DDoS 防护 AAD

API 参考

文档版本 05

发布日期 2024-01-08

华为技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：
https://www.huawei.com/cn/psirt/vul-response-process
如企业客户须获取漏洞信息，请参见如下网址：
https://securitybulletin.huawei.com/enterprise/cn/security-advisory

目 录

1 使用前必读

1.1 概述

针对DDoS攻击，华为云提供多种安全防护方案，您可以根据您的实际业务选择合适的防护方案。华为云DDoS防护服务（Anti-DDoS Service，简称AAD）提供了DDoS原生基础防护（Anti-DDoS流量清洗）、DDoS原生高级防护和DDoS高防三个子服务。

DDoS原生基础防护（Anti-DDoS流量清洗）服务（以下简称Anti-DDoS）为华为云内公网IP资源（弹性云服务器、弹性负载均衡），提供网络层和应用层的DDoS攻击防护（如泛洪流量型攻击防护、资源消耗型攻击防护），并提供攻击拦截实时告警，有效提升用户带宽利用率，保障业务稳定可靠。

您可以使用本文档提供的API对Anti-DDoS进行相关操作，如查询、更新Anti-DDoS服务等。支持的全部操作请参见**API概览**。

在调用Anti-DDoS API之前，请确保已经充分了解Anti-DDoS相关概念，详细信息请参见《DDoS防护用户指南》的“产品介绍”章节。

1.2 调用说明

Anti-DDoS提供了REST（Representational State Transfer）风格API，支持您通过HTTPS请求调用，调用方法请参见**如何调用API**。

1.3 终端节点

终端节点（Endpoint）即调用API的**请求地址**，不同服务不同区域的终端节点不同，您可以从**地区和终端节点**中查询服务的终端节点。

1.4 约束与限制

详细的约束限制请参见具体API的说明。

1.5 基本概念

- 账号

  用户注册时的账号，账号对其所拥有的资源及云服务具有完全的访问权限，可以重置用户密码、分配用户权限等。由于账号是付费主体，为了确保账号安全，建议您不要直接使用账号进行日常管理工作，而是创建用户并使用他们进行日常管理工作。

- 用户

  由账号在IAM中创建的用户，是云服务的使用人员，具有身份凭证（密码和访问密钥）。

  在**我的凭证**下，您可以查看账号ID和用户ID。通常在调用API的鉴权过程中，您需要用到账号、用户和密码等信息。

- 区域（Region）

  从地理位置和网络时延维度划分，同一个Region内共享弹性计算、块存储、对象存储、VPC网络、弹性公网IP、镜像等公共服务。Region分为通用Region和专属Region，通用Region指面向公共租户提供通用云服务的Region；专属Region指只承载同一类业务或只面向特定租户提供业务服务的专用Region。

  详情请参见**区域和可用区**。

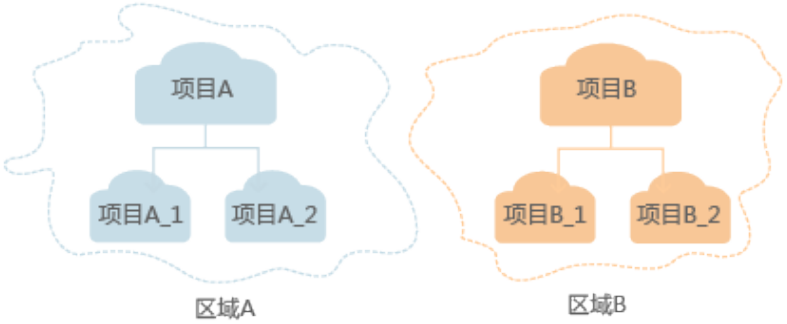
- 可用区（AZ，Availability Zone）

  一个AZ是一个或多个物理数据中心的集合，有独立的风火水电，AZ内逻辑上再将计算、网络、存储等资源划分成多个集群。一个Region中的多个AZ间通过高速光纤相连，以满足用户跨AZ构建高可用性系统的需求。

- 项目

  区域默认对应一个项目，这个项目由系统预置，用来隔离物理区域间的资源（计算资源、存储资源和网络资源），以默认项目为单位进行授权，用户可以访问您账号中该区域的所有资源。如果您希望进行更加精细的权限控制，可以在区域默认的项目中创建子项目，并在子项目中创建资源，然后以子项目为单位进行授权，使得仅能访问特定子项目中资源，使得资源的权限控制更加精确。

  图 1-1 项目隔离模型



2 API 概览

通过使用Anti-DDoS提供的接口，您可以完整的使用Anti-DDoS的所有功能。

类型	说明
Anti-DDoS API接口	Anti-DDoS接口，包括查询、更新Anti-DDoS服务等接口。
告警提醒API接口	告警提醒API接口，包括查询告警配置信息接口。

3 如何调用 API

3.1 构造请求

本节介绍如何构造REST API的请求，并以调用IAM服务的**获取用户Token**说明如何调用API，该API获取用户的Token，Token可以用于调用其他API时鉴权。

您还可以通过这个视频教程了解如何构造请求调用API：**https://bbs.huaweicloud.com/videos/102987**。

请求 URI

请求URI由如下部分组成。

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

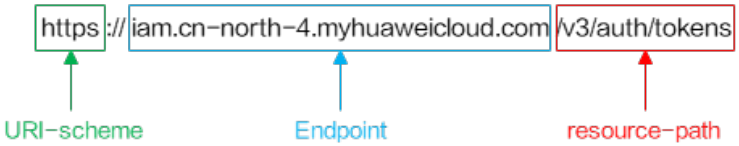
尽管请求URI包含在请求消息头中，但大多数语言或框架都要求您从请求消息中单独传递它，所以在此单独强调。

- **URI-scheme**：
  表示用于传输请求的协议，当前所有API均采用**HTTPS**协议。
- **Endpoint**：
  指定承载REST服务端点的服务器域名或IP，不同服务不同区域的Endpoint不同，您可以从**地区和终端节点**获取。
  例如IAM服务在“华北-北京四”区域的Endpoint为“iam.cn-north-4.myhuaweicloud.com”。
- **resource-path**：
  资源路径，也即API访问路径。从具体API的URI模块获取，例如“获取用户Token”API的resource-path为“/v3/auth/tokens”。
- **query-string**：
  查询参数，是可选部分，并不是每个API都有查询参数。查询参数前面需要带一个“？”，形式为“参数名=参数取值”，例如“limit=10”，表示查询不超过10条数据。

例如您需要获取IAM在“华北-北京四”区域的Token，则需使用“华北-北京四”区域的Endpoint（iam.cn-north-4.myhuaweicloud.com），并在**获取用户Token**的URI部分找到resource-path（/v3/auth/tokens），拼接起来如下所示。

```
https://iam.cn-north-4.myhuaweicloud.com/v3/auth/tokens
```

图 3-1 URI 示意图



说明

为查看方便，在每个具体API的URI部分，只给出resource-path部分，并将请求方法写在一起。这是因为URI-scheme都是HTTPS，同一个服务的Endpoint在同一个区域也相同，所以简洁起见将这两部分省略。

请求方法

HTTP请求方法（也称为操作或动词），它告诉服务你正在请求什么类型的操作。

- **GET**：请求服务器返回指定资源。
- **PUT**：请求服务器更新指定资源。
- **POST**：请求服务器新增资源或执行特殊操作。
- **DELETE**：请求服务器删除指定资源，如删除对象等。
- **HEAD**：请求服务器资源头部。
- **PATCH**：请求服务器更新资源的部分内容。当资源不存在的时候，PATCH可能会去创建一个新的资源。

在**获取用户Token**的URI部分，您可以看到其请求方法为“POST”，则其请求为：

```
POST https://iam.cn-north-4.myhuaweicloud.com/v3/auth/tokens
```

请求消息头

附加请求头字段，如指定的URI和HTTP方法所要求的字段。例如定义消息体类型的请求头“Content-Type”，请求鉴权信息等。

如下公共消息头需要添加到请求中。

- **Content-Type**：消息体的类型（格式），必选，默认取值为“application/json”，有其他取值时会在具体接口中专门说明。
- **X-Auth-Token**：用户Token，可选，当使用Token方式认证时，必须填充该字段。用户Token也就是调用**获取用户Token**接口的响应值，该接口是唯一不需要认证的接口。

  说明

  API同时支持使用AK/SK认证，AK/SK认证是使用SDK对请求进行签名，签名过程会自动往请求中添加Authorization（签名认证信息）和X-Sdk-Date（请求发送的时间）请求头。

  AK/SK认证的详细说明请参见**AK/SK认证**。

对于**获取用户Token**接口，由于不需要认证，所以只添加“Content-Type”即可，添加消息头后的请求如下所示。


```
POST https://iam.cn-north-4.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

请求消息体

请求消息体通常以结构化格式发出，与请求消息头中Content-type对应，传递除请求消息头之外的内容。若请求消息体中参数支持中文，则中文字符必须为UTF-8编码。

每个接口的请求消息体内容不同，也并不是每个接口都需要有请求消息体（或者说消息体为空），GET、DELETE操作类型的接口就不需要消息体，消息体具体内容需要根据具体接口而定。

对于**获取用户Token**接口，您可以从接口的请求部分看到所需的请求参数及参数说明。将消息体加入后的请求如下所示，加粗的斜体字段需要根据实际值填写，其中***username***为用户名，***domainname***为用户所属的账号名称，*********为用户登录密码，***xxxxxxxxxxxxxxxxxx***为project的名称，如“cn-north-4”，您可以从**地区和终端节点**获取，对应地区和终端节点页面的“区域”字段的值。

说明

scope参数定义了Token的作用域，下面示例中获取的Token仅能访问project下的资源。您还可以设置Token作用域为某个账号下所有资源或账号的某个project下的资源，详细定义请参见**获取用户Token**。

```
POST https://iam.cn-north-4.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "username",
                    "password": "********",
                    "domain": {
                        "name": "domainname"
                    }
                }
            }
        },
        "scope": {
            "project": {
                "name": "xxxxxxxxxxxxxxxxxx"
            }
        }
    }
}
```

到这里为止这个请求需要的内容就具备齐全了，您可以使用**curl**、**Postman**或直接编写代码等方式发送请求调用API。对于获取用户Token接口，返回的响应消息头中“x-subject-token”就是需要获取的用户Token。有了Token之后，您就可以使用Token认证调用其他API。

3.2 认证鉴权

调用接口有如下两种认证方式，您可以选择其中一种进行认证鉴权。

- Token认证：通过Token认证调用请求。
- AK/SK认证：通过AK（Access Key ID）/SK（Secret Access Key）加密调用请求。推荐使用AK/SK认证，其安全性比Token认证要高。

Token 认证

> **说明**
>
> Token的有效期为24小时，需要使用一个Token鉴权时，可以先缓存起来，避免频繁调用。

Token在计算机系统中代表令牌（临时）的意思，拥有Token就代表拥有某种权限。Token认证就是在调用API的时候将Token加到请求消息头，从而通过身份认证，获得操作API的权限。

Token可通过调用**获取用户Token**接口获取，调用本服务API需要project级别的Token，即调用**获取用户Token**接口时，请求body中auth.scope的取值需要选择project，如下所示。

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

获取Token后，再调用其他接口时，您需要在请求消息头中添加“X-Auth-Token”，其值即为Token。例如Token值为“ABCDEFJ....”，则调用接口时将“X-Auth-Token: ABCDEFJ....”加到请求消息头即可，如下所示。

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

您还可以通过这个视频教程了解如何使用Token认证：**https://bbs.huaweicloud.com/videos/101333**。

AK/SK 认证

> **说明**
>
> AK/SK签名认证方式仅支持消息体大小12MB以内，12MB以上的请求请使用Token认证。

AK/SK认证就是使用AK/SK对请求进行签名，在请求时将签名信息添加到消息头，从而通过身份认证。

- AK(Access Key ID)：访问密钥ID。与私有访问密钥关联的唯一标识符；访问密钥ID和私有访问密钥一起使用，对请求进行加密签名。
- SK(Secret Access Key)：与访问密钥ID结合使用的密钥，对请求进行加密签名，可标识发送方，并防止请求被修改。

使用AK/SK认证时，您可以基于签名算法使用AK/SK对请求进行签名，也可以使用专门的签名SDK对请求进行签名。详细的签名方法和SDK使用方法请参见**API签名指南**。

须知

签名SDK只提供签名功能，与服务提供的SDK不同，使用时请注意。

3.3 返回结果

状态码

请求发送以后，您会收到响应，包含状态码、响应消息头和消息体。

状态码是一组从1xx到5xx的数字代码，状态码表示了请求响应的状态，完整的状态码列表请参见**DDoS原生基础防护状态码**。

对于**获取用户Token**接口，如果调用后返回状态码为“201”，则表示请求成功。

响应消息头

对应请求消息头，响应同样也有消息头，如“Content-type”。

对于**获取用户Token**接口，返回如**图3-2**所示的消息头，其中“x-subject-token”就是需要获取的用户Token。有了Token之后，您就可以使用Token认证调用其他API。

图 3-2 获取用户 Token 响应消息头

响应消息体（可选）

响应消息体通常以结构化格式返回，与响应消息头中Content-type对应，传递除响应消息头之外的内容。

对于**获取用户Token**接口，返回如下消息体。为篇幅起见，这里只展示部分内容。

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

当接口调用出错时，会返回错误码及错误信息说明，错误响应的Body体格式如下所示。

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

其中，error_code表示错误码，error_msg表示错误描述信息。

4 DDoS 原生基础防护 API

4.1 DDoS 任务管理

4.1.1 查询 Anti-DDoS 任务

功能介绍

用户查询指定的Anti-DDoS防护配置任务，得到任务当前执行的状态。

调用方法

请参见**如何调用API**。

URI

GET /v2/{project_id}/query-task-status

表 4-1 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id 最小长度：**32** 最大长度：**64**

表 4-2 Query 参数

参数	是否必选	参数类型	描述
task_id	是	String	任务ID（非负整数）的字符串 最小长度：**32** 最大长度：**64**

请求参数

表 4-3 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type请求头 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

响应参数

状态码： 200

表 4-4 响应 Body 参数

参数	参数类型	描述
task_status	String	任务状态，可选范围： ● success: 表示成功 ● failed: 表示失败 ● waiting: 表示等待 ● running: 表示运行中 ● preprocess: 表示预处理 ● ready: 表示准备
task_msg	String	任务的附加信息 最小长度：**0** 最大长度：**255**

请求示例

无

响应示例

状态码： 200

请求已成功

```
{
  "task_status" : "success",
  "task_msg" : ""
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.antiddos.v1.region.AntiDDoSRegion;
import com.huaweicloud.sdk.antiddos.v1.*;
import com.huaweicloud.sdk.antiddos.v1.model.*;


public class ShowNewTaskStatusSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        AntiDDoSClient client = AntiDDoSClient.newBuilder()
                .withCredential(auth)
                .withRegion(AntiDDoSRegion.valueOf("cn-north-4"))
                .build();
        ShowNewTaskStatusRequest request = new ShowNewTaskStatusRequest();
        request.withTaskId("<task_id>");
        try {
            ShowNewTaskStatusResponse response = client.showNewTaskStatus(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkantiddos.v1.region.antiddos_region import AntiDDoSRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkantiddos.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = AntiDDoSClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AntiDDoSRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ShowNewTaskStatusRequest()
        request.task_id = "<task_id>"
        response = client.show_new_task_status(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    antiddos "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := antiddos.NewAntiDDoSClient(
        antiddos.AntiDDoSClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ShowNewTaskStatusRequest{}
```

```
    request.TaskId = "<task_id>"
    response, err := client.ShowNewTaskStatus(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	请求已成功

错误码

请参见**错误码**。

4.2 DDoS 防护管理

4.2.1 查询 EIP 防护状态列表

功能介绍

查询用户所有EIP的Anti-DDoS防护状态信息，用户的EIP无论是否绑定到云服务器，都可以进行查询。

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/antiddos

表 4-5 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id 最小长度：**32** 最大长度：**64**

表 4-6 Query 参数

参数	是否必选	参数类型	描述
status	否	String	可选范围： ● normal：表示正常 ● configging：表示设置中 ● notConfig：表示未设置 ● packetcleaning：表示清洗 ● packetdropping：表示黑洞 不带此参数默认所有列表，以neutron查询到的顺序为准。
limit	否	String	返回结果个数限制，取值范围：1~100 最小长度：**1** 最大长度：**3**
offset	否	String	偏移量，取值范围：0~2147483647 最小长度：**1** 最大长度：**10**
ip	否	String	IP地址，支持IPv4格式和IPv6格式输入，支持部分查询。例如“？ip=192.168”，会返回192.168.111.1和10.192.168.8所对应的EIP防护状态。

请求参数

表 4-7 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type请求头 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

响应参数

状态码： 200

表 4-8 响应 Body 参数

参数	参数类型	描述
total	Integer	弹性IP总数 最小值：**0** 最大值：**2147483647**
ddosStatus	Array of **DDosStatus** objects	防护状态列表

表 4-9 DDosStatus

参数	参数类型	描述
floating_ip_id	String	EIP的ID
floating_ip_address	String	浮动IP地址
network_type	String	EIP所属类型，可选范围： • EIP：未绑定到ECS的EIP或绑定到ECS的EIP • ELB：绑定到ELB的EIP
status	String	防护状态，可选范围： • normal：表示正常 • configging：表示设置中 • notConfig：表示未设置 • packetcleaning：表示清洗 • packetdropping：表示黑洞
blackhole_endtime	Long	黑洞结束时间
protect_type	String	防护类型
traffic_threshold	Long	流量阈值
http_threshold	Long	http流量阈值

