

Anti-DDoS

API Reference

Issue 01
Date 2020-11-06



Copyright © Huawei Technologies Co., Ltd. 2021. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks 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 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.

Contents

1 Before You Start.....	1
1.1 Overview.....	1
1.2 API Calling.....	1
1.3 Endpoints.....	1
1.4 Notes and Constraints.....	1
1.5 Concepts.....	2
2 API Overview.....	4
3 API Calling.....	5
3.1 Making an API Request.....	5
3.2 Authentication.....	8
3.3 Response.....	9
4 API.....	11
4.1 Anti-DDoS APIs.....	11
4.1.1 Querying Optional Anti-DDoS Defense Policies.....	11
4.1.2 Querying Configured Anti-DDoS Defense Policies.....	16
4.1.3 Updating Anti-DDoS Defense Policies.....	17
4.1.4 Querying Anti-DDoS Tasks.....	19
4.1.5 Querying the List of Defense Statuses of EIPs.....	20
4.1.6 Querying the Defense Status of a Specified EIP.....	23
4.1.7 Querying the Traffic of a Specified EIP.....	24
4.1.8 Querying Events of a Specified EIP.....	26
4.1.9 Querying Weekly Defense Statistics.....	28
4.2 Alarm Reminding APIs.....	31
4.2.1 Querying Alarm Configuration.....	31
4.2.2 Updating Alarm Configuration.....	33
A Appendix.....	35
A.1 Status Code.....	35
A.2 Error Codes.....	36
A.3 Obtaining a Project ID.....	37
B Change History.....	39

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 policy. 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

- **Account**

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

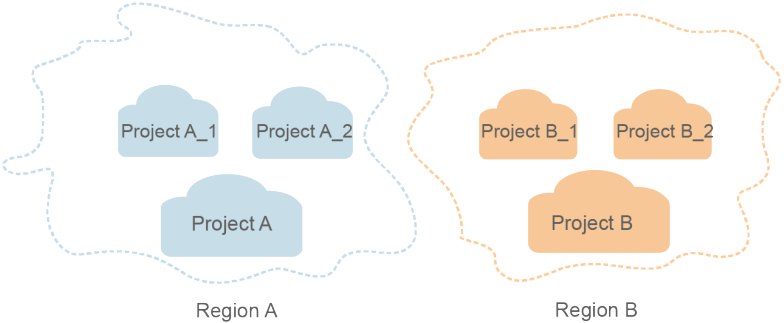
The domain name, username, and password will be required for API authentication.
- **Region**

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 resources (including 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



2 API Overview

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

Type	Description
APIs of Anti-DDoS	Anti-DDoS APIs, including the APIs for querying and updating the Anti-DDoS defense policies.
Alarm Reminding APIs	Alarm reminding APIs, including the API for querying alarm configuration information

3 API Calling

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.

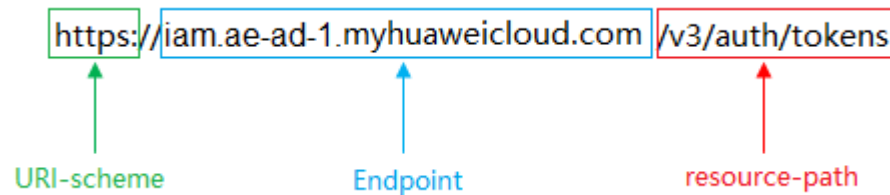
- **URI-scheme:**
Protocol used to transmit requests. All APIs use HTTPS.
- **Endpoint:**
Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from [Regions and Endpoints](#).
For example, the endpoint of IAM in the **ae-ad-1** region is **iam.ae-ad-1.myhuaweicloud.com**.
- **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 **ae-ad-1** region, obtain the endpoint of IAM (**iam.ae-ad-1.myhuaweicloud.com**) for this region and the **resource-path**

(/v3/auth/tokens) in the URI of the API used to **obtain a user token**. Then, construct the URI as follows:

```
https://iam.ae-ad-1.myhuaweicloud.com/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 case of the API used to **obtain a user token**, the request method is POST. The request is as follows:

```
POST https://iam.ae-ad-1.myhuaweicloud.com/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 only for token-based API authentication. 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 requests for calling the API. An example of such requests is as follows:

```
POST https://iam.ae-ad-1.myhuaweicloud.com/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. The following provides an example request with a body included. Set **username** to the name of a user, **domainname** to the name of the account that the user belongs to, ********* to the user's login password, and **xxxxxxxxxxxxxxxxxxxx** to the project name. You can learn more information about projects from [Regions and Endpoints](#).

 NOTE

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

```
POST https://iam.ae-ad-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

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

```
}  
}  
}
```

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.ae-ad-1.myhuaweicloud.com/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 Signature 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 , 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 , 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

```
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 → MIIVXQYJKoZIhvcNAQcCoIIVTjCCGEoCAQExDTALBgIghkgBZQMEAgEwgharBgkqhkiG9w0BBwGgghacBIWmHsidG9rZW4iOansiZXhwaXJlc19hdCI6IjwMTktMDItMTNUMC
fj3Kjs6YgKnpVNRbW2eZ5eb78SZOkajACgklqO1wi4JIGzrpd18LGXK5bdfq4lqHCYb8P4NaYONYejcAgzVefYtLWT1GSO0zxKZmlQHq82HBqHdgIZO9fuEbl5dMhdavj+33wEl
xHRCE9I87o+k9-
j+CMZSEB7bUGd5Uj6eRASXl1jipPEGA270g1FruooL6jggIFkNPQuFSOU8+uSsttVwRtnfsC+qTp22Rkd5MCqFGQ8LcuUxC3a+9CMBnOintWW7oeRUUVhVpxk8pxiX1wTEboX-
RzT6MUbpvGw-oPNFYxJECKnoH3HRozv0vN--n5d6Nbxg==
x-xss-protection → 1; mode=block;
```

