Anti-DDoS

API Reference

Issue 01

Date 2024-06-30

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud 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 Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 Before You Start

1.1 Overview

The Anti-DDoS service protects public IP addresses against Layer 4 to Layer 7 distributed denial of service (DDoS) attacks and sends alarms immediately when detecting an attack. Anti-DDoS improves the bandwidth utilization and ensures the stable running of user services.

Anti-DDoS monitors the service traffic from the Internet to public IP addresses and detects attack traffic in real time. It then scrubs attack traffic based on user-configured defense policies without interrupting service running. It also generates monitoring reports that provide visibility into the network traffic security.

This document describes how to use application programming interfaces (APIs) to perform operations to Anti-DDoS, such as querying or updating Anti-DDoS protection policies. For details about all supported operations, see **API Overview**.

Before calling Anti-DDoS APIs, ensure that you are familiar with Anti-DDoS concepts. For details, see section "Service Overview" of the *Anti-DDoS User Guide*.

1.2 API Calling

Anti-DDoS provides Representational State Transfer (REST) APIs, allowing you to use HTTPS requests to call them. For details, see **API Calling**.

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

For details about the constraints, see the API description.

1.5 Concepts

- Region

  Regions are divided based on geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Object Storage Service (OBS), Virtual Private Cloud (VPC), Elastic IP (EIP), and Image Management Service (IMS), are shared within the same region. Regions are classified as universal regions and dedicated regions. A universal region provides universal cloud services for common tenants. A dedicated region provides services of the same type only or for specific tenants.

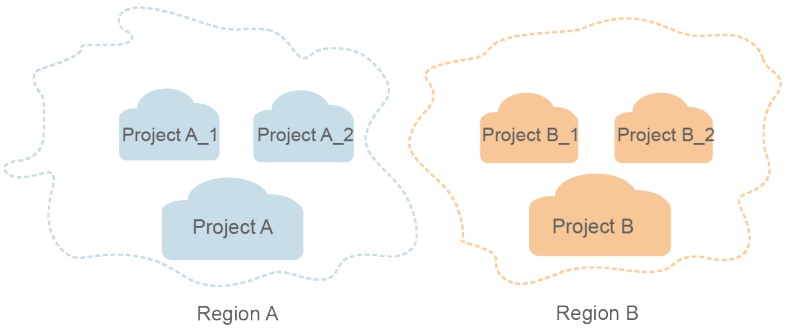
- Availability Zone (AZ)

  An AZ comprises one or multiple physical data centers equipped with independent ventilation, fire, water, and electricity facilities. Compute, 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 support cross-AZ high-availability systems.

- Project

  Projects group and isolate compute, storage, and network resources across physical regions. A default project is provided for each region, and subprojects can be created under each default project. Users can be granted permissions to access all resources in a specific project. For more refined access control, create subprojects under a project and create resources in the subprojects. Users can then be assigned permissions to access only specific resources in the subprojects.

  Figure 1-1 Project isolation model

- Enterprise project

  Enterprise projects group and manage resources across regions. Resources in enterprise projects are logically isolated from each other. An enterprise project can contain resources in multiple regions, and resources can be directly transferred between enterprise projects.

  For more information about enterprise projects and how to obtain enterprise project IDs, see **Enterprise Project Management**.

2 API Overview

You can use all functions of Anti-DDoS through its APIs.

Type	Description
Anti-DDoS service management	These APIs are used to enable and update the Anti-DDoS service, and query Anti-DDoS running information, including EIP defense status, defense statistics, defense traffic, and abnormal events.
Anti-DDoS task management	These APIs are used to query Anti-DDoS tasks.
Anti-DDoS alarm configurations	These APIs are used to query and update Anti-DDoS alarm notification configurations.

3 API Calling

3.1 Making an API Request

This section describes the structure of a REST API request, and uses the API for **obtaining a user token** as an example to demonstrate how to call an API. A token is a user's access credential, which contains the user identity and permission information. 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.

- **URI-scheme**:

  Protocol used to transmit requests. All APIs use HTTPS.

- **Endpoint**:

  Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from **Regions and Endpoints**.

  For example, the endpoint of IAM in the **my-kualalumpur-1** region is **iam.my-kualalumpur-1.alphaedge.tmone.com.my**.

- **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**.