请求示例

无

响应示例

状态码： 200

请求已成功

```
{
  "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
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.antiddos.v1.region.AntiDDoSRegion;
import com.huaweicloud.sdk.antiddos.v1.*;
import com.huaweicloud.sdk.antiddos.v1.model.*;


public class ListDDosStatusSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        AntiDDoSClient client = AntiDDoSClient.newBuilder()
                .withCredential(auth)
                .withRegion(AntiDDoSRegion.valueOf("cn-north-4"))
                .build();
        ListDDosStatusRequest request = new ListDDosStatusRequest();
        request.withStatus("<status>");
        request.withLimit("<limit>");
        request.withOffset("<offset>");
        request.withIp("<ip>");
        try {
            ListDDosStatusResponse response = client.listDDosStatus(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
```

```
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkantiddos.v1.region.antiddos_region import AntiDDoSRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkantiddos.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = AntiDDoSClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AntiDDoSRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ListDDosStatusRequest()
        request.status = "<status>"
        request.limit = "<limit>"
        request.offset = "<offset>"
        request.ip = "<ip>"
        response = client.list_d_dos_status(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    antiddos "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := antiddos.NewAntiDDoSClient(
        antiddos.AntiDDoSClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ListDDosStatusRequest{}
    statusRequest:= "<status>"
    request.Status = &statusRequest
    limitRequest:= "<limit>"
    request.Limit = &limitRequest
    offsetRequest:= "<offset>"
    request.Offset = &offsetRequest
    ipRequest:= "<ip>"
    request.Ip = &ipRequest
    response, err := client.ListDDosStatus(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	请求已成功

错误码

请参见**错误码**。

4.2.2 查询 Anti-DDoS 配置可选范围

功能介绍

查询系统支持的Anti-DDoS防护策略配置的可选范围，用户根据范围列表选择适合自己业务的防护策略进行Anti-DDoS流量清洗。

调用方法

请参见**如何调用API**。

URI

GET /v2/{project_id}/antiddos/query-config-list

表 4-10 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id 最小长度：**32** 最大长度：**64**

请求参数

表 4-11 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type请求头 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

响应参数

状态码： 200

表 4-12 响应 Body 参数

参数	参数类型	描述
traffic_limited_list	Array of **TriggerBpsDict** objects	流量限制列表
http_limited_list	Array of **TriggerQpsDict** objects	HTTP限制列表

参数	参数类型	描述
connection_limited_list	Array of **CleanLimitDict** objects	连接数限制列表
extend_ddos_config	Array of **ExtendDDoSSet** objects	扩展配置列表

表 4-13 TriggerBpsDict

参数	参数类型	描述
traffic_pos_id	Long	流量分段ID
traffic_per_second	Long	每秒流量（Mbit/s）阈值
packet_per_second	Long	每秒报文数（个/s）阈值

表 4-14 TriggerQpsDict

参数	参数类型	描述
http_request_pos_id	Long	HTTP请求数分段ID
http_packet_per_second	Long	每秒HTTP请求数（个/s）阈值

表 4-15 CleanLimitDict

参数	参数类型	描述
cleaning_access_pos_id	Long	清洗时访问限制分段ID
new_connection_limited	Long	单一源IP新建连接个数
total_connection_limited	Long	单一源IP连接数总个数

表 4-16 ExtendDDoSSet

参数	参数类型	描述
SetID	Long	配置分段ID
new_connection_limited	Long	单一源IP新建连接个数
total_connection_limited	Long	单一源IP连接数总个数
http_packet_per_second	Long	每秒HTTP请求数（个/s）阈值
traffic_per_second	Long	每秒流量（Mbit/s）阈值
packet_per_second	Long	每秒报文数（个/s）阈值

请求示例

无

响应示例

状态码： 200

请求已成功

```
{
  "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
}, {
  "traffic_pos_id" : 88,
  "traffic_per_second" : 1000,
  "packet_per_second" : 300000
} ],
"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" : [ ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.antiddos.v1.region.AntiDDoSRegion;
import com.huaweicloud.sdk.antiddos.v1.*;
import com.huaweicloud.sdk.antiddos.v1.model.*;


public class ListNewConfigsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        AntiDDoSClient client = AntiDDoSClient.newBuilder()
                .withCredential(auth)
                .withRegion(AntiDDoSRegion.valueOf("cn-north-4"))
                .build();
        ListNewConfigsRequest request = new ListNewConfigsRequest();
        try {
            ListNewConfigsResponse response = client.listNewConfigs(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
```

```
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkantiddos.v1.region.antiddos_region import AntiDDoSRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkantiddos.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = AntiDDoSClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AntiDDoSRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ListNewConfigsRequest()
        response = client.list_new_configs(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    antiddos "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
```

```
        WithAk(ak).
        WithSk(sk).
        Build()

    client := antiddos.NewAntiDDoSClient(
        antiddos.AntiDDoSClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ListNewConfigsRequest{}
    response, err := client.ListNewConfigs(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	请求已成功

错误码

请参见**错误码**。

4.2.3 查询周防护统计情况

功能介绍

查询用户所有Anti-DDoS防护周统计情况，包括一周内DDoS拦截次数和攻击次数、以及按照被攻击次数进行的排名信息等统计数据。系统支持当前时间之前四周的周统计数据查询，超过这个时间的请求是查询不到统计数据的。

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/antiddos/weekly

表 4-17 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id 最小长度：**32** 最大长度：**64**

表 4-18 Query 参数

参数	是否必选	参数类型	描述
period_start_date	否	String	每周的起始时间 最小长度：**13** 最大长度：**13**

请求参数

表 4-19 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type请求头 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

响应参数

状态码： 200

表 4-20 响应 Body 参数

参数	参数类型	描述
ddos_intercept_times	Integer	一周内DDoS拦截次数 最小值：**0** 最大值：**2147483647**
weekdata	Array of **WeeklyCount** objects	一周的攻击次数统计数据 数组长度：**0 - 2147483647**
top10	Array of **WeeklyTop10** objects	被攻击次数排名前10的IP地址 数组长度：**0 - 2147483647**

表 4-21 WeeklyCount

参数	参数类型	描述
ddos_intercept_times	Integer	DDoS拦截次数 最小值：**0** 最大值：**2147483647**
ddos_blackhole_times	Integer	DDoS黑洞次数 最小值：**0** 最大值：**2147483647**
max_attack_bps	Integer	最大攻击流量 最小值：**0** 最大值：**2147483647**
max_attack_conns	Integer	最大攻击连接数 最小值：**0** 最大值：**2147483647**
period_start_date	Long	开始时间 最小值：**0** 最大值：**999999999**

表 4-22 WeeklyTop10

参数	参数类型	描述
floating_ip_address	String	弹性IP地址 最小长度：**7** 最大长度：**128**

参数	参数类型	描述
times	Integer	DDoS拦截次数，包括清洗和黑洞 最小值：**0** 最大值：**2147483647**

请求示例

无

响应示例

状态码： 200

请求已成功


```
{
  "ddos_intercept_times" : 0,
  "weekdata" : [ {
    "ddos_intercept_times" : 0,
    "ddos_blackhole_times" : 0,
    "max_attack_bps" : 0,
    "max_attack_conns" : 0,
    "period_start_date" : 1605496722606
  }, {
    "ddos_intercept_times" : 0,
    "ddos_blackhole_times" : 0,
    "max_attack_bps" : 0,
    "max_attack_conns" : 0,
    "period_start_date" : 1605583122606
  }, {
    "ddos_intercept_times" : 0,
    "ddos_blackhole_times" : 0,
    "max_attack_bps" : 0,
    "max_attack_conns" : 0,
    "period_start_date" : 1605669522606
  }, {
    "ddos_intercept_times" : 0,
    "ddos_blackhole_times" : 0,
    "max_attack_bps" : 0,
    "max_attack_conns" : 0,
    "period_start_date" : 1605755922606
  }, {
    "ddos_intercept_times" : 0,
    "ddos_blackhole_times" : 0,
    "max_attack_bps" : 0,
    "max_attack_conns" : 0,
    "period_start_date" : 1605842322606
  }, {
    "ddos_intercept_times" : 0,
    "ddos_blackhole_times" : 0,
    "max_attack_bps" : 0,
    "max_attack_conns" : 0,
    "period_start_date" : 1605928722606
  }, {
    "ddos_intercept_times" : 0,
    "ddos_blackhole_times" : 0,
    "max_attack_bps" : 0,
    "max_attack_conns" : 0,
    "period_start_date" : 1606015122606
  } ],
  "top10" : [ ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.antiddos.v1.region.AntiDDoSRegion;
import com.huaweicloud.sdk.antiddos.v1.*;
import com.huaweicloud.sdk.antiddos.v1.model.*;


public class ListWeeklyReportsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        AntiDDoSClient client = AntiDDoSClient.newBuilder()
                .withCredential(auth)
                .withRegion(AntiDDoSRegion.valueOf("cn-north-4"))
                .build();
        ListWeeklyReportsRequest request = new ListWeeklyReportsRequest();
        request.withPeriodStartDate("<period_start_date>");
        try {
            ListWeeklyReportsResponse response = client.listWeeklyReports(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkantiddos.v1.region.antiddos_region import AntiDDoSRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkantiddos.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```

```
variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = AntiDDoSClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AntiDDoSRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ListWeeklyReportsRequest()
        request.period_start_date = "<period_start_date>"
        response = client.list_weekly_reports(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    antiddos "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := antiddos.NewAntiDDoSClient(
        antiddos.AntiDDoSClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ListWeeklyReportsRequest{}
    periodStartDateRequest:= "<period_start_date>"
    request.PeriodStartDate = &periodStartDateRequest
    response, err := client.ListWeeklyReports(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	请求已成功

错误码

请参见**错误码**。

4.2.4 查询 Anti-DDoS 服务

功能介绍

查询配置的Anti-DDoS防护策略，用户可以查询指定EIP的Anti-DDoS防护策略。

调用方法

请参见**如何调用API**。

URI

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

表 4-23 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id 最小长度：**32** 最大长度：**64**
floating_ip_id	是	String	用户EIP对应的ID 最小长度：**32** 最大长度：**64**

表 4-24 Query 参数

参数	是否必选	参数类型	描述
ip	否	String	用户EIP 最小长度：**7** 最大长度：**128**

请求参数

表 4-25 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type请求头 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

响应参数

状态码： 200

表 4-26 响应 Body 参数

参数	参数类型	描述
enable_L7	Boolean	是否开启L7层防护，固定值为fasle
traffic_pos_id	Long	流量分段ID，取值范围：1～9、88、99 最小值：**1** 最大值：**99**
http_request_pos_id	Long	HTTP请求数分段ID，取值范围：1～15 最小值：**1** 最大值：**15**

参数	参数类型	描述
cleaning_access_pos_id	Long	清洗时访问限制分段ID，取值范围：1～8、88、99 最小值：**1** 最大值：**99**
app_type_id	Integer	应用类型ID，可选取值： • 0 • 1

请求示例

无

响应示例

状态码： 200

请求已成功

```
{
  "enable_L7" : false,
  "traffic_pos_id" : 8,
  "http_request_pos_id" : 8,
  "cleaning_access_pos_id" : 8,
  "app_type_id" : 1
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.antiddos.v1.region.AntiDDoSRegion;
import com.huaweicloud.sdk.antiddos.v1.*;
import com.huaweicloud.sdk.antiddos.v1.model.*;


public class ShowDDosSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
```

```
        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        AntiDDoSClient client = AntiDDoSClient.newBuilder()
                .withCredential(auth)
                .withRegion(AntiDDoSRegion.valueOf("cn-north-4"))
                .build();
        ShowDDosRequest request = new ShowDDosRequest();
        request.withIp("<ip>");
        try {
            ShowDDosResponse response = client.showDDos(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkantiddos.v1.region.antiddos_region import AntiDDoSRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkantiddos.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = AntiDDoSClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AntiDDoSRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ShowDDosRequest()
        request.ip = "<ip>"
        response = client.show_d_dos(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
```

```
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    antiddos "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := antiddos.NewAntiDDoSClient(
        antiddos.AntiDDoSClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ShowDDosRequest{}
    ipRequest:= "<ip>"
    request.Ip = &ipRequest
    response, err := client.ShowDDos(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	请求已成功

错误码

请参见**错误码**。

4.2.5 更新 Anti-DDoS 服务

功能介绍

更新指定EIP的Anti-DDoS防护策略配置。调用成功，只是说明服务节点收到了关闭更新配置请求，操作是否成功需要通过任务查询接口查询该任务的执行状态，具体请参考查询Anti-DDoS任务。

调用方法

请参见**如何调用API**。

URI

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

表 4-27 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id 最小长度：**32** 最大长度：**64**
floating_ip_id	是	String	用户EIP对应的ID

表 4-28 Query 参数

参数	是否必选	参数类型	描述
ip	否	String	ip

请求参数

表 4-29 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type请求头 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

表 4-30 请求 Body 参数

参数	是否必选	参数类型	描述
app_type_id	是	Integer	应用类型ID，可选取值： • 0 • 1
cleaning_access_pos_id	是	Integer	清洗时访问限制分段ID，取值范围：1～8、88、99 最小值：**1** 最大值：**99**
enable_L7	是	Boolean	是否开启L7层防护，固定值为fasle
http_request_pos_id	是	Integer	HTTP请求数分段ID，取值范围：1～15 最小值：**1** 最大值：**15**
traffic_pos_id	是	Integer	流量分段ID，取值范围：1～9、88、99 最小值：**1** 最大值：**99**

响应参数

状态码： 200

表 4-31 响应 Body 参数

参数	参数类型	描述
error_code	String	内部错误码 最小长度：**1** 最大长度：**20**
error_msg	String	内部错误描述 最小长度：**1** 最大长度：**255**
task_id	String	任务ID，后续可根据该ID查询本任务状态。 本字段为后续的任务审计扩展，暂时不需要，先保留。

请求示例

更新指定EIP的Anti-DDoS防护策略，清洗时访问限制分段ID设置为8，流量分段ID设置为1。

```
PUT https://{endpoint}/v1/{project_id}/antiddos/{floating_ip_id}

{
  "app_type_id" : 0,
  "cleaning_access_pos_id" : 8,
  "enable_L7" : false,
  "http_request_pos_id" : 1,
  "traffic_pos_id" : 1
}
```

响应示例

状态码： 200

请求已成功

```
{
  "error_code" : "10000000",
  "error_msg" : "The task has been received and is being handled",
  "task_id" : "59385d2a-6266-4d3a-9122-a228c530f557"
}
```

SDK 代码示例

SDK代码示例如下。

Java

更新指定EIP的Anti-DDoS防护策略，清洗时访问限制分段ID设置为8，流量分段ID设置为1。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.antiddos.v1.region.AntiDDoSRegion;
import com.huaweicloud.sdk.antiddos.v1.*;
import com.huaweicloud.sdk.antiddos.v1.model.*;


public class UpdateDDosSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        AntiDDoSClient client = AntiDDoSClient.newBuilder()
                .withCredential(auth)
```

```
            .withRegion(AntiDDoSRegion.valueOf("cn-north-4"))
            .build();
        UpdateDDosRequest request = new UpdateDDosRequest();
        request.withIp("<ip>");
        UpdateAntiDDosServiceRequestBody body = new UpdateAntiDDosServiceRequestBody();
        body.withTrafficPosId(1);
        body.withHttpRequestPosId(1);
        body.withEnableL7(false);
        body.withCleaningAccessPosId(8);
        body.withAppTypeId(0);
        request.withBody(body);
        try {
            UpdateDDosResponse response = client.updateDDos(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

更新指定EIP的Anti-DDoS防护策略，清洗时访问限制分段ID设置为8，流量分段ID设置为1。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkantiddos.v1.region.antiddos_region import AntiDDoSRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkantiddos.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = AntiDDoSClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AntiDDoSRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = UpdateDDosRequest()
        request.ip = "<ip>"
        request.body = UpdateAntiDDosServiceRequestBody(
            traffic_pos_id=1,
            http_request_pos_id=1,
            enable_l7=False,
            cleaning_access_pos_id=8,
            app_type_id=0
        )
        response = client.update_d_dos(request)
        print(response)
    except exceptions.ClientRequestException as e:
```

```
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

更新指定EIP的Anti-DDoS防护策略，清洗时访问限制分段ID设置为8，流量分段ID设置为1。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    antiddos "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := antiddos.NewAntiDDoSClient(
        antiddos.AntiDDoSClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.UpdateDDosRequest{}
    ipRequest:= "<ip>"
    request.Ip = &ipRequest
    request.Body = &model.UpdateAntiDDosServiceRequestBody{
        TrafficPosId: int32(1),
        HttpRequestPosId: int32(1),
        EnableL7: false,
        CleaningAccessPosId: int32(8),
        AppTypeId: int32(0),
    }
    response, err := client.UpdateDDos(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	请求已成功

错误码

请参见**错误码**。

4.2.6 查询指定 EIP 防护流量

功能介绍

查询指定EIP在过去24小时之内的防护流量信息，流量的间隔时间单位为5分钟。

调用方法

请参见**如何调用API**。

URI

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

表 4-32 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id 最小长度：**32** 最大长度：**64**
floating_ip_id	是	String	用户EIP对应的ID 最小长度：**32** 最大长度：**64**

表 4-33 Query 参数

参数	是否必选	参数类型	描述
ip	否	String	用户EIP 最小长度：**7** 最大长度：**128**

请求参数

表 4-34 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type请求头 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

响应参数

状态码： 200

表 4-35 响应 Body 参数

参数	参数类型	描述
data	Array of **DailyData** objects	24小时内的流量数据 数组长度：**288 - 288**

表 4-36 DailyData

参数	参数类型	描述
period_start	Long	开始时间 最小值：**1** 最大值：**9999999999999**
bps_in	Integer	入流量（bit/s） 最小值：**0** 最大值：**2147483647**
bps_attack	Long	攻击流量（bit/s） 最小值：**0** 最大值：**2147483647**

参数	参数类型	描述
total_bps	Long	总流量 最小值：**0** 最大值：**2147483647**
pps_in	Long	入报文速率（个/s） 最小值：**0** 最大值：**2147483647**
pps_attack	Long	攻击文速率（个/s） 最小值：**0** 最大值：**2147483647**
total_pps	Long	总报文速率 最小值：**0** 最大值：**2147483647**

请求示例

无

响应示例

状态码： 200

请求已成功

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

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.antiddos.v1.region.AntiDDoSRegion;
import com.huaweicloud.sdk.antiddos.v1.*;
```

```
import com.huaweicloud.sdk.antiddos.v1.model.*;


public class ListDailyReportSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        AntiDDoSClient client = AntiDDoSClient.newBuilder()
                .withCredential(auth)
                .withRegion(AntiDDoSRegion.valueOf("cn-north-4"))
                .build();
        ListDailyReportRequest request = new ListDailyReportRequest();
        request.withIp("<ip>");
        try {
            ListDailyReportResponse response = client.listDailyReport(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkantiddos.v1.region.antiddos_region import AntiDDoSRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkantiddos.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = AntiDDoSClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AntiDDoSRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ListDailyReportRequest()
        request.ip = "<ip>"
```

```
        response = client.list_daily_report(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    antiddos "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := antiddos.NewAntiDDoSClient(
        antiddos.AntiDDoSClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ListDailyReportRequest{}
    ipRequest:= "<ip>"
    request.Ip = &ipRequest
    response, err := client.ListDailyReport(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	请求已成功

错误码

请参见**错误码**。

4.2.7 查询指定 EIP 异常事件

功能介绍

查询指定EIP在过去24小时之内的异常事件信息，异常事件包括清洗事件和黑洞事件，查询延迟在5分钟之内。

调用方法

请参见**如何调用API**。

URI

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

表 4-37 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id 最小长度：**32** 最大长度：**64**
floating_ip_id	是	String	用户EIP对应的ID 最小长度：**32** 最大长度：**64**

表 4-38 Query 参数

参数	是否必选	参数类型	描述
sort_dir	否	String	可选范围： • desc：表示时间降序 • asc：表示时间升序 默认值为“desc”。
limit	否	String	返回结果个数限制，此次查询返回数量最大值，取值范围：1~100，与offset配合使用。 若“limit”与“offset”均不携带则返回所有主机列表。 最小长度：**1** 最大长度：**3**

参数	是否必选	参数类型	描述
offset	否	String	偏移量，“limit”携带时此字段有效。 最小长度：**1** 最大长度：**10**
ip	否	String	用户EIP 最小长度：**7** 最大长度：**128**

请求参数

表 4-39 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type请求头 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

响应参数

状态码： 200

表 4-40 响应 Body 参数

参数	参数类型	描述
total	Long	弹性IP总数 最小值：**0** 最大值：**2147483647**
logs	Array of **DailyLog** objects	异常事件列表 数组长度：**0 - 2147483647**

表 4-41 DailyLog

参数	参数类型	描述
start_time	Long	开始时间 最小值：**1** 最大值：**9999999999999**
end_time	Long	结束时间 最小值：**1** 最大值：**9999999999999**
status	Integer	防护状态，可选范围： ● 1：表示清洗 ● 2：表示黑洞
trigger_bps	Integer	触发时流量 最小值：**0** 最大值：**2147483647**
trigger_pps	Integer	触发时报文速率 最小值：**0** 最大值：**2147483647**
trigger_http_pps	Integer	触发时HTTP请求速率 最小值：**0** 最大值：**2147483647**

请求示例

无

响应示例

状态码： 200

请求已成功

