

云防火墙

API 参考

文档版本 03
发布日期 2023-12-15



版权所有 © 华为云计算技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI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云计算技术有限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，华为云计算技术有限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。

华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

目录

1 使用前必读.....	1
1.1 概述.....	1
1.2 调用说明.....	1
1.3 终端节点.....	1
1.4 基本概念.....	1
2 API 概览.....	3
3 如何调用 API.....	4
3.1 构造请求.....	4
3.2 认证鉴权.....	6
3.3 返回结果.....	8
4 API.....	10
4.1 防火墙管理.....	10
4.1.1 创建防火墙.....	10
4.1.2 获取 CFW 任务执行状态.....	21
4.1.3 删除防火墙.....	25
4.1.4 查询防火墙列表.....	29
4.1.5 修改东西向防火墙防护状态.....	36
4.1.6 查询防火墙详细信息.....	42
4.1.7 获取东西向防火墙信息.....	52
4.1.8 创建东西向防火墙.....	60
4.1.9 查询防护 VPC 数.....	65
4.2 EIP 管理.....	71
4.2.1 弹性 IP 开启关闭.....	71
4.2.2 查询 Eip 个数.....	81
4.2.3 弹性 IP 列表查询.....	86
4.3 ACL 规则管理.....	94
4.3.1 创建 ACL 规则.....	94
4.3.2 删除 ACL 规则.....	107
4.3.3 批量删除 Acl 规则.....	112
4.3.4 删除规则击中次数.....	118
4.3.5 更新 ACL 规则.....	123
4.3.6 批量更新规则动作.....	135

4.3.7 ACL 防护规则优先级设置.....	140
4.3.8 查询防护规则.....	145
4.3.9 查询规则标签.....	157
4.3.10 获取规则击中次数.....	162
4.4 黑白名单管理.....	167
4.4.1 创建黑白名单规则.....	167
4.4.2 更新黑白名单列表.....	174
4.4.3 删除黑白名单规则.....	180
4.4.4 查询黑白名单列表.....	185
4.5 地址组管理.....	191
4.5.1 添加地址组.....	191
4.5.2 添加地址组成员.....	197
4.5.3 删除地址组.....	204
4.5.4 删除地址组成员.....	209
4.5.5 批量删除地址组成员.....	214
4.5.6 更新地址组信息.....	219
4.5.7 查询地址组列表.....	224
4.5.8 查询地址组详细信息.....	231
4.5.9 查询地址组成员.....	236
4.6 服务组管理.....	242
4.6.1 新建服务组.....	242
4.6.2 新建服务成员.....	248
4.6.3 删除服务组.....	254
4.6.4 删除服务成员.....	259
4.6.5 批量删除服务组成员信息.....	264
4.6.6 修改服务组.....	269
4.6.7 获取服务组列表.....	274
4.6.8 查询服务组详情.....	281
4.6.9 查询服务成员列表.....	286
4.7 域名解析及域名组管理.....	292
4.7.1 添加域名组.....	292
4.7.2 删除域名组.....	298
4.7.3 更新域名组.....	302
4.7.4 更新 dns 服务器列表.....	308
4.7.5 查询域名组列表.....	314
4.7.6 查询 dns 服务器列表.....	320
4.7.7 查询域名解析 ip 地址.....	325
4.7.8 获取域名组下域名列表.....	330
4.7.9 添加域名列表.....	336
4.7.10 删除域名列表.....	342
4.8 IPS 管理.....	347
4.8.1 查询 IPS 特性开关状态.....	347

4.8.2 IPS 特性开关操作.....	352
4.8.3 查询防护模式.....	357
4.8.4 切换防护模式.....	362
4.9 日志管理.....	368
4.9.1 获取日志配置.....	368
4.9.2 创建日志配置.....	372
4.9.3 更新日志配置.....	377
4.9.4 查询流日志.....	382
4.9.5 查询访问控制日志.....	390
4.9.6 查询攻击日志.....	398
A 附录.....	408
A.1 状态码.....	408
A.2 错误码.....	408
A.3 获取项目 ID.....	412
B 修订记录.....	413

1 使用前必读

1.1 概述

云防火墙（Cloud Firewall，CFW）是新一代的云原生防火墙，提供云上互联网边界和VPC边界的防护，包括实时入侵检测与防御、全局统一访问控制、全流量分析可视化、日志审计与溯源分析等，同时支持按需弹性扩容、AI提升智能防御能力、灵活扩展满足云上业务的变化和扩张需求，极简应用让用户快速灵活应对威胁。云防火墙服务是为用户业务上云提供网络安全防护的基础服务。