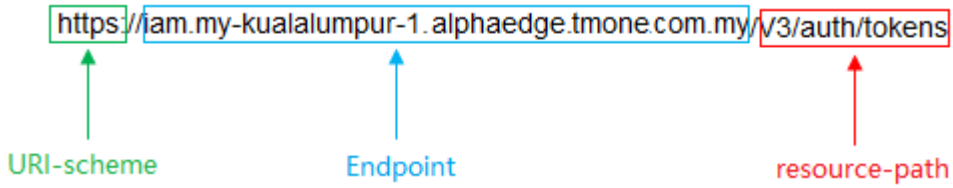
- **query-string**:

  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.

For example, to obtain an IAM token in the **my-kualalumpur-1** region, obtain the endpoint of IAM (**iam.my-kualalumpur-1.alphaedge.tmone.com.my**) for this region and the **resource-path** (**/v3/auth/tokens**) in the URI of the API in **Regions and Endpoints**. Then, construct the URI as follows:

```
https://iam.my-kualalumpur-1.alphaedge.tmone.com.my/v3/auth/tokens
```

Figure 3-1 Example URI



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:

- **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 URI for **obtaining a user token**, the request method is **POST**, and the request is as follows:

```
POST https://iam.my-kualalumpur-1.alphaedge.tmone.com.my/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:

- **Content-Type**: specifies the request body type or format. This field is mandatory and its default value is **application/json**. Other values of this field will be provided for specific APIs if any.
- **X-Auth-Token**: specifies a user token, which is optional. This field is mandatory when token authentication is used. The user token is a response to

the API used to **obtain a user token**. This API is the only one that does not require authentication.

NOTE

In addition to supporting token-based authentication, APIs also support authentication using access key ID/secret access key (AK/SK). During AK/SK-based authentication, an SDK is used to sign the request, and the **Authorization** (signature information) and **X-Sdk-Date** (time when the request is sent) header fields are automatically added to the request.

For more information, see **AK/SK-based Authentication**.

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

```
POST https://iam.my-kualalumpur-1.alphaedge.tmone.com.my/v3/auth/tokens
Content-Type: application/json
```

Request Body

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

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

In the case of the API used to **obtain a user token**, the request parameters and parameter description can be obtained from the API request. Below is a sample request that includes a body. Replace ***username***, ***domainname***, ********, and ***xxxxxxxxxxxxxxxxx*** with the actual values. ***username*** indicates the username, ***domainname*** indicates the name of the account to which the user belongs, ******** indicates the login password, and ***xxxxxxxxxxxxxxxxx*** indicates the project name, which can be obtained from .

NOTE

The **scope** parameter specifies where a token takes effect. In the following example, the token takes effect only for the resources in a specified project. In the following example, the token takes effect only for the resources in the specified project. For more information about this API, see **Obtaining a User Token Through Password Authentication**.

```
POST https://iam.my-kualalumpur-1.alphaedge.tmone.com.my/v3/auth/tokens
Content-Type: application/json
{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "username",
                    "password": "********",
                    "domain": {
                        "name": "domainname"
                    }
                }
            }
        },
        "scope": {
            "project": {
```



```
            "name": "xxxxxxxxxxxxxxxxx"
        }
    }
  }
}
```

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

3.2 Authentication

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

- Token-based authentication: Requests are authenticated using a token.
- AK/SK-based authentication: Requests are authenticated by encrypting the request body using an AK/SK pair. This method is recommended because it provides higher security than token-based authentication.

Token-based Authentication

NOTE

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

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

The token can be obtained by calling the required API. For more information, see **Obtaining a User Token**. A project-level token is required for calling this API, that is, **auth.scope** must be set to **project** in the request body. Example:

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


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

```
POST https://iam.my-kualalumpur-1.alphaedge.tmone.com.my/v3/auth/tokens
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

AK/SK-based Authentication

NOTE

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

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

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

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

NOTICE

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

3.3 Response

Status Code

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

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

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**.

The following shows the response header for the API to **obtain a user token**, in which **x-subject-token** is the desired user token. This token can then be used to authenticate the calling of other APIs.

Figure 3-2 Header fields of the response to the request for obtaining a user token

(Optional) Response Body

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

The following shows part of the response body for the API to **obtain a user token**. For the sake of space, only part of the content is displayed here.

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

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

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

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

4 API

4.1 DDoS Protection Management

4.1.1 Querying the List of Defense Statuses of EIPs

Functions

This API enables you to query the defense statuses of all EIPs, regardless whether an EIP has been bound to an Elastic Cloud Server (ECS) or not.

URI

- URI format

  GET /v1/{project_id}/antiddos

  NOTE

  You can use **?** and **&** behind the URI to add query conditions, as shown in the request example.

- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID.

