Cloud Eye

API Reference

Issue 01

Date 2024-04-15

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

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process*. For details about this process, visit the following web page:
https://www.huawei.com/en/psirt/vul-response-process
For vulnerability information, enterprise customers can visit the following web page:
https://securitybulletin.huawei.com/enterprise/en/security-advisory

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?" in the *Cloud Eye User Guide*.

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

- Domain

  A domain 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 domain 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 domain 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.

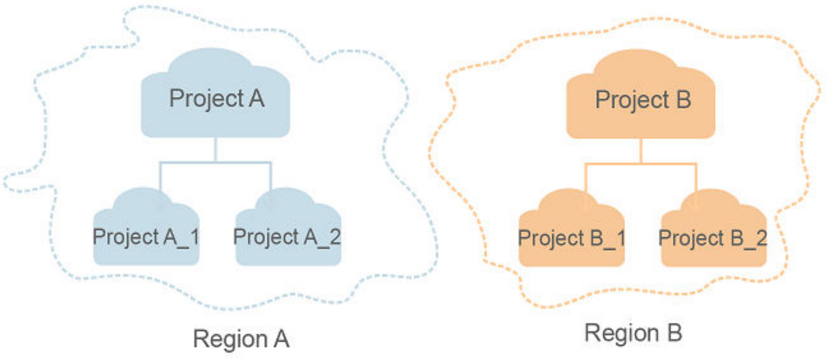
- AZ

  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 domains 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 versions	**Querying All API Versions**	Query all API versions supported by Cloud Eye.
		Querying a Specified API Version	Query a specified API version of Cloud Eye.
	Metrics	**Querying Metrics**	Query metrics supported by Cloud Eye.
	Alarm rules	**Querying Alarm Rules**	Query alarm rules.
		Querying Details of an Alarm Rule	Query details of an alarm rule based on its 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 its ID.
		Creating an Alarm Rule	Create an alarm rule.
	Monitoring data	**Querying Monitoring Data of a Metric**	Query the monitoring data of a specified metric at a specified granularity in a specified time range.

Type	Subtype	API	Description
		Adding Monitoring Data	Add one or more pieces of metric monitoring data.
		Querying Monitoring Data of Multiple Metrics	Query the monitoring data of specified metrics within a specified time range and at a specified granularity. You can query the monitoring data of up to 10 metrics in one batch.
	Quotas	**Querying Quotas**	Query the alarm rule quota.
	Event monitoring	**Reporting Events**	Report custom events.

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 the administrator.
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	e9993fc787d94b6c886cbaa340f9c0f4
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 (for example, JSON or XML) 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*, *$ADMIN_PASS* (login password), and *xxxxxxxxxxxxxxxxxx* (project name) with the actual values. Obtain a project name from the administrator.

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": "$ADMIN_PASS",      //You are advised to store it in ciphertext in the 
configuration file or an environment variable and decrypt it when needed to ensure security.
                    "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 authentication: Requests are authenticated using tokens.
- AK/SK authentication: Requests are encrypted using AK/SK pairs. AK/SK authentication is recommended because it is more secure than token authentication.

Token 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.

Cloud Eye is a project-level service. When you call the API, set **auth.scope** in the request body to **project**.

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username",   //IAM user name
          "password": "********",  //IAM user password
          "domain": {
            "name": "domainname"  //Name of the account to which the IAM user belongs
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "xxxxxxxx"    //Project Name
      }
    }
  }
}
```

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:

```
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

AK/SK Authentication

An AK/SK is used to verify the identity of a request sender. In AK/SK authentication, a signature needs to be obtained and then added to requests.

NOTE

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, which is used in conjunction with an AK to sign requests cryptographically. It identifies a request sender and prevents the request from being modified.

The following uses a demo project to show how to sign a request and use an HTTP client to send an HTTPS request.

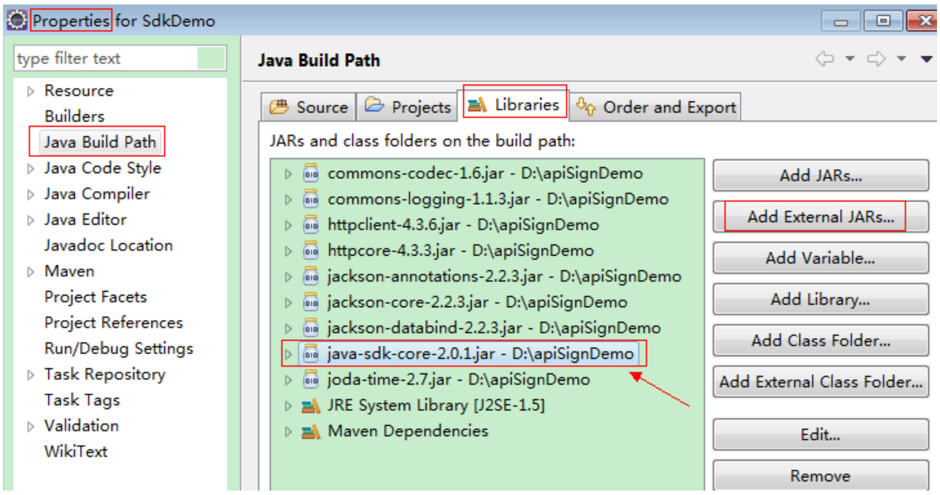
Download the demo project at **https://github.com/api-gate-way/SdkDemo**.

If you do not need the demo project, visit the following URL to download the API Gateway signing SDK:

Obtain the API Gateway signing SDK from the enterprise administrator.

Decompress the downloaded package and reference the obtained JAR files as dependencies.

Figure 3-1 Introducing the API Gateway signing SDK



JARs and class folders on the build path:

- commons-codec-1.6.jar - D:\apiSignDemo
- commons-logging-1.1.3.jar - D:\apiSignDemo
- httpclient-4.3.6.jar - D:\apiSignDemo
- httpcore-4.3.3.jar - D:\apiSignDemo
- jackson-annotations-2.2.3.jar - D:\apiSignDemo
- jackson-core-2.2.3.jar - D:\apiSignDemo
- jackson-databind-2.2.3.jar - D:\apiSignDemo
- java-sdk-core-2.0.1.jar - D:\apiSignDemo
- joda-time-2.7.jar - D:\apiSignDemo
- JRE System Library [J2SE-1.5]
- Maven Dependencies

Step 1 Generate an AK/SK. (If an AK/SK file has already been obtained, skip this step and locate the downloaded AK/SK file. Generally, the file name will be **credentials.csv**.)

1. Log in to the management console.
2. Click the username and select **My Credentials** from the drop-down list.
3. In the navigation tree on the left, click **Access Keys**.
4. Click **Add Access Key**.

5. Enter an access key description and click **OK**.
6. Enter the verification code received by email, SMS message, or MFA application. Click **OK**.

   NOTE

   If you have enabled operation protection (**Security Settings** > **Critical Operations** > **Operation Protection**), enter the verification code.

   For users created in IAM that have not bound with any email address or mobile number, only the login password needs to be entered.
7. Click **OK** to download the access key.

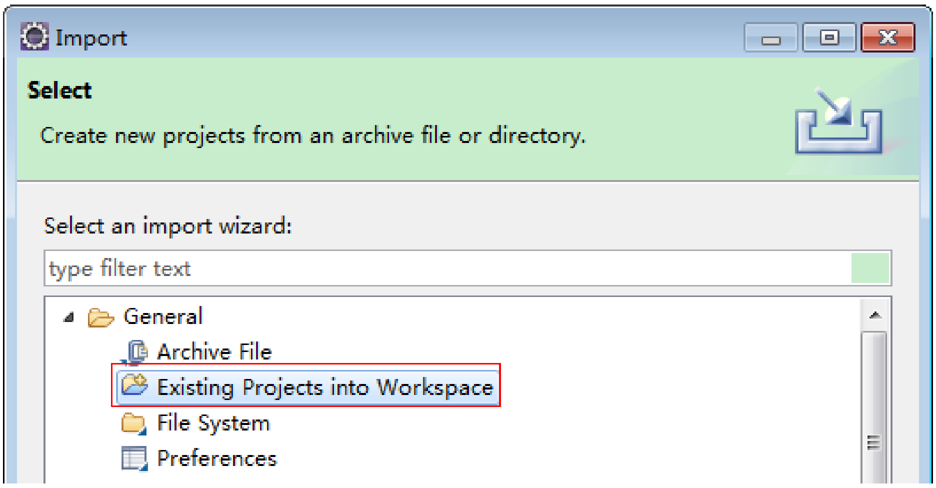
   NOTE

   Keep the access key secure.

Step 2 Download and decompress the demo project.

Step 3 Import the demo project to Eclipse.