您可以使用本文档提供的API对CFW进行相关操作，如查询、更新等操作。

在调用云防火墙API之前，请确保已经充分了解云防火墙服务，有关云防火墙服务的详细介绍，请参见[产品介绍](#)。

1.2 调用说明

CFW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 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工作，而是通过创建用户来进行日常工作。
- 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、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](#)。

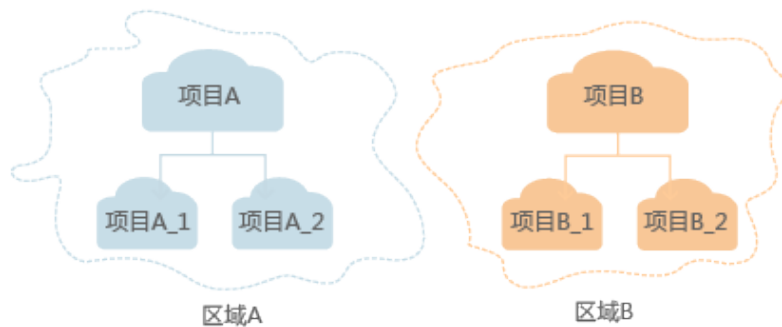
- 可用区（AZ，Availability Zone）

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，使得用户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CFW提供的接口，您可以完整的使用CFW的所有功能。

类型	说明
防火墙管理	查询防火墙信息接口，包括查询防火墙列表、查询防火墙详细信息、修改防火墙防护状态等。
EIP管理	管理EIP接口，包括开启/关闭EIP、查询EIP个数，查询EIP列表等。
ACL规则管理	ACL规则接口，包括创建、更新、删除ACL规则等接口。
黑白名单管理	管理黑白名单，包括创建、更新、删除黑/白名单等接口。
地址组管理	管理地址组，包括添加、查询、更新地址组等接口。
服务组管理	管理服务组，包括新增、查询、修改服务组等接口。
域名解析及域名组管理	管理域名组，包括添加、查询、更新域名组等接口。
IPS管理	管理IPS特性开关，包括查询IPS状态、IPS开关、查询防护模式等操作。
日志管理	管理日志接口，包括查询三类日志等接口。

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}

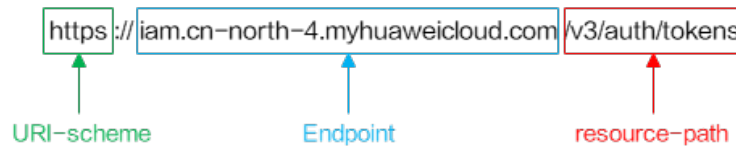
尽管请求URI包含在请求消息头中，但大多数语言或框架都要求您从请求消息中单独传递它，所以在此单独强调。

- **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，***xxxxxxxxxxxxxxxxxxxx***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": "xxxxxxxxxxxxxxxxxxxx"
      }
    }
  }
}
```

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": "xxxxxxx"
      }
    }
  }
}
```

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[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

```
connection → keep-alive
content-type → application/json
date → Tue, 12 Feb 2019 06:52:13 GMT
server → Web Server
strict-transport-security → max-age=31536000; includeSubdomains;
transfer-encoding → chunked
via → proxy A
x-content-type-options → nosniff
x-download-options → noopen
x-frame-options → SAMEORIGIN
x-iam-trace-id → 218d45ab-d674-4995-af3a-2d0255ba41b5
x-subject-token → MIIVXQYJKoZIhvcNAQcCoIIYJCCEGoCAQExDTALBglghkgBZQMEAgEwgharBgkqhkiG9w0BBwGgghacBIIIWmHsidG9rZW4iOnsiZlhwXlJlc19hdCI6IjIwMTk0MTU0MCBj3Kjs6YgKnpVNRbW2eZ5eb785Z0kqjACgkqO1wi4JlGzrpd18LGXK5tdfdq4lqHCYb8P4NaY0NYejcAgz/VeFYtLWT1GSO0zxKZmlQHQj82HBqHdgIZO9fuEbL5dMhdavj+33wElxHRCe9I87o+k9-j+CMZSEB7bUGd5Uj6eRASXl1jipPEGA270g1FruooL6jqglFkNPQuFSOU8+uSsttVwRtNfsC+qTp22Rkd5MCqFGQ8LcuUxC3a+9CMBnOintWW7oeRUVhVpxk8pxiX1wTEboX-RzT6MUbupvGw-oPNFYxJECKnoH3HRozv0vN--n5d6Nbxg==
x-xss-protection → 1; mode=block;
```