Request

Table 4-1 Parameter description

Parameter	Mandatory	Type	Description
status	No	String	Possible values: • **normal**: indicates that the defense status is normal. • **configuring**: indicates that defense is being configured. • **notConfig**: indicates that defense is not configured. • **packetcleaning**: indicates that traffic cleaning is underway. • **packetdropping**: indicates that traffic is discarded. If this parameter is not used, the defense statuses of all ECSs are displayed in the Neutron-queried order by default.
limit	No	Integer	Maximum number of returned results. The value ranges from 1 to 100.
offset	No	Integer	Offset. The value ranges from 0 to 2147483647.
ip	No	String	IP address. Both IPv4 and IPv6 addresses are supported. For example, if you enter **?ip=192.168**, the defense status of EIPs corresponding to 192.168.111.1 and 10.192.168.8 is returned.

Response

- Parameter description

Name	Type	Description
total	Integer	Total number of EIPs
ddosStatus	List data structure	List of protection statuses

- Data structure description of **ddosStatus**

Parameter	Type	Description
floating_ip_address	String	Floating IP address
floating_ip_id	String	ID of an EIP
network_type	String	EIP type. The value can be: • **EIP**: EIP that is bound or not bound with ECS. • **ELB**: EIP that is bound with ELB.
status	String	Defense status, the possible value of which is one of the following: • **normal**: indicates that the defense status is normal. • **configuring**: indicates that defense is being configured. • **notConfig**: indicates that defense is not configured. • **packetcleaning**: indicates that traffic cleaning is underway. • **packetdropping**: indicates that traffic is discarded.
blackhole_endtime	Integer	End time of a black hole.
protect_type	String	Protection type
traffic_threshold	Integer	Traffic threshold
http_threshold	Integer	HTTP traffic threshold.

Example

- Example request
  ```
  GET /v1/67641fe6886f43fcb78edbbf0ad0b99f/antiddos?status=packetdropping
  ```
- Example response
  ```
  {
     "total": 1,
     "ddosStatus": [
        {
           "floating_ip_id": "18e6ace5-eb36-4196-a15e-1e000c24e026",
           "floating_ip_address": "139.9.116.167",
           "network_type": "EIP",
           "status": "normal",
           "blackhole_endtime": 0,
           "protect_type": "default",
           "traffic_threshold": 99,
           "http_threshold": 0
        }
     ]
  }
  ```


Status Code

See **Status Code**.

4.1.2 Querying Anti-DDoS specifications

Functions

This API allows you to query optional Anti-DDoS defense policies. Based on your service, you can select a policy for Anti-DDoS traffic cleaning.

URI

- URI format

  GET /v1/{project_id}/antiddos/query_config_list

- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID.

Request

Request parameters

None

Response

- Parameter description

Parameter	Type	Description
traffic_limited_list	List data structure	List of traffic limits
http_limited_list	List data structure	List of HTTP limits
connection_limited_list	List data structure	List of connection limits

- Data structure description of **traffic_limited_list**

Parameter	Type	Description
traffic_pos_id	Integer	Position ID of traffic
traffic_per_second	Integer	Threshold of traffic per second (Mbit/s)
packet_per_second	Integer	Threshold of number of packets per second

- Data structure description of **http_limited_list**

Parameter	Type	Description
http_request_pos_id	Integer	Position ID of number of HTTP requests
http_packet_per_second	Integer	Threshold of number of HTTP requests per second

- Data structure description of **connection_limited_list**

Parameter	Type	Description
cleaning_access_pos_id	Integer	Position ID of access limit during cleaning
new_connection_limited	Integer	Number of new connections of a source IP address
total_connection_limited	Integer	Total number of connections of a source IP address

Example

- Example request
  ```
  GET /v1/67641fe6886f43fcb78edbbf0ad0b99f/antiddos/query_config_list
  ```
- Example response
  ```
  {
    "traffic_limited_list": [
      {
        "traffic_pos_id": 1,
        "traffic_per_second": 10,
        "packet_per_second": 2000
      },
      {
        "traffic_pos_id": 2,
        "traffic_per_second": 30,
        "packet_per_second": 6000
      },
    ],
    "http_limited_list": [
      {
        "http_request_pos_id": 1,
        "http_packet_per_second": 100
      },
      {
        "http_request_pos_id": 2,
        "http_packet_per_second": 150
      },
    ],
    "connection_limited_list": [
      {
        "cleaning_access_pos_id": 1,
        "new_connection_limited": 10,
        "total_connection_limited": 30
      },
      {
        "cleaning_access_pos_id": 2,
  ```



```
            "new_connection_limited": 20,
            "total_connection_limited": 100
        },
    ],
    "extend_ddos_config": []
}
```


NOTE

The **extend_ddos_config** field displays information about Anti-DDoS defense policies set by users based on their needs.