Figure 3-2 Selecting Existing Projects into Workspace



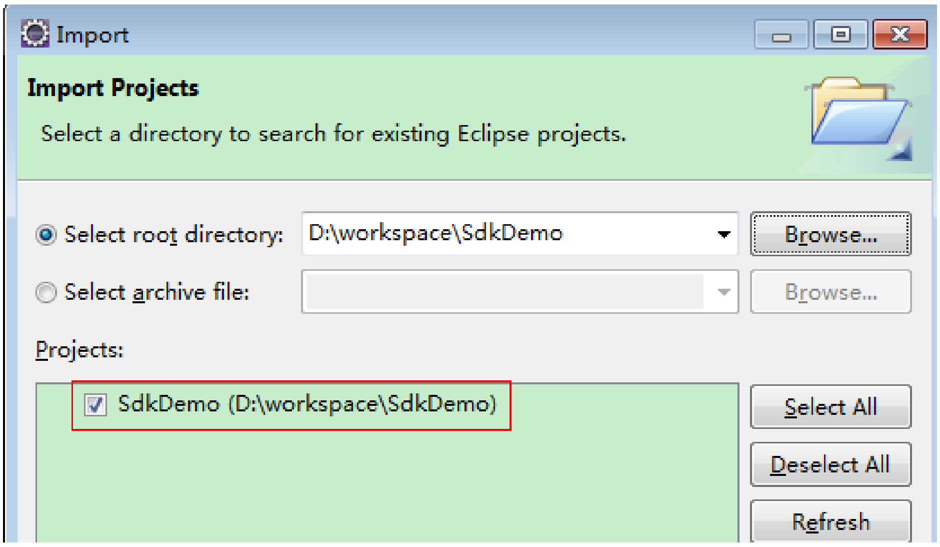
Import

Select

Create new projects from an archive file or directory.

Select an import wizard:

Figure 3-3 Selecting the demo project



Import

Import Projects

Select a directory to search for existing Eclipse projects.

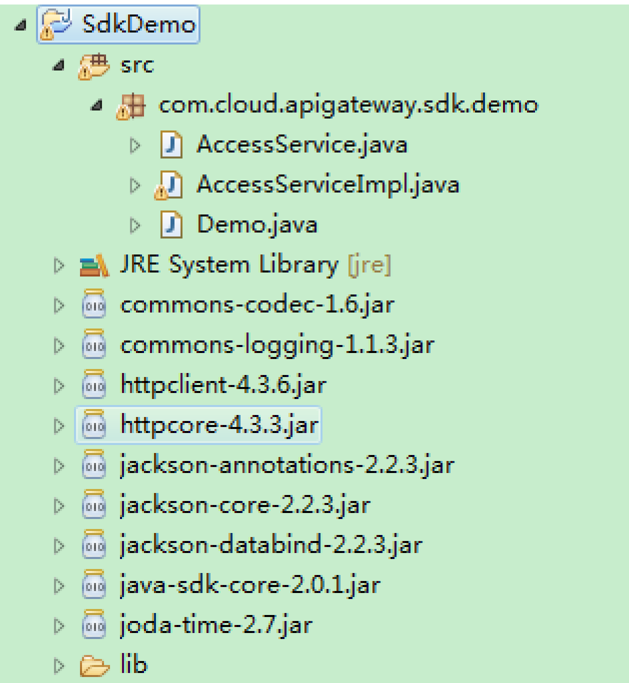
● Select root directory: D:\workspace\SdkDemo

○ Select archive file:

Projects:

☑ SdkDemo (D:\workspace\SdkDemo)

Figure 3-4 Structure of the demo project



```
SdkDemo
  src
    com.cloud.apigateway.sdk.demo
      AccessService.java
      AccessServiceImpl.java
      Demo.java
  JRE System Library [jre]
  commons-codec-1.6.jar
  commons-logging-1.1.3.jar
  httpclient-4.3.6.jar
  httpcore-4.3.3.jar
  jackson-annotations-2.2.3.jar
  jackson-core-2.2.3.jar
  jackson-databind-2.2.3.jar
  java-sdk-core-2.0.1.jar
  joda-time-2.7.jar
  lib
```

Step 4 Sign the request.

The request signing method is integrated in the JAR files imported in **Step 3**. The request needs to be signed before it is sent. The signature will then be added as part of the HTTP header to the request.

The demo code is classified into the following classes to demonstrate signing and sending the HTTP request:

- **AccessService**: An abstract class that merges the GET, POST, PUT, and DELETE methods into the **access** method.
- **Demo**: Execution entry used to simulate the sending of GET, POST, PUT, and DELETE requests.
- **AccessServiceImpl**: Implements the **access** method, which contains the code required for communication with API Gateway.

1. Edit the main method in the **Demo.java** file, and replace the bold text with actual values.

   If you use other methods such as POST, PUT, and DELETE, see the corresponding comment.

   Specify **region**, **serviceName**, **ak/sk**, and **url** as the actual values. In this demo, the URLs for accessing VPC resources are used.

   To obtain the project ID in the URLs, see **Obtaining a Project ID**.

   To obtain the endpoint, contact the enterprise administrator.

```
//TODO: Replace region with the name of the region in which the service to be accessed is located.
private static final String region = "";

//TODO: Replace vpc with the name of the service you want to access. For example, ecs, vpc, iam, and elb.
private static final String serviceName = "";

public static void main(String[] args) throws UnsupportedEncodingException
```

```
{
//TODO: Replace the AK and SK with those obtained on the My Credentials page.
String ak = "ZIRRKMTWP******1WKNKB";
String sk = "Us0mdMNHk******YrRCnW0ecfzl";

//TODO: To specify a project ID (multi-project scenarios), add the X-Project-Id header.
//TODO: To access a global service, such as IAM, DNS, CDN, and TMS, add the X-Domain-Id header to specify an account ID.
//TODO: To add a header, find "Add special headers" in the AccessServiceImple.java file.

//TODO: Test the API
String url = "https://{Endpoint}/v1/{project_id}/vpcs";
get(ak, sk, url);

//TODO: When creating a VPC, replace {project_id} in postUrl with the actual value.
//String postUrl = "https://serviceEndpoint/v1/{project_id}/cloudservers";
//String postbody ="{\"vpc\": {\"name\": \"vpc\",\"cidr\": \"192.168.0.0/16\"}}";
//post(ak, sk, postUrl, postbody);

//TODO: When querying a VPC, replace {project_id} in url with the actual value.
//String url = "https://serviceEndpoint/v1/{project_id}/vpcs/{vpc_id}";
//get(ak, sk, url);

//TODO: When updating a VPC, replace {project_id} and {vpc_id} in putUrl with the actual values.
//String putUrl = "https://serviceEndpoint/v1/{project_id}/vpcs/{vpc_id}";
//String putbody ="{\"vpc\":{\"name\": \"vpc1\",\"cidr\": \"192.168.0.0/16\"}}";
//put(ak, sk, putUrl, putbody);

//TODO: When deleting a VPC, replace {project_id} and {vpc_id} in deleteUrl with the actual values.
//String deleteUrl = "https://serviceEndpoint/v1/{project_id}/vpcs/{vpc_id}";
//delete(ak, sk, deleteUrl);
}
```

2. Compile the code and call the API.

   In the **Package Explorer** area on the left, right-click **Demo.java**, choose **Run AS** > **Java Application** from the shortcut menu to run the demo code.

   You can view API call logs on the console.

----End

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

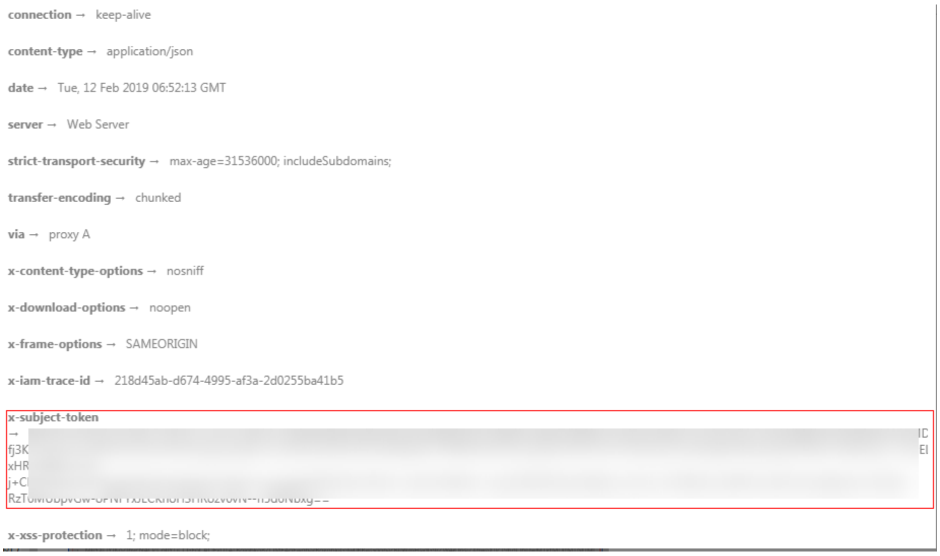
Similar to a request, a response also has a header, for example, **Content-Type**.

Figure 3-5 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.

NOTE

For security purposes, you are advised to set the token in ciphertext in configuration files or environment variables and decrypt it when using it.