```
{
  "total" : 0,
  "logs" : [ ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
```

```
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.antiddos.v1.region.AntiDDoSRegion;
import com.huaweicloud.sdk.antiddos.v1.*;
import com.huaweicloud.sdk.antiddos.v1.model.*;


public class ListDailyLogSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        AntiDDoSClient client = AntiDDoSClient.newBuilder()
                .withCredential(auth)
                .withRegion(AntiDDoSRegion.valueOf("cn-north-4"))
                .build();
        ListDailyLogRequest request = new ListDailyLogRequest();
        request.withSortDir("<sort_dir>");
        request.withLimit("<limit>");
        request.withOffset("<offset>");
        request.withIp("<ip>");
        try {
            ListDailyLogResponse response = client.listDailyLog(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkantiddos.v1.region.antiddos_region import AntiDDoSRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkantiddos.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \
```

```
    client = AntiDDoSClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AntiDDoSRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ListDailyLogRequest()
        request.sort_dir = "<sort_dir>"
        request.limit = "<limit>"
        request.offset = "<offset>"
        request.ip = "<ip>"
        response = client.list_daily_log(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    antiddos "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := antiddos.NewAntiDDoSClient(
        antiddos.AntiDDoSClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ListDailyLogRequest{}
    sortDirRequest:= "<sort_dir>"
    request.SortDir = &sortDirRequest
    limitRequest:= "<limit>"
    request.Limit = &limitRequest
    offsetRequest:= "<offset>"
    request.Offset = &offsetRequest
    ipRequest:= "<ip>"
    request.Ip = &ipRequest
    response, err := client.ListDailyLog(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	请求已成功

错误码

请参见**错误码**。

4.2.8 查询指定 EIP 防护状态

功能介绍

查询指定EIP的Anti-DDoS防护状态。

调用方法

请参见**如何调用API**。

URI

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

表 4-42 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id 最小长度：**32** 最大长度：**64**
floating_ip_id	是	String	用户EIP对应的ID 最小长度：**32** 最大长度：**64**

表 4-43 Query 参数

参数	是否必选	参数类型	描述
ip	否	String	用户EIP 最小长度：**7** 最大长度：**128**

请求参数

表 4-44 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type请求头 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

响应参数

状态码： 200

表 4-45 响应 Body 参数

参数	参数类型	描述
status	String	防护状态，可选范围： ● normal：表示正常 ● configging：表示设置中 ● notConfig：表示未设置 ● packetcleaning：表示清洗 ● packetdropping：表示黑洞

请求示例

无

响应示例

状态码： 200

请求已成功

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

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.antiddos.v1.region.AntiDDoSRegion;
import com.huaweicloud.sdk.antiddos.v1.*;
import com.huaweicloud.sdk.antiddos.v1.model.*;


public class ShowDDosStatusSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        AntiDDoSClient client = AntiDDoSClient.newBuilder()
                .withCredential(auth)
                .withRegion(AntiDDoSRegion.valueOf("cn-north-4"))
                .build();
        ShowDDosStatusRequest request = new ShowDDosStatusRequest();
        request.withIp("<ip>");
        try {
            ShowDDosStatusResponse response = client.showDDosStatus(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkantiddos.v1.region.antiddos_region import AntiDDoSRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkantiddos.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = AntiDDoSClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AntiDDoSRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ShowDDosStatusRequest()
        request.ip = "<ip>"
        response = client.show_d_dos_status(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    antiddos "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := antiddos.NewAntiDDoSClient(
        antiddos.AntiDDoSClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ShowDDosStatusRequest{}
```

```
    ipRequest:= "<ip>"
    request.Ip = &ipRequest
    response, err := client.ShowDDosStatus(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	请求已成功

错误码

请参见**错误码**。

4.3 告警配置管理

4.3.1 查询告警配置信息

功能介绍

查询用户配置信息，用户可以通过此接口查询是否接收某类告警，同时可以配置是手机短信还是电子邮件接收告警信息。

调用方法

请参见**如何调用API**。

URI

GET /v2/{project_id}/warnalert/alertconfig/query

表 4-46 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id 最小长度：**32** 最大长度：**64**

请求参数

表 4-47 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type请求头 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

响应参数

状态码： 200

表 4-48 响应 Body 参数

参数	参数类型	描述
topic_urn	String	告警群组的唯一标识 最小长度：**1** 最大长度：**255**
display_name	String	告警群组描述 最小长度：**1** 最大长度：**255**
warn_config	**warn_config** object	告警配置信息

表 4-49 warn_config

参数	参数类型	描述
antiDDoS	Boolean	DDoS攻击
back_doors	Boolean	网页后门
bruce_force	Boolean	暴力破解（系统登录，FTP，DB）

参数	参数类型	描述
high_privilege	Boolean	数据库进程权限过高
remote_login	Boolean	异地登录提醒
send_frequency	Integer	取值范围： ● 0：表示每天一次 ● 1：表示半小时一次 对于HID必选。
waf	Boolean	保留字段
weak_password	Boolean	弱口令（系统，数据库）

请求示例

无

响应示例

状态码： 200

请求已成功

```
{
  "warn_config" : {
    "antiDDoS" : false,
    "bruce_force" : false,
    "remote_login" : false,
    "weak_password" : false,
    "high_privilege" : false,
    "back_doors" : false,
    "waf" : false,
    "send_frequency" : 0
  },
  "topic_urn" : null,
  "display_name" : null
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.antiddos.v1.region.AntiDDoSRegion;
import com.huaweicloud.sdk.antiddos.v1.*;
import com.huaweicloud.sdk.antiddos.v1.model.*;
```

```
public class ShowAlertConfigSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        AntiDDoSClient client = AntiDDoSClient.newBuilder()
                .withCredential(auth)
                .withRegion(AntiDDoSRegion.valueOf("cn-north-4"))
                .build();
        ShowAlertConfigRequest request = new ShowAlertConfigRequest();
        try {
            ShowAlertConfigResponse response = client.showAlertConfig(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkantiddos.v1.region.antiddos_region import AntiDDoSRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkantiddos.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = AntiDDoSClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AntiDDoSRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ShowAlertConfigRequest()
        response = client.show_alert_config(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```

```
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    antiddos "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := antiddos.NewAntiDDoSClient(
        antiddos.AntiDDoSClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ShowAlertConfigRequest{}
    response, err := client.ShowAlertConfig(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	请求已成功

错误码

请参见**错误码**。

4.3.2 更新告警配置信息

功能介绍

更新用户配置信息，用户可以通过此接口更新是否接收某类告警，同时可以配置是手机短信还是电子邮件接收告警信息。

调用方法

请参见**如何调用API**。

URI

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

表 4-50 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id 最小长度：**32** 最大长度：**64**

请求参数

表 4-51 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type请求头 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

表 4-52 请求 Body 参数

参数	是否必选	参数类型	描述
display_name	是	String	告警群组描述。 最小长度：**1** 最大长度：**255**
topic_urn	是	String	告警群组的唯一标识。 最小长度：**1** 最大长度：**255**
warn_config	是	**WarnConfig** object	告警配置信息。

表 4-53 WarnConfig

参数	是否必选	参数类型	描述
antiDDoS	是	Boolean	DDoS攻击
back_doors	否	Boolean	网页后门
bruce_force	否	Boolean	暴力破解（系统登录，FTP，DB）
high_privilege	否	Boolean	数据库进程权限过高
remote_login	否	Boolean	异地登录提醒
send_frequency	否	Integer	取值范围： • 0：表示每天一次 • 1：表示半小时一次 对于HID必选。
waf	否	Boolean	保留字段
weak_password	否	Boolean	弱口令（系统，数据库）

响应参数

状态码： 200

表 4-54 响应 Body 参数

参数	参数类型	描述
error_code	String	内部错误码 最小长度：**1** 最大长度：**20**
error_msg	String	内部错误描述 最小长度：**1** 最大长度：**255**

请求示例

设置消息通知的主题为“urn:smn:cn-north-7:2d2d90a56a3243bdb909f6a24a27be8d:cnad-test-intl”，并配置需要告警通知的攻击类型为DDoS攻击

```
POST https://{endpoint}/v2/{project_id}/warnalert/alertconfig/update

{
  "display_name" : "",
  "topic_urn" : "urn:smn:cn-north-7:2d2d90a56a3243bdb909f6a24a27be8d:cnad-test-intl",
  "warn_config" : {
    "antiDDoS" : true,
    "back_doors" : false,
    "bruce_force" : false,
    "high_privilege" : false,
    "remote_login" : false,
    "send_frequency" : 1,
    "waf" : false,
    "weak_password" : false
  }
}
```

响应示例

状态码： 200

请求已成功

```
{
  "error_code" : "10000000",
  "error_msg" : "Ok",
  "task_id" : ""
}
```

SDK 代码示例

SDK代码示例如下。

Java

设置消息通知的主题为“urn:smn:cn-north-7:2d2d90a56a3243bdb909f6a24a27be8d:cnad-test-intl”，并配置需要告警通知的攻击类型为DDoS攻击

```
package com.huaweicloud.sdk.test;
```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.antiddos.v1.region.AntiDDoSRegion;
import com.huaweicloud.sdk.antiddos.v1.*;
import com.huaweicloud.sdk.antiddos.v1.model.*;


public class UpdateAlertConfigSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        AntiDDoSClient client = AntiDDoSClient.newBuilder()
                .withCredential(auth)
                .withRegion(AntiDDoSRegion.valueOf("cn-north-4"))
                .build();
        UpdateAlertConfigRequest request = new UpdateAlertConfigRequest();
        UpdateAlertConfigRequestBody body = new UpdateAlertConfigRequestBody();
        UpdateAlertConfigRequestBodyWarnConfig warnConfigbody = new UpdateAlertConfigRequestBodyWarnConfig();
        warnConfigbody.withAntiDDoS(true)
            .withBackDoors(false)
            .withBruceForce(false)
            .withHighPrivilege(false)
            .withRemoteLogin(false)
            .withSendFrequency(1)
            .withWaf(false)
            .withWeakPassword(false);
        body.withWarnConfig(warnConfigbody);
        body.withTopicUrn("urn:smn:cn-north-7:2d2d90a56a3243bdb909f6a24a27be8d:cnad-test-intl");
        body.withDisplayName("");
        request.withBody(body);
        try {
            UpdateAlertConfigResponse response = client.updateAlertConfig(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

设置消息通知的主题为“urn:smn:cn-north-7:2d2d90a56a3243bdb909f6a24a27be8d:cnad-test-intl”，并配置需要告警通知的攻击类型为DDoS攻击

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkantiddos.v1.region.antiddos_region import AntiDDoSRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkantiddos.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = AntiDDoSClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AntiDDoSRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = UpdateAlertConfigRequest()
        warnConfigbody = UpdateAlertConfigRequestBodyWarnConfig(
            anti_d_do_s=True,
            back_doors=False,
            bruce_force=False,
            high_privilege=False,
            remote_login=False,
            send_frequency=1,
            waf=False,
            weak_password=False
        )
        request.body = UpdateAlertConfigRequestBody(
            warn_config=warnConfigbody,
            topic_urn="urn:smn:cn-north-7:2d2d90a56a3243bdb909f6a24a27be8d:cnad-test-intl",
            display_name=""
        )
        response = client.update_alert_config(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

设置消息通知的主题为“urn:smn:cn-north-7:2d2d90a56a3243bdb909f6a24a27be8d:cnad-test-intl”，并配置需要告警通知的攻击类型为DDoS攻击

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    antiddos "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
```

```
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := antiddos.NewAntiDDoSClient(
        antiddos.AntiDDoSClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.UpdateAlertConfigRequest{}
    backDoorsWarnConfig:= false
    bruceForceWarnConfig:= false
    highPrivilegeWarnConfig:= false
    remoteLoginWarnConfig:= false
    sendFrequencyWarnConfig:= int32(1)
    wafWarnConfig:= false
    weakPasswordWarnConfig:= false
    warnConfigbody := &model.UpdateAlertConfigRequestBodyWarnConfig{
        AntiDDoS: true,
        BackDoors: &backDoorsWarnConfig,
        BruceForce: &bruceForceWarnConfig,
        HighPrivilege: &highPrivilegeWarnConfig,
        RemoteLogin: &remoteLoginWarnConfig,
        SendFrequency: &sendFrequencyWarnConfig,
        Waf: &wafWarnConfig,
        WeakPassword: &weakPasswordWarnConfig,
    }
    request.Body = &model.UpdateAlertConfigRequestBody{
        WarnConfig: warnConfigbody,
        TopicUrn: "urn:smn:cn-north-7:2d2d90a56a3243bdb909f6a24a27be8d:cnad-test-intl",
        DisplayName: "",
    }
    response, err := client.UpdateAlertConfig(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	请求已成功

错误码

请参见**错误码**。

4.4 默认防护策略管理

4.4.1 配置 Anti-DDoS 默认防护策略

功能介绍

配置用户的默认防护策略。配置防护策略后，新购买的资源在自动开启防护时，会按照该默认防护策略进行配置。

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/antiddos/default-config

表 4-55 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id 最小长度：**32** 最大长度：**64**

请求参数

表 4-56 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type请求头 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

表 4-57 请求 Body 参数

参数	是否必选	参数类型	描述
enable_L7	是	Boolean	是否开启L7层防护，固定值为fasle
traffic_pos_id	是	Long	流量分段ID，取值范围：1～9、88、99 最小值：**1** 最大值：**99**
http_request_pos_id	是	Long	HTTP请求数分段ID，取值范围：1～15 最小值：**1** 最大值：**15**
cleaning_access_pos_id	是	Long	清洗时访问限制分段ID，取值范围：1～8、88、99 最小值：**1** 最大值：**99**
app_type_id	是	Integer	应用类型ID，可选取值： ● 0 ● 1

响应参数

状态码： 200

表 4-58 响应 Body 参数

参数	参数类型	描述
error_code	String	内部错误码 最小长度：**1** 最大长度：**20**
error_msg	String	内部错误描述 最小长度：**1** 最大长度：**255**

请求示例

配置用户默认的防护策略，清洗时访问限制分段ID设置为8，流量分段ID设置为1。

```
POST https://{endpoint}/v1/{project_id}/antiddos/default-config

{
```

```
  "app_type_id" : 0,
  "cleaning_access_pos_id" : 8,
  "enable_L7" : false,
  "http_request_pos_id" : 1,
  "traffic_pos_id" : 1
}
```

响应示例

状态码： 200

请求已成功

```
{
  "error_code" : "10000000",
  "error_msg" : "Ok",
  "task_id" : ""
}
```

SDK 代码示例

SDK代码示例如下。

Java

配置用户默认的防护策略，清洗时访问限制分段ID设置为8，流量分段ID设置为1。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.antiddos.v1.region.AntiDDoSRegion;
import com.huaweicloud.sdk.antiddos.v1.*;
import com.huaweicloud.sdk.antiddos.v1.model.*;


public class CreateDefaultConfigSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        AntiDDoSClient client = AntiDDoSClient.newBuilder()
                .withCredential(auth)
                .withRegion(AntiDDoSRegion.valueOf("cn-north-4"))
                .build();
        CreateDefaultConfigRequest request = new CreateDefaultConfigRequest();
        DdosConfig body = new DdosConfig();
        body.withAppTypeId(0L);
        body.withCleaningAccessPosId(8L);
        body.withHttpRequestPosId(1L);
        body.withTrafficPosId(1L);
        body.withEnableL7(false);
        request.withBody(body);
```

```
        try {
            CreateDefaultConfigResponse response = client.createDefaultConfig(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

配置用户默认的防护策略，清洗时访问限制分段ID设置为8，流量分段ID设置为1。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkantiddos.v1.region.antiddos_region import AntiDDoSRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkantiddos.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = AntiDDoSClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AntiDDoSRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = CreateDefaultConfigRequest()
        request.body = DdosConfig(
            app_type_id=0,
            cleaning_access_pos_id=8,
            http_request_pos_id=1,
            traffic_pos_id=1,
            enable_l7=False
        )
        response = client.create_default_config(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

配置用户默认的防护策略，清洗时访问限制分段ID设置为8，流量分段ID设置为1。

```
package main

import (
```

```
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    antiddos "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := antiddos.NewAntiDDoSClient(
        antiddos.AntiDDoSClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.CreateDefaultConfigRequest{}
    request.Body = &model.DdosConfig{
        AppTypeId: int64(0),
        CleaningAccessPosId: int64(8),
        HttpRequestPosId: int64(1),
        TrafficPosId: int64(1),
        EnableL7: false,
    }
    response, err := client.CreateDefaultConfig(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	请求已成功

错误码

请参见**错误码**。

4.4.2 查询 Anti-DDoS 默认防护策略

功能介绍

查询用户配置的默认防护策略。

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/antiddos/default-config

表 4-59 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id 最小长度：**32** 最大长度：**64**

请求参数

表 4-60 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type请求头 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

响应参数

状态码： 200

表 4-61 响应 Body 参数

参数	参数类型	描述
enable_L7	Boolean	是否开启L7层防护，固定值为fasle
traffic_pos_id	Long	流量分段ID，取值范围：1～9、88、99 最小值：**1** 最大值：**99**
http_request_pos_id	Long	HTTP请求数分段ID，取值范围：1～15 最小值：**1** 最大值：**15**
cleaning_access_pos_id	Long	清洗时访问限制分段ID，取值范围：1～8、88、99 最小值：**1** 最大值：**99**
app_type_id	Integer	应用类型ID，可选取值： • 0 • 1

请求示例

无

响应示例

状态码： 200

请求已成功

```
{
  "app_type_id" : 1,
  "cleaning_access_pos_id" : 8,
  "enable_L7" : false,
  "http_request_pos_id" : 8,
  "traffic_pos_id" : 8
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.antiddos.v1.region.AntiDDoSRegion;
```

```
import com.huaweicloud.sdk.antiddos.v1.*;
import com.huaweicloud.sdk.antiddos.v1.model.*;


public class ShowDefaultConfigSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        AntiDDoSClient client = AntiDDoSClient.newBuilder()
                .withCredential(auth)
                .withRegion(AntiDDoSRegion.valueOf("cn-north-4"))
                .build();
        ShowDefaultConfigRequest request = new ShowDefaultConfigRequest();
        try {
            ShowDefaultConfigResponse response = client.showDefaultConfig(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkantiddos.v1.region.antiddos_region import AntiDDoSRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkantiddos.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = AntiDDoSClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AntiDDoSRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ShowDefaultConfigRequest()
        response = client.show_default_config(request)
```

```
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    antiddos "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := antiddos.NewAntiDDoSClient(
        antiddos.AntiDDoSClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ShowDefaultConfigRequest{}
    response, err := client.ShowDefaultConfig(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	请求已成功

错误码

请参见**错误码**。

4.4.3 删除 Anti-DDoS 默认防护策略

功能介绍

删除用户配置的默认防护策略。

调用方法

请参见**如何调用API**。

URI

DELETE /v1/{project_id}/antiddos/default-config

表 4-62 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id 最小长度：**32** 最大长度：**64**

请求参数

表 4-63 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type请求头 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

