored object	≥ 0	Monitored object: ECS Monitored instance type: PostgreSQL instance

Table A-3 Microsoft SQL Server metrics

Metric	Name	Description	Value Range	Remarks
rds001_cpu_usage	CPU Usage	CPU usage of the monitored object Unit: Ratio	0-1	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance
rds002_mem_usage	Memory Usage	Memory usage of the monitored object Unit: Ratio	0-1	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance

Metric	Name	Description	Value Range	Remarks
rds003_iops	IOPS	Average rate at which I/O requests are processed during a specified period Unit: Count/s	≥ 0 counts/s	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance
rds004_bytes_in	Network Input Throughput	Rate at which all incoming traffic passes through the network adapter Unit: byte/s	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance
rds005_bytes_out	Network Output Throughput	Rate at which all outgoing traffic passes through the network adapter Unit: byte/s	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance
rds039_disk_usage	Disk Utilization	Disk usage of the monitored object Unit: Ratio	0-1	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance
rds047_disk_total_size	Total Disk Size	Total disk size of the monitored object Unit: GB	40-2000 GB	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance
rds048_disk_used_size	Storage Space Used	Amount of used storage space size of the monitored object Unit: GB	0-2000 GB	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance
rds049_disk_read_throughput	Disk Read Throughput	Rate at which data is read from a disk Unit: byte/s	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance
rds050_disk_write_throughput	Disk Write Throughput	Rate at which data is written to a disk Unit: byte/s	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance
rds051_avg_disk_sec_per_read	Average Time per Disk Read	Time required for reading 1 KB disk data Unit: Second	> 0s	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance

Metric	Name	Description	Value Range	Remarks
rds052_avg_disk_sec_per_write	Average Time per Disk Write	Time required for writing 1 KB data to a disk Unit: Second	> 0s	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance
rds053_avg_disk_queue_length	Average Disk Queue Length	Number of processes waiting to be written to the monitored object	≥ 0	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance
rds054_db_connections_in_use	Number of Database Connections In Use	Number of database connections in use Unit: Count	≥ 0 counts	Monitored object: database Monitored instance type: Microsoft SQL Server instance

Dimension

Key	Value
rds_instance_id	MySQL instance ID
postgresql_instance_id	PostgreSQL instance ID
rds_instance_sqlserver_id	Microsoft SQL Server instance ID

A.8 Workspace Metrics

Function

This section describes metrics reported by Workspace to Cloud Eye as well as their namespaces, list, and dimensions. You can use APIs provided by Cloud Eye to query the metric information generated for Workspace.

Namespace

SYS.Workspace

Metrics

Metric	Name	Description	Value Range	Monitored Object
cpu_util	CPU Usage	CPU usage of the monitored object Unit: Percent	0-100%	Infrastructure server
mem_util	Memory Usage	Memory usage of the monitored object Unit: Percent	0-100%	Infrastructure server
disk_util_inband	Disk Utilization	Disk usage of the monitored object Unit: Percent	0-100%	Infrastructure server
iops	IOPS	Number of I/O operations per second on the monitored object Unit: Count	≥ 0 counts	Infrastructure server
bytes_in	Network Input Throughput	Number of incoming bytes per second on the monitored object Unit: byte/s	≥ 0 bytes/s	Infrastructure server
bytes_out	Network Output Throughput	Number of outgoing bytes per second on the monitored object Unit: byte/s	≥ 0 bytes/s	Infrastructure server

Dimension

Key	Value
instance_id	Instance ID of the infrastructure server

B Change History

Released On	Description
2020-03-24	This release incorporates the following changes: ● Added Reporting Events.
2019-08-31	This release incorporates the following changes: ● Optimized the document structure. ● Added **Obtaining a Project ID**.
2018-09-30	This release incorporates the following changes: ● Added the API for querying all API versions. ● Added the API for querying a specified API version.
2018-05-30	This release incorporates the following change: Added metrics for the enhanced load balancer and enhanced load balancer listener.
2018-04-30	This release incorporates the following changes: Added Relational Database Service (RDS) metrics.
2018-02-28	This release incorporates the following change: Updated Virtual Private Cloud (VPC) metrics.
2018-01-30	This release incorporates the following change: Added the API for querying the host configuration.
2017-12-30	This release incorporates the following changes: ● Added the API for creating alarm rules. ● Optimized descriptions of error codes.
2017-11-30	This release incorporates the following changes: Added Auto Scaling (AS) metrics.
2017-10-30	This release incorporates the following changes: Added Relational Database Service (RDS) metrics.

Released On	Description
2017-07-30	This release incorporates the following change: Marked the section about querying followed metrics as deprecated.
2017-04-28	This release incorporates the following change: Added Virtual Private Cloud (VPC) traffic metrics.
2017-02-27	This release incorporates the following change: Added the setting description for the **from** parameter.
2016-12-30	This issue is the first official release.