Status Code

See **Status Code**.

4.1.3 Querying Weekly Defense Statistics

Functions

This API allows you to query weekly defense statistics about all your EIPs, including the number of intercepted DDoS attacks, number of attacks, and ranking by the number of attacks. Currently, you can query weekly statistics up to four weeks before the current time. Data older than four weeks cannot be queried.

URI

- URI format

  GET /v1/{project_id}/antiddos/weekly

  NOTE

  You can use **?** and **&** behind the URI to add query conditions, as shown in the request example.

- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID.

Request

Table 4-2 Parameter description

Parameter	Mandatory	Type	Description
period_start_date	No	String	Start date of a week

Response

- Parameter description

Name	Type	Description
ddos_intercept_times	Integer	Number of DDoS attacks blocked in a week
weekdata	Data structure	Number of attacks in a week
top10	Data structure	Top 10 attacked IP addresses

- Data structure description of **weekdata**

Parameter	Type	Description
ddos_intercept_times	Integer	Number of DDoS attacks blocked
ddos_blackhole_times	Integer	Number of DDoS black holes
max_attack_bps	Integer	Maximum attack traffic
max_attack_conns	Integer	Maximum number of attack connections
period_start_date	Long integer	Start time

- Data structure description of **top10**

Parameter	Type	Description
floating_ip_address	String	EIP
times	Integer	Number of DDoS attacks intercepted, including cleaning operations and black holes

Example

- Example request

```
GET /v1/67641fe6886f43fcb78edbbf0ad0b99f/antiddos/weekly?
period_start_date=1006510306
```

- Example response

```
{
  "ddos_intercept_times": 23,
  "weekdata": [
    {
      "ddos_intercept_times": 0,
      "ddos_blackhole_times": 0,
      "max_attack_bps": 0,
      "max_attack_conns": 0,
      "period_start_date": 1474214461651
    },
    {
      "ddos_intercept_times": 0,
      "ddos_blackhole_times": 0,
      "max_attack_bps": 0,
```



```
      "max_attack_conns": 0,
      "period_start_date": 1474300861651
    },
    {
      "ddos_intercept_times": 0,
      "ddos_blackhole_times": 0,
      "max_attack_bps": 0,
      "max_attack_conns": 0,
      "period_start_date": 1474387261651
    },
    {
      "ddos_intercept_times": 0,
      "ddos_blackhole_times": 0,
      "max_attack_bps": 0,
      "max_attack_conns": 0,
      "period_start_date": 1474473661651
    },
    {
      "ddos_intercept_times": 0,
      "ddos_blackhole_times": 0,
      "max_attack_bps": 0,
      "max_attack_conns": 0,
      "period_start_date": 1474560061651
    },
    {
      "ddos_intercept_times": 2,
      "ddos_blackhole_times": 0,
      "max_attack_bps": 16375,
      "max_attack_conns": 0,
      "period_start_date": 1474646461651
    },
    {
      "ddos_intercept_times": 1,
      "ddos_blackhole_times": 0,
      "max_attack_bps": 0,
      "max_attack_conns": 0,
      "period_start_date": 1474732861651
    }
  ],
  "top10": [
    {
      "floating_ip_address": "192.168.44.69",
      "times": 23
    }
  ]
}
```


Status Code

See **Status Code**.

4.1.4 Querying Configured Anti-DDoS Defense Policies

Functions

This API enables you to query configured Anti-DDoS defense policies. You can query the policy of a specified EIP.

URI

- URI format

  GET /v1/{project_id}/antiddos/{floating_ip_id}

- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
floating_ip_id	Yes	String	ID corresponding to the EIP of a user

Request

Table 4-3 Parameter description

Parameter	Mandatory	Type	Description
ip	No	String	EIP of a user

Response

Table 4-4 Parameter description

Parameter	Type	Description
enable_L7	Boolean	Whether to enable layer-7 protection. • **true**: Enable layer 7 protection. • **false**: Disable layer 7 protection.
traffic_pos_id	Integer	Position ID of traffic. The value ranges from 1 to 9, or 99.
http_request_pos_id	Integer	Position ID of number of HTTP requests. The value ranges from 1 to 15.
cleaning_access_pos_id	Integer	Position ID of access limit during cleaning. The value ranges from 1 to 8, or 99.
app_type_id	Integer	Application type ID. Possible values: • 0 • 1

Example

- Example request
  ```
  GET /v1/67641fe6886f43fcb78edbbf0ad0b99f/antiddos/1df977c2-fdc6-4483-bc1c-ba46829f57b8
  ```
- Example response
  ```
  {
    "enable_L7": true,
    "traffic_pos_id": 1,
    "http_request_pos_id": 1,
  ```



```
  "cleaning_access_pos_id": 1,
  "app_type_id": 1
}
```

Status Code

See **Status Code**.