响应参数

状态码： 200

表 4-64 响应 Body 参数

参数	参数类型	描述
error_code	String	内部错误码 最小长度：**1** 最大长度：**20**
error_msg	String	内部错误描述 最小长度：**1** 最大长度：**255**

请求示例

无

响应示例

状态码： 200

请求已成功

```
{
  "error_code" : "10000000",
  "error_msg" : "Ok",
  "task_id" : ""
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.antiddos.v1.region.AntiDDoSRegion;
import com.huaweicloud.sdk.antiddos.v1.*;
import com.huaweicloud.sdk.antiddos.v1.model.*;


public class DeleteDefaultConfigSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
```

```
            .withSk(sk);

    AntiDDoSClient client = AntiDDoSClient.newBuilder()
            .withCredential(auth)
            .withRegion(AntiDDoSRegion.valueOf("cn-north-4"))
            .build();
    DeleteDefaultConfigRequest request = new DeleteDefaultConfigRequest();
    try {
        DeleteDefaultConfigResponse response = client.deleteDefaultConfig(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
  }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkantiddos.v1.region.antiddos_region import AntiDDoSRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkantiddos.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = AntiDDoSClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AntiDDoSRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = DeleteDefaultConfigRequest()
        response = client.delete_default_config(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    antiddos "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/antiddos/v1/region"
```

```
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := antiddos.NewAntiDDoSClient(
        antiddos.AntiDDoSClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.DeleteDefaultConfigRequest{}
    response, err := client.DeleteDefaultConfig(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	请求已成功

错误码

请参见**错误码**。

5 DDoS 原生高级防护 API

5.1 DDoS 原生高级防护-告警配置管理

5.1.1 查询告警配置

功能介绍

查询告警配置

调用方法

请参见**如何调用API**。

URI

GET /v1/cnad/alarm-config

请求参数

表 5-1 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**

参数	是否必选	参数类型	描述
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

响应参数

状态码： 200

表 5-2 响应 Body 参数

参数	参数类型	描述
topic_urn	String	SMN的topic urn 最小长度：**1** 最大长度：**255**

请求示例

无

响应示例

状态码： 200

OK

```
{
  "topic_urn" : "urn:smn:cn-north-7:4229e3e5b9014d978b208b971f2e0e0c:DDos_test"
}
```

状态码

状态码	描述
200	OK
401	Unauthorized
403	Forbidden
404	Not Found

错误码

请参见**错误码**。

5.1.2 设置告警配置

功能介绍

设置告警配置

调用方法

请参见**如何调用API**。

URI

POST /v1/cnad/alarm-config

请求参数

表 5-3 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

表 5-4 请求 Body 参数

参数	是否必选	参数类型	描述
topic_urn	是	String	SMN的topic urn 最小长度：**1** 最大长度：**255**

响应参数

状态码： 200

表 5-5 响应 Body 参数

参数	参数类型	描述
topic_urn	String	SMN的topic urn 最小长度：**1** 最大长度：**255**

请求示例

开启告警通知，并指定名为“urn:smn:cn-north-7:2d2d90a56a3243bdb909f6a24a27be8d:cnad-test-intl”的消息通知的主题

```
POST https://{endpoint}/v1/cnad/alarm-config

{
  "topic_urn" : "urn:smn:cn-north-7:2d2d90a56a3243bdb909f6a24a27be8d:cnad-test-intl"
}
```

响应示例

状态码： 200

OK

```
{
  "topic_urn" : "urn:smn:cn-north-7:4229e3e5b9014d978b208b971f2e0e0c:DDos_test"
}
```

状态码

状态码	描述
200	OK
401	Unauthorized
403	Forbidden
404	Not Found

错误码

请参见**错误码**。

5.1.3 删除告警配置

功能介绍

删除告警配置

调用方法

请参见**如何调用API**。

URI

DELETE /v1/cnad/alarm-config

请求参数

表 5-6 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

响应参数

状态码： 200

表 5-7 响应 Body 参数

参数	参数类型	描述
topic_urn	String	SMN的topic urn 最小长度：**1** 最大长度：**255**

请求示例

无

响应示例

状态码： 200

OK

```
{
  "topic_urn" : "urn:smn:cn-north-7:4229e3e5b9014d978b208b971f2e0e0c:DDos_test"
}
```

状态码

状态码	描述
200	OK
401	Unauthorized
403	Forbidden

错误码

请参见**错误码**。

5.2 DDoS 原生高级防护-防护包管理

5.2.1 查询防护包列表

功能介绍

查询防护包列表

调用方法

请参见**如何调用API**。

URI

GET /v1/cnad/packages

请求参数

表 5-8 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**

参数	是否必选	参数类型	描述
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

响应参数

状态码： 200

表 5-9 响应 Body 参数

参数	参数类型	描述
total	Integer	总数 最小值：**0** 最大值：**2147483647**
items	Array of **PackageResponse** objects	数据 数组长度：**0 - 2147483647**

表 5-10 PackageResponse

参数	参数类型	描述
package_id	String	防护包id 最小长度：**32** 最大长度：**64**
package_name	String	防护包名 最小长度：**1** 最大长度：**255**
region_id	String	资源所属region 最小长度：**1** 最大长度：**255**
protection_type	Integer	防护类型 最小值：**0** 最大值：**1**

参数	参数类型	描述
instance_type	String	防护包类型。cnad_pro：专业版；cnad_ip：标准版；cnad_ep：铂金版；cnad_full_high：全力防高级版；cnad_vic：按需版；cnad_intl_ep：国际站铂金版
resource_id	String	资源id 最小长度：**32** 最大长度：**64**
count_down_code	String	倒计时相关信息 最小长度：**1** 最大长度：**255**
count_down_infos	String	倒计时相关信息 最小长度：**1** 最大长度：**255**
count_down_tips	String	倒计时相关信息 最小长度：**1** 最大长度：**255**
order_id	String	订单id 最小长度：**17** 最大长度：**17**
subscription_id	String	续费用的id 最小长度：**32** 最大长度：**64**
ip_num	Integer	ip数 最小值：**0** 最大值：**2147483647**
ip_num_now	Integer	当前IP数 最小值：**0** 最大值：**2147483647**
protection_num_now	Integer	当前防护次数 最小值：**0** 最大值：**9999**
protection_num	Integer	防护次数，9999为无限次 最小值：**0** 最大值：**9999**

参数	参数类型	描述
basic_bandwidth	Integer	保底带宽 最小值：**0** 最大值：**2147483647**
elastic_bandwidth	Integer	弹性带宽 最小值：**0** 最大值：**2147483647**
service_bandwidth	Integer	业务带宽 最小值：**0** 最大值：**2147483647**
clean_bandwidth	Integer	回源带宽 最小值：**0** 最大值：**2147483647**
policy_num	Integer	策略模板数 最小值：**0** 最大值：**2147483647**
is_old	Boolean	是否旧防护包（旧防护包不支持升级规格）,默认不传为否
new_flag	Boolean	专业版铂金版合并之后购买的专业版和铂金版均标识为true
create_time	Long	创建时间 最小值：**0** 最大值：**9223372036854775807**

请求示例

无

响应示例

状态码： 200

OK


```
{
  "items" : [ {
    "package_id" : "0c613ff9-2b63-4aa2-9333-e03541fe0d9a",
    "package_name" : "name",
    "region_id" : "cn-north4",
    "protection_type" : 0,
    "instance_type" : "cnad_pro",
    "resource_id" : "0c613ff9-2b63-4aa2-9333-e03541fe0d9a",
    "count_down_code" : "hws_countdown_period_using",
    "count_down_infos" : {
      "status" : 2,
```



```
    "nextOperationPolicy" : 0,
    "nextOperationRemainingDay" : 10
  },
  "count_down_tips" : {
    "effTime" : "2022-10-09T10:00:17Z",
    "expTime" : "2022-12-09T15:59:59Z"
  },
  "order_id" : "CS2211020327RKFEA",
  "subscription_id" : "0c613ff9-2b63-4aa2-9333-e03541fe0d9a",
  "ip_num" : 100,
  "ip_num_now" : 100,
  "protection_num_now" : 1,
  "protection_num" : 9999,
  "basic_bandwidth" : 10,
  "elastic_bandwidth" : 200,
  "service_bandwidth" : 100,
  "clean_bandwidth" : 100,
  "policy_num" : 1,
  "is_old" : false,
  "new_flag" : true,
  "create_time" : 1665309611045
 } ],
 "total" : 1
}
```


状态码

状态码	描述
200	OK
401	Unauthorized
403	Forbidden
404	Not Found

错误码

请参见**错误码**。

5.2.2 更新防护包名字

功能介绍

更新防护包名字

调用方法

请参见**如何调用API**。

URI

PUT /v1/cnad/packages/{package_id}/name

表 5-11 路径参数

参数	是否必选	参数类型	描述
package_id	是	String	防护包id 最小长度：**32** 最大长度：**64**

请求参数

表 5-12 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

表 5-13 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	名字 最小长度：**1** 最大长度：**255**

响应参数

无

请求示例

将指定防护包名称更新为“package_name”。

```
PUT https://{endpoint}/v1/cnad/packages/{package_id}/name

{
```

```
  "name" : "package_name"
}
```

响应示例

无

状态码

状态码	描述
200	OK
401	Unauthorized
403	Forbidden
404	Not Found

错误码

请参见**错误码**。

5.2.3 更新防护包绑定的全量防护对象

功能介绍

更新防护包绑定的全量防护对象

调用方法

请参见**如何调用API**。

URI

POST /v1/cnad/packages/{package_id}/protected-ips

表 5-14 路径参数

参数	是否必选	参数类型	描述
package_id	是	String	防护包id 最小长度：**32** 最大长度：**64**

请求参数

表 5-15 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

表 5-16 请求 Body 参数

参数	是否必选	参数类型	描述
protected_ip_list	是	Array of **UpdateProtectedIpInPolicyBody** objects	全量防护ip列表 数组长度：**0 - 2147483647**

表 5-17 UpdateProtectedIpInPolicyBody

参数	是否必选	参数类型	描述
id	是	String	防护ip的id 最小长度：**32** 最大长度：**64**
ip	是	String	防护ip 最小长度：**7** 最大长度：**128**
type	是	String	类型。VPN：VPN；NAT：NAT；VIP：VIP；CCI：CCI；EIP：弹性公网IP；ELB：弹性负载均衡；REROUTING_IP：REROUTING_IP；SubEni：SubEni；NetInterFace：NetInterFace；

参数	是否必选	参数类型	描述
name	否	String	名字 最小长度：**1** 最大长度：**255**

响应参数

无

请求示例

将指定防护包的全量防护对象更改为EIP“1.1.1.1”（ID标识为“0c613ff9-2b63-4aa2-9333-e03541fe0d9a”）。

```
POST https://{endpoint}/v1/cnad/packages/{package_id}/protected-ips

{
  "protected_ip_list" : [ {
    "ip" : "1.1.1.1",
    "id" : "0c613ff9-2b63-4aa2-9333-e03541fe0d9a",
    "type" : "EIP"
  } ]
}
```

响应示例

无

状态码

状态码	描述
200	OK
401	Unauthorized
403	Forbidden
404	Not Found

错误码

请参见**错误码**。

5.2.4 查询可绑定的防护对象列表

功能介绍

查询可绑定的防护对象列表

调用方法

请参见**如何调用API**。

URI

GET /v1/cnad/packages/{package_id}/unbound-protected-ips

表 5-18 路径参数

参数	是否必选	参数类型	描述
package_id	是	String	防护包id 最小长度：**32** 最大长度：**64**

表 5-19 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	开始查询的偏移量,默认值:0 最小值：**0** 最大值：**65535**
limit	否	Integer	每页显示的条目数量,默认值:2000 最小值：**1** 最大值：**2000**

请求参数

表 5-20 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**

参数	是否必选	参数类型	描述
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

响应参数

状态码： 200

表 5-21 响应 Body 参数

参数	参数类型	描述
total	Integer	总数 最小值：**0** 最大值：**2147483647**
ips	Array of **ProtectedIpResponse** objects	防护ip列表 数组长度：**0 - 2147483647**

表 5-22 ProtectedIpResponse

参数	参数类型	描述
id	String	防护IP的Id 最小长度：**32** 最大长度：**64**
ip	String	防护IP 最小长度：**7** 最大长度：**128**
type	String	类型。VPN：VPN；NAT：NAT；VIP：VIP；CCI：CCI；EIP：弹性公网IP；ELB：弹性负载均衡；REROUTING_IP：REROUTING_IP；SubEni：SubEni；NetInterFace：NetInterFace；
name	String	名字 最小长度：**1** 最大长度：**255**

参数	参数类型	描述
status	Integer	状态：0 - 正常， 1 - 清洗中， 2 - 黑洞中 最小值：**0** 最大值：**2**
status_detail	IpStatusDetail object	封堵信息
policy_name	String	策略名 最小长度：**1** 最大长度：**255**
region	String	所属region 最小长度：**1** 最大长度：**255**
package_id	String	防护包id 最小长度：**32** 最大长度：**64**
package_name	String	防护包名 最小长度：**0** 最大长度：**255**
tags	String	TMS标签 最小长度：**1** 最大长度：**255**
tag	String	本地标签 最小长度：**0** 最大长度：**255**
is_resale	Boolean	默认false，表示是否转售版的IP，不需要展示策略和报表
package_version	String	package_version。cnad_pro：专业版；cnad_ip：标准版；cnad_ep：铂金版；cnad_full_high：全力防高级版；cnad_vic：按需版；cnad_intl_ep：国际站铂金版

表 5-23 IpStatusDetail

参数	参数类型	描述
block_time	Long	封堵时间 最小值：**0** 最大值：**9223372036854775807**

参数	参数类型	描述
unblock_time	Long	解封时间 最小值：**0** 最大值：**9223372036854775807**

请求示例

无

响应示例

状态码： 200

OK

```
{
  "ips" : [ {
    "id" : "2b93b3ed-7aed-4b73-be77-d04318cbda5e",
    "ip" : "100.85.112.111",
    "is_resale" : false,
    "tags" : "{}",
    "type" : "EIP"
  } ],
  "total" : 1
}
```

状态码

状态码	描述
200	OK
401	Unauthorized
403	Forbidden
404	Not Found

错误码

请参见**错误码**。

5.3 DDoS 原生高级防护-策略管理

5.3.1 查询策略列表

功能介绍

查询策略列表

调用方法

请参见**如何调用API**。

URI

GET /v1/cnad/policies

表 5-24 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	开始查询的偏移量,默认值:0 最小值：**0** 最大值：**65535**
limit	否	Integer	每页显示的条目数量,默认值:2000 最小值：**1** 最大值：**2000**
policy_name	否	String	策略名 最小长度：**0** 最大长度：**50**

请求参数

表 5-25 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

响应参数

状态码： 200

表 5-26 响应 Body 参数

参数	参数类型	描述
total	Integer	总数 最小值：**0** 最大值：**2147483647**
items	Array of **PolicyResponse** objects	策略列表 数组长度：**0 - 2147483647**

表 5-27 PolicyResponse

参数	参数类型	描述
id	String	id 最小长度：**32** 最大长度：**64**
package_id	String	防护包id 最小长度：**32** 最大长度：**64**
package_name	String	防护包名 最小长度：**0** 最大长度：**255**
name	String	策略名 最小长度：**0** 最大长度：**255**
description	String	描述 最小长度：**0** 最大长度：**255**
region	String	所属region的id 最小长度：**1** 最大长度：**255**
clean_threshold	Integer	清洗阈值 最小值：**100** 最大值：**1000**
num_protected_ip	Integer	防护ip数 最小值：**0** 最大值：**2147483647**

请求示例

无

响应示例

状态码： 200

OK


```
{
  "items" : [ {
    "clean_threshold" : 120,
    "id" : "798d8975-f7cf-490f-b990-747ebaf41248",
    "name" : "sadf",
    "num_protected_ip" : 0,
    "package_id" : "09b5ae8e-ae05-49ff-80fc-8bdf68c16499",
    "package_name" : "测试",
    "region" : "cn-north-4"
  } ],
  "total" : 1
}
```


状态码

状态码	描述
200	OK
401	Unauthorized
403	Forbidden
404	Not Found

错误码

请参见**错误码**。

5.3.2 创建策略

功能介绍

创建策略

调用方法

请参见**如何调用API**。

URI

POST /v1/cnad/policies

请求参数

表 5-28 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

表 5-29 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	策略名 最小长度：**1** 最大长度：**255**
package_id	是	String	防护包id 最小长度：**32** 最大长度：**64**
description	否	String	描述 最小长度：**0** 最大长度：**255**

响应参数

状态码： 200

表 5-30 响应 Body 参数

参数	参数类型	描述
id	String	策略id 最小长度：**32** 最大长度：**64**

参数	参数类型	描述
name	String	策略名 最小长度：**0** 最大长度：**255**
package_id	String	防护包id 最小长度：**32** 最大长度：**64**
description	String	描述 最小长度：**0** 最大长度：**255**
clean_threshold	Integer	清洗阈值 最小值：**100** 最大值：**1000**

请求示例

为id为“0c613ff9-2b63-4aa2-9333-e03541fe0d9a”的防护包创建一个名称为“name”的防护策略。

```
POST https://{endpoint}/v1/cnad/policies

{
  "name" : "name",
  "package_id" : "0c613ff9-2b63-4aa2-9333-e03541fe0d9a",
  "description" : ""
}
```

响应示例

状态码： 200

OK

```
{
  "id" : "0c613ff9-2b63-4aa2-9333-e03541fe0d9a",
  "name" : "名字",
  "package_id" : "0c613ff9-2b63-4aa2-9333-e03541fe0d9a",
  "description" : "描述",
  "clean_threshold" : 100
}
```

状态码

状态码	描述
200	OK
401	Unauthorized
403	Forbidden

状态码	描述
404	Not Found

错误码

请参见**错误码**。

5.3.3 查询策略详情

功能介绍

查询策略详情

调用方法

请参见**如何调用API**。

URI

GET /v1/cnad/policies/{policy_id}

表 5-31 路径参数

参数	是否必选	参数类型	描述
policy_id	是	String	策略id 最小长度：**32** 最大长度：**64**

请求参数

表 5-32 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**

参数	是否必选	参数类型	描述
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

响应参数

状态码： 200

表 5-33 响应 Body 参数

参数	参数类型	描述
id	String	策略id 最小长度：**32** 最大长度：**64**
package_id	String	防护包id 最小长度：**32** 最大长度：**64**
name	String	策略名 最小长度：**0** 最大长度：**255**
clean_threshold	Integer	清洗阈值 最小值：**100** 最大值：**1000**
pop_policy	**PopPolicy** object	策略阈值详情

表 5-34 PopPolicy

参数	参数类型	描述
block_location	Array of strings	位置封禁列表 最小长度：**1** 最大长度：**255** 数组长度：**0 - 255**

参数	参数类型	描述
block_protocol	Array of strings	协议封禁列表 最小长度：**1** 最大长度：**255** 数组长度：**0 - 255**
bw_list	**Bw** object	黑白名单详情
connection_protection	Boolean	是否开启连接防护
connection_protection_list	Array of strings	连接防护列表 最小长度：**1** 最大长度：**255** 数组长度：**0 - 255**
fingerprint_count	Integer	指纹数 最小值：**0** 最大值：**255**
port_block_count	Integer	端口封禁数 最小值：**0** 最大值：**255**
watermark_count	Integer	水印数 最小值：**0** 最大值：**255**
if_exist_traffic	Boolean	是否存在流量
pop	String	固定值ALL

表 5-35 Bw

参数	参数类型	描述
black_ip_list	Array of strings	黑名单列表 最小长度：**7** 最大长度：**128** 数组长度：**0 - 255**
white_ip_list	Array of strings	白名单列表 最小长度：**7** 最大长度：**128** 数组长度：**0 - 255**

请求示例

无

响应示例

状态码： 200

OK