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": "xxxxxxx",
            .....

```

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": {
    "message": "The request you have made requires authentication.",
    "title": "Unauthorized"
  }
}
```

其中，error_code表示错误码，error_msg表示错误描述信息。

4 API

4.1 防火墙管理

4.1.1 创建防火墙

功能介绍

创建防火墙

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/firewall

表 4-1 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

请求参数

表 4-2 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	请求token

参数	是否必选	参数类型	描述
X-Client-Token	否	String	保证客户端请求幂等性的标识。该标识为32位UUID格式，由客户端生成，且需确保不同请求之间该标识具有唯一性。
X-Trace-Id	否	String	-

表 4-3 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	防火墙名称
enterprise_project_id	否	String	企业项目ID，租户未开启企业项目时传0 缺省值： 0
tags	否	Array of tags objects	资源标签
flavor	否	flavor object	规格信息
charge_info	是	charge_info object	计费类型信息，支持包年/包月和按需，默认为按需。

表 4-4 tags

参数	是否必选	参数类型	描述
key	是	String	标签KEY
value	是	String	标签VALUE

表 4-5 flavor

参数	是否必选	参数类型	描述
version	否	String	防火墙版本 “charge_mode” 为 “prePaid” 时，仅支持专业版。“charge_mode” 为 “postPaid” 时，支持标准版、专业版。 Standard - 标准版 Professional - 专业版 枚举值： <ul style="list-style-type: none">• Standard• Professional
extend_eip_count	否	Integer	扩展EIP数量，仅包周期场景下生效 最小值：0 最大值：2000 缺省值：0
extend_bandwidth	否	Integer	扩展带宽，步长为5，仅包周期场景下生效 最小值：0 最大值：5000
extend_vpc_count	否	Integer	扩展VPC数量，仅包周期场景下生效 最小值：0 最大值：100

表 4-6 charge_info

参数	是否必选	参数类型	描述
charge_mode	是	String	计费模式。 取值范围： prePaid: 预付费，即包年/包月。postPaid: 后付费，即按需付费。
period_type	否	String	订购周期类型。 取值范围： month: 包月。year: 包年。 说明：“charge_mode” 为 “prePaid” 时生效，且为必选值。

参数	是否必选	参数类型	描述
period_num	否	Integer	“charge_mode”为“prePaid”时生效，且为必选值，指定订购的时间。 取值范围： 当“period_type”为“month”时，取值为1~9。当“period_type”为“year”时，取值为1~3。
is_auto_renew	是	Boolean	创建包周期实例时可指定，表示是否自动续订，续订的周期和原周期相同，且续订时会自动支付。 true，为自动续订。false，为不自动续订，默认该方式。 缺省值： false
is_auto_pay	是	Boolean	创建包周期时可指定，表示是否自动从客户的账户中支付，此字段不影响自动续订的支付方式。 true，为自动支付。（会自动选择折扣和优惠券进行优惠，然后自动从客户华为云账户中支付），自动支付失败后会生成订单成功(该订单应付金额是优惠后金额)、但订单状态为“待支付”，等待客户手动支付(手动支付时，客户还可以修改系统自动选择的折扣和优惠券) false，为手动支付，默认该方式。（需要客户手动去支付，客户可以选择折扣和优惠券） 缺省值： false

响应参数

状态码： 200

表 4-7 响应 Body 参数

参数	参数类型	描述
job_id	String	实例创建的任务id。仅创建按需实例时会返回该参数。
order_id	String	订单号，创建包年包月时返回该参数。

参数	参数类型	描述
data	CreateFirewallReq object	按需防火墙实体

表 4-8 CreateFirewallReq

参数	参数类型	描述
name	String	防火墙名称
enterprise_project_id	String	企业项目ID，租户未开启企业项目时传0 缺省值： 0
tags	Array of tags objects	资源标签
flavor	flavor object	规格信息
charge_info	charge_info object	计费类型信息，支持包年/包月和按需，默认为按需。

表 4-9 tags

参数	参数类型	描述
key	String	标签KEY
value	String	标签VALUE

表 4-10 flavor

参数	参数类型	描述
version	String	防火墙版本 “charge_mode” 为 “prePaid” 时，仅支持专业版。“charge_mode” 为 “postPaid” 时，支持标准版、专业版。 Standard - 标准版 Professional - 专业版 枚举值： <ul style="list-style-type: none">• Standard• Professional
extend_eip_count	Integer	扩展EIP数量，仅包周期场景下生效 最小值： 0 最大值： 2000 缺省值： 0