4.1.5 Enabling Anti-DDoS

Functions

This API is used to enable the Anti-DDoS defense. Successfully invoking this API only means that the service node has received the enabling request. You need to use the task querying API to check the task execution status. For details about the task querying API, see **Querying Anti-DDoS Tasks**.

URI

- URI format

  POST /v1/{project_id}/antiddos/{floating_ip_id}

- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	User ID
floating_ip_id	Yes	String	ID corresponding to the Elastic IP Address (EIP) of a user

Request

Table 4-5 Parameter description

Parameter	Mandatory	Type	Description
enable_L7	No	Boolean	Whether to enable L7 defense
traffic_pos_id	Yes	Integer	Position ID of traffic. The value ranges from 1 to 9.
http_request_pos_id	Yes	Integer	Position ID of number of HTTP requests. The value ranges from 1 to 15.
cleaning_access_pos_id	Yes	Integer	Position ID of access limit during cleaning. The value ranges from 1 to 8.

Parameter	Mandatory	Type	Description
app_type_id	No	Integer	Application type ID. Possible values: • 0 • 1

Response

Table 4-6 Parameter description

Name	Type	Description
error_code	String	Internal error code
error_description	String	Internal error description
task_id	String	ID of a task. This ID can be used to query the status of the task. This field is reserved for use in task auditing later. It is temporarily unused.

Example

- Example request

```
POST /v1/67641fe6886f43fcb78edbbf0ad0b99f/antiddos/1df977c2-fdc6-4483-bc1c-ba46829f57b8
{
   "enable_L7":true,
   "traffic_pos_id":1,
   "http_request_pos_id":1,
   "cleaning_access_pos_id":1,
   "app_type_id":1
}
```

- Example response

```
{
  "error_code": "10000000",
  "error_description": "Task has been received and is being processed.",
  "task_id": "94e17e18-5b2c-40c6-a218-8ec5134e32a5"
}
```

Status Code

See **Status Code**.

4.1.6 Updating Anti-DDoS Defense Policies

Functions

This API enables you to update the Anti-DDoS defense policy of a specified EIP. Successfully invoking this API only means that the service node has received the

update request. You need to use the task querying API to check the task execution status. For details about the task querying API, see **Querying Anti-DDoS Tasks**.

URI

- URI format

  PUT /v1/{project_id}/antiddos/{floating_ip_id}

- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
floating_ip_id	Yes	String	ID corresponding to the EIP of a user

Request Parameters

Table 4-7 Parameter description

Parameter	Mandatory	Type	Description
enable_L7	No	Boolean	Whether to enable layer-7 protection. • **true**: Enable layer 7 protection. • **false**: Disable layer 7 protection.
traffic_pos_id	Yes	Integer	Position ID of traffic. The value ranges from 1 to 9, or 99.
http_request_pos_id	Yes	Integer	Position ID of number of HTTP requests. The value ranges from 1 to 15.
cleaning_access_pos_id	Yes	Integer	Position ID of access limit during cleaning. The value ranges from 1 to 8, or 99.
app_type_id	No	Integer	Application type ID. Possible values: • 0 • 1

Response

Table 4-8 Parameter description

Name	Type	Description
error_code	String	Internal error code
error_msg	String	Internal error description
task_id	String	ID of a task. This ID can be used to query the status of the task. This field is reserved for use in task auditing later. It is temporarily unused.

Example

- Example request

```
PUT /v1/67641fe6886f43fcb78edbbf0ad0b99f/antiddos/ee0c854e-082f-499e-b7d8-1b42c22781af
{
    "enable_L7":false,
    "traffic_pos_id":2,
    "http_request_pos_id":1,
    "cleaning_access_pos_id":1,
    "app_type_id":1
}
```

- Example response

```
{
  "error_code": "10000000",
  "error_msg": "The task has been received and is being handled",
  "task_id": "4a4fefe7-34a1-40e2-a87c-16932af3ac4a"
}
```

Status Code

See **Status Code**.