```
{
  "clean_threshold" : 120,
  "id" : "798d8975-f7cf-490f-b990-747ebaf41248",
  "name" : "sadf",
  "package_id" : "09b5ae8e-ae05-49ff-80fc-8bdf68c16499",
  "pop_policy" : [ {
    "block_location" : [ ],
    "block_protocol" : [ ],
    "bw_list" : [ {
      "black_ip_list" : [ ],
      "white_ip_list" : [ ]
    } ],
    "connection_protection" : false,
    "connection_protection_list" : [ ],
    "fingerprint_count" : 0,
    "if_exist_traffic" : false,
    "pop" : "ALL",
    "port_block_count" : 0,
    "watermark_count" : 0
  } ]
}
```

状态码

状态码	描述
200	OK
401	Unauthorized
403	Forbidden
404	Not Found

错误码

请参见**错误码**。

5.3.4 更新策略

功能介绍

更新策略

调用方法

请参见**如何调用API**。

URI

PUT /v1/cnad/policies/{policy_id}

表 5-36 路径参数

参数	是否必选	参数类型	描述
policy_id	是	String	策略id 最小长度：**32** 最大长度：**64**

请求参数

表 5-37 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

表 5-38 请求 Body 参数

参数	是否必选	参数类型	描述
name	否	String	策略名 最小长度：**1** 最大长度：**255**
threshold	否	Integer	清洗阈值 最小值：**100** 最大值：**1000**
description	否	String	描述 最小长度：**0** 最大长度：**255**

参数	是否必选	参数类型	描述
udp	否	String	udp协议封禁。block：封禁，unblock：不封禁

响应参数

无

请求示例

将指定防护策略的名称更新为“name”，打开UDP协议封禁功能，清洗阈值设置为“100Mbps”。

```
PUT https://{endpoint}/v1/cnad/policies/{policy_id}

{
  "name" : "name",
  "udp" : "block",
  "description" : "",
  "threshold" : 100
}
```

响应示例

无

状态码

状态码	描述
200	OK
401	Unauthorized
403	Forbidden
404	Not Found

错误码

请参见**错误码**。

5.3.5 删除策略

功能介绍

删除策略

调用方法

请参见**如何调用API**。

URI

DELETE /v1/cnad/policies/{policy_id}

表 5-39 路径参数

参数	是否必选	参数类型	描述
policy_id	是	String	策略id 最小长度：**32** 最大长度：**64**

请求参数

表 5-40 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

响应参数

无

请求示例

无

响应示例

无

状态码

状态码	描述
200	OK
401	Unauthorized
403	Forbidden

错误码

请参见**错误码**。

5.3.6 策略添加黑白名单

功能介绍

策略添加黑白名单

调用方法

请参见**如何调用API**。

URI

POST /v1/cnad/policies/{policy_id}/ip-list/add

表 5-41 路径参数

参数	是否必选	参数类型	描述
policy_id	是	String	策略id 最小长度：**32** 最大长度：**64**

请求参数

表 5-42 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**

参数	是否必选	参数类型	描述
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

表 5-43 请求 Body 参数

参数	是否必选	参数类型	描述
type	是	String	类型。white：白名单，black：黑名单
ip_list	是	Array of strings	ip列表 最小长度：**7** 最大长度：**128** 数组长度：**0 - 2147483647**

响应参数

无

请求示例

将IP“1.1.1.1”和“2.2.2.2”加入到指定防护策略的白名单。

```
POST https://{endpoint}/v1/cnad/policies/{policy_id}/ip-list/add

{
  "type" : "white",
  "ip_list" : [ "1.1.1.1", "2.2.2.2" ]
}
```

响应示例

无

状态码

状态码	描述
200	OK
401	Unauthorized
403	Forbidden
404	Not Found

错误码

请参见**错误码**。

5.3.7 策略删除黑白名单

功能介绍

策略删除黑白名单

调用方法

请参见**如何调用API**。

URI

POST /v1/cnad/policies/{policy_id}/ip-list/delete

表 5-44 路径参数

参数	是否必选	参数类型	描述
policy_id	是	String	策略id 最小长度：**32** 最大长度：**64**

请求参数

表 5-45 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

表 5-46 请求 Body 参数

参数	是否必选	参数类型	描述
type	是	String	类型。white：白名单，black：黑名单
ip_list	是	Array of strings	ip列表 最小长度：**7** 最大长度：**128** 数组长度：**0 - 2147483647**

响应参数

无

请求示例

将IP“1.1.1.1”和“2.2.2.2”从指定防护策略的白名单中删除。

```
POST https://{endpoint}/v1/cnad/policies/{policy_id}/ip-list/delete

{
  "type" : "white",
  "ip_list" : [ "1.1.1.1", "2.2.2.2" ]
}
```

响应示例

无

状态码

状态码	描述
200	OK
401	Unauthorized
403	Forbidden
404	Not Found

错误码

请参见**错误码**。

5.3.8 策略绑定防护对象

功能介绍

策略绑定防护对象

调用方法

请参见**如何调用API**。

URI

POST /v1/cnad/policies/{policy_id}/bind

表 5-47 路径参数

参数	是否必选	参数类型	描述
policy_id	是	String	策略id 最小长度：**32** 最大长度：**64**

请求参数

表 5-48 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

表 5-49 请求 Body 参数

参数	是否必选	参数类型	描述
package_id	是	String	防护包id 最小长度：**32** 最大长度：**64**

参数	是否必选	参数类型	描述
id_list	是	Array of strings	防护ip的id列表 最小长度：**7** 最大长度：**128** 数组长度：**0 - 2147483647**

响应参数

无

请求示例

为ID为634f346e-a291-4124-a6be-ac57e6152463的防护包下指定防护策略绑定防护ID为“634f346e-a291-4124-a6be-ac57e6152464”的IP。

```
POST https://{endpoint}/v1/cnad/policies/{policy_id}/bind

{
  "package_id" : "634f346e-a291-4124-a6be-ac57e6152463",
  "id_list" : [ "634f346e-a291-4124-a6be-ac57e6152464" ]
}
```

响应示例

无

状态码

状态码	描述
200	OK
401	Unauthorized
403	Forbidden
404	Not Found

错误码

请参见**错误码**。

5.3.9 策略解绑防护对象

功能介绍

策略解绑防护对象

调用方法

请参见**如何调用API**。

URI

POST /v1/cnad/policies/{policy_id}/unbind

表 5-50 路径参数

参数	是否必选	参数类型	描述
policy_id	是	String	策略id 最小长度：**32** 最大长度：**64**

请求参数

表 5-51 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

表 5-52 请求 Body 参数

参数	是否必选	参数类型	描述
package_id	是	String	防护包id 最小长度：**32** 最大长度：**64**

参数	是否必选	参数类型	描述
id_list	是	Array of strings	防护ip的id列表 最小长度：**7** 最大长度：**128** 数组长度：**0 - 2147483647**

响应参数

无

请求示例

将ID为“634f346e-a291-4124-a6be-ac57e6152463”的防护包下指定防护策略中绑定的ID为“634f346e-a291-4124-a6be-ac57e6152464”的防护IP解除绑定。

```
POST https://{endpoint}/v1/cnad/policies/{policy_id}/unbind

{
  "package_id" : "634f346e-a291-4124-a6be-ac57e6152463",
  "id_list" : [ "634f346e-a291-4124-a6be-ac57e6152464" ]
}
```

响应示例

无

状态码

状态码	描述
200	OK
401	Unauthorized
403	Forbidden
404	Not Found

错误码

请参见**错误码**。

5.4 DDoS 原生高级防护-防护对象管理

5.4.1 查询防护对象列表

功能介绍

查询防护对象列表

调用方法

请参见**如何调用API**。

URI

GET /v1/cnad/protected-ips

表 5-53 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	开始查询的偏移量,默认值:0 最小值：**0** 最大值：**65535**
limit	否	Integer	每页显示的条目数量,默认值:2000 最小值：**1** 最大值：**2000**
package_id	否	String	防护包id 最小长度：**32** 最大长度：**64**
policy_id	否	String	策略id 最小长度：**32** 最大长度：**64**
ip	否	String	防护ip 最小长度：**7** 最大长度：**128**
tag	否	String	本地标签 最小长度：**1** 最大长度：**255**

请求参数

表 5-54 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

响应参数

状态码： 200

表 5-55 响应 Body 参数

参数	参数类型	描述
total	Integer	总数 最小值：**0** 最大值：**2147483647**
items	Array of **ProtectedIpResponse** objects	防护ip列表 数组长度：**0 - 2147483647**

表 5-56 ProtectedIpResponse

参数	参数类型	描述
id	String	防护IP的Id 最小长度：**32** 最大长度：**64**
ip	String	防护IP 最小长度：**7** 最大长度：**128**

参数	参数类型	描述
type	String	类型。VPN：VPN；NAT：NAT；VIP：VIP；CCI：CCI；EIP：弹性公网IP；ELB：弹性负载均衡；REROUTING_IP：REROUTING_IP；SubEni：SubEni；NetInterFace：NetInterFace；
name	String	名字 最小长度：**1** 最大长度：**255**
status	Integer	状态：0 - 正常， 1 - 清洗中， 2 - 黑洞中 最小值：**0** 最大值：**2**
status_detail	IpStatusDetail object	封堵信息
policy_name	String	策略名 最小长度：**1** 最大长度：**255**
region	String	所属region 最小长度：**1** 最大长度：**255**
package_id	String	防护包id 最小长度：**32** 最大长度：**64**
package_name	String	防护包名 最小长度：**0** 最大长度：**255**
tags	String	TMS标签 最小长度：**1** 最大长度：**255**
tag	String	本地标签 最小长度：**0** 最大长度：**255**
is_resale	Boolean	默认false，表示是否转售版的IP，不需要展示策略和报表
package_version	String	package_version。cnad_pro：专业版；cnad_ip：标准版；cnad_ep：铂金版；cnad_full_high：全力防高级版；cnad_vic：按需版；cnad_intl_ep：国际站铂金版

表 5-57 IpStatusDetail

参数	参数类型	描述
block_time	Long	封堵时间 最小值：**0** 最大值：**9223372036854775807**
unblock_time	Long	解封时间 最小值：**0** 最大值：**9223372036854775807**

请求示例

无

响应示例

状态码： 200

OK

```
{
  "items" : [ {
    "id" : "7f0e3af2-455a-4a47-b95c-49137794f0c5",
    "ip" : "100.85.222.60",
    "is_resale" : false,
    "package_id" : "17f8f630-0714-4b3a-aedf-3983d7b98124",
    "package_name" : "kxnkxn99",
    "package_version" : "cnad_ip",
    "policy_name" : "tesy",
    "region" : "cn-north-7",
    "status" : 0,
    "type" : "EIP"
  } ],
  "total" : 1
}
```

状态码

状态码	描述
200	OK
401	Unauthorized
403	Forbidden
404	Not Found

错误码

请参见**错误码**。

5.4.2 防护对象设置标签

功能介绍

防护对象设置标签

调用方法

请参见**如何调用API**。

URI

PUT /v1/cnad/protected-ips/tags

请求参数

表 5-58 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

表 5-59 请求 Body 参数

参数	是否必选	参数类型	描述
id	是	String	防护ip的id 最小长度：**32** 最大长度：**64**
tag	是	String	本地标签 最小长度：**0** 最大长度：**2147483647**

响应参数

无

请求示例

将ID为“79189f77-bc57-46ff-a69d-17168d95c970”的防护IP的标签设置为“test”。

```
PUT https://{endpoint}/v1/cnad/protected-ips/tags

{
  "id" : "79189f77-bc57-46ff-a69d-17168d95c970",
  "tag" : "test"
}
```

响应示例

无

状态码

状态码	描述
200	OK
401	Unauthorized
403	Forbidden
404	Not Found

错误码

请参见**错误码**。

5.5 解封中心-解封管理

5.5.1 查询封堵统计数据

功能介绍

查询封堵统计数据

调用方法

请参见**如何调用API**。

URI

GET /v1/unblockservice/{domain_id}/block-statistics

表 5-60 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	租户id 最小长度：**32** 最大长度：**64**

请求参数

表 5-61 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户token 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	application/json;charset=utf8 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

响应参数

状态码： 200

表 5-62 响应 Body 参数

参数	参数类型	描述
total_unblocking_times	Integer	总解封次数 最小值：**0** 最大值：**2097152**
manual_unblocking_times	Integer	人工解封次数 最小值：**0** 最大值：**2097152**
automatic_unblocking_times	Integer	自动解封次数 最小值：**0** 最大值：**2097152**

参数	参数类型	描述
current_blocked_ip_numbers	Integer	当前封堵Ip数 最小值：**0** 最大值：**2097152**

请求示例

无

响应示例

状态码： 200

OK

```
{
  "total_unblocking_times" : 8,
  "manual_unblocking_times" : 3,
  "automatic_unblocking_times" : 5,
  "current_blocked_ip_numbers" : 1
}
```

状态码

状态码	描述
200	OK

错误码

请参见**错误码**。

5.5.2 查询租户封堵列表

功能介绍

查询租户封堵列表

调用方法

请参见**如何调用API**。

URI

GET /v1/unblockservice/{domain_id}/block-list

表 5-63 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	租户id 最小长度：**32** 最大长度：**64**

请求参数

表 5-64 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户token 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	application/json;charset=utf8 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

响应参数

状态码： 200

表 5-65 响应 Body 参数

参数	参数类型	描述
blocking_list	Array of **blocking_list** objects	封堵列表响应体 数组长度：**0 - 2000**
total	Integer	总数 最小值：**0** 最大值：**2000**

表 5-66 blocking_list

参数	参数类型	描述
ip	String	ip地址 最小长度：**7** 最大长度：**128**
blocking_time	Long	封堵时间 最小值：**0** 最大值：**9223372036854775807**
estimated_unblocking_time	Long	预计解封时间 最小值：**0** 最大值：**9223372036854775807**
status	String	状态。unblocking：解封中；success：成功；failed：失败

请求示例

无

响应示例

状态码： 200

OK

```
{
  "total" : 1,
  "blocking_list" : [ {
    "ip" : "100.93.15.45",
    "status" : "blocking",
    "blocking_time" : 1672110604844,
    "estimated_unblocking_time" : 1672114204823
  } ]
}
```

状态码

状态码	描述
200	OK

错误码

请参见**错误码**。

5.5.3 查询租户解封记录

功能介绍

查询租户解封记录

调用方法

请参见**如何调用API**。

URI

GET /v1/unblockservice/{domain_id}/unblock-record

表 5-67 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	租户id 最小长度：**32** 最大长度：**64**

表 5-68 Query 参数

参数	是否必选	参数类型	描述
start_time	是	Long	开始时间 最小值：**0** 最大值：**9223372036854775807**
end_time	是	Long	结束时间 最小值：**0** 最大值：**9223372036854775807**

请求参数

表 5-69 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户token 最小长度：**32** 最大长度：**2097152**

参数	是否必选	参数类型	描述
Content-Type	是	String	application/json;charset=utf8 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

响应参数

状态码： 200

表 5-70 响应 Body 参数

参数	参数类型	描述
unblock_record	Array of **unblock_record** objects	解封记录 数组长度：**0 - 2000**
total	Integer	总数 最小值：**1** 最大值：**2000**
domain_id	String	租户id 最小长度：**32** 最大长度：**64**

表 5-71 unblock_record

参数	参数类型	描述
ip	String	ip地址 最小长度：**7** 最大长度：**128**
executor	String	执行者 最小长度：**7** 最大长度：**255**
block_id	Long	封堵id 最小值：**0** 最大值：**9223372036854775807**

参数	参数类型	描述
blocking_time	Long	封堵时间 最小值：**0** 最大值：**9223372036854775807**
unblocking_time	Long	解封时间 最小值：**0** 最大值：**9223372036854775807**
unblock_type	String	解封类型。manual：人工；automatic：自动
status	String	状态。unblocking：解封中；success：成功；failed：失败
sort_time	Long	时间 最小值：**0** 最大值：**9223372036854775807**

请求示例

无

响应示例

状态码： 200

OK

```
{
  "total" : 8,
  "domain_id" : "02753903d8994d8ea664166ed6b9cc24",
  "unblock_record" : [ {
    "ip" : "100.93.15.45",
    "executor" : "pianjiangtao",
    "status" : "success",
    "block_id" : 37090,
    "blocking_time" : 1671069809245,
    "unblocking_time" : 1671070811966,
    "unblock_type" : "manual",
    "sort_time" : 1671070811966
  }, {
    "ip" : "100.93.3.136",
    "executor" : "System",
    "status" : "success",
    "block_id" : 37089,
    "blocking_time" : 1671006261093,
    "unblocking_time" : 1671009811163,
    "unblock_type" : "automatic",
    "sort_time" : 167100981116
  } ]
}
```


状态码

状态码	描述
200	OK

错误码

请参见**错误码**。

5.5.4 解封 IP

功能介绍

解封IP

调用方法

请参见**如何调用API**。

URI

POST /v1/unblockservice/{domain_id}/unblock

表 5-72 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	租户id 最小长度：**32** 最大长度：**64**

请求参数

表 5-73 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户token 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	application/json;charset=utf8 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

表 5-74 请求 Body 参数

参数	是否必选	参数类型	描述
ip	是	String	ip地址 最小长度：**7** 最大长度：**128**
blocking_time	否	Long	用于查询IP的封堵时间 最小值：**0** 最大值：**9223372036854775807**

响应参数

状态码： 200

表 5-75 响应 Body 参数

参数	参数类型	描述
error_code	String	业务错误码 最小长度：**8** 最大长度：**128**
message	String	描述信息 最小长度：**1** 最大长度：**1024**

请求示例

解封防护IP“...”。

```
POST https://{endpoint}/v1/unblockservice/{domain_id}/unblock

{
  "ip" : "*.*.*.*",
  "blocking_time" : 1688105685078
}
```

响应示例

状态码： 200

OK

```
{
  "error_code" : "200 OK",
```

```
  "message" : "success"
}
```

状态码

状态码	描述
200	OK

错误码

请参见**错误码**。

5.5.5 查询解封配额

功能介绍

查询解封配额

调用方法

请参见**如何调用API**。

URI

GET /v1/unblockservice/{domain_id}/unblock-quota-statistics

表 5-76 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	租户id 最小长度：**32** 最大长度：**64**

请求参数

表 5-77 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户token 最小长度：**32** 最大长度：**2097152**

参数	是否必选	参数类型	描述
Content-Type	是	String	application/json;charset=utf8 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**255**

响应参数

状态码： 200

表 5-78 响应 Body 参数

参数	参数类型	描述
type	String	用户类型：common_user , native_protection_user
total_unblocking_quota	Integer	解封总配额 最小值：**0** 最大值：**2097152**
remaining_unblocking_quota	Integer	剩余解封配额 最小值：**0** 最大值：**2097152**
unblocking_quota_today	Integer	今日解封配额 最小值：**0** 最大值：**2097152**
remaining_unblocking_quota_today	Integer	今日剩余解封配额 最小值：**0** 最大值：**2097152**

请求示例

无

响应示例

状态码： 200

OK