Figure 3-5 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
→ [redacted]

x-xss-protection → 1; mode=block;
```

(Optional) Response Body

The body of a response is often returned in a structured format (for example, JSON or XML) 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 request message format is invalid.",
    "error_code": "IMG.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 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.

Procedure

1. Obtain the token by referring to **Authentication**.
2. Query the list of metrics that can be monitored.

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

   Add **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 Parameter description

Parameter	Type	Description
versions	Array of objects	Specifies the list of all versions. For details, see **Table 5-2**.

Table 5-2 versions 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. **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 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	Access to the requested page is forbidden.
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 Codes

See **Error Codes**.

5.1.2 Querying a Specified API Version

Function

This API is used to query a specified API version of 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 Parameter description

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

Table 5-6 versions 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. **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 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 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	Access to the requested page is forbidden.
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 Codes

See **Error Codes**.

5.2 Metrics

5.2.1 Querying Metrics

Function

This API is used to query metrics supported by Cloud Eye. 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	Specifies the namespace of a service. For example, see **Namespace** for the ECS namespace. The namespace must be in the **service.item** format and contain 3 to 32 characters. **service** and **item** each must start with a letter and contain only 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 ECS **dimension** is **instance_id**. A maximum of three dimensions are supported, and the dimensions are numbered from 0 in **dim.{i}=key,value** format. **key** cannot exceed 32 characters and **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	Supported range: **1** to **1000** (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 method is **desc**. ● **asc**: The query results are displayed in the ascending order. ● **desc**: The query results are displayed in the descending order.

- Example requests

  Example request 1: Query all metrics that can be monitored.

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

  Example request 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 Parameter description

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 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 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. Enter 1 to 256 characters.

Table 5-13 meta_data 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 Value	Description
400 Bad Request	Request error.
401 Unauthorized	The authentication information is not provided or is incorrect.
403 Forbidden	Access to the requested page is forbidden.
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 Value	Description
503 Service Unavailable	The service is currently unavailable.

Error Codes

See **Error Codes**.

5.3 Alarm Rules

5.3.1 Querying Alarm Rules

Function

This API is used to query alarm rules. 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. For details, see **Table 5-16**.

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	Supported range: **1** to **100** (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 method 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 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, which can be **1**, **2** (default), **3** or **4**, indicating 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 to be 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 when the alarm status changed. The time is a UNIX timestamp and the unit is ms.
alarm_state	String	Specifies the alarm status, which can be ● **ok**: The alarm status is normal. ● **alarm**: An alarm is generated. ● **insufficient_data**: The required data is insufficient.

Table 5-19 metric data structure description

Parameter	Type	Description
namespace	String	Specifies the namespace of a service. For example, see **Namespace** for the 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 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. Enter 1 to 256 characters.

Table 5-21 alarm_actions data structure description

Parameter	Type	Description
type	String	Specifies the alarm notification type. ● **notification**: indicates that a notification will be sent. ● **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 strings.

Table 5-22 ok_actions 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. ● **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 strings.

Table 5-23 condition 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, which can be ● **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 alarm threshold operator, which can be **>**, **=**, **<**, **>=**, or **<=**.
value	Double	Specifies the alarm threshold. Supported range: **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. Enter up to 32 characters.
count	Integer	Specifies the number of consecutive occurrence times that the alarm policy was met. Supported range: **1** to **5**

Table 5-24 meta_data 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": "Byte/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	Access to the requested page is forbidden.
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 Value	Description
503 Service Unavailable	The service is currently unavailable.

Error Codes

See **Error Codes**.

5.3.2 Querying Details of an Alarm Rule

Function

This API is used to query details of an alarm rule based on its 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 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 5-27**.
condition	Object	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, which can be **1**, **2** (default), **3** or **4**, indicating 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 to be 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 when the alarm status changed. The time is a UNIX timestamp and the unit is ms.
alarm_state	String	Specifies the alarm status, which can be ● **ok**: The alarm status is normal. ● **alarm**: An alarm is generated. ● **insufficient_data**: The required data is insufficient.

Table 5-27 metric data structure description

Parameter	Type	Description
namespace	String	Specifies the namespace of a service. For example, see **Namespace** for the ECS namespace.

Parameter	Type	Description
dimensions	Array of objects	Specifies the list of metric dimensions. For details, see **Table 5-28**.
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 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. Enter 1 to 256 characters.

Table 5-29 alarm_actions data structure description

Parameter	Type	Description
type	String	Specifies the alarm notification type. ● **notification**: indicates that a notification will be sent. ● **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 strings.

Table 5-30 ok_actions 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. ● **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 strings.

Table 5-31 condition 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, which can be ● **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 alarm threshold operator, which can be **>**, **=**, **<**, **>=**, or **<=**.
value	Double	Specifies the alarm threshold. Supported range: **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. Enter up to 32 characters.
count	Integer	Specifies the number of consecutive occurrence times that the alarm policy was met. Supported range: **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	Access to the requested page is forbidden.
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 Value	Description
503 Service Unavailable	The service is currently unavailable.

Error Codes

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	Access to the requested page is forbidden.
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 Codes

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	Access to the requested page is forbidden.
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 Codes

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. Enter 1 to 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
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 5-41**.
alarm_enabled	No	Boolean	Specifies whether to enable the alarm. The default value is **true**.

Parameter	Mandatory	Type	Description
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** must be the same. (You do not need to configure the deprecated parameter **insufficientdata_actions**.)
alarm_level	No	Integer	Specifies the alarm severity, which can be **1**, **2** (default), **3** or **4**, indicating critical, major, minor, and informational, respectively.
alarm_type	No	String	Specifies the alarm rule type. **EVENT.SYS**: The alarm rule is created for system events. **EVENT.CUSTOM**: The alarm rule is created for custom events.

alarm_actions	No	Array of objects	Specifies the action to be 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	Array 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 data structure description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Specifies the namespace of a service. For example, see **Namespace** for the ECS namespace. The namespace must be in the **service.item** format and contain 3 to 32 characters. **service** and **item** each must start with a letter and contain only letters, digits, and underscores (_).
dimensions	No	Array of objects	Specifies the metric dimension list. When **resource_group_id** is not used, **dimensions** is mandatory. For details, see **Table 5-38**.
metric_name	Yes	String	Specifies the metric name. Start with a letter. Enter 1 to 64 characters. Only letters, digits, and underscores (_) are allowed. For details, see the metric name queried in **Querying Metrics**.
resource_group_id	No	String	Specifies the resource group ID selected during the alarm rule creation, for example, **rg1603786526428bWbVmk4rP**. **NOTE** If you create alarm rules for resource groups, you must specify **resource_group_id** and **name**, enter at least one dimension for **dimensions**, and set **alarm_type** to **RESOURCE_GROUP**.

Table 5-38 dimensions data structure description

Parameter	Mandatory	Type	Description
name	Yes	String	Specifies the dimension. For example, the ECS dimension is **instance_id**, which is listed in the **key** column in **Dimension**. Start with a letter. Enter 1 to 32 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
value	Yes	String	Specifies the dimension value, for example, an ECS ID. Start with a letter or a digit. Enter 1 to 256 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Table 5-39 alarm_actions data structure description

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

Parameter	Mandatory	Type	Description
notificationList	Yes	Array of strings	Specifies the list of objects to be notified if the alarm status changes. You can add up to 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 you set **type** to **notification**, you must specify **notificationList**. If you set **type** to **autoscaling**, you must set **notificationList** to **[]**. **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** must be the same. (You do not need to configure the deprecated parameter **insufficientdata_actions**.) ● The IDs in the list are strings.

Table 5-40 ok_actions 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. ● **autoscaling**: indicates that a scaling action will be triggered.

Parameter	Mandatory	Type	Description
notificationList	Yes	Array of objects	Specifies the list of objects to be notified if the alarm status changes. You can add up to 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** must be the same. (You do not need to configure the deprecated parameter **insufficientdata_actions**.)

Table 5-41 condition 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 periods 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 methods are **max**, **min**, **average**, **sum**, or **variance**.
comparison_operator	Yes	String	Specifies the alarm threshold operator. Possible operators are **>**, **=**, **<**, **>=**, and **<=**.
value	Yes	Double	Specifies the alarm threshold. Supported range: **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. Enter up to 32 characters.

Parameter	Mandatory	Type	Description
count	Yes	Integer	Specifies the number of consecutive occurrence times that the alarm policy was met. Supported range: **1** to **5**

- Example request 1

  Creating an alarm rule to monitor a metric

```
{
    "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": "Byte/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"]
        }
    ]
}
```

- Example request 2

  Creating an alarm rule to monitor an event