(Optional) Response Body

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

The following shows part of the response body for the API to . 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": "xxxxxxx",
            .....

```

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 Anti-DDoS APIs

4.1.1 Querying Optional Anti-DDoS Defense Policies

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 scrubbing.

URI

- URI format
GET /v1/{project_id}/antiddos/query_config_list
- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	User ID

Request

Request parameters

None

Response

- Parameter description

Parameter	Mandatory	Type	Description
traffic_limited_list	Yes	List data structure	List of traffic limits
http_limited_list	Yes	List data structure	List of HTTP limits
connection_limited_list	Yes	List data structure	List of limits of numbers of connections

- Data structure description of **traffic_limited_list**

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

- Data structure description of **http_limited_list**

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

- Data structure description of **connection_limited_list**

Parameter	Mandatory	Type	Description
cleaning_access_pos_id	Yes	Integer	Position ID of access limit during cleaning
new_connection_limited	Yes	Integer	Number of new connections of a source IP address
total_connection_limited	Yes	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
    },
    {
      "traffic_pos_id": 3,
      "traffic_per_second": 50,
      "packet_per_second": 10000
    },
    {
      "traffic_pos_id": 4,
      "traffic_per_second": 70,
      "packet_per_second": 15000
    },
    {
      "traffic_pos_id": 5,
      "traffic_per_second": 100,
      "packet_per_second": 20000
    },
    {
      "traffic_pos_id": 6,
      "traffic_per_second": 150,
      "packet_per_second": 25000
    },
    {
      "traffic_pos_id": 7,
      "traffic_per_second": 200,
      "packet_per_second": 35000
    },
    {
      "traffic_pos_id": 8,
      "traffic_per_second": 250,
      "packet_per_second": 50000
    },
    {
      "traffic_pos_id": 9,
      "traffic_per_second": 300,
      "packet_per_second": 70000
    }
  ],
  "http_limited_list": [
    {
      "http_request_pos_id": 1,
      "http_packet_per_second": 100
    },
    {
      "http_request_pos_id": 2,
      "http_packet_per_second": 150
    },
    {
      "http_request_pos_id": 3,
      "http_packet_per_second": 240
    },
    {
      "http_request_pos_id": 4,
```



```
"http_packet_per_second": 350
},
{
  "http_request_pos_id": 5,
  "http_packet_per_second": 480
},
{
  "http_request_pos_id": 6,
  "http_packet_per_second": 550
},
{
  "http_request_pos_id": 7,
  "http_packet_per_second": 700
},
{
  "http_request_pos_id": 8,
  "http_packet_per_second": 850
},
{
  "http_request_pos_id": 9,
  "http_packet_per_second": 1000
},
{
  "http_request_pos_id": 10,
  "http_packet_per_second": 1500
},
{
  "http_request_pos_id": 11,
  "http_packet_per_second": 2000
},
{
  "http_request_pos_id": 12,
  "http_packet_per_second": 3000
},
{
  "http_request_pos_id": 13,
  "http_packet_per_second": 5000
},
{
  "http_request_pos_id": 14,
  "http_packet_per_second": 10000
},
{
  "http_request_pos_id": 15,
  "http_packet_per_second": 20000
}
],
"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
  },
  {
    "cleaning_access_pos_id": 3,
    "new_connection_limited": 30,
    "total_connection_limited": 200
  },
  {
    "cleaning_access_pos_id": 4,
    "new_connection_limited": 40,
    "total_connection_limited": 250
  },
  {
```

```
    "cleaning_access_pos_id": 5,
    "new_connection_limited": 50,
    "total_connection_limited": 300
  },
  {
    "cleaning_access_pos_id": 6,
    "new_connection_limited": 60,
    "total_connection_limited": 500
  },
  {
    "cleaning_access_pos_id": 7,
    "new_connection_limited": 70,
    "total_connection_limited": 600
  },
  {
    "cleaning_access_pos_id": 8,
    "new_connection_limited": 80,
    "total_connection_limited": 700
  }
],
"extend_ddos_config": [
  {
    "new_connection_limited": 80,
    "total_connection_limited": 700,
    "http_packet_per_second": 500000,
    "traffic_per_second": 1000,
    "packet_per_second": 200000,
    "setID": 33
  },
  {
    "new_connection_limited": 80,
    "total_connection_limited": 700,
    "http_packet_per_second": 500000,
    "traffic_per_second": 2000,
    "packet_per_second": 200000,
    "setID": 34
  },
  {
    "new_connection_limited": 80,
    "total_connection_limited": 700,
    "http_packet_per_second": 500000,
    "traffic_per_second": 5000,
    "packet_per_second": 400000,
    "setID": 35
  },
  {
    "new_connection_limited": 80,
    "total_connection_limited": 700,
    "http_packet_per_second": 0,
    "traffic_per_second": 0,
    "packet_per_second": 0,
    "setID": 36
  }
]
}
```

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](#).

4.1.2 Querying Configured Anti-DDoS Defense Policies

Functions

This API enables you to query the Anti-DDoS defense 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	User ID
floating_ip_id	Yes	String	ID corresponding to the EIP of a user

Request

Request parameters

None

Response

Table 4-1 Parameter description

Parameter	Mandatory	Type	Description
enable_L7	Yes	Boolean	Whether L7 defense has been enabled.
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.
app_type_id	Yes	Integer	Application type ID. Possible values: <ul style="list-style-type: none"> • 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](#).

4.1.3 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	User ID
floating_ip_id	Yes	String	ID corresponding to the EIP of a user

Request

Table 4-2 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.

Parameter	Mandatory	Type	Description
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.
app_type_id	No	Integer	Application type ID. Possible values: <ul style="list-style-type: none"> • 0 • 1

Response

Table 4-3 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

```
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_description": "Task has been received and is being processed.",
  "task_id": "4a4fefe7-34a1-40e2-a87c-16932af3ac4a"
}
```

Status Code

See [Status Code](#).