```
{
  "type" : "self_unblocking_user",
  "total_unblocking_quota" : 103,
  "remaining_unblocking_quota" : 100,
```

```
  "unblocking_quota_today" : 4,
  "remaining_unblocking_quota_today" : 4
}
```

状态码

状态码	描述
200	OK

错误码

请参见**错误码**。

6 DDoS 高防 API

6.1 DDoS 高防-概览

6.1.1 查询流量峰值、攻击计数

功能介绍

查询流量峰值、攻击计数

调用方法

请参见**如何调用API**。

URI

GET /v1/aad/instances/{instance_id}/{ip}/ddos-statistics

表 6-1 路径参数

参数	是否必选	参数类型	描述
instance_id	是	String	实例Id 最小长度：**16** 最大长度：**64**
ip	是	String	单个 IP 最小长度：**1** 最大长度：**45**

表 6-2 Query 参数

参数	是否必选	参数类型	描述
start_time	是	String	开始时间，毫秒时间戳
end_time	是	String	结束时间，毫秒时间戳

请求参数

表 6-3 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	token 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**128**

响应参数

状态码： 200

表 6-4 响应 Body 参数

参数	参数类型	描述
attack_kbps_peak	Number	攻击峰值 最小值：**0** 最大值：**100000000**
in_kbps_peak	Number	流量峰值 最小值：**0** 最大值：**100000000**
ddos_count	Number	攻击次数 最小值：**0** 最大值：**100000000**
timestamp	Number	攻击峰值发生时间点 最小值：**0** 最大值：**4803811200000**

参数	参数类型	描述
vip	String	高防IP

请求示例

无

响应示例

状态码： 200

攻击流量峰值、攻击峰值时间、入流量峰值、攻击次数

```
{
  "attack_kbps_peak" : 200000000,
  "ddos_count" : 24691,
  "in_kbps_peak" : 0,
  "timestamp" : 1671415200000,
  "vip" : "100.10.10.10"
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.aad.v1.region.AadRegion;
import com.huaweicloud.sdk.aad.v1.*;
import com.huaweicloud.sdk.aad.v1.model.*;


public class ListPeakSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new GlobalCredentials()
                .withAk(ak)
                .withSk(sk);

        AadClient client = AadClient.newBuilder()
                .withCredential(auth)
                .withRegion(AadRegion.valueOf("<YOUR REGION>"))
                .build();
        ListPeakRequest request = new ListPeakRequest();
        request.withStartTime("<start_time>");
        request.withEndTime("<end_time>");
```



```
        try {
            ListPeakResponse response = client.listPeak(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkaad.v1.region.aad_region import AadRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkaad.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = GlobalCredentials(ak, sk) \

    client = AadClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AadRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListPeakRequest()
        request.start_time = "<start_time>"
        request.end_time = "<end_time>"
        response = client.list_peak(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    aad "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```

```
variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := aad.NewAadClient(
        aad.AadClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListPeakRequest{}
    request.StartTime = "<start_time>"
    request.EndTime = "<end_time>"
    response, err := client.ListPeak(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	攻击流量峰值、攻击峰值时间、入流量峰值、攻击次数

错误码

请参见**错误码**。

6.2 DDoS 高防-实例列表

6.2.1 查询高防实例列表

功能介绍

查询高防实例列表

调用方法

请参见**如何调用API**。

URI

GET /v1/aad/instances

请求参数

表 6-5 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	token 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**128**

响应参数

状态码： 200

表 6-6 响应 Body 参数

参数	参数类型	描述
count	Integer	实例总数
items	Array of **InstanceInfo** objects	实例

表 6-7 InstanceInfo

参数	参数类型	描述
instance_id	String	实例ID
instance_name	String	实例名
ips	Array of **InstanceIpInfo** objects	实例IP
expire_time	Long	实例过期时间

参数	参数类型	描述
service_bandwidth	Integer	业务带宽
instance_status	Integer	实例状态
enterprise_project_id	String	企业项目ID
overseas_type	Integer	实例类型，0-大陆，1-海外

表 6-8 InstanceIpInfo

参数	参数类型	描述
ip_id	String	IP ID
ip	String	IP
basic_bandwidth	Integer	保底带宽
elastic_bandwidth	Integer	弹性带宽
ip_status	Integer	IP状态

请求示例

无

响应示例

状态码： 200

高防实例列表

```
{
  "count" : 2,
  "items" : [ {
    "ips" : [ {
      "ip" : "103.239.246.6",
      "ip_id" : "4f1XXXX-XXXX-XXXX-XXXX-XXXXd4b1",
      "basic_bandwidth" : 10,
      "elastic_bandwidth" : 20,
      "ip_status" : 0
    } ],
    "instance_id" : "74acXXXX-XXXX-XXXX-XXXX-XXXXd012",
    "instance_name" : "example 1",
    "expire_time" : 1675926787000,
    "service_bandwidth" : 0,
    "instance_status" : 0,
    "enterprise_project_id" : "0"
  }, {
    "ips" : [ {
```

```
    "ip" : "103.239.246.156",
    "ip_id" : "d6f9XXXX-XXXX-XXXX-XXXX-XXXXd4b1",
    "basic_bandwidth" : 20,
    "elastic_bandwidth" : 30,
    "ip_status" : 0
   } ],
   "instance_id" : "6a7fXXXX-XXXX-XXXX-XXXX-XXXXd4a6",
   "instance_name" : "example 2",
   "expire_time" : 1674909266000,
   "service_bandwidth" : 0,
   "instance_status" : 0,
   "enterprise_project_id" : "0"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.aad.v1.region.AadRegion;
import com.huaweicloud.sdk.aad.v1.*;
import com.huaweicloud.sdk.aad.v1.model.*;


public class ListInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new GlobalCredentials()
                .withAk(ak)
                .withSk(sk);

        AadClient client = AadClient.newBuilder()
                .withCredential(auth)
                .withRegion(AadRegion.valueOf("<YOUR REGION>"))
                .build();
        ListInstanceRequest request = new ListInstanceRequest();
        try {
            ListInstanceResponse response = client.listInstance(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
```



```
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkaad.v1.region.aad_region import AadRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkaad.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = GlobalCredentials(ak, sk) \

    client = AadClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AadRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListInstanceRequest()
        response = client.list_instance(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    aad "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := aad.NewAadClient(
        aad.AadClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```
    request := &model.ListInstanceRequest{}
    response, err := client.ListInstance(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	高防实例列表
400	Error

错误码

请参见**错误码**。

6.3 DDoS 高防-转发规则管理

6.3.1 查询高防实例 IP 的转发规则列表

功能介绍

查询高防实例IP的转发规则列表

调用方法

请参见**如何调用API**。

URI

GET /v1/aad/instances/{instance_id}/{ip}/rules

表 6-9 路径参数

参数	是否必选	参数类型	描述
instance_id	是	String	实例Id 最小长度：**16** 最大长度：**64**

参数	是否必选	参数类型	描述
ip	是	String	单个 IP 最小长度：**1** 最大长度：**45**

请求参数

表 6-10 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	token 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**128**

响应参数

状态码： 200

表 6-11 响应 Body 参数

参数	参数类型	描述
total	Integer	转发规则总数
rules	Array of **TransferRuleInfo** objects	转发规则列表

表 6-12 TransferRuleInfo

参数	参数类型	描述
rule_id	String	转发规则名ID
forward_protocol	String	转发协议，tcp或udp

参数	参数类型	描述
forward_port	Integer	转发端口 最小值：**1** 最大值：**65536**
source_port	Integer	源站端口 最小值：**1** 最大值：**65536**
lb_method	String	LVS转发规则
source_ip	String	源站IP，多个IP用逗号隔开，限制20个IP
status	Integer	源站状态 1 正常，2 异常

请求示例

无

响应示例

状态码： 200

高防实例IP的转发规则列表

```
{
  "total" : 2,
  "rules" : [ {
    "rule_id" : "81c8XXXX-XXXX-XXXX-XXXX11c9",
    "forward_protocol" : "tcp",
    "forward_port" : 31,
    "source_port" : 31,
    "lb_method" : "rr",
    "source_ip" : "4.0.1.1",
    "status" : 1
  }, {
    "rule_id" : "ea86XXXX-XXXX-XXXX-XXXXfa13",
    "forward_protocol" : "tcp",
    "forward_port" : 61,
    "source_port" : 61,
    "lb_method" : "rr",
    "source_ip" : "6.0.1.1",
    "status" : 1
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
```



```
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.aad.v1.region.AadRegion;
import com.huaweicloud.sdk.aad.v1.*;
import com.huaweicloud.sdk.aad.v1.model.*;


public class ListInstanceIpRuleSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new GlobalCredentials()
                .withAk(ak)
                .withSk(sk);

        AadClient client = AadClient.newBuilder()
                .withCredential(auth)
                .withRegion(AadRegion.valueOf("<YOUR REGION>"))
                .build();
        ListInstanceIpRuleRequest request = new ListInstanceIpRuleRequest();
        try {
            ListInstanceIpRuleResponse response = client.listInstanceIpRule(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkaad.v1.region.aad_region import AadRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkaad.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = GlobalCredentials(ak, sk) \

    client = AadClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AadRegion.value_of("<YOUR REGION>")) \
        .build()
```

```
    try:
        request = ListInstanceIpRuleRequest()
        response = client.list_instance_ip_rule(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    aad "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := aad.NewAadClient(
        aad.AadClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListInstanceIpRuleRequest{}
    response, err := client.ListInstanceIpRule(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	高防实例IP的转发规则列表
401	Unauthorized

错误码

请参见**错误码**。

6.3.2 修改高防实例转发配置的源站 IP

功能介绍

修改高防实例转发配置的源站IP

调用方法

请参见**如何调用API**。

URI

PUT /v1/aad/instances/{instance_id}/{ip}/rules/{rule_id}

表 6-13 路径参数

参数	是否必选	参数类型	描述
instance_id	是	String	实例Id 最小长度：**16** 最大长度：**64**
ip	是	String	单个 IP 最小长度：**1** 最大长度：**45**
rule_id	是	String	规则ID 最小长度：**16** 最大长度：**64**

请求参数

表 6-14 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	token 最小长度：**32** 最大长度：**2097152**

参数	是否必选	参数类型	描述
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**128**

表 6-15 请求 Body 参数

参数	是否必选	参数类型	描述
forward_protocol	否	String	转发协议，tcp或udp
forward_port	否	Integer	转发端口 最小值：**1** 最大值：**65536**
source_port	否	Integer	源站端口 最小值：**1** 最大值：**65536**
source_ip	否	String	源站IP，多个IP用逗号隔开，限制20个IP 最小值：**1** 最大值：**256**

响应参数

无

请求示例

将指定高防实例IP的转发协议修改为tcp，转发端口和源站端口设置为31，源站IP修改为4.1.0.0。

```
PUT https://{endpoint}/v1/aad/instances/{instance_id}/{ip}/rules/{rule_id}

{
  "forward_protocol" : "tcp",
  "forward_port" : 31,
  "source_port" : 31,
  "source_ip" : "4.1.0.0"
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

将指定高防实例IP的转发协议修改为tcp，转发端口和源站端口设置为31，源站IP修改为4.1.0.0。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.aad.v1.region.AadRegion;
import com.huaweicloud.sdk.aad.v1.*;
import com.huaweicloud.sdk.aad.v1.model.*;


public class UpdateInstanceIpRuleSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new GlobalCredentials()
                .withAk(ak)
                .withSk(sk);

        AadClient client = AadClient.newBuilder()
                .withCredential(auth)
                .withRegion(AadRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateInstanceIpRuleRequest request = new UpdateInstanceIpRuleRequest();
        TransferRuleBody body = new TransferRuleBody();
        body.withSourceIp("4.1.0.0");
        body.withSourcePort(31);
        body.withForwardPort(31);
        body.withForwardProtocol("tcp");
        request.withBody(body);
        try {
            UpdateInstanceIpRuleResponse response = client.updateInstanceIpRule(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```


Python

将指定高防实例IP的转发协议修改为tcp，转发端口和源站端口设置为31，源站IP修改为4.1.0.0。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkaad.v1.region.aad_region import AadRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkaad.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = GlobalCredentials(ak, sk) \

    client = AadClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AadRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateInstanceIpRuleRequest()
        request.body = TransferRuleBody(
            source_ip="4.1.0.0",
            source_port=31,
            forward_port=31,
            forward_protocol="tcp"
        )
        response = client.update_instance_ip_rule(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

将指定高防实例IP的转发协议修改为tcp，转发端口和源站端口设置为31，源站IP修改为4.1.0.0。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    aad "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
```



```
    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := aad.NewAadClient(
        aad.AadClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateInstanceIpRuleRequest{}
    sourceIpTransferRuleBody:= "4.1.0.0"
    sourcePortTransferRuleBody:= int32(31)
    forwardPortTransferRuleBody:= int32(31)
    forwardProtocolTransferRuleBody:= "tcp"
    request.Body = &model.TransferRuleBody{
        SourceIp: &sourceIpTransferRuleBody,
        SourcePort: &sourcePortTransferRuleBody,
        ForwardPort: &forwardPortTransferRuleBody,
        ForwardProtocol: &forwardProtocolTransferRuleBody,
    }
    response, err := client.UpdateInstanceIpRule(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	OK
400	Error

错误码

请参见**错误码**。

6.3.3 批量创建高防实例 IP 的转发规则

功能介绍

批量创建高防实例IP的转发规则

调用方法

请参见**如何调用API**。

URI

POST /v1/aad/instances/{instance_id}/{ip}/rules/batch-create

表 6-16 路径参数

参数	是否必选	参数类型	描述
instance_id	是	String	实例Id 最小长度：**16** 最大长度：**64**
ip	是	String	单个 IP 最小长度：**1** 最大长度：**45**

请求参数

表 6-17 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	token 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**128**

表 6-18 请求 Body 参数

参数	是否必选	参数类型	描述
rules	否	Array of **TransferRuleBody** objects	批量转发规则

表 6-19 TransferRuleBody

参数	是否必选	参数类型	描述
forward_protocol	否	String	转发协议，tcp或udp
forward_port	否	Integer	转发端口 最小值：**1** 最大值：**65536**
source_port	否	Integer	源站端口 最小值：**1** 最大值：**65536**
source_ip	否	String	源站IP，多个IP用逗号隔开，限制20个IP 最小值：**1** 最大值：**256**

响应参数

状态码： 200

表 6-20 响应 Body 参数

参数	参数类型	描述
success_num	Integer	数量

请求示例

为指定高防实例IP创建两个TCP转发协议。

```
POST https://{endpoint}/v1/aad/instances/{instance_id}/{ip}/rules/batch-create

{
  "rules" : [ {
    "forward_protocol" : "tcp",
    "forward_port" : 66,
    "source_port" : 66,
    "source_ip" : "6.6.1.1"
  }, {
    "forward_protocol" : "tcp",
    "forward_port" : 33,
    "source_port" : 33,
    "source_ip" : "3.3.1.1"
  } ]
}
```

响应示例

状态码： 200

批量创建转发规则响应

```
{
  "success_num" : 1
}
```

SDK 代码示例

SDK代码示例如下。

Java

为指定高防实例IP创建两个TCP转发协议。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.aad.v1.region.AadRegion;
import com.huaweicloud.sdk.aad.v1.*;
import com.huaweicloud.sdk.aad.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchCreateInstanceIpRuleSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new GlobalCredentials()
                .withAk(ak)
                .withSk(sk);

        AadClient client = AadClient.newBuilder()
                .withCredential(auth)
                .withRegion(AadRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchCreateInstanceIpRuleRequest request = new BatchCreateInstanceIpRuleRequest();
        BatchTransferRuleBody body = new BatchTransferRuleBody();
        List<TransferRuleBody> listbodyRules = new ArrayList<>();
        listbodyRules.add(
            new TransferRuleBody()
                .withForwardProtocol("tcp")
                .withForwardPort(66)
                .withSourcePort(66)
                .withSourceIp("6.6.1.1")
        );
        listbodyRules.add(
            new TransferRuleBody()
                .withForwardProtocol("tcp")
                .withForwardPort(33)
                .withSourcePort(33)
                .withSourceIp("3.3.1.1")
        );
        body.withRules(listbodyRules);
        request.withBody(body);
        try {
```

```
            BatchCreateInstanceIpRuleResponse response = client.batchCreateInstanceIpRule(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

为指定高防实例IP创建两个TCP转发协议。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkaad.v1.region.aad_region import AadRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkaad.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = GlobalCredentials(ak, sk) \

    client = AadClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AadRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchCreateInstanceIpRuleRequest()
        listRulesbody = [
            TransferRuleBody(
                forward_protocol="tcp",
                forward_port=66,
                source_port=66,
                source_ip="6.6.1.1"
            ),
            TransferRuleBody(
                forward_protocol="tcp",
                forward_port=33,
                source_port=33,
                source_ip="3.3.1.1"
            )
        ]
        request.body = BatchTransferRuleBody(
            rules=listRulesbody
        )
        response = client.batch_create_instance_ip_rule(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

为指定高防实例IP创建两个TCP转发协议。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    aad "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := aad.NewAadClient(
        aad.AadClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateInstanceIpRuleRequest{}
    forwardProtocolRules:= "tcp"
    forwardPortRules:= int32(66)
    sourcePortRules:= int32(66)
    sourceIpRules:= "6.6.1.1"
    forwardProtocolRules1:= "tcp"
    forwardPortRules1:= int32(33)
    sourcePortRules1:= int32(33)
    sourceIpRules1:= "3.3.1.1"
    var listRulesbody = []model.TransferRuleBody{
        {
            ForwardProtocol: &forwardProtocolRules,
            ForwardPort: &forwardPortRules,
            SourcePort: &sourcePortRules,
            SourceIp: &sourceIpRules,
        },
        {
            ForwardProtocol: &forwardProtocolRules1,
            ForwardPort: &forwardPortRules1,
            SourcePort: &sourcePortRules1,
            SourceIp: &sourceIpRules1,
        },
    }
    request.Body = &model.BatchTransferRuleBody{
        Rules: &listRulesbody,
    }
    response, err := client.BatchCreateInstanceIpRule(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	批量创建转发规则响应
400	Error

错误码

请参见**错误码**。

6.3.4 批量删除高防实例 IP 的转发规则

功能介绍

批量删除高防实例IP的转发规则

调用方法

请参见**如何调用API**。

URI

POST /v1/aad/instances/{instance_id}/{ip}/rules/batch-delete

表 6-21 路径参数

参数	是否必选	参数类型	描述
instance_id	是	String	实例Id 最小长度：**16** 最大长度：**64**
ip	是	String	单个 IP 最小长度：**1** 最大长度：**45**

请求参数

表 6-22 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	token 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**128**

表 6-23 请求 Body 参数

参数	是否必选	参数类型	描述
ids	否	Array of strings	批量id

响应参数

状态码： 200

表 6-24 响应 Body 参数

参数	参数类型	描述
success_num	Integer	数量

请求示例

删除指定高防实例IP下，ID为“e6e4a678-XXXX-XXXX-XXXX-a47db59553bf”的转发规则。

```
POST https://{endpoint}/v1/aad/instances/{instance_id}/{ip}/rules/batch-delete

{
  "ids" : [ "e6e4a678-XXXX-XXXX-XXXX-a47db59553bf" ]
}
```