```
{
 "alarm_name": "alarm-test",
 "metric": {
  "namespace": "SYS.ECS",
  "metric_name": "instance_resize_scheduled",
  "dimensions": [
   {
    "name": "instance_id",
    "value": "d53692e5-828b-495b-a5e2-a1b227f6034c"
   }
  ]
 },
 "condition": {
  "comparison_operator": ">=",
  "count": 1,
  "filter": "average",
  "period": 0,
  "unit": "count",
  "value": 1
 },
 "alarm_enabled": true,
 "alarm_action_enabled": true,
 "alarm_level": 2,
 "alarm_type": "EVENT.SYS",
 "alarm_actions": [
  {
   "type": "notification",
   "notificationList": ["urn:smn:region:ce8476c174f94c6991ea7885e3380d99:sd"]
  }
 ],
 "ok_actions": [
  {
   "type": "notification",
   "notificationList": ["urn:smn:region:ce8476c174f94c6991ea7885e3380d99:sd"]
  }
 ]
}
```

Response

- Response parameters

Table 5-42 Response parameters

Parameter	Type	Description
alarm_id	String	Specifies the alarm rule ID.

- Example response

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

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	Access to the requested page is forbidden.
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 Value	Description
503 Service Unavailable	The service is currently unavailable.

Error Codes

See **Error Codes**.

5.4 Monitoring Data

5.4.1 Querying Monitoring Data of a Metric

Function

This API is used to query the monitoring data of a specified metric at a specified granularity in a specified time range. 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 5-43 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 5-44 Query parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Specifies the namespace of a service. For example, see **Namespace** for the ECS namespace. The namespace must be in the **service.item** format and contain 3 to 32 characters. **service** and **item** each must start with a letter and contain only 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 Metrics**.
from	Yes	String	Specifies the start time of the query. The time is a UNIX timestamp and the unit is ms. Rollup aggregates the raw data generated within a period to the start time of the period. If **from** and **to** are within a period, the query result will be empty due to the rollup failure. Set **from** to at least one period earlier than the current time. 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. In this example, if **period** is 5 minutes, **from** should be 10:30. **NOTE** Cloud Eye rounds up **from** based on the level of granularity required to perform the rollup.
to	Yes	String	Specifies the end time of the query. The time is a UNIX timestamp and the unit is ms. **from** must be earlier than **to**.

Parameter	Mandatory	Type	Description
period	Yes	Integer	Specifies how often Cloud Eye aggregates data, which can be ● **1**: Cloud Eye 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.
filter	Yes	String	Specifies the data rollup method, which can be ● **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.

Parameter	Mandatory	Type	Description
dim	Yes	String	A maximum of three metric dimensions are supported, and the dimensions are numbered from zero in the **dim.{i}=key,value** format. **key** cannot exceed 32 characters and **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 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**

NOTE

- **dimensions** can be obtained from the response body by calling the API for **Querying Metrics**.
- OBS metric data can be queried only when the related OBS APIs are called.

- 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 5-45 Response parameters

Parameter	Type	Description
datapoints	Array of objects	Specifies the metric data list. For details, see **Table 5-46**. Since Cloud Eye rounds up **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 5-46 datapoints data structure description

Parameter	Type	Description
average	Double	Specifies the average value of metric data within a rollup period.
max	Double	Specifies the maximum value of metric data within a rollup period.
min	Double	Specifies the minimum value of metric data within a rollup period.
sum	Double	Specifies the sum of metric data within a rollup period.
variance	Double	Specifies the variance of metric data within a rollup period.
timestamp	Long	Specifies when the metric is collected. It is a UNIX timestamp in milliseconds.
unit	String	Specifies the metric unit.

- Example response

  Example response 1: The dimension is SYS.ECS, and the average CPU usage of ECSs is displayed.

```
{
    "datapoints": [
        {
            "average": 0.23,
            "timestamp": 1442341200000,
            "unit": "%"
        }
    ],
    "metric_name": "cpu_util"
}
```

  Example response 2: The dimension is SYS.ECS, and the sum CPU usage of ECSs is displayed.

```
{
    "datapoints": [
        {
            "sum": 0.53,
            "timestamp": 1442341200000,
            "unit": "%"
        }
    ],
    "metric_name": "cpu_util"
}
```

Example response 3: The dimension is SYS.ECS, and the maximum CPU usage of ECSs is displayed.

```
{
    "datapoints": [
        {
            "max": 0.13,
            "timestamp": 1442341200000,
            "unit": "%"
        }
    ],
    "metric_name": "cpu_util"
}
```

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	Access to the requested page is forbidden.
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 Codes

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 5-47 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 5-48 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. For details, see **Table 5-49**.

Table 5-49 Array elements

Parameter	Mandatory	Type	Description
metric	Yes	Object	Specifies the metric data. For details, see **Table 5-50**.
ttl	Yes	Integer	Specifies the data validity period. The unit is second. Supported range: 1 to 604800 If the validity period expires, the data will be automatically deleted.
collect_time	Yes	Long	Specifies 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, which 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. Possible types: ● **int** ● **float**

Table 5-50 metric data structure description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Specifies the customized namespace. For example, see **Namespace** for the customized ECS namespace. The namespace must be in the **service.item** format and contain 3 to 32 characters. **service** and **item** each must start with a letter and contain only letters, digits, and underscores (_). In addition, **service** cannot start with **SYS**, **AGT**, or **SRE**, and **namespace** cannot be **SERVICE.BMS** because this namespace has been used by the system. You can leave this parameter blank when you set **alarm_type** 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 5-51**.
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 5-51 dimensions data structure description

Parameter	Mandatory	Type	Description
name	Yes	String	Specifies the dimension. For example, the ECS dimension is **instance_id**, which is listed in the **key** column in **Dimension**. Start with a letter. Enter 1 to 32 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
value	Yes	String	Specifies the dimension value, for example, an ECS ID. Start with a letter or a digit. Enter 1 to 256 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

- Example request

  Example request 1: Add **cpu_util** data of a custom dimension. The instance ID is **6f3c6f91-4b24-4e1b-b7d1-a94ac1cb011d**.

```
[
    {
        "metric": {
            "namespace": "MINE.APP",
            "dimensions": [
                {
                    "name": "instance_id",
                    "value": "6f3c6f91-4b24-4e1b-b7d1-a94ac1cb011d"
                }
            ],
            "metric_name": "cpu_util"
        },
        "ttl": 172800,
        "collect_time": 1463598260000,
        "type": "float",
        "value": 0.09,
        "unit": "%"
    },
    {
        "metric": {
            "namespace": "MINE.APP",
            "dimensions": [
                {
                    "name": "instance_id",
                    "value": "6f3c6f91-4b24-4e1b-b7d1-a94ac1cb011d"
                }
            ],
            "metric_name": "cpu_util"
        },
        "ttl": 172800,
        "collect_time": 1463598270000,
        "type": "float",
        "value": 0.12,
        "unit": "%"
    }
]
```

  Example request 2: Add **rds021_myisam_buf_usage** data of the RDS instance whose **rds_cluster_id** is **3c8cc15614ab46f5b8743317555e0de2in01**.

```
[
    {
        "metric": {
            "namespace": "SYS.RDS",
            "dimensions": [
                {
                    "name": "rds_cluster_id",
                    "value": "3c8cc15614ab46f5b8743317555e0de2in01"
                }
            ],
            "metric_name": "rds021_myisam_buf_usage"
        },
        "ttl": 172800,
        "collect_time": 1463598260000,
        "type": "float",
        "value": 0.01,
        "unit": "Ratio"
    }
]
```

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	Access to the requested page is forbidden.
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 Codes

See **Error Codes**.

5.4.3 Querying Monitoring Data of Multiple Metrics

Function

You can query the monitoring data of specified metrics within a specified time range and at a specified granularity. You can query the monitoring data of up to 10 metrics in one batch.

URI

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

- Parameter description

Table 5-52 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 default maximum query intervals of different periods are different.

   If **period** is **1**, the maximum interval between **from** and **to** is 4 hours. If the interval between **from** and **to** is longer than 4 hours, adjust the value of **from** to **to - 4*3600*1000**.

   If **period** is **300**, the maximum interval between **from** and **to** is one day. If the interval between **from** and **to** is longer than 1 day, adjust the value of **from** to **to - 24*3600*1000**.

   If **period** is **1200**, the maximum interval between **from** and **to** is three days. If the interval between **from** and **to** is longer than three days, adjust the value of **from** to **to - 3*24*3600*1000**.

   If **period** is **3600**, the maximum interval between **from** and **to** is 10 days. If the interval between **from** and **to** is longer than 10 days, adjust the value of **from** to **to - 10*24*3600*1000**.

   If **period** is **14400**, the maximum interval between **from** and **to** is 30 days. If the interval between **from** and **to** is longer than 30 days, adjust the value of **from** to **to - 30*24*3600*1000**.

   If **period** is **86400,** the maximum interval between **from** and **to** is 180 days. If the interval between **from** and **to** is longer than 180 days, adjust the value of **from** to **to - 180*24*3600*1000**.

- Request parameters

Table 5-53 Request parameters

Parameter	Mandatory	Type	Description
metrics	Yes	Array of objects	Specifies the metric data. The maximum length of the array is 10. For details, see **Table 5-54**.

Parameter	Mandatory	Type	Description
from	Yes	Long	Specifies the start time of the query. The time is a UNIX timestamp and the unit is ms. Set **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. If **from** and **to** are within a period, the query result will be empty due to the rollup failure. Set **from** to at least one period earlier than the current time. 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. In this example, if **period** is 5 minutes, **from** should be 10:30. **NOTE** Cloud Eye rounds up **from** based on the level of granularity required to perform the rollup.
to	Yes	Long	Specifies the end time of the query. The time is a UNIX timestamp and the unit is ms. **from** must be earlier than **to**.
period	Yes	String	Specifies how often Cloud Eye aggregates data, which can be ● **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 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, which can be ● **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. **filter** does not affect the query result of raw data. (The period is **1**.)

Table 5-54 metrics data structure description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Specifies the metric namespace, which must be in the **service.item** format and contain 3 to 32 characters. **service** and **item** each must start with a letter and contain only letters, digits, and underscores (_).

Parameter	Mandatory	Type	Description

dimensions	Yes	Array of objects	Specifies metric dimensions. **dimensions** is an array consisting of a maximum of four JSON objects.
			One dimension is a JSON object, and its structure is as follows:
			{
			"name": "instance_id",
			"value": "33328f02-3814-422e-b688-bfdba93d4050"
			}
			For details, see **Table 5-55**.

metric_name	Yes	String	Specifies the metric name. Start with a letter. Enter 1 to 64 characters. Only letters, digits, and underscores (_) are allowed.

Table 5-55 dimensions data structure description

Parameter	Mandatory	Type	Description
name	Yes	String	Specifies the dimension. For example, the ECS dimension is **instance_id**, which is listed in the **key** column in **Dimension**. Start with a letter. Enter 1 to 32 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
value	Yes	String	Specifies the dimension value, for example, an ECS ID. Start with a letter or a digit. Enter 1 to 256 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

NOTE

- **dimensions** can be obtained from the response body by calling the API for **Querying Metrics**.
- OBS metric data can be queried only when the related OBS APIs are called.

- Example request

Request example 1: View the average **cpu_util** of the ECS whose **instance_id** is **faea5b75-e390-4e2b-8733-9226a9026070** and the average **network_vm_connections** of the ECS whose **instance_id** is **06b4020f-461a-4a52-84da-53fa71c2f42b**. The monitoring data was collected from 20:00:00 to 22:00:00 on April 30, 2019.