4.1.7 Querying the Traffic of a Specified EIP

Functions

This API allows you to query the traffic of a specified EIP in the last 24 hours. Traffic is detected in five-minute intervals.

URI

- URI format

  GET /v1/{project_id}/antiddos/{floating_ip_id}/daily

- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Parameter	Mandatory	Type	Description
floating_ip_id	Yes	String	ID corresponding to the EIP of a user

Request

Table 4-9 Parameter description

Parameter	Mandatory	Type	Description
ip	No	String	EIP of a user

Response

- Parameter description

Name	Type	Description
data	Data structure	Traffic in the last 24 hours

- Data structure description of **data**

Parameter	Type	Description
period_start	Long integer	Start time
bps_in	Integer	Inbound traffic (bit/s)
bps_attack	Integer	Attack traffic (bit/s)
total_bps	Integer	Total traffic
pps_in	Integer	Inbound packet rate (number of packets per second)
pps_attack	Integer	Attack packet rate (number of packets per second)
total_pps	Integer	Total packet rate

Example

- Example request

```
GET /v1/67641fe6886f43fcb78edbbf0ad0b99f/antiddos/1df977c2-fdc6-4483-bc1c-ba46829f57b8/daily
```

- Example response

```
{
   "data": [
      {
```



```
            "period_start": 1606188642720,
            "bps_in": 0,
            "bps_attack": 0,
            "total_bps": 0,
            "pps_in": 0,
            "pps_attack": 0,
            "total_pps": 0
        }
    ]
}
```

Status Code

See **Status Code**.

4.1.8 Querying Events of a Specified EIP

Functions

This API allows you to query events of a specified EIP in the last 24 hours. Events include cleaning and blackhole events, and the query delay is within five minutes.

URI

- URI format

  GET /v1/{project_id}/antiddos/{floating_ip_id}/logs

  NOTE

  You can use **?** and **&** behind the URI to add query conditions, as shown in the request example.

- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
floating_ip_id	Yes	String	ID corresponding to the EIP of a user

Request

Table 4-10 Parameter description

Parameter	Mandatory	Type	Description
limit	No	Integer	Limit of number of returned results or the maximum number of returned results of a query. The value ranges from 1 to 100, and this parameter is used together with the **offset** parameter. If neither **limit** nor **offset** is used, query results of all ECSs are returned.
offset	No	Integer	Offset. This parameter is valid only when used together with the **limit** parameter.
sort_dir	No	String	Possible values: • **desc**: indicates that query results are given and sorted by time in descending order. • **asc**: indicates that query results are given and sorted by time in ascending order. The default value is **desc**.
ip	No	String	EIP of a user

Response

- Parameter description

Name	Type	Description
total	Integer	Total number of EIPs
logs	Data structure	List of events

- Data structure description of **logs**

Parameter	Type	Description
start_time	Long integer	Start time
end_time	Long integer	End time

Parameter	Type	Description
status	Integer	Defense status, the possible value of which is one of the following: • **1**: indicates traffic scrubbing. • **2**: indicates blackhole.
trigger_bps	Integer	Traffic at the triggering point
trigger_pps	Integer	Packet rate at the triggering point
trigger_http_pps	Integer	HTTP request rate at the triggering point

Example

- Example request
  ```
  GET /v1/67641fe6886f43fcb78edbbf0ad0b99f/antiddos/1df977c2-fdc6-4483-bc1c-ba46829f57b8/logs
  ```
- Example response
  ```
  {
    "total": 1,
    "logs": [
      {
        "start_time": 1473217200000,
        "end_time": 1473242400000,
        "status": 1,
        "trigger_bps": 51106,
        "trigger_pps": 2600,
        "trigger_http_pps": 3589
      }
    ]
  }
  ```

Status Code

See **Status Code**.

4.1.9 Querying the Defense Status of a Specified EIP

Functions

This API is used to query the defense status of a specified EIP.

URI

- URI format

  GET /v1/{project_id}/antiddos/{floating_ip_id}/status
- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Parameter	Mandatory	Type	Description
floating_ip_id	Yes	String	ID corresponding to the EIP of a user

Request

Table 4-11 Parameter description

Parameter	Mandatory	Type	Description
ip	No	String	EIP of a user

Response

- Parameter description

Parameter	Type	Description
status	String	Defense status, the possible value of which is one of the following: • **normal**: indicates that the defense status is normal. • **configuring**: indicates that defense is being configured. • **notConfig**: indicates that defense is not configured. • **packetcleaning**: indicates traffic scrubbing. • **packetdropping**: indicates blackhole.