参数	参数类型	描述
extend_band_width	Integer	扩展带宽，步长为5，仅包周期场景下生效 最小值： 0 最大值： 5000
extend_vpc_count	Integer	扩展VPC数量，仅包周期场景下生效 最小值： 0 最大值： 100

表 4-11 charge_info

参数	参数类型	描述
charge_mode	String	计费模式。 取值范围： prePaid：预付费，即包年/包月。postPaid：后付费，即按需付费。
period_type	String	订购周期类型。 取值范围： month：包月。year：包年。说明： “charge_mode”为“prePaid”时生效，且为必选值。
period_num	Integer	“charge_mode”为“prePaid”时生效，且为必选值，指定订购的时间。 取值范围： 当“period_type”为“month”时，取值为1~9。当“period_type”为“year”时，取值为1~3。
is_auto_renew	Boolean	创建包周期实例时可指定，表示是否自动续订，续订的周期和原周期相同，且续订时会自动支付。 true，为自动续订。false，为不自动续订，默认该方式。 缺省值： false

参数	参数类型	描述
is_auto_pay	Boolean	创建包周期时可指定，表示是否自动从客户的账户中支付，此字段不影响自动续订的支付方式。 true，为自动支付。（会自动选择折扣和优惠券进行优惠，然后自动从客户华为云账户中支付），自动支付失败后会生成订单成功(该订单应付金额是优惠后金额)、但订单状态为“待支付”，等待客户手动支付(手动支付时，客户还可以修改系统自动选择的折扣和优惠券) false，为手动支付，默认该方式。（需要客户手动去支付，客户可以选择折扣和优惠券） 缺省值： false

状态码： 400

表 4-12 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

开通标准版防火墙，扩展防护EIP2000个，扩展防护带宽5000Mbps，扩展防护VCP100个，包周期计费模式，启用自动续费，启用自动付款，开通1个月

<https://{Endpoint}/v2/124147da-5b08-471a-93d2-bc82acc290c6/firewall>

```
{
  "name": "CFW-TEST",
  "enterprise_project_id": "0",
  "tags": [ {
    "key": "TagKey",
    "value": "TagVal"
  } ],
  "flavor": {
    "version": "standard",
    "extend_eip_count": 2000,
    "extend_bandwidth": 5000,
    "extend_vpc_count": 100
  },
  "charge_info": {
    "charge_mode": "prePaid",
    "period_type": "month",
    "period_num": 1,
    "is_auto_renew": true,
  }
}
```

```
"is_auto_pay" : true
}
}
```

响应示例

状态码： 200

执行成功，返回JobID

```
{
  "data" : {
    "charge_info" : {
      "charge_mode" : "prePaid",
      "is_auto_pay" : true,
      "is_auto_renew" : true,
      "period_num" : 1,
      "period_type" : "month"
    },
    "enterprise_project_id" : "0",
    "flavor" : {
      "extend_bandwidth" : 5000,
      "extend_eip_count" : 2000,
      "extend_vpc_count" : 100,
      "version" : "Standard"
    },
    "name" : "CFW-TEST",
    "tags" : [ {
      "key" : "TagKey",
      "value" : "TagVal"
    } ]
  },
  "order_id" : "CS2403271050ZEM0L"
}
```