响应示例

状态码： 200

批量删除转发规则响应

```
{
  "success_num" : 1
}
```

SDK 代码示例

SDK代码示例如下。

Java

删除指定高防实例IP下，ID为“e6e4a678-XXXX-XXXX-XXXX-a47db59553bf”的转发规则。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.aad.v1.region.AadRegion;
import com.huaweicloud.sdk.aad.v1.*;
import com.huaweicloud.sdk.aad.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchDeleteInstanceIpRuleSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new GlobalCredentials()
                .withAk(ak)
                .withSk(sk);

        AadClient client = AadClient.newBuilder()
                .withCredential(auth)
                .withRegion(AadRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchDeleteInstanceIpRuleRequest request = new BatchDeleteInstanceIpRuleRequest();
        BatchIdBody body = new BatchIdBody();
        List<String> listbodyIds = new ArrayList<>();
        listbodyIds.add("e6e4a678-XXXX-XXXX-XXXX-a47db59553bf");
        body.withIds(listbodyIds);
        request.withBody(body);
        try {
            BatchDeleteInstanceIpRuleResponse response = client.batchDeleteInstanceIpRule(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

删除指定高防实例IP下，ID为“e6e4a678-XXXX-XXXX-XXXX-a47db59553bf”的转发规则。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkaad.v1.region.aad_region import AadRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkaad.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = GlobalCredentials(ak, sk) \

    client = AadClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AadRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchDeleteInstanceIpRuleRequest()
        listIdsbody = [
            "e6e4a678-XXXX-XXXX-XXXX-a47db59553bf"
        ]
        request.body = BatchIdBody(
            ids=listIdsbody
        )
        response = client.batch_delete_instance_ip_rule(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

删除指定高防实例IP下，ID为“e6e4a678-XXXX-XXXX-XXXX-a47db59553bf”的转发规则。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    aad "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
```

```
    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := aad.NewAadClient(
        aad.AadClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteInstanceIpRuleRequest{}
    var listIdsbody = []string{
        "e6e4a678-XXXX-XXXX-XXXX-a47db59553bf",
    }
    request.Body = &model.BatchIdBody{
        Ids: &listIdsbody,
    }
    response, err := client.BatchDeleteInstanceIpRule(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	批量删除转发规则响应
400	Error

错误码

请参见**错误码**。

6.4 DDoS 高防-域名管理

6.4.1 查询域名列表

功能介绍

查询域名列表

调用方法

请参见**如何调用API**。

URI

GET /v1/aad/protected-domains

请求参数

表 6-25 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	token 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**128**

响应参数

状态码： 200

表 6-26 响应 Body 参数

参数	参数类型	描述
count	Integer	域名总数
items	Array of **DomainInfo** objects	域名列表

表 6-27 DomainInfo

参数	参数类型	描述
domain_id	String	域名ID
domain_name	String	域名
cname	String	域名cname
protocol	Array of strings	域名协议
real_server_type	Integer	源站类型

参数	参数类型	描述
real_servers	String	源站
waf_status	Integer	waf防护状态
enterprise_project_id	String	企业项目ID

请求示例

无

响应示例

状态码： 200

Domain Information

```
{
  "count" : 2,
  "items" : [ {
    "cname" : "8bacXXXX73fd.huaweisafedns.cn",
    "protocol" : [ "HTTP" ],
    "domain_id" : "0d3eXXXX6c42",
    "domain_name" : "www.test2.com",
    "waf_status" : 1,
    "real_server_type" : 1,
    "real_servers" : "*.test2.com",
    "enterprise_project_id" : "0"
  }, {
    "cname" : "1604XXXXc833.huaweisafedns.cn",
    "protocol" : [ "HTTPS", "HTTP" ],
    "domain_id" : "a115XXXXfafe",
    "domain_name" : "test1.com",
    "waf_status" : 1,
    "real_server_type" : 1,
    "real_servers" : "test1.com",
    "enterprise_project_id" : "51bcXXXX-XXXX-XXXX-XXXX-XXXXa06b"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.aad.v1.region.AadRegion;
import com.huaweicloud.sdk.aad.v1.*;
import com.huaweicloud.sdk.aad.v1.model.*;


public class ListDomainSolution {
```



```
    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new GlobalCredentials()
                .withAk(ak)
                .withSk(sk);

        AadClient client = AadClient.newBuilder()
                .withCredential(auth)
                .withRegion(AadRegion.valueOf("<YOUR REGION>"))
                .build();
        ListDomainRequest request = new ListDomainRequest();
        try {
            ListDomainResponse response = client.listDomain(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkaad.v1.region.aad_region import AadRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkaad.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = GlobalCredentials(ak, sk) \

    client = AadClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AadRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListDomainRequest()
        response = client.list_domain(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

```
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    aad "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := aad.NewAadClient(
        aad.AadClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDomainRequest{}
    response, err := client.ListDomain(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	Domain Information

错误码

请参见**错误码**。

6.4.2 查询域名关联的实例 ID

功能介绍

查询域名关联的实例ID

调用方法

请参见**如何调用API**。

URI

GET /v1/aad/protected-domains/{domain_id}

表 6-28 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	域名ID

请求参数

表 6-29 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	token 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**128**

响应参数

状态码： 200

表 6-30 响应 Body 参数

参数	参数类型	描述
instance_ids	Array of strings	实例ID列表

请求示例

无

响应示例

状态码： 200

实例ID列表

```
{
  "instance_ids" : [ "c81cXXXX-XXXX-XXXX-XXXX-XXXXb984" ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.aad.v1.region.AadRegion;
import com.huaweicloud.sdk.aad.v1.*;
import com.huaweicloud.sdk.aad.v1.model.*;


public class ListInstanceIdSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new GlobalCredentials()
                .withAk(ak)
                .withSk(sk);

        AadClient client = AadClient.newBuilder()
                .withCredential(auth)
                .withRegion(AadRegion.valueOf("<YOUR REGION>"))
                .build();
        ListInstanceIdRequest request = new ListInstanceIdRequest();
        try {
            ListInstanceIdResponse response = client.listInstanceId(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
```



```
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkaad.v1.region.aad_region import AadRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkaad.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = GlobalCredentials(ak, sk) \

    client = AadClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AadRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListInstanceIdRequest()
        response = client.list_instance_id(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    aad "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := aad.NewAadClient(
        aad.AadClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```
    request := &model.ListInstanceIdRequest{}
    response, err := client.ListInstanceId(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	实例ID列表
400	Error

错误码

请参见**错误码**。

6.4.3 更新域名信息

功能介绍

更新域名源站配置信息

调用方法

请参见**如何调用API**。

URI

PUT /v1/aad/protected-domains/{domain_id}

表 6-31 路径参数

参数	是否必选	参数类型	描述
domain_id	是	String	域名ID 最小长度：**16** 最大长度：**64**

请求参数

表 6-32 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	token 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**128**

表 6-33 请求 Body 参数

参数	是否必选	参数类型	描述
real_server_type	否	Integer	源站类型 0：源站IP 1：源站域名
real_servers	否	String	源站

响应参数

无

请求示例

修改指定域名ID的源站类型为0-源站IP，源站为3.3.3.3,1.1.1.1

```
PUT https://{endpoint}/v1/aad/protected-domains/{domain_id}

{
  "real_servers" : "3.3.3.3,1.1.1.1",
  "real_server_type" : 0
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

修改指定域名ID的源站类型为0-源站IP，源站为3.3.3.3,1.1.1.1

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.aad.v1.region.AadRegion;
import com.huaweicloud.sdk.aad.v1.*;
import com.huaweicloud.sdk.aad.v1.model.*;


public class UpdateDomainSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new GlobalCredentials()
                .withAk(ak)
                .withSk(sk);

        AadClient client = AadClient.newBuilder()
                .withCredential(auth)
                .withRegion(AadRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateDomainRequest request = new UpdateDomainRequest();
        DomainRealServerInfo body = new DomainRealServerInfo();
        body.withRealServers("3.3.3.3,1.1.1.1");
        body.withRealServerType(0);
        request.withBody(body);
        try {
            UpdateDomainResponse response = client.updateDomain(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

修改指定域名ID的源站类型为0-源站IP，源站为3.3.3.3,1.1.1.1

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkaad.v1.region.aad_region import AadRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkaad.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
```



```
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = GlobalCredentials(ak, sk) \

    client = AadClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AadRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateDomainRequest()
        request.body = DomainRealServerInfo(
            real_servers="3.3.3.3,1.1.1.1",
            real_server_type=0
        )
        response = client.update_domain(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

修改指定域名ID的源站类型为0-源站IP，源站为3.3.3.3,1.1.1.1

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    aad "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := aad.NewAadClient(
        aad.AadClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateDomainRequest{}
    realServersDomainRealServerInfo:= "3.3.3.3,1.1.1.1"
    realServerTypeDomainRealServerInfo:= int32(0)
    request.Body = &model.DomainRealServerInfo{
        RealServers: &realServersDomainRealServerInfo,
        RealServerType: &realServerTypeDomainRealServerInfo,
    }
    response, err := client.UpdateDomain(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
```

```
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	OK

错误码

请参见**错误码**。

6.4.4 创建防护域名

功能介绍

创建防护域名

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/aad/external/domains

表 6-34 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，对应华为云控制台用户名->我的凭证->项目列表->项目ID

请求参数

表 6-35 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**128**

表 6-36 请求 Body 参数

参数	是否必选	参数类型	描述
domain_name	是	String	域名
enterprise_project_id	是	String	企业项目id，与接入的高防实例所属企业项目保持一致。可在华为云EPS服务中查看企业项目id，default企业项目id为"0"。
vips	是	Array of strings	高防实例ip列表。多个高防实例ip必须属于同一企业项目。
real_server_type	是	Integer	源站类型。 0 - 源站IP， 1 - 源站域名。
port_http	否	Array of integers	HTTP端口，与port_https不能同时为空。DDoS高防支持的HTTP端口可在控制台查看。
port_https	否	Array of integers	HTTPS端口，与port_http不能同时为空。DDoS高防支持的HTTPS端口可在控制台查看。
real_server	是	String	源站ip/源站域名

响应参数

状态码： 200

表 6-37 响应 Body 参数

参数	参数类型	描述
cname	String	高防提供的CNAME地址
domainId	String	域名id

请求示例

无

响应示例

状态码： 200

HostInfo

```
{
  "domainId" : "81c8xxxxxxxxxxxxxxx11c9",
  "cname" : "29xxxxxxxxxxx1e.huaweisafedns.cn"
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.aad.v1.region.AadRegion;
import com.huaweicloud.sdk.aad.v1.*;
import com.huaweicloud.sdk.aad.v1.model.*;


public class CreateAadDomainSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new GlobalCredentials()
                .withAk(ak)
                .withSk(sk);

        AadClient client = AadClient.newBuilder()
                .withCredential(auth)
                .withRegion(AadRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateAadDomainRequest request = new CreateAadDomainRequest();
        HostBody body = new HostBody();
```



```
        request.withBody(body);
        try {
            CreateAadDomainResponse response = client.createAadDomain(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkaad.v1.region.aad_region import AadRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkaad.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = GlobalCredentials(ak, sk) \

    client = AadClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AadRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateAadDomainRequest()
        request.body = HostBody(
        )
        response = client.create_aad_domain(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    aad "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```

```
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := aad.NewAadClient(
        aad.AadClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateAadDomainRequest{}
    request.Body = &model.HostBody{
    }
    response, err := client.CreateAadDomain(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	HostInfo
400	Error

错误码

请参见**错误码**。

6.4.5 上传/修改域名对应证书

功能介绍

上传/修改域名对应证书

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/aad/external/domains/certificate

表 6-38 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，对应华为云控制台用户名->我的凭证->项目列表->项目ID

请求参数

表 6-39 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**128**

表 6-40 请求 Body 参数

参数	是否必选	参数类型	描述
op_type	是	Integer	操作类型。0 - 上传， 1 - 更换
cert_name	是	String	证书名称
cert_file	否	String	证书文件。上传新证书(op_type为0)时必填，更换为已有证书(op_type为1)时置空
cert_key_file	否	String	私钥文件。上传新证书(op_type为0)时必填，更换为已有证书(op_type为1)时置空
domain_id	是	String	域名id

响应参数

无

请求示例

无

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.aad.v1.region.AadRegion;
import com.huaweicloud.sdk.aad.v1.*;
import com.huaweicloud.sdk.aad.v1.model.*;


public class CreateCertificateSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new GlobalCredentials()
                .withAk(ak)
                .withSk(sk);

        AadClient client = AadClient.newBuilder()
                .withCredential(auth)
                .withRegion(AadRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateCertificateRequest request = new CreateCertificateRequest();
        CertificateBody body = new CertificateBody();
        request.withBody(body);
        try {
            CreateCertificateResponse response = client.createCertificate(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
```



```
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkaad.v1.region.aad_region import AadRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkaad.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = GlobalCredentials(ak, sk) \

    client = AadClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AadRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateCertificateRequest()
        request.body = CertificateBody(
        )
        response = client.create_certificate(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    aad "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := aad.NewAadClient(
        aad.AadClientBuilder().
```

```
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

    request := &model.CreateCertificateRequest{}
    request.Body = &model.CertificateBody{
    }
    response, err := client.CreateCertificate(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	OK
400	Error

错误码

请参见**错误码**。

6.4.6 修改域名 WEB 基础防护开关/CC 防护开关

功能介绍

修改域名WEB基础防护开关/CC防护开关

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/aad/external/domains/switch

表 6-41 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，对应华为云控制台用户名->我的凭证->项目列表->项目ID

请求参数

表 6-42 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**128**

表 6-43 请求 Body 参数

参数	是否必选	参数类型	描述
domain_id	是	String	域名id
waf_switch	是	Integer	是否开启WEB基础防护开关。0 - 开启， 1 - 关闭
cc_switch	是	Integer	是否开启CC防护开关。0 - 开启， 1 - 关闭。开启CC开关必须同时开启WEB基础防护开关

响应参数

无

请求示例

无

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.aad.v1.region.AadRegion;
import com.huaweicloud.sdk.aad.v1.*;
import com.huaweicloud.sdk.aad.v1.model.*;


public class ModifyDomainWebSwitchSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new GlobalCredentials()
                .withAk(ak)
                .withSk(sk);

        AadClient client = AadClient.newBuilder()
                .withCredential(auth)
                .withRegion(AadRegion.valueOf("<YOUR REGION>"))
                .build();
        ModifyDomainWebSwitchRequest request = new ModifyDomainWebSwitchRequest();
        CadDomainSwitchRequest body = new CadDomainSwitchRequest();
        request.withBody(body);
        try {
            ModifyDomainWebSwitchResponse response = client.modifyDomainWebSwitch(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkaad.v1.region.aad_region import AadRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkaad.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
```

```
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = GlobalCredentials(ak, sk) \

    client = AadClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AadRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ModifyDomainWebSwitchRequest()
        request.body = CadDomainSwitchRequest(
        )
        response = client.modify_domain_web_switch(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    aad "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := aad.NewAadClient(
        aad.AadClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ModifyDomainWebSwitchRequest{}
    request.Body = &model.CadDomainSwitchRequest{
    }
    response, err := client.ModifyDomainWebSwitch(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	OK
400	Error

错误码

请参见**错误码**。

6.4.7 查询高防回源 IP 段列表

功能介绍

查询高防回源IP段列表

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/aad/external/source-ip

表 6-44 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，对应华为云控制台用户名->我的凭证->项目列表->项目ID

请求参数

表 6-45 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**

参数	是否必选	参数类型	描述
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**128**

响应参数

状态码： 200

表 6-46 响应 Body 参数

参数	参数类型	描述
ips	Array of strings	查询高防回源IP段列表

请求示例

无

响应示例

状态码： 200

SourceIpList

```
{
  "ips" : [ "1.1.1.1", "2.2.2.2" ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.aad.v1.region.AadRegion;
import com.huaweicloud.sdk.aad.v1.*;
import com.huaweicloud.sdk.aad.v1.model.*;


public class ListSourceIpsSolution {

    public static void main(String[] args) {
```

```
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new GlobalCredentials()
                .withAk(ak)
                .withSk(sk);

        AadClient client = AadClient.newBuilder()
                .withCredential(auth)
                .withRegion(AadRegion.valueOf("<YOUR REGION>"))
                .build();
        ListSourceIpsRequest request = new ListSourceIpsRequest();
        try {
            ListSourceIpsResponse response = client.listSourceIps(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkaad.v1.region.aad_region import AadRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkaad.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = GlobalCredentials(ak, sk) \

    client = AadClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AadRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListSourceIpsRequest()
        response = client.list_source_ips(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    aad "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := aad.NewAadClient(
        aad.AadClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSourceIpsRequest{}
    response, err := client.ListSourceIps(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	SourceIpList
400	Error

错误码

请参见**错误码**。

6.5 DDoS 高防-防护策略

6.5.1 高防实例添加黑白名单

功能介绍

高防实例添加黑白名单

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/aad/external/bwlist

表 6-47 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，对应华为云控制台用户名->我的凭证->项目列表->项目ID

请求参数

表 6-48 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**128**

表 6-49 请求 Body 参数

参数	是否必选	参数类型	描述
instance_id	是	String	实例id

参数	是否必选	参数类型	描述
type	是	String	规则类型。black - 黑名单，white - 白名单
ips	是	Array of strings	ip列表

响应参数

无

请求示例

无

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.aad.v1.region.AadRegion;
import com.huaweicloud.sdk.aad.v1.*;
import com.huaweicloud.sdk.aad.v1.model.*;


public class AddBlackWhiteIpListSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new GlobalCredentials()
                .withAk(ak)
                .withSk(sk);

        AadClient client = AadClient.newBuilder()
                .withCredential(auth)
                .withRegion(AadRegion.valueOf("<YOUR REGION>"))
                .build();
        AddBlackWhiteIpListRequest request = new AddBlackWhiteIpListRequest();
        BlackWhiteIpListRequest body = new BlackWhiteIpListRequest();
        request.withBody(body);
```



```
        try {
            AddBlackWhiteIpListResponse response = client.addBlackWhiteIpList(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkaad.v1.region.aad_region import AadRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkaad.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = GlobalCredentials(ak, sk) \

    client = AadClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AadRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = AddBlackWhiteIpListRequest()
        request.body = BlackWhiteIpListRequest(
        )
        response = client.add_black_white_ip_list(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    aad "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```

```
variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := aad.NewAadClient(
        aad.AadClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddBlackWhiteIpListRequest{}
    request.Body = &model.BlackWhiteIpListRequest{
    }
    response, err := client.AddBlackWhiteIpList(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	ok
400	Error

错误码

请参见**错误码**。

6.5.2 高防实例删除黑白名单

功能介绍

高防实例删除黑白名单

调用方法

请参见**如何调用API**。

URI

DELETE /v1/{project_id}/aad/external/bwlist

表 6-50 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，对应华为云控制台用户名->我的凭证->项目列表->项目ID

请求参数

表 6-51 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**32** 最大长度：**2097152**
Content-Type	是	String	Content-Type 缺省值：**application/json;charset=utf8** 最小长度：**1** 最大长度：**128**

表 6-52 请求 Body 参数

参数	是否必选	参数类型	描述
instance_id	是	String	实例id
type	是	String	规则类型。black - 黑名单，white - 白名单
ips	是	Array of strings	ip列表