```
{
    "metrics": [
        {
            "namespace": "SYS.ECS",
            "dimensions": [
                {
                    "name": "instance_id",
                    "value": "faea5b75-e390-4e2b-8733-9226a9026070"
                }
            ],
            "metric_name": "cpu_util"
        },
        {
            "namespace": "SYS.ECS",
            "dimensions": [
                {
                    "name": "instance_id",
                    "value": "06b4020f-461a-4a52-84da-53fa71c2f42b"
                }
            ],
            "metric_name": "network_vm_connections"
        }
    ],
    "from": 1556625600000,
    "to": 1556632800000,
    "period": "1",
    "filter": "average"
}
```

Request example 2: View the sums of **rds021_myisam_buf_usage** of the RDS instance whose **rds_cluster_id** is **3c8cc15614ab46f5b8743317555e0de2in01** and the RDS instance whose **rds_cluster_id** is **3b2fa8b55a9b4adca3713962a9d31884in01**. The monitoring data was collected from 20:00:00 to 22:00:00 on April 30, 2019.

```
{
    "metrics": [
        {
            "namespace": "SYS.RDS",
            "dimensions": [
                {
                    "name": "rds_cluster_id",
                    "value": "3c8cc15614ab46f5b8743317555e0de2in01"
                }
            ],
            "metric_name": "rds021_myisam_buf_usage"
        },
        {
            "namespace": "SYS.RDS",
            "dimensions": [
                {
                    "name": "rds_cluster_id",
                    "value": "3b2fa8b55a9b4adca3713962a9d31884in01"
                }
            ],
            "metric_name": "rds021_myisam_buf_usage"
        }
    ],
    "from": 1556625600000,
    "to": 1556632800000,
    "period": "1",
    "filter": "sum"
}
```

Example request 3: View the minimum **proc_specified_count** of the server whose **instance_id** is **cd841102-f6b1-407d-a31f-235db796dcbb** and **proc** is **b28354b543375bfa94dabaeda722927f**. The monitoring data is collected from 20:00:00 to 22:00:00 on April 30, 2019 and the rollup period is 20 minutes.

```
{
    "metrics": [
        {
            "namespace": "AGT.ECS",
            "dimensions": [
                {
                    "name": "instance_id",
                    "value": "cd841102-f6b1-407d-a31f-235db796dcbb"
                },
                {
                    "name": "proc",
                    "value": "b28354b543375bfa94dabaeda722927"
                }
            ],
            "metric_name": "proc_specified_count"
        }
    ],
    "from": 1556625600000,
    "to": 1556632800000,
    "period": "1200",
    "filter": "min"
}
```

Response

- Response parameters

Table 5-56 Response parameters

Parameter	Type	Description
metrics	Array of objects	Specifies the metric data. For details, see **Table 5-57**.

Table 5-57 metrics data structure description

Parameter	Type	Description
unit	String	Specifies the metric unit.
datapoints	Array of objects	Specifies the metric data list. Cloud Eye rounds up the value of **from** based on the selected granularity for data query, so **datapoints** may contain more data points than expected. Up to 3,000 data points can be returned. For details, see **Table 5-59**.
namespace	String	Specifies the metric namespace, which must be in the **service.item** format and contain 3 to 32 characters. **service** and **item** each must start with a letter and contain only letters, digits, and underscores (_).

Parameter	Type	Description

dimensions	Array of objects	Specifies the list of metric dimensions.
		Each dimension is a JSON object, and its structure is as follows:
		{
		"name": "instance_id",
		"value": "33328f02-3814-422e-b688-bfdba93d4050"
		}
		For details, see **Table 5-58**.

metric_name	String	Specifies the metric name. Start with a letter. Enter 1 to 64 characters. Only letters, digits, and underscores (_) are allowed.

Table 5-58 dimensions 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**. Start with a letter. Enter 1 to 32 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
value	String	Specifies the dimension value, for example, an ECS ID. Start with a letter or a digit. Enter 1 to 256 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Table 5-59 datapoints data structure description

Parameter	Type	Description
average	Double	Specifies the average value of metric data within a rollup period.
max	Double	Specifies the maximum value of metric data within a rollup period.
min	Double	Specifies the minimum value of metric data within a rollup period.
sum	Double	Specifies the sum of metric data within a rollup period.

Parameter	Type	Description
variance	Double	Specifies the variance of metric data within a rollup period.
timestamp	Long	Specifies when the metric is collected. It is a UNIX timestamp in milliseconds.

- Example response

  Example response 1: The average **cpu_util** of the ECS whose **instance_id** is **faea5b75-e390-4e2b-8733-9226a9026070** and the average **network_vm_connections** of the ECS whose **instance_id** is **06b4020f-461a-4a52-84da-53fa71c2f42b** are displayed.

```
{
    "metrics": [
        {
            "namespace": "SYS.ECS",
            "metric_name": "cpu_util",
            "dimensions": [
                {
                    "name": "instance_id",
                    "value": "faea5b75-e390-4e2b-8733-9226a9026070"
                }
            ],
            "datapoints": [
                {
                    "average": 0.69,
                    "timestamp": 1556625610000
                },
                {
                    "average": 0.7,
                    "timestamp": 1556625715000
                }
            ],
            "unit": "%"
        },
        {
            "namespace": "SYS.ECS",
            "metric_name": "network_vm_connections",
            "dimensions": [
                {
                    "name": "instance_id",
                    "value": "06b4020f-461a-4a52-84da-53fa71c2f42b"
                }
            ],
            "datapoints": [
                {
                    "average": 1,
                    "timestamp": 1556625612000
                },
                {
                    "average": 3,
                    "timestamp": 1556625717000
                }
            ],
            "unit": "count"
        }
    ]
}
```

  Response example 2: The **rds021_myisam_buf_usage** sums of the RDS instance whose **rds_cluster_id** are **3c8cc15614ab46f5b8743317555e0de2in01** is displayed, and those of the

RDS instance whose **rds_cluster_id** is **3b2fa8b55a9b4adca3713962a9d31884in01** are displayed.

```
{
    "metrics": [
        {
            "unit": "Ratio",
            "datapoints": [
                {
                    "sum": 0.07,
                    "timestamp": 1556625628000
                },
                {
                    "sum": 0.07,
                    "timestamp": 1556625688000
                }
            ],
            "namespace": "SYS.RDS",
            "dimensions": [
                {
                    "name": "rds_cluster_id",
                    "value": "3c8cc15614ab46f5b8743317555e0de2in01"
                }
            ],
            "metric_name": "rds021_myisam_buf_usage"
        },
        {
            "unit": "Ratio",
            "datapoints": [
                {
                    "sum": 0.06,
                    "timestamp": 1556625614000
                },
                {
                    "sum": 0.07,
                    "timestamp": 1556625674000
                }
            ],
            "namespace": "SYS.RDS",
            "dimensions": [
                {
                    "name": "rds_cluster_id",
                    "value": "3b2fa8b55a9b4adca3713962a9d31884in01"
                }
            ],
            "metric_name": "rds021_myisam_buf_usage"
        }
    ]
}
```

Response example 3: The minimum **rds021_myisam_buf_usage** of the server whose **instance_id** is **cd841102-f6b1-407d-a31f-235db796dcbb** and **proc** is **b28354b543375bfa94dabaeda722927f** is displayed.

```
{
    "metrics": [
        {
            "unit": "Ratio",
            "datapoints": [
                {
                    "min": 0,
                    "timestamp": 1556625612000
                },
                {
                    "min": 0,
                    "timestamp": 1556625672000
                }
            ],
            "namespace": "AGT.ECS",
            "dimensions": [
                {
                    "name": "instance_id",
                    "value": "cd841102-f6b1-407d-a31f-235db796dcbb"
                },
                {
                    "name": "proc",
                    "value": "b28354b543375bfa94dabaeda722927f"
                }
            ],
            "metric_name": "rds021_myisam_buf_usage"
        }
    ]
}
```

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	Access to the requested page is forbidden.
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 Codes

See **Error Codes**.

5.5 Quotas

5.5.1 Querying Quotas

Function

This API is used to query the alarm rule quota and the number of alarm rules that have been created.

URI

GET /V1.0/{project_id}/quotas

- Parameter description

Table 5-60 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 5-61 Response parameters

Parameter	Type	Description
quotas	Object	Specifies the quota list. For details, see **Table 5-62**.

Table 5-62 Data structure description of **quotas**

Parameter	Type	Description
resources	Array of objects	Specifies the resource quota list. For details, see **Table 5-63**.

Table 5-63 Data structure description of **resources**

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

Parameter	Type	Description
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	Access to the requested page is forbidden.
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 Codes

See **Error Codes**.

5.6 Event Monitoring

5.6.1 Reporting Events

Function

An API for reporting custom events is provided, which helps you collect and report abnormal events or important change events to Cloud Eye.

URI

POST /V1.0/{project_id}/events

- Parameter description

Table 5-64 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}/events
```

Request

- Request parameters

Table 5-65 Parameter description

Parameter	Type	Mandatory	Description
[Array element]	Array of **EventItem** objects	Yes	Specifies the event list.