状态码： 400

请求体错误

```
{
  "error_code" : "CFW.00100001",
  "error_msg" : "系统繁忙，请稍后重试"
}
```

SDK 代码示例

SDK代码示例如下。

Java

开通标准版防火墙，扩展防护EIP2000个，扩展防护带宽5000Mbps，扩展防护VCP100个，包周期计费模式，启用自动续费，启用自动付款，开通1个月

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

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

```
public class CreateFirewallSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateFirewallRequest request = new CreateFirewallRequest();
        CreateFirewallReq body = new CreateFirewallReq();
        CreateFirewallReqChargeInfo chargeInfoBody = new CreateFirewallReqChargeInfo();
        chargeInfoBody.withChargeMode("prePaid")
            .withPeriodType("month")
            .withPeriodNum(1)
            .withIsAutoRenew(true)
            .withIsAutoPay(true);
        CreateFirewallReqFlavor flavorBody = new CreateFirewallReqFlavor();
        flavorBody.withVersion(CreateFirewallReqFlavor.VersionEnum.fromValue("standard"))
            .withExtendEipCount(2000)
            .withExtendBandwidth(5000)
            .withExtendVpcCount(100);
        List<CreateFirewallReqTags> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new CreateFirewallReqTags()
                .withKey("TagKey")
                .withValue("TagVal")
        );
        body.withChargeInfo(chargeInfoBody);
        body.withFlavor(flavorBody);
        body.withTags(listbodyTags);
        body.withEnterpriseProjectId("0");
        body.withName("CFW-TEST");
        request.withBody(body);
        try {
            CreateFirewallResponse response = client.createFirewall(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

开通标准版防火墙，扩展防护EIP2000个，扩展防护带宽5000Mbps，扩展防护VCP100个，包周期计费模式，启用自动续费，启用自动付款，开通1个月

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreateFirewallRequest()
        chargeInfobody = CreateFirewallReqChargeInfo(
            charge_mode="prePaid",
            period_type="month",
            period_num=1,
            is_auto_renew=True,
            is_auto_pay=True
        )
        flavorbody = CreateFirewallReqFlavor(
            version="standard",
            extend_eip_count=2000,
            extend_bandwidth=5000,
            extend_vpc_count=100
        )
        listTagsbody = [
            CreateFirewallReqTags(
                key="TagKey",
                value="TagVal"
            )
        ]
        request.body = CreateFirewallReq(
            charge_info=chargeInfobody,
            flavor=flavorbody,
            tags=listTagsbody,
            enterprise_project_id="0",
            name="CFW-TEST"
        )
        response = client.create_firewall(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

开通标准版防火墙，扩展防护EIP2000个，扩展防护带宽5000Mbps，扩展防护VPC100个，包周期计费模式，启用自动续费，启用自动付款，开通1个月

```
package main

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



```
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateFirewallRequest{
        periodTypeChargeInfo:= "month"
        periodNumChargeInfo:= int32(1)
        chargeInfobody := &model.CreateFirewallReqChargeInfo{
            ChargeMode: "prePaid",
            PeriodType: &periodTypeChargeInfo,
            PeriodNum: &periodNumChargeInfo,
            IsAutoRenew: true,
            IsAutoPay: true,
        }
        versionFlavor:= model.GetCreateFirewallReqFlavorVersionEnum().STANDARD
        extendEipCountFlavor:= int32(2000)
        extendBandwidthFlavor:= int32(5000)
        extendVpcCountFlavor:= int32(100)
        flavorbody := &model.CreateFirewallReqFlavor{
            Version: &versionFlavor,
            ExtendEipCount: &extendEipCountFlavor,
            ExtendBandwidth: &extendBandwidthFlavor,
            ExtendVpcCount: &extendVpcCountFlavor,
        }
        var listTagsbody = []model.CreateFirewallReqTags{
            {
                Key: "TagKey",
                Value: "TagVal",
            },
        }
        enterpriseProjectIdCreateFirewallReq:= "0"
        request.Body = &model.CreateFirewallReq{
            ChargeInfo: chargeInfobody,
            Flavor: flavorbody,
            Tags: &listTagsbody,
            EnterpriseProjectId: &enterpriseProjectIdCreateFirewallReq,
            Name: "CFW-TEST",
        }
    }
    response, err := client.CreateFirewall(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	执行成功，返回JobID
400	请求体错误

错误码

请参见[错误码](#)。

4.1.2 获取 CFW 任务执行状态

功能介绍

获取CFW任务执行状态

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/jobs/{job_id}

表 4-13 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
job_id	是	String	任务ID

请求参数

表 4-14 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	请求token

响应参数

状态码： 200

表 4-15 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512
data	data object	执行结果

表 4-16 data

参数	参数类型	描述
id	String	任务ID
status	String	任务执行状态。 取值： 值为“Running”，表示任务正在执行。 值为“Success”，表示任务执行成功。 值为“Failed”，表示任务执行失败。 枚举值： <ul style="list-style-type: none">● Running● Success● Failed
begin_time	String	创建时间，格式为“yyyy-mm-ddThh:mm:ssZ”。 其中，T指某个时间的开始；Z指时区偏移量，例如北京时间偏移显示为+0800。
end_time	String	结束时间，格式为“yyyy-mm-ddThh:mm:ssZ”。 其中，T指某个时间的开始；Z指时区偏移量，例如北京时间偏移显示为+0800。

请求示例

获取项目09bb24e6fe80d23d2fa2c010b53b418c下的jobid为的f588ce71-e26c-400d-8981-f854355f6849的任务情况