4.1.4 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	User ID

Request

Table 4-4 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: <ul style="list-style-type: none">• 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](#).

4.1.5 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	User ID

Request

Table 4-5 Parameter description

Parameter	Mandatory	Type	Description
status	No	String	<p>Possible values:</p> <ul style="list-style-type: none"> • normal: indicates that the defense status is normal. • configging: 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. <p>If this parameter is not used, the defense statuses of all ECSs are displayed in the Neutron-queried order by default.</p>
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 defense statuses

- Data structure description of **ddosStatus**

Parameter	Mandatory	Type	Description
floating_ip_address	Yes	String	Floating IP address
floating_ip_id	Yes	String	ID of an EIP
network_type	Yes	String	EIP type. The value can be: <ul style="list-style-type: none"> • EIP: EIP that is bound or not bound with ECS. • ELB: EIP that is bound with ELB.
status	Yes	String	Defense status, the possible value of which is one of the following: <ul style="list-style-type: none"> • normal: indicates that the defense status is normal. • configging: 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.

Example

- Example request
GET /v1/67641fe6886f43fcb78edbbf0ad0b99f/antiddos?status=packetdropping

- Example response

```
{
  "total": 5,
  "ddosStatus": [
    {
      "floating_ip_id": "1867f954-fc11-4202-8247-6af2144867ea",
      "floating_ip_address": "192.168.42.221",
      "network_type": "EIP",
      "status": "notConfig"
    },
    {
      "floating_ip_id": "49c6af49-9ace-42e6-ab89-1eee1f4ac821",
      "floating_ip_address": "192.168.35.152",
      "network_type": "EIP",
      "status": "normal"
    },
    {
      "floating_ip_id": "7a8dc957-083b-499d-b7cf-6fa48f4880c5",
      "floating_ip_address": "192.168.42.222",
      "network_type": "EIP",
      "status": "notConfig"
    }
  ],
  {
```

```
    "floating_ip_id": "7c6676a0-b281-4163-9d0d-cb6485ae9860",  
    "floating_ip_address": "192.168.44.69",  
    "network_type": "EIP",  
    "status": "normal"  
  },  
  {  
    "floating_ip_id": "969c1d48-6a92-4ef1-b66c-b17c7e7d7ce7",  
    "floating_ip_address": "192.168.47.192",  
    "network_type": "EIP",  
    "status": "notConfig"  
  }  
]  
}
```

Status Code

See [Status Code](#).

4.1.6 Querying the Defense Status of a Specified EIP

Functions

This API allows you to query the defense status of a specified EIP.

URI

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

Parameter	Mandator y	Type	Description
project_id	Yes	String	User ID
floating_ip_id	Yes	String	ID corresponding to the EIP of a user

Request

Request parameters

None

Response

- Parameter description

Parameter	Mandatory	Type	Description
status	Yes	String	Defense status, the possible value of which is one of the following: <ul style="list-style-type: none">• normal: indicates that the defense status is normal.• configging: 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.

Example

- Example request
GET /v1/67641fe6886f43fcb78edbbf0ad0b99f/antiddos/1df977c2-fdc6-4483-bc1c-ba46829f57b8/status
- Example response
{ "status": "normal" }

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	User ID
floating_ip_id	Yes	String	ID corresponding to the EIP of a user