Table 5-66 Parameter description of the **EventItem** field

Parameter	Mandatory	Type	Description
event_name	Yes	String	Specifies the event name. Start with a letter. Enter 1 to 64 characters. Only letters, digits, and underscores (_) are allowed.

Parameter	Mandatory	Type	Description
event_source	Yes	String	Specifies the event source. The format is service.item. Set this parameter based on the site requirements. **service** and **item** each must be a string that starts with a letter and contains 3 to 32 characters, including only letters, digits, and underscores (_).
time	Yes	Long	Specifies when the event occurred, which is a UNIX timestamp (ms). **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 one hour 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. For example, if the current time is 2020.01.30 12:00:30, the timestamp inserted must be within the range [2020.01.30 11:00:50, 2020.01.30 12:10:10]. The corresponding UNIX timestamp is [1580353250, 1580357410].
detail	Yes	Detail object	Specifies the event details. For details, see **Table 5-67**.

Table 5-67 detail data structure description

Parameter	Mandatory	Type	Description
content	No	String	Specifies the event content. Enter up to 4,096 characters.
group_id	No	String	Specifies the resource group the event belongs to. This ID must be an existing resource group ID. To query the group ID, perform the following steps: 1. Log in to the management console. 2. Click **Cloud Eye**. 3. Choose **Resource Groups**. Obtain the resource group ID in the **Name /ID** column.

Parameter	Mandatory	Type	Description
resource_id	No	String	Specifies the resource ID. Enter up to 128 characters, including letters, digits, underscores (_), hyphens (-), and colon (:). Example: 6a69bf28-ee62-49f3-9785-845dacd799ec To query the resource ID, perform the following steps: 1. Log in to the management console. 2. Under **Computing**, select **Elastic Cloud Server**. On the **Resource Overview** page, obtain the resource ID.
resource_name	No	String	Specifies the resource name. Enter up to 128 characters, including letters, digits, underscores (_), hyphens (-), and periods (.).
event_state	No	String	Specifies the event status. The value can be **normal**, **warning**, or **incident**.
event_level	No	String	Specifies the event severity. The value can be **Critical**, **Major**, **Minor**, or **Info**.
event_user	No	String	Specifies the event user. Enter up to 64 characters, including letters, digits, underscores (_), hyphens (-), slashes (/), and spaces.
event_type	No	String	Specifies the event type. Its value can be **EVENT.SYS** or **EVENT.CUSTOM**. **EVENT.SYS** indicates system events that cannot be reported by users. Only custom events can be reported.

- Example request

```
[{
    "event_name":"systemInvaded",
    "event_source":"financial.System",
    "time":1522121194000,
    "detail":{
        "content":"The financial system was invaded",
        "group_id":"rg15221211517051YWWkEnVd",
        "resource_id":"1234567890sjgggad",
        "resource_name":"ecs001",
        "event_state":"normal",
        "event_level":"Major",
        "event_user":"xiaokong",
        "event_type": "EVENT.CUSTOM"
    }
},
{
    "event_name":"systemInvaded",
    "event_source":"financial.System",
    "time":1522121194020,
    "detail":{
        "content":"The financial system was invaded",
        "group_id":"rg15221211517051YWWkEnVd",
        "resource_id":"1234567890sjgggad",
        "resource_name":"ecs001",
        "event_state":"normal",
        "event_level":"Major",
        "event_user":"xihong",
        "event_type": "EVENT.CUSTOM"
    }
}]
```

Response

- Response parameters

Table 5-68 Parameter description

Parameter	Type	Description
Array elements	Array of objects	Specifies the event list. For details, see **Table 5-69**.

Table 5-69 Response parameters

Parameter	Mandatory	Type	Description
event_id	Yes	String	Specifies the event ID.
event_name	Yes	String	Specifies the event name. Start with a letter. Enter 1 to 64 characters. Only letters, digits, and underscores (_) are allowed.

- Example response

```
[
    {
        "event_id":"evdgiqwgedkkcvhdjcdu346",
        "event_name":"systemInvaded"
    },
    {
        "event_id":"evdgiqwgedkkcvhdjcdu347",
        "event_name":"systemParalysis"
    }
]
```

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	Access to the requested page is forbidden.
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 Codes

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 correct pagination 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 valid 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.
Zookeeper	500	ces.0021	Internal locking error	Internal locking error	Contact technical support.
Blueflood	500	ces.0019	The metric processing engine is abnormal.	The metric processing engine is abnormal.	Contact technical support.

Module	HTTP Status Code	Error Code	Error Code Description	Error Message	Measure
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

Scenarios

A project ID is required for some URLs when an API is called. Therefore, you need to obtain a project ID in advance. Two methods are available:

- **Obtain the Project ID by Calling an API**
- **Obtain the Project ID from the Console**

Obtain the Project ID by Calling an API

You can obtain the project ID by calling the IAM API used to query project information based on the specified criteria.

The API used to obtain a project ID is GET https://{Endpoint}/v3/projects. {Endpoint} is the IAM endpoint and can be obtained from **Regions and Endpoints**. For details about API authentication, see **Authentication**.

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

```
{
    "projects": [
        {
            "domain_id": "65ewtrgaggshhk1223245sghjlse684b",
            "is_domain": false,
            "parent_id": "65ewtrgaggshhk1223245sghjlse684b",
            "name": "project_name",
            "description": "",
            "links": {
                "next": null,
                "previous": null,
                "self": "https://www.example.com/v3/projects/a4adasfjljaaaakla12334jklga9sasfg"
            },
            "id": "a4adasfjljaaaakla12334jklga9sasfg",
            "enabled": true
        }
    ],
    "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects"
    }
}
```

Obtain a Project ID from the Console

To obtain a project ID from the console, perform the following operations:

1. Log in to the management console.
2. Click the username and select **My Credentials** from the drop-down list.

   On the **My Credentials** page, view the project ID (value in the **Project ID** column).

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 UVP VMTools are accurate.
mem_util UVP VMTools	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 UVP 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 UVP 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 UVP VMTools installed. Otherwise, the **Memory Usage** and **Disk Usage** metrics are unavailable. For details about how to install the UVP VMTools, visit **https://github.com/UVP-Tools/UVP-Tools/**.

Dimension

Key	Value
instance_id	Specifies the ECS ID.

A.2 ECS Metrics Under OS Monitoring (Agent Installed)

Prerequisites

The Agent has been installed and is running properly.

Function

This topic describes OS metrics reported by the ECS plug-in to Cloud Eye as well as their namespaces and dimensions. You can use APIs provided by Cloud Eye to query the metrics and alarms generated by the ECS plug-in.

Namespace

AGT.ECS

Metrics

Metric	Name	Description	Collection Mode (Linux)	Collection Mode (Windows)
cpu_usage_idle	(Agent) Idle CPU Usage	Percentage of time that CPU is idle Unit: Percent	Monitored object: ECS Check the metric value changes in file **/proc/stat** in a collection period. Run the **top** command to check the **%Cpu(s) id** value.	Monitored object: ECS Obtain its value by invoking WindowsAPI GetSystemTimes.
cpu_usage_other	(Agent) Other Process CPU Usage	Percentage of time that the CPU is used by other processes Unit: Percent	Monitored object: ECS **Other Process CPU Usage** = 1- **Idle CPU Usage** - **Kernel Space CPU Usage** - **User Space CPU Usage**	Monitored object: ECS **Other Process CPU Usage** = 1- **Idle CPU Usage** - **Kernel Space CPU Usage** - **User Space CPU Usage**
cpu_usage_system	(Agent) Kernel Space CPU Usage	Percentage of time that the CPU is used by kernel space Unit: Percent	Monitored object: ECS Check the metric value changes in file **/proc/stat** in a collection period. Run the **top** command to check the **%Cpu(s) sy** value.	Monitored object: ECS Obtain its value by invoking WindowsAPI GetSystemTimes.