```
/v3/09bb24e6fe80d23d2fa2c010b53b418c/jobs/f588ce71-e26c-400d-8981-f854355f6849
```

响应示例

状态码： 200

获取任务情况接口返回值

```
{
  "data" : {
    "begin_time" : 1641370501000,
    "end_time" : 1641370515000,
    "id" : "f588ce71-e26c-400d-8981-f854355f6849",
    "status" : "Success"
  }
}
```

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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class ListJobSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListJobRequest request = new ListJobRequest();
        try {
            ListJobResponse response = client.listJob(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.getStatusCode());
            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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")  
    sk = __import__('os').getenv("CLOUD_SDK_SK")  
  
    credentials = BasicCredentials(ak, sk) \  
  
    client = CfwClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(CfwRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ListJobRequest()  
        response = client.list_job(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"  
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(  
        cfw.CfwClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())
```

```
request := &model.ListJobRequest{}
response, err := client.ListJob(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	获取任务情况接口返回值

错误码

请参见[错误码](#)。

4.1.3 删除防火墙

功能介绍

删除防火墙，仅按需生效

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/firewall/{resource_id}

表 4-17 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id
resource_id	是	String	资源id

请求参数

表 4-18 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	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）

响应参数

状态码： 200

表 4-19 响应 Body 参数

参数	参数类型	描述
data	String	删除防火墙返回data

请求示例

删除项目id为06217ebc876e427a80a2c05d51264ab1下的防火墙08065281-860a-4c98-aeb5-82cf65c44c46。

```
https://{Endpoint}/v2/06217ebc876e427a80a2c05d51264ab1/firewall/08065281-860a-4c98-aeb5-82cf65c44c46
```

响应示例

状态码： 200

删除防火墙返回值

```
{  
  "data" : "56884cd0-cf3c-4cb7-bbeb-59d8722a2671"  
}
```

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.cfw.v1.region.CfwRegion;  
import com.huaweicloud.sdk.cfw.v1.*;
```

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

public class DeleteFirewallSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteFirewallRequest request = new DeleteFirewallRequest();
        try {
            DeleteFirewallResponse response = client.deleteFirewall(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = CfwClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CfwRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteFirewallRequest()
        response = client.delete_firewall(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteFirewallRequest{}
    response, err := client.DeleteFirewall(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	删除防火墙返回值

错误码

请参见[错误码](#)。

4.1.4 查询防火墙列表

功能介绍

查询防火墙列表

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/firewalls/list

表 4-20 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目ID

表 4-21 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id，用户支持企业项目后，由企业项目生成的id。

请求参数

表 4-22 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）

表 4-23 请求 Body 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id，用户支持企业项目后，由企业项目生成的id
key_word	否	String	查询关键字

参数	是否必选	参数类型	描述
tags	否	Array of TagInfo objects	标签列表
limit	是	Integer	每页显示个数，范围为1-1024
offset	是	Integer	偏移量：指定返回记录的开始位置，必须为数字，取值范围为大于或等于0，默认0

表 4-24 TagInfo

参数	是否必选	参数类型	描述
key	否	String	标签键
values	否	Array of strings	标签值列表

响应参数

状态码： 200

表 4-25 响应 Body 参数

参数	参数类型	描述
user_support_eps	Boolean	是否支持eps
has_ndr	Boolean	是否存在ndr
is_support_postpaid	Boolean	是否支持按需购买
is_support_basic_version	Boolean	是否支持基础版
is_support_buy_professional	Boolean	是否支持购买专业版
data	HttpFirewallInstanceListResponseData object	查询防火墙列表返回值data信息

表 4-26 HttpFirewallInstanceListResponseData

参数	参数类型	描述
limit	Integer	每页显示个数，范围为1-1024
offset	Integer	偏移量：指定返回记录的开始位置，必须为数字，取值范围为大于或等于0，默认0
project_id	String	租户项目ID
total	Integer	总数
records	Array of FirewallInstanceVO objects	查询防火墙列表记录