响应参数

无

请求示例

无

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.GlobalCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.aad.v1.region.AadRegion;
import com.huaweicloud.sdk.aad.v1.*;
import com.huaweicloud.sdk.aad.v1.model.*;


public class DeleteBlackWhiteIpListSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new GlobalCredentials()
                .withAk(ak)
                .withSk(sk);

        AadClient client = AadClient.newBuilder()
                .withCredential(auth)
                .withRegion(AadRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteBlackWhiteIpListRequest request = new DeleteBlackWhiteIpListRequest();
        BlackWhiteIpListRequest body = new BlackWhiteIpListRequest();
        request.withBody(body);
        try {
            DeleteBlackWhiteIpListResponse response = client.deleteBlackWhiteIpList(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkaad.v1.region.aad_region import AadRegion
```



```
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkaad.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = GlobalCredentials(ak, sk) \

    client = AadClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(AadRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteBlackWhiteIpListRequest()
        request.body = BlackWhiteIpListRequest(
        )
        response = client.delete_black_white_ip_list(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/global"
    aad "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/aad/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := global.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := aad.NewAadClient(
        aad.AadClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteBlackWhiteIpListRequest{}
    request.Body = &model.BlackWhiteIpListRequest{
    }
    response, err := client.DeleteBlackWhiteIpList(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
```

```
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	ok
400	Error

错误码

请参见**错误码**。

7 DDoS 原生基础防护应用示例

7.1 示例 1：更新 DDoS 防护

场景描述

当您需要更新某个IP的防护策略时，您可以通过Anti-DDoS界面在线修改，或者调用Anti-DDoS的API接口修改。

流程如下：

1. 用户先查询名下所有主机的配置监控状态。
2. 用户再查询Anti-DDoS服务的配置策略列表。
3. 用户根据配置参数，修改IP的防护策略为目标策略。
4. 用户根据3返回的任务ID，查询更新防护DDoS防护策略任务的执行情况。

涉及接口

更新DDoS防护时，涉及的接口如下：

- **查询EIP防护状态列表**：批量查询用户所有主机的配置监控状态。
- **查询Anti-DDoS配置可选范围**：查询AntiDDOS服务的配置策略列表。
- **更新Anti-DDoS服务**：更新IP的AntiDDoS安全服务配置策略。
- **查询Anti-DDoS任务**：根据任务ID获取任务状态。

操作步骤

步骤1 批量查询用户所有主机的配置监控状态。

- 接口相关信息
  URI格式：GET /v1/{project_id}/antiddos
  详情请参见“**查询EIP防护状态列表**”。
- 请求示例
  GET: https://*{endpoint}*/v1/1858a4e1f99d4454bd6a539d5477f5de/antiddos
  {endpoint}信息请从**地区和终端节点**获取。

Body：

```
{
}
```

- 响应示例

```
{
"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
}
]
}
```

步骤2 查询AntiDDOS服务的配置策略列表。

- 接口相关信息

  URI格式：GET /v2/{project_id}/antiddos/query-config-list

  详情请参见“**查询Anti-DDoS配置可选范围**”。

- 请求示例

  GET: https://*{endpoint}*/v2/1858a4e1f99d4454bd6a539d5477f5de/antiddos/query-config-list

  {endpoint}信息请从**地区和终端节点**获取。

  Body：

```
{
}
```

- 响应示例

```
{
"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
},
{
"traffic_pos_id": 88,
"traffic_per_second": 1000,
"packet_per_second": 300000
}
],
"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": []
}
```


步骤3 更新IP的AntiDDoS安全服务配置策略。

- 接口相关信息

  URI格式：PUT /v1/{project_id}/antiddos/{floating_ip_id}

  详情请参见“**更新Anti-DDoS服务**”。

- 请求示例

PUT: https://*{endpoint}*/v1/1858a4e1f99d4454bd6a539d5477f5de/antiddos/18e6ace5-eb36-4196-a15e-1e000c24e026

{endpoint}信息请从**地区和终端节点**获取。

Body：

```
{
"app_type_id": 1,
"cleaning_access_pos_id": 8,
"enable_L7": false,
"http_request_pos_id": 8,
"traffic_pos_id": 8
}
```

- 响应示例

```
{
"error_code": "10000000",
"error_msg": "The task has been received and is being handled",
"task_id": "59385d2a-6266-4d3a-9122-a228c530f557"
}
```

步骤4 根据**步骤3**返回的任务ID获取任务状态。

- 接口相关信息

  URI格式：GET /v2/{project_id}/query-task-status

  详情请参见“**查询Anti-DDoS任务**”。

- 请求示例

  GET: https://*{endpoint}*/v2/1858a4e1f99d4454bd6a539d5477f5de/query-task-status?task_id=59385d2a-6266-4d3a-9122-a228c530f557

  {endpoint}信息请从**地区和终端节点**获取。

  Body：

```
{
}
```

- 响应示例

```
{
"task_status": "success",
"task_msg": ""
}
```

----结束

7.2 示例 2：配置默认防护策略

场景描述

当您需要配置默认防护策略时，您可以通过Anti-DDoS界面在线配置，或者调用Anti-DDoS的API接口配置。

流程如下：

1. 用户先查询默认防护策略。
2. 用户再配置默认防护策略。

涉及接口

配置默认防护策略时，涉及的接口如下：

- **查询Anti-DDoS默认防护策略**
- **配置Anti-DDoS默认防护策略**

操作步骤

步骤1 查询默认防护策略。

- 接口相关信息

  URI格式：GET /v1/{project_id}/antiddos/default-config

  详情请参见“**查询Anti-DDoS默认防护策略**”。

- 请求示例

  GET: https://*{endpoint}*/v1/1858a4e1f99d4454bd6a539d5477f5de/antiddos/default/config

  {endpoint}信息请从**地区和终端节点**获取。

  Body：

  ```
  {
  }
  ```

- 响应示例

  ```
  {
  "app_type_id": 1,
  "cleaning_access_pos_id": 8,
  "enable_L7": false,
  "http_request_pos_id": 8,
  "traffic_pos_id": 8
  }
  ```

步骤2 配置默认防护策略。

- 接口相关信息

  URI格式：POST /v1/{project_id}/antiddos/default-config

  详情请参见“**配置Anti-DDoS默认防护策略**”。

- 请求示例

  POST: https://*{endpoint}*/v1/1858a4e1f99d4454bd6a539d5477f5de/antiddos/default-config

  {endpoint}信息请从**地区和终端节点**获取。

  Body：

  ```
  {
  "app_type_id": 1,
  "cleaning_access_pos_id": 8,
  "enable_L7": false,
  "http_request_pos_id": 8,
  "traffic_pos_id": 8
  }
  ```

- 响应示例

  ```
  {
  }
  ```

----结束

A 附录

A.1 DDoS 原生基础防护状态码

- 正常

返回值	说明
200	请求成功。

- 异常

状态码	编码	说明
400	Bad Request	服务器未能处理请求。
401	Unauthorized	被请求的页面需要用户名和密码。
403	Forbidden	对被请求页面的访问被禁止。
404	Not Found	服务器无法找到被请求的页面。
405	Method Not Allowed	请求中指定的方法不被允许。
406	Not Acceptable	服务器生成的响应无法被客户端所接受。
407	Proxy Authentication Required	用户必须首先使用代理服务器进行验证，这样请求才会被处理。
408	Request Timeout	请求超出了服务器的等待时间。
409	Conflict	由于冲突，请求无法被完成。
500	Internal Server Error	请求未完成，服务异常。
501	Not Implemented	请求未完成，服务器不支持所请求的功能。

状态码	编码	说明
502	Bad Gateway	请求未完成，服务器从上游服务器收到一个无效的响应。
503	Service Unavailable	请求未完成，系统暂时异常。
504	Gateway Timeout	网关超时。

A.2 Anti-DDoS 错误码

更多服务错误码请参见**API错误中心**。

状态码	错误码	错误信息	描述	处理措施
200	Anti-DDoS.0	Succeeded	成功	无需处理
200	Anti-DDoS.10000000	The task has been received and is being handled	任务已接收，正在处理	无需处理
400	Anti-DDoS.10000001	Enter a valid request message	请求消息格式非法	检查参数
400	Anti-DDoS.10001008	An incorrect task ID is used	不正确的任务ID	检查参数
400	Anti-DDoS.10001010	Invalid time	时间非法	检查参数
401	Anti-DDoS.10000004	Public test service denied	公测服务被拒绝	申请公测
403	Anti-DDoS.10000002	Failed to authenticate the token in the request	请求所带的Token认证失败	重新申请Token
403	Anti-DDoS.10000009	The account is restricted	账户受限	申请权限
403	Anti-DDoS.10000010	The account is frozen	账户冻结	申请解冻
403	Anti-DDoS.10000012	Unknown user type	未知用户类型	申请权限

状态码	错误码	错误信息	描述	处理措施
403	Anti-DDoS.10000016	VPC access failed or EIP is not exist	访问VPC平台异常或EIP不存在	联系管理员
403	Anti-DDoS.10000030	You have not been authenticated. Perform real-name authentication first.	您尚未认证，请先进行实名认证。	实名认证
403	Anti-DDoS.10001009	The operation permission is restricted	操作权限受限	申请权限
403	Anti-DDoS.11000001	Access to the database is rejected	数据库访问被拒绝	联系管理员
500	Anti-DDoS.11000000	Internal system exception. Contact technical support engineers	系统内部异常，请联系技术支持人员	联系管理员

A.3 CNAD 错误码

更多服务错误码请参见**API错误中心**。

状态码	错误码	错误信息	描述	处理措施
400	CNAD.00001000	Invalid IP Address.	IP地址不合法。	检查参数
400	CNAD.00001001	Invalid SMN Topic.	告警通知主题不合法。	检查参数
400	CNAD.00001002	Package Name Already Exists.	防护包名已存在。	检查参数
400	CNAD.00001003	IP: {0} is in another package.	IP: {0} 已经在其他防护包里。	检查参数
400	CNAD.00001004	IP: {0} has policy associated.	IP: {0} 已经绑定了其他策略。	检查参数

状态码	错误码	错误信息	描述	处理措施
400	CNAD.00001005	Policy Name Already Exists.	策略名重复。	检查参数
400	CNAD.00001006	Duplicate IP.	IP重复。	检查参数
400	CNAD.00001007	Duplicate CleanThreshold.	清洗阈值挡位重复。	检查参数
400	CNAD.00001008	Request Illegal.	请求非法。	检查参数
400	CNAD.00001009	The IPs in list don't belong to the same region or aren't multicast IPs	ip列表不属于同一region或者不是代播ip	检查参数
400	CNAD.00004000	Resource is Being Used.	资源正在被使用。	稍后重试
400	CNAD.00004001	IP is Being Used.	防护IP正在被使用。	检查参数
400	CNAD.00004003	Application Has Existed.	您提交的申请已存在。	无需处理
400	CNAD.00004004	Application is Being Approved.	您之前提交的申请正在审批中，请稍后再试！	无需处理
400	CNAD.00004005	Application Denied.	您提交的申请已被驳回！	联系管理员
400	CNAD.00004006	Application Approved.	您提交的申请已通过，请勿重复提交！	无需处理
400	CNAD.00005000	Package Unavailable.	防护包不可用。	检查参数
400	CNAD.00005001	Quota Not Enough.	配额不足。	联系管理员
400	CNAD.00005004	The remaining traffic cleaning threshold of the instance is {0} Mbps.	该实例增加流量清洗阈值剩余额度为：{0}Mbps	无需处理

状态码	错误码	错误信息	描述	处理措施
400	CNAD.00005006	Resource Not Purchased.	未订购CNAD服务。	订购CNAD服务
400	CNAD.00005007	Cannot create policy.	不能创建策略	检查参数
401	CNAD.00002000	User Role Not Permitted.	没有权限，拒绝访问。	申请权限
403	CNAD.00005002	5 change times has reached the maximum in a month.	1个月内可设置防护对象的次数（5次）已满。	等待次数刷新
403	CNAD.00005003	1 change times has reached the maximum in a day.	1天内可设置防护对象的次数（1次）已满。	等待次数刷新
403	CNAD.00005005	Instances of the non-VIC edition cannot be manually deleted.	不支持手动删除非大客户版实例。	检查参数
404	CNAD.00003000	Resource Not Found.	资源不存在。	检查参数
404	CNAD.00003001	Package Not Found.	防护包不存在。	检查参数
404	CNAD.00003002	Policy Not Found.	策略不存在。	检查参数
404	CNAD.00003003	IP Not Found.	防护IP不存在。	检查参数
404	CNAD.00003004	IP Not in Package.	防护IP不属于该防护包。	检查参数
404	CNAD.00003005	Policy Not in Package.	防护策略不属于该防护包。	检查参数
404	CNAD.00003006	Application Not Found.	申请不存在。	检查参数
404	CNAD.00003007	Tenant Not Found.	租户不存在。	检查参数
404	CNAD.00003008	Clean Policy Not Found.	清洗策略不存在。	检查参数

状态码	错误码	错误信息	描述	处理措施
500	CNAD.00000001	Dependent Service Error.	依赖的其他服务错误。	联系管理员
400	TRAFFICHUB.10000400	Param invalid	参数无效	检查参数
400	TRAFFICHUB.10060001	Insufficient unblocking quota, The current unblocking quota is 0	解封配额不足,今日解封配额已用完或本月剩余解封配额为0	等待配额刷新
400	TRAFFICHUB.10060002	Unblocking is not allowed because the tenant risk is too high	不允许解封,因为租户风险等级过高	联系管理员
400	TRAFFICHUB.10060003	Unblocking is not allowed because the tenant level is insufficient	不允许解封,因为租户等级过低	联系管理员
400	TRAFFICHUB.10060005	Unblocking is not allowed because unblock the ip address need to exceed the interval time, see the unblocking policy	IP {0} 不允许解封,可手动点击解封的时间为 {1} 之后,解封间隔计算方法详见上方的自助解封策略	稍后重试
400	TRAFFICHUB.10060006	Unblocking is not allowed because unblock the ip address firstly need to exceed the interval time	不允许解封，因为解封ip地址需要超过间隔时间	稍后重试
400	TRAFFICHUB.10060007	Unblocking is not allowed because blocking ip address list is empty	不允许解封,因为封堵ip列表为空	刷新重试或者联系管理员

状态码	错误码	错误信息	描述	处理措施
400	TRAFFICHUB.10060008	Unblocking is not allowed because ip address is not blocked	不允许解封,因为IP {0} 未被封堵	刷新重试或者联系管理员
401	TRAFFICHUB.10010001	Auth failed	身份验证失败	申请权限
404	TRAFFICHUB.10060004	Tenant config is not found	未找到租户配置	检查参数

A.4 获取项目 ID

调用 API 获取项目 ID

项目ID可以通过调用**查询指定条件下的项目信息**API获取。

获取项目ID的接口为“GET https://{Endpoint}/v3/projects”，其中{Endpoint}为IAM的终端节点，可以从**地区和终端节点**获取。接口的认证鉴权请参见**认证鉴权**。

响应示例如下，其中projects下的“id”即为项目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"
    }
}
```

从控制台获取项目 ID

在调用接口的时候，部分URL中需要填入项目编号，所以需要获取到项目编号。项目编号获取步骤如下：

1. 登录管理控制台。
2. 单击用户名，在下拉列表中单击“我的凭证”。
3. 在“API凭证”页面的项目列表中查看项目ID。

图 A-1 查看项目 ID

B DDoS 原生基础防护历史 API

B.1 查询 Anti-DDoS 配置可选范围

说明

当前页面API为历史版本API，未来可能停止维护。请使用**查询Anti-DDoS配置可选范围**。

功能介绍

查询系统支持的Anti-DDoS防护策略配置的可选范围，用户根据范围列表选择适合自己业务的防护策略进行Anti-DDoS流量清洗。

URI

- URI格式

  GET /v1/{project_id}/antiddos/query_config_list

- 参数说明

参数	是否必选	类型	说明
project_id	是	String	用户的ID

请求

请求样例

```
GET /v1/67641fe6886f43fcb78edbbf0ad0b99f/antiddos/query_config_list
```

响应

- 要素说明

参数	是否必选	类型	说明
traffic_limited_list	是	列表数据结构	流量限制列表
http_limited_list	是	列表数据结构	HTTP限制列表

参数	是否必选	类型	说明
connection_limited_list	是	列表数据结构	连接数限制列表

- traffic_limited_list字段数据结构说明

参数	是否必选	类型	说明
traffic_pos_id	是	Integer	流量分段ID
traffic_per_second	是	Integer	每秒流量（Mbit/s）阈值
packet_per_second	是	Integer	每秒报文数（个/s）阈值

- http_limited_list字段数据结构说明

参数	是否必选	类型	说明
http_request_pos_id	是	Integer	HTTP请求数分段ID
http_packet_per_second	是	Integer	每秒HTTP请求数（个/s）阈值

- connection_limited_list字段数据结构说明

参数	是否必选	类型	说明
cleaning_access_pos_id	是	Integer	清洗时访问限制分段ID
new_connection_limited	是	Integer	单一源IP新建连接个数
total_connection_limited	是	Integer	单一源IP连接数总个数

- 响应样例

```
{
  "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
   }
  ]
}
```


说明

“extend_ddos_config”字段显示用户根据实际需求设置的Anti-DDoS防护策略信息。

返回值

请参考**DDoS原生基础防护状态码**。

B.2 查询 Anti-DDoS 任务

说明

当前页面API为历史版本API，未来可能停止维护。请使用**查询Anti-DDoS任务**。

功能介绍

用户查询指定的Anti-DDoS防护配置任务，得到任务当前执行的状态。

URI

- URI格式

  GET /v1/{project_id}/query_task_status

  说明

  可以在URI后面用“?”和“&”添加不同的查询条件组合，请参考请求样例。

- 参数说明

参数	是否必选	类型	说明
project_id	是	String	用户的ID

请求

- 参数说明

参数	是否必选	类型	说明
task_id	是	String	任务ID（非负整数）的字符串

- 请求样例

```
GET /v1/67641fe6886f43fcb78edbbf0ad0b99f/query_task_status?
task_id=4a4fefe7-34a1-40e2-a87c-16932af3ac4a
```

响应

- 要素说明

名称	类型	说明
task_status	String	任务状态，有以下几种： • success • failed • waiting • running • preprocess • ready
task_msg	String	任务的附加信息

- 响应样例

```
{
  "task_status": "running",
  "task_msg": ""
}
```

返回值

请参考**DDoS原生基础防护状态码**。

C 修订记录

发布日期	修改说明
2023-01-08	第五次正式发布。 • 新增**上传/修改域名对应证书**接口。 • 新增**修改域名WEB基础防护开关/CC防护开关**接口。 • 新增**查询高防回源IP段列表**接口。 • 新增**高防实例添加黑白名单**接口。 • 新增**高防实例删除黑白名单**接口。
2022-06-17	第四次正式发布。 修改参数： • **查询Anti-DDoS任务** • **更新Anti-DDoS服务** • **配置Anti-DDoS默认防护策略** • **查询Anti-DDoS默认防护策略** 参数“traffic_pos_id”取值范围修改为“1～9或取值99”；参数“cleaning_access_pos_id”取值范围修改为“1～9或取值99”
2021-12-06	第三次正式发布。 “查询Anti-DDoS任务”章节，修改内容描述。
2021-04-19	第二次正式发布。 修改章节标题。
2020-12-31	第一次正式发布。