Metric	Name	Description	Collection Mode (Linux)	Collection Mode (Windows)
cpu_usage_user	(Agent) User Space CPU Usage	Percentage of time that the CPU is used by user space Unit: Percent	Monitored object: ECS Check the metric value changes in file **/proc/stat** in a collection period. Run the **top** command to check the **%Cpu(s) us** value.	Monitored object: ECS Obtain its value by invoking WindowsAPI GetSystemTimes.
cpu_usage	(Agent) CPU Usage	CPU usage of the monitored object Unit: Percent	Monitored object: ECS Check the metric value changes in file **/proc/stat** in a collection period. Run the **top** command to check the **%Cpu(s)** value.	Monitored object: ECS Obtain its value by invoking WindowsAPI GetSystemTimes.
cpu_usage_nice	(Agent) Nice Process CPU Usage	Percentage of time during which the CPU runs in user mode with low-priority processes that can easily be interrupted by higher-priority processes Unit: Percent	Monitored object: ECS Check the metric value changes in file **/proc/stat** in a collection period. Run the **top** command to check the **%Cpu(s) ni** value.	Not supported
cpu_usage_iowait	(Agent) iowait Process CPU Usage	Percentage of time during which the CPU is waiting for I/O operations to complete Unit: Percent	Monitored object: ECS Check the metric value changes in file **/proc/stat** in a collection period. Run the **top** command to check the **%Cpu(s) wa** value.	Not supported
cpu_usage_irq	(Agent) CPU Interrupt Time	Percentage of time that the CPU is servicing interrupts Unit: Percent	Monitored object: ECS Check the metric value changes in file **/proc/stat** in a collection period. Run the **top** command to check the **%Cpu(s) hi** value.	Not supported

Metric	Name	Description	Collection Mode (Linux)	Collection Mode (Windows)
cpu_usage_softirq	(Agent) CPU Software Interrupt Time	Percentage of time that the CPU is servicing software interrupts Unit: Percent	Monitored object: ECS Check the metric value changes in file **/proc/stat** in a collection period. Run the **top** command to check the **%Cpu(s) si** value.	Not supported
load_average1	(Agent) 1-Minute Load Average	CPU load averaged from the last 1 minute	Monitored object: ECS Obtain its value by using the **load1/** value to divide the number of logical CPUs. Run the **top** command to check the **load1** value in the **/proc/loadavg** file.	Not supported
load_average5	(Agent) 5-Minute Load Average	CPU load averaged from the last 5 minutes	Monitored object: ECS Obtain its value by using the **load5/** value to divide the number of logical CPUs. Run the **top** command to check the **load5** value in the **/proc/loadavg** file.	Not supported
load_average15	(Agent) 15-Minute Load Average	CPU load averaged from the last 15 minutes	Monitored object: ECS Obtain its value by using the **load15/** value to divide the number of logical CPUs. Run the **top** command to check the **load15** value in the **/proc/loadavg** file.	Not supported
mem_available	(Agent) Available Memory	Available memory size of the monitored object Unit: GB	Monitored object: ECS Obtain the **MemAvailable** value by checking file **/proc/meminfo**. If it is not displayed in the file, **MemAvailable** = **MemFree** + **Buffers +Cached**	Monitored object: ECS Available Memory = Total memory - Used memory. Obtain its value by invoking WindowsAPI GlobalMemoryStatusEx.

Metric	Name	Description	Collection Mode (Linux)	Collection Mode (Windows)
mem_usedPercent	(Agent) Memory Usage	Memory usage of the monitored object Unit: Percent	Monitored object: ECS Check file **/proc/meminfo**. **AGT. Memory Usage** = (**MemTotal-MemAvailable**)/ **MemTotal**	Monitored object: ECS **Memory Usage** = Used memory/ Total memory *100%
mem_free	(Agent) Idle Memory	Amount of memory that is not being used Unit: GB	Monitored object: ECS Obtain its value by checking file **/proc/meminfo**.	Not supported
mem_buffers	(Agent) Buffer	Memory that is being used for buffers Unit: GB	Monitored object: ECS Obtain its value by checking file **/proc/meminfo**. Run the **top** command to check the **KiB Mem:buffers** value.	Not supported
mem_cached	(Agent) Cache	Memory that is being used for file caches Unit: GB	Monitored object: ECS Obtain its value by checking file **/proc/meminfo**. Run the **top** command to check the **KiB Swap:cached Mem** value.	Not supported

Metric	Name	Description	Collection Mode (Linux)	Collection Mode (Windows)
mountPointPrefix_disk_free	(Agent) Available Disk Space	Available disk space of the monitored object Unit: GB	Monitored object: ECS **mountPointPrefix** is the prefix of the mount point. / in the path is replaced by the word SlAsH. The new path can contain a maximum of 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~). Run the **df -h** command to check data in the **Avail** column.	Monitored object: ECS Use the WMI interface to invoke GetDiskFreeSpaceExW to obtain disk space data. **mountPointPrefix** is the prefix of the mount point. / in the path is replaced by the word SlAsH. The new path can contain a maximum of 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).
mountPointPrefix_disk_total	(Agent) Disk Storage Capacity	Disk storage capacity of the monitored object Unit: GB	Monitored object: ECS **mountPointPrefix** is the prefix of the mount point. / in the path is replaced by the word SlAsH. The new path can contain a maximum of 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~). Run the **df -h** command to check data in the **Size** column.	Monitored object: ECS Use the WMI interface to invoke GetDiskFreeSpaceExW to obtain disk space data. **mountPointPrefix** is the prefix of the mount point. / in the path is replaced by the word SlAsH. The new path can contain a maximum of 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).

Metric	Name	Description	Collection Mode (Linux)	Collection Mode (Windows)
mountPointPrefix_disk_used	(Agent) Used Disk Space	Used disk space of the monitored object Unit: GB	Monitored object: ECS **mountPointPrefix** is the prefix of the mount point. / in the path is replaced by the word SlAsH. The new path can contain a maximum of 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~). Run the **df -h** command to check data in the **Used** column.	Monitored object: ECS Use the WMI interface to invoke GetDiskFreeSpaceExW to obtain disk space data. **mountPointPrefix** is the prefix of the mount point. / in the path is replaced by the word SlAsH. The new path can contain a maximum of 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).
mountPointPrefix_disk_usedPercent	(Agent) Disk I/O Usage	Disk usage of the monitored object Unit: Percent	Monitored object: ECS **mountPointPrefix** is the prefix of the mount point. / in the path is replaced by the word SlAsH. The new path can contain a maximum of 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~). **AGT. Disk Usage** = **AGT. Used Disk Space**/**AGT. Disk Storage Capacity**	Monitored object: ECS Use the WMI interface to invoke GetDiskFreeSpaceExW to obtain disk space data. **mountPointPrefix** is the prefix of the mount point. / in the path is replaced by the word SlAsH. The new path can contain a maximum of 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).

Metric	Name	Description	Collection Mode (Linux)	Collection Mode (Windows)
disk_queue_length	(Agent) Disk Queue Length	Average number of read or write requests queued up for completion for the monitored disk in the monitoring period Unit: Count	Monitored object: ECS The average disk queue length is calculated by calculating the data changes in the fourteenth column of the corresponding device in file **/proc/diskstats** in a collection period. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).	Not supported
disk_write_bytes_per_operation	(Agent) Average Disk Write Size	Average number of bytes in an I/O write for the monitored disk in the monitoring period Unit: KB/op	Monitored object: ECS The average disk write size is calculated by using the data changes in the tenth column of the corresponding device to divide that of the eighth column in file **/proc/diskstats** in a collection period. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).	

Metric	Name	Description	Collection Mode (Linux)	Collection Mode (Windows)
disk_read_bytes_per_operation	(Agent) Average Disk Read Size	Average number of bytes in an I/O read for the monitored disk in the monitoring period Unit: KB/op	Monitored object: ECS The average disk read size is calculated by using the data changes in the sixth column of the corresponding device to divide that of the fourth column in file **/proc/diskstats** in a collection period. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).	
disk_io_svctm	(Agent) Disk I/O Service Time	Average time in an I/O read or write for the monitored disk in the monitoring period Unit: ms/op	Monitored object: ECS The average disk I/O service time is calculated by using the data changes in the thirteenth column of the corresponding device to divide the sum of data changes in the fourth and eighth columns in file **/proc/diskstats** in a collection period. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).	

Metric	Name	Description	Collection Mode (Linux)	Collection Mode (Windows)
disk_inodesTotal	(Agent) Disk inode Total	Total number of index nodes on the disk	Monitored object: ECS **mountPointPrefix** is the prefix of the mount point. / in the path is replaced by the word SlAsH. The new path can contain a maximum of 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~). Run the **df -i** command to check data in the **Inodes** column.	Not supported
disk_inodesUsed	(Agent) Total inode Used	Number of used index nodes on the disk	Monitored object: ECS **mountPointPrefix** is the prefix of the mount point. / in the path is replaced by the word SlAsH. The new path can contain a maximum of 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~). Run the **df -i** command to check data in the **IUsed** column.	Not supported
mountPointPrefix_disk_inodesUsedPercent	(Agent) Percentage of Total inode Used	Ratio of used index nodes to the total index nodes on the disk Unit: Percent	Monitored object: ECS **mountPointPrefix** is the prefix of the mount point. / in the path is replaced by the word SlAsH. The new path can contain a maximum of 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~). Run the **df -i** command to check data in the **IUse%** column.	Not supported

Metric	Name	Description	Collection Mode (Linux)	Collection Mode (Windows)
mountPointPrefix_disk_agt_read_bytes_rate	(Agent) Disks Read Rate	Number of bytes read from the monitored object per second Unit: byte/s	Monitored object: ECS The disks read rate is calculated by calculating the data changes in the sixth column of the corresponding device in file **/proc/diskstats** in a collection period. **mountPointPrefix** is the prefix of the mount point. / in the path is replaced by the word SlAsH. The new path can contain a maximum of 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).	Monitored object: ECS Use the Win32_PerfFormattedData_PerfDisk_LogicalDisk object in WMI to obtain the disk I/O data. **mountPointPrefix** is the prefix of the mount point. / in the path is replaced by the word SlAsH. The new path can contain a maximum of 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).

Metric	Name	Description	Collection Mode (Linux)	Collection Mode (Windows)
mountPointPrefix_disk_agt_read_requests_rate	(Agent) Disks Read Requests	Number of read requests sent to the monitored object per second Unit: Request/s	Monitored object: ECS The disks read requests are calculated by calculating the data changes in the fourth column of the corresponding device in file **/proc/diskstats** in a collection period. **mountPointPrefix** is the prefix of the mount point. / in the path is replaced by the word SlAsH. The new path can contain a maximum of 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).	Monitored object: ECS Use the Win32_PerfFormattedData_PerfDisk_LogicalDisk object in WMI to obtain the disk I/O data. **mountPointPrefix** is the prefix of the mount point. / in the path is replaced by the word SlAsH. The new path can contain a maximum of 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).