表 4-27 FirewallInstanceVO

参数	参数类型	描述
fw_instance_id	String	防火墙实例id，创建云防火墙后用于标志防火墙由系统自动生成的标志id，可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下，fw_instance_id为空时，返回账号下第一个墙的信息；fw_instance_id非空时，返回与fw_instance_id对应墙的信息。
resource_id	String	资源id
name	String	防火墙创建时间戳
fw_instance_name	String	防火墙名称
enterprise_project_id	String	企业项目id，用户支持企业项目后，由企业项目生成的id。
ha_type	Integer	集群类型
charge_mode	Integer	计费模式 0: 包年/包月 1: 按需
service_type	Integer	服务类型
engine_type	Integer	引擎类型
flavor	Flavor object	防火墙规格信息
status	Integer	防火墙状态列表，包括-1：等待支付，0：创建中，1，删除中，2：运行中，3：升级中，4：删除完成：5：冻结中，6：创建失败，7：删除失败，8：冻结失败，9：存储中，10：存储失败，11：升级失败
tags	String	标签列表

表 4-28 Flavor

参数	参数类型	描述
version	Integer	防火墙版本，0：标准版，1：专业版，2：铂金版，3：基础版，购买时，当防火墙“charge_mode”为“postPaid”时，仅支持专业版。“charge_mode”为“prePaid”时，支持标准版、专业版。 枚举值： <ul style="list-style-type: none">• 0• 1• 2• 3
eip_count	Integer	eip数量 最小值：1
vpc_count	Integer	vpc数量 最小值：1
bandwidth	Integer	带宽 最小值：1
log_storage	Integer	日志存储
session_concurrent	Integer	目前的会话数
session_create	Integer	创建会话数
total_rule_count	Integer	总计规则数
used_rule_count	Integer	已使用规则数
vpc_bandwidth	Integer	vpc间带宽
default_bandwidth	Integer	默认防火墙带宽
default_eip_count	Integer	默认eip数
default_log_storage	Integer	默认日志存储
default_vpc_count	Integer	默认vpc数

请求示例

查询项目id为14181c1245cf4fd786824efe1e2b9388，企业项目id为all_granted_eps的第一页的防火墙列表

```
https://{Endpoint}/v1/14181c1245cf4fd786824efe1e2b9388/firewalls/list?  
enterprise_project_id=all_granted_eps  
  
{  
  "limit" : 10,  
  "offset" : 0  
}
```

响应示例

状态码： 200

查询防火墙列表返回值

```
{  
  "data" : {  
    "limit" : 10,  
    "offset" : 0,  
    "project_id" : "14181c1245cf4fd786824efe1e2b9388",  
    "records" : [ {  
      "charge_mode" : 0,  
      "engine_type" : 1,  
      "enterprise_project_id" : "default",  
      "flavor" : {  
        "bandwidth" : 60,  
        "eip_count" : 51,  
        "log_storage" : 0,  
        "version" : 1,  
        "vpc_count" : 8,  
        "default_eip_count" : 20,  
        "default_vpc_count" : 0,  
        "default_bandwidth" : 10,  
        "default_log_storage" : 0  
      },  
      "fw_instance_id" : "546af3f8-88e9-47f2-a205-2346d7090925",  
      "fw_instance_name" : "test",  
      "ha_type" : 1,  
      "name" : "1680054140516",  
      "resource_id" : "546af3f8-88e9-47f2-a205-2346d7090925",  
      "service_type" : 0,  
      "status" : 2,  
      "tags" : "{\"key1234\":\"1234\",\"key122\":\"2222\"}"  
    } ],  
    "total" : 1  
  },  
  "user_support_eps" : false,  
  "has_ndr" : false,  
  "is_support_postpaid" : false,  
  "is_support_basic_version" : false,  
  "is_support_buy_professional" : false  
}
```

SDK 代码示例

SDK代码示例如下。

Java

查询项目id为14181c1245cf4fd786824efe1e2b9388，企业项目id为all_granted_eps的第一页的防火墙列表

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class ListFirewallListSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListFirewallListRequest request = new ListFirewallListRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        QueryFireWallInstanceDto body = new QueryFireWallInstanceDto();
        body.withOffset(0);
        body.withLimit(10);
        request.withBody(body);
        try {
            ListFirewallListResponse response = client.listFirewallList(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.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

查询项目id为14181c1245cf4fd786824efe1e2b9388，企业项目id为all_granted_eps的第一页的防火墙列表

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

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

try:
    request = ListFirewallListRequest()
    request.enterprise_project_id = "<enterprise_project_id>"
    request.body = QueryFireWallInstanceDto(
        offset=0,
        limit=10
    )
    response = client.list_firewall_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

查询项目id为14181c1245cf4fd786824efe1e2b9388，企业项目id为all_granted_eps的第一页的防火墙列表

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListFirewallListRequest{}
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    request.Body = &model.QueryFireWallInstanceDto{
        Offset: int32(0),
        Limit: int32(10),
```