Example

- Example request

```
GET /v1/67641fe6886f43fcb78edbbf0ad0b99f/antiddos/1df977c2-fdc6-4483-bc1c-ba46829f57b8/status
```

- Example response

```
{
  "status": "normal"
}
```

Status Code

See **Status Code**.

4.2 Anti-DDoS Task Management

4.2.1 Querying Anti-DDoS Tasks

Functions

This API enables you to query the execution status of a specified Anti-DDoS configuration task.

URI

- URI format

  GET /v1/{project_id}/query_task_status

  NOTE

  You can use **?** and **&** behind the URI to add query conditions, as shown in the request example.

- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request

Table 4-12 Parameter description

Parameter	Mandatory	Type	Description
task_id	Yes	String	Task ID (nonnegative integer) character string

Response

- Parameter description

Name	Type	Description
task_status	String	Status of a task, which can be one of the following: • **success** • **failed** • **waiting** • **running** • **preprocess** • **ready**
task_msg	String	Additional information about a task

Example

- Example request
```
GET /v1/67641fe6886f43fcb78edbbf0ad0b99f/query_task_status?
task_id=4a4fefe7-34a1-40e2-a87c-16932af3ac4a
```
- Example response
```
{
  "task_status": "running",
  "task_msg": ""
}
```

Status Code

See **Status Code**.

4.3 Alarm Configuration Management

4.3.1 Querying Alarm Configuration

Functions

This API allows you to query alarm configuration, such as whether a certain type of alarms will be received, and whether alarms are received through SMS messages or emails.

URI

- URI format

  GET /v2/{project_id}/warnalert/alertconfig/query
- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request

None

Response

- Parameter description

Parameter	Type	Description
warn_config	List data structure	Alarm configuration
topic_urn	String	ID of an alarm group
display_name	String	Specifies the name of the SMN topic used for sending alarm notifications.

- Data structure description of **warn_config**

Parameter	Type	Description
antiDDoS	Boolean	DDoS attacks
bruce_force	Boolean	Brute force cracking (system logins, FTP, and DB)
remote_login	Boolean	Alarms about remote logins
weak_password	Boolean	Weak passwords (system and database)
high_privilege	Boolean	Overly high rights of a database process
back_doors	Boolean	Webshells
waf	Boolean	Reserved field
send_frequency	Integer	Frequency • **0**: indicates that alarms are sent once a day. • **1**: indicates that alarms are sent once every half hour.

Example

- Example request

```
GET /v2/67641fe6886f43fcb78edbbf0ad0b99f/warnalert/alertconfig/query
```

- Example response

```
{
  "warn_config":    {
    "antiDDoS": true,
    "bruce_force": false,
    "remote_login": false,
    "weak_password": false,
    "high_privilege": false,
```



```
    "back_doors": false,
    "waf": false
  },
  "topic_urn": "urn:smn::67641fe6886f43fcb78edbbf0ad0b99f:test_soft",
  "display_name": "group_1"
}
```

NOTE

SFTP is more secure than FTP. To secure data transmission, use SFTP to transfer files.

Status Code

See **Status Code**.

4.3.2 Updating Alarm Configuration

Functions

This API allows you to update alarm configuration, such as whether a certain type of alarms will be received, and whether alarms are received through SMS messages or emails.

URI

- URI format

  POST /v2/{project_id}/warnalert/alertconfig/update

- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request

- Parameter description

Parameter	Mandatory	Type	Description
warn_config	Yes	List data structure	Alarm configuration
topic_urn	Yes	String	ID of an alarm group
display_name	Yes	String	Specifies the name of the SMN topic used for sending alarm notifications.

- Data structure description of **warn_config**

Parameter	Mandatory	Type	Description
antiDDoS	No	Boolean	DDoS attacks
bruce_force	No	Boolean	Brute force cracking (system logins, FTP, and DB)
remote_login	No	Boolean	Alarms about remote logins
weak_password	No	Boolean	Weak passwords (system and database)
high_privilege	No	Boolean	Overly high rights of a database process
back_doors	No	Boolean	Webshells
waf	No	Boolean	Reserved field

NOTE

SFTP is more secure than FTP. To secure data transmission, use SFTP to transfer files.