Metric	Name	Description	Collection Mode (Linux)	Collection Mode (Windows)
mountPointPrefix_disk_agt_write_bytes_rate	(Agent) Disks Write Rate	Number of bytes written to the monitored disk per second Unit: byte/s	Monitored object: ECS The disks write rate is calculated by calculating the data changes in the tenth column of the corresponding device in file **/proc/diskstats** in a collection period. **mountPointPrefix** is the prefix of the mount point. / in the path is replaced by the word SlAsH. The new path can contain a maximum of 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).	Monitored object: ECS Use the Win32_PerfFormattedData_PerfDisk_LogicalDisk object in WMI to obtain the disk I/O data. **mountPointPrefix** is the prefix of the mount point. / in the path is replaced by the word SlAsH. The new path can contain a maximum of 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).

Metric	Name	Description	Collection Mode (Linux)	Collection Mode (Windows)
mountPointPrefix_disk_agt_write_requests_rate	(Agent) Disks Write Requests	Number of write requests sent to the monitored disk per second Unit: Request/s	Monitored object: ECS The disks write requests are calculated by calculating the data changes in the eighth column of the corresponding device in file **/proc/diskstats** in a collection period. **mountPointPrefix** is the prefix of the mount point. / in the path is replaced by the word SlAsH. The new path can contain a maximum of 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).	Monitored object: ECS Use the Win32_PerfFormattedData_PerfDisk_LogicalDisk object in WMI to obtain the disk I/O data. **mountPointPrefix** is the prefix of the mount point. / in the path is replaced by the word SlAsH. The new path can contain a maximum of 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).

Metric	Name	Description	Collection Mode (Linux)	Collection Mode (Windows)
disk_readTime	(Agent) Average Read Request Time	Average amount of time that read requests have waited on the disks Unit: ms/count	Monitored object: ECS The average read request time is calculated by calculating the data changes in the seventh column of the corresponding device in file **/proc/diskstats** in a collection period.	Monitored object: ECS Use the Win32_PerfFormattedData_PerfDisk_LogicalDisk object in WMI to obtain the disk I/O data. **mountPointPrefix** is the prefix of the mount point. / in the path is replaced by the word SlAsH. The new path can contain a maximum of 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).

Metric	Name	Description	Collection Mode (Linux)	Collection Mode (Windows)
disk_writeTime	(Agent) Average Write Request Time	Average amount of time that write requests have waited on the disks Unit: ms/count	Monitored object: ECS The average write request time is calculated by calculating the data changes in the eleventh column of the corresponding device in file **/proc/diskstats** in a collection period.	Monitored object: ECS Use the Win32_PerfFormattedData_PerfDisk_LogicalDisk object in WMI to obtain the disk I/O data. **mountPointPrefix** is the prefix of the mount point. / in the path is replaced by the word SlAsH. The new path can contain a maximum of 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).

Metric	Name	Description	Collection Mode (Linux)	Collection Mode (Windows)
mountPointPrefix_disk_ioUtils	(Agent) Disk I/O Usage	Percentage of the time that the disk has had I/O requests queued to the total disk operation time Unit: Percent	Monitored object: ECS The disk I/O usage is obtained by calculating the data changes in the thirteenth column of the corresponding device in file **/proc/diskstats** in a collection period.	Monitored object: ECS Use the Win32_PerfFormattedData_PerfDisk_LogicalDisk object in WMI to obtain the disk I/O data. **mountPointPrefix** is the prefix of the mount point. / in the path is replaced by the word SlAsH. The new path can contain a maximum of 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).
disk_fs_rwstate	(Agent) File System Read/ Write Status	Read and write status of the mounted file system of the monitored object Possible statuses are **0** (read and write) and **1** (read only).	Monitored object: ECS Check file system information in the fourth column in file **/proc/mounts**.	Not supported
net_bitSent	(Agent) Inbound Bandwidth	Number of bits sent by this NIC per second Unit: bit/s	Monitored object: ECS Check the metric value changes in file **/proc/net/dev** in a collection period.	Monitored object: ECS Use the MibIfRow object in WMI to obtain the network metric data.
net_bitRecv	(Agent) Outbound Bandwidth	Number of bits received by this NIC per second Unit: bit/s		
net_packetRecv	(Agent) NIC Packet Receive Rate	Number of packets received by this NIC per second Unit: Count/s		

Metric	Name	Description	Collection Mode (Linux)	Collection Mode (Windows)
net_packetSent	(Agent) NIC Packet Send Rate	Number of packets sent by this NIC per second Unit: Count/s		
net_tcp_total	(Agent) TCP TOTAL	Total number of TCP connections of the target NIC		
net_tcp_established	(Agent) TCP ESTABLISHED	Number of ESTABLISHED TCP connections of the target NIC		
net_errin	(Agent) Receive Error Rate	Percentage of receive errors detected by this NIC per second Unit: Percent	Monitored object: ECS Check the metric value changes in file **/proc/net/dev** in a collection period.	Not supported
net_errout	(Agent) Transmit Error Rate	Percentage of transmit errors detected by this NIC per second Unit: Percent		
net_dropin	(Agent) Received Packet Drop Rate	Percentage of packets discarded by this NIC to the total number of packets received by the NIC per second Unit: Percent		
net_dropout	(Agent) Transmitted Packet Drop Rate	Percentage of packets transmitted by this NIC which were dropped per second Unit: Percent		

Dimension

Key	Value
instance_id	ECS ID

A.3 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 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.

Dimension

Key	Value
AutoScalingGroup	AS group ID

A.4 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

Dimension

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

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

Dimension

Key	Value
publicip_id	EIP ID
bandwidth_id	Bandwidth ID

A.6 Monitoring Metrics

Overview

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

Namespace

SYS.ELB

Metrics

Table A-1 Metrics supported by ELB

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: N/A	≥ 0	● Load balancer ● Listener	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
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: N/A	≥ 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: N/A	≥ 0		
m4_ncps	New Connections	Number of connections established between clients and the monitored object per second Unit: Count/s	≥ 0/second		
m5_in_pps	Incoming Packets	Number of packets received by the monitored object per second Unit: Packet/s	≥ 0/second		

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
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 per second Unit: byte/s	≥ 0 bytes/s		
m8_out_Bps	Outbound Rate	Traffic used by the monitored object to access the Internet per second Unit: byte/s	≥ 0 bytes/s		
m9_abnormal_servers	Unhealthy Servers	Number of unhealthy backend servers associated with the monitored object Unit: N/A	≥ 0	● Load balancer	1 minute
ma_normal_servers	Healthy Servers	Number of healthy backend servers associated with the monitored object Unit: N/A	≥ 0		
mb_l7_qps	Layer-7 Query Rate	Number of requests the monitored object receives per second Unit: Query/s	≥ 0 query/s	● Load balancer ● Listener	1 minute
md_l7_http_3xx	Layer-7 3xx Status Codes	Number of 3xx status codes returned by the monitored object Unit: Count/s	≥ 0/second	● Load balancer ● Listener	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
mc_l7_http_2xx	Layer-7 2xx Status Codes	Number of 2xx status codes returned by the monitored object Unit: Count/s	≥ 0/second	- Load balancer - Listener	1 minute
me_l7_http_4xx	Layer-7 4xx Status Codes	Number of 4xx status codes returned by the monitored object Unit: Count/s	≥ 0/second		
mf_l7_http_5xx	Layer-7 5xx Status Codes	Number of 5xx status codes returned by the monitored object Unit: Count/s	≥ 0/second		
m10_l7_http_other_status	Layer-7 Other Status Codes	Number of status codes returned by the monitored object except 2xx, 3xx, 4xx, and 5xx status codes Unit: Count/s	≥ 0/second		
m11_l7_http_404	Layer-7 404 Not Found	Number of 404 Not Found status codes returned by the monitored object Unit: Count/s	≥ 0/second		
m12_l7_http_499	Layer-7 499 Client Closed Request	Number of 499 Client Closed Request status codes returned by the monitored object Unit: Count/s	≥ 0/second		
m13_l7_http_502	Layer-7 502 Bad Gateway	Number of 502 Bad Gateway status codes returned by the monitored object Unit: Count/s	≥ 0/second		

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m14_l7_rt	Average Layer-7 Response Time	Average response time of the monitored object The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. Unit: ms **NOTE** The average response time it takes to establish a WebSocket connection may be very high. This metric cannot be used as a reference.	≥ 0 ms		

a: If a service is being monitored from multiple dimensions, include 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
lbaas_instance_id	Load balancer ID

Key	Value
lbaas_listener_id	ID of a listener added to a load balancer
lbaas_pool_id	ID of the backend server group

A.7 NAT Gateway Metrics

Function

This section describes metrics reported by NAT Gateway 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 NAT Gateway.

Namespace

SYS.NAT

Metrics

Metric	Name	Description	Value Range	Monitored Object
snat_connection	SNAT Connections	Number of SNAT connections initiated by the current user Unit: Count	≥ 0 counts	Active node of NAT Gateway

Dimension

Key	Value
nat_gateway_id	NAT Gateway instance ID

B Change History

Released On	Description
2024-04-15	This issue is the first official release.