Request

Request parameters

None

Response

- Parameter description

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

- Data structure description of **data**

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

Example

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

- Example response

```
{"data": [  
  {  
    "period_start": 1472713370609,  
    "bps_in": 0,  
    "bps_attack": 0,  
    "total_bps": 0,  
    "pps_in": 0,  
    "pps_attack": 0,  
    "total_pps": 0  
  },  
]
```

```

...
{
  "period_start": 1472713670609,
  "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	User ID
floating_ip_id	Yes	String	ID corresponding to the EIP of a user

Request

Table 4-6 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: <ul style="list-style-type: none">• 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 .

Response

- Parameter description

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

- Data structure description of **logs**

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

Parameter	Mandatory	Type	Description
status	Yes	Integer	Defense status, the possible value of which is one of the following: <ul style="list-style-type: none">• 1: indicates that traffic cleaning is underway.• 2: indicates that traffic is discarded.
trigger_bps	Yes	Integer	Traffic at the triggering point
trigger_pps	Yes	Integer	Packet rate at the triggering point
trigger_http_pps	Yes	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 Weekly Defense Statistics

Functions

This API allows you to query weekly defense statistics about all your IP addresses, 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	User ID

Request

Table 4-7 Parameter description

Parameter	Mandatory	Type	Description
period_start_date	No	String	Start date of a seven-day period

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	Mandatory	Type	Description
ddos_intercept_times	Yes	Integer	Number of DDoS attacks blocked
ddos_blackhole_times	Yes	Integer	Number of DDoS black holes
max_attack_bps	Yes	Integer	Maximum attack traffic
max_attack_conns	Yes	Integer	Maximum number of attack connections

Parameter	Mandatory	Type	Description
period_start_date	Yes	Long integer	Start time

- Data structure description of **top10**

Parameter	Mandatory	Type	Description
floating_ip_addresses	Yes	String	EIP
times	Yes	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](#).

4.2 Alarm Reminding APIs

4.2.1 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	User ID

Request

Request parameters

None

Response

- 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	Description of an alarm group

- Data structure description of **warn_config**

Parameter	Mandatory	Type	Description
antiDDoS	Yes	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

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:aaa: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](#).

4.2.2 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	User 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	Description of an alarm group

- Data structure description of **warn_config**

Parameter	Mandatory	Type	Description
antiDDoS	Yes	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)

Parameter	Mandatory	Type	Description
high_privilege	No	Boolean	Overly high rights of a database process
back_doors	No	Boolean	Webshells
waf	No	Boolean	Reserved

 **NOTE**

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

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:aaa:67641fe6886f43fcb78edbbf0ad0b99f:test_soft",
  "display_name": "group_1"
}
```

Status Code

For details, see [Status Code](#).

A Appendix

A.1 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.

Status Code	Status	Description
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.
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.

A.2 Error Codes

Table A-1 Error codes

Code	Description
0	Success
10000000	The task has been received and is being processed.
10000001	The request message is in an invalid format.
10000002	In the request, the token has not been authenticated or the authentication fails.
10000003	The access to the authentication service (IAM) has been denied.
10000005	The network access has been denied.
10000006	The access to Neutron has been denied.
10000007	The access to Nova has been denied.
10000008	The service is not enabled.
10000009	The account is limited.
10000010	The account is frozen.
10000011	The account is abnormal.
10000012	The account is of an unknown type.

Code	Description
10000016	VPC cannot be accessed or the EIP does not exist.
10001001	The network access has been denied.
10001002	An invalid traffic triggering parameter is used.
10001003	An invalid HTTP request triggering parameter is used.
10001004	An invalid cleaning limit parameter is used.
10001005	An invalid application type parameter is used.
10001006	The server is busy processing other requests, please try again later.
10001007	This operation is not allowed in the current status.
10001008	An incorrect task ID is used.
10001009	The operation rights are limited.
10001020	ID of the IP address is invalid.
10005001	An invalid sending frequency parameter is used.
10005002	The alarm configuration fails to be updated.
10005003	An update operation is being performed. Try again later.
10005004	Invalid topic_urn .
10005005	Invalid display_name .
10001010	The entered time range is invalid.
11000000	A system error occurs, please contact technical support engineers.
11000001	The access to the database has been denied.
11000002	The access to ATIC has been denied.
12100001	The specified ECS does not exist.

A.3 Obtaining a Project ID

Obtaining a 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](#).

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

```
{
  "projects": [
    {
      "domain_id": "65382450e8f64ac0870cd180d14e684b",
      "is_domain": false,
      "parent_id": "65382450e8f64ac0870cd180d14e684b",
      "name": "xxxxxxx",
      "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.

B Change History

Release Date	Description
2020-11-06	This is the first official release.