Response Message

Parameter	Type	Description
error_code	String	Internal error code
error_msg	String	Internal error description
task_id	String	Task ID

Example

- Example request

```
{
  "warn_config":    {
     "antiDDoS": true,
     "bruce_force": false,
     "remote_login": false,
     "weak_password": false,
     "high_privilege": false,
     "back_doors": false,
     "waf": false
  },
  "topic_urn": "urn:smn::67641fe6886f43fcb78edbbf0ad0b99f:test_soft",
  "display_name": "group_1"
}
```

- Example response

```
{
 "error_code" : "10000000",
```



```
  "error_msg" : "Ok",
  "task_id" : ""
}
```

Status Code

For details, see **Status Code**.

A Status Code

- Normal

Returned Value	Description
200	The request is successfully processed.

- Abnormal

Status Code	Status	Description
400	Bad Request	The server fails to process the request.
401	Unauthorized	The requested page requires a username and a password.
403	Forbidden	Access to the requested page is denied.
404	Not Found	The server fails to find the requested page.
405	Method Not Allowed	Method specified in the request is not allowed.
406	Not Acceptable	Response generated by the server is not acceptable to the client.
407	Proxy Authentication Required	Proxy authentication is required before the request is processed.
408	Request Timeout	A timeout error occurs because the request is not processed within the specified waiting period of the server.
409	Conflict	The request cannot be processed due to a conflict.
500	Internal Server Error	The request is not processed due to a server error.

Status Code	Status	Description
501	Not Implemented	The request is not processed because the server does not support the requested function.
502	Bad Gateway	The request is not processed, and the server receives an invalid response from the upstream server.
503	Service Unavailable	The request is not processed due to a temporary system abnormality.
504	Gateway Timeout	A gateway timeout error occurs.

B Anti-DDoS Error Codes

Status Code	Error Code	Message	Description	Solution
200	Anti-DDoS.0	Succeeded	Execution succeeded.	No action is required.
200	Anti-DDoS.10000000	The task has been received and is being handled	The task has been received and is being processed.	No action is required.
400	Anti-DDoS.10000001	Enter a valid request message	The request is invalid.	Check parameters.
400	Anti-DDoS.10001008	An incorrect task ID is used	An incorrect task ID is used.	Check parameters
400	Anti-DDoS.10001010	Invalid time	The time is invalid.	Check parameters.
401	Anti-DDoS.10000004	Public test service denied	The OBT service is rejected.	Apply for OBT.
403	Anti-DDoS.10000002	Failed to authenticate the token in the request	Failed to authenticate the token carried in the request.	Apply for a new token.
403	Anti-DDoS.10000009	The account is restricted	The account is restricted.	Apply for permissions.

Status Code	Error Code	Message	Description	Solution
403	Anti-DDoS.10000010	The account is frozen	The account is frozen.	Apply for unfreezing.
403	Anti-DDoS.10000012	Unknown user type	The account is of an unknown type.	Apply for permissions.
403	Anti-DDoS.10000016	VPC access failed or EIP is not exist	VPC cannot be accessed or the EIP does not exist.	Contact the administrator.
403	Anti-DDoS.10000030	You have not been authenticated. Perform real-name authentication first.	You have not been authenticated. Perform real-name authentication first.	Complete real-name authentication.
403	Anti-DDoS.10001009	The operation permission is restricted	The operation is restricted.	Apply for permissions.
403	Anti-DDoS.11000001	Access to the database is rejected	Access to the database has been denied.	Contact the administrator.
500	Anti-DDoS.11000000	Internal system exception. Contact technical support engineers	A system error occurs, please contact technical support engineers.	Contact the administrator.

C Obtaining a Project ID

Obtaining a Project ID by Calling an API

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**.

In the following example, **id** indicates the project ID.

```
{
    "projects": [
        {
            "domain_id": "65382450e8f64ac0870cd180d14e684b",
            "is_domain": false,
            "parent_id": "65382450e8f64ac0870cd180d14e684b",
            "name": "xxxxxxxx",
            "description": "",
            "links": {
                "next": null,
                "previous": null,
                "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f897d6b99"
            },
            "id": "a4a5d4098fb4474fa22cd05f897d6b99",
            "enabled": true
        }
    ],
    "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects"
    }
}
```

Obtaining a Project ID from the Console

A project ID is required for some URLs when an API is called. To obtain a project ID, perform the following operations:

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

   On the **My Credential** page, view project IDs in the project list.

D Change History

Release Date	Change Description
2024-06-30	This is the second official release. • Added **Enabling Anti-DDoS**. • Optimized descriptions in **API Overview**. • Optimized descriptions in **API Calling**. • Optimized the document structure.
2020-09-30	This is the first official release.