```
}  
response, err := client.ListFirewallList(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	查询防火墙列表返回值

错误码

请参见[错误码](#)。

4.1.5 修改东西向防火墙防护状态

功能介绍

东西向防护资源防护开启/关闭

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/firewall/east-west/protect

表 4-29 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-30 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id，用户支持企业项目后，由企业项目生成的id。

参数	是否必选	参数类型	描述
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

请求参数

表 4-31 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）

表 4-32 请求 Body 参数

参数	是否必选	参数类型	描述
object_id	是	String	防护对象id, 是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id, 可通过调用查询防火墙实例接口获得, 注意type为0的为互联网边界防护对象id, type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
status	是	Integer	防护状态: 0 开启, 1 关闭 枚举值: <ul style="list-style-type: none">• 0• 1

响应参数

状态码: 200

表 4-33 响应 Body 参数

参数	参数类型	描述
data	data object	响应体
trace_id	String	trace id

表 4-34 data

参数	参数类型	描述
id	String	标识ID

状态码： 400

表 4-35 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

给项目id为09bb24e6fe80d23d2fa2c010b53b418c的项目的防护对象
74820b38-1cc0-4f0b-8cce-32490fa840a3修改东西向防火墙防护状态为开启

```
https://{Endpoint}/v1/09bb24e6fe80d23d2fa2c010b53b418c/firewall/east-west/protect
```

```
{  
  "object_id" : "74820b38-1cc0-4f0b-8cce-32490fa840a3",  
  "status" : 1  
}
```

响应示例

状态码： 200

更新东西向防护状态响应体

```
{  
  "data" : {  
    "id" : "5c539816-7a94-4833-9df0-944b362f0797"  
  }  
}
```

状态码： 400

Bad Request

```
{
  "error_code" : "CFW.00200005",
  "error_msg" : "操作内容不存在"
}
```

SDK 代码示例

SDK代码示例如下。

Java

给项目id为09bb24e6fe80d23d2fa2c010b53b418c的项目的防护对象
74820b38-1cc0-4f0b-8cce-32490fa840a3修改东西向防火墙防护状态为开启

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class ChangeEastWestFirewallStatusSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ChangeEastWestFirewallStatusRequest request = new ChangeEastWestFirewallStatusRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        ChangeProtectStatusRequestBody body = new ChangeProtectStatusRequestBody();
        body.withStatus(ChangeProtectStatusRequestBody.StatusEnum.NUMBER_1);
        body.withObjectId("74820b38-1cc0-4f0b-8cce-32490fa840a3");
        request.withBody(body);
        try {
            ChangeEastWestFirewallStatusResponse response = client.changeEastWestFirewallStatus(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

给项目id为09bb24e6fe80d23d2fa2c010b53b418c的项目的防护对象
74820b38-1cc0-4f0b-8cce-32490fa840a3修改东西向防火墙防护状态为开启

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ChangeEastWestFirewallStatusRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        request.body = ChangeProtectStatusRequestBody(
            status=1,
            object_id="74820b38-1cc0-4f0b-8cce-32490fa840a3"
        )
        response = client.change_east_west_firewall_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

给项目id为09bb24e6fe80d23d2fa2c010b53b418c的项目的防护对象
74820b38-1cc0-4f0b-8cce-32490fa840a3修改东西向防火墙防护状态为开启

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ChangeEastWestFirewallStatusRequest{
    enterpriseProjectIdRequest:= "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest:= "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    request.Body = &model.ChangeProtectStatusRequestBody{
        Status: model.GetChangeProtectStatusRequestBodyStatusEnum().E_1,
        Objectid: "74820b38-1cc0-4f0b-8cce-32490fa840a3",
    }
}
response, err := client.ChangeEastWestFirewallStatus(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	更新东西向防护状态响应体
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.1.6 查询防火墙详细信息

功能介绍

查询防火墙实例

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/firewall/exist

表 4-36 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-37 Query 参数

参数	是否必选	参数类型	描述
offset	是	Integer	偏移量：指定返回记录的开始位置，必须为数字，取值范围为大于或等于0，默认0
limit	是	Integer	每页显示个数，范围为1-1024
service_type	是	Integer	服务类型 0 南北向防火墙 1 东西向防火墙 最小值：0 最大值：1 枚举值： <ul style="list-style-type: none">• 0• 1
enterprise_project_id	否	String	企业项目id，用户支持企业项目后，由企业项目生成的id。

参数	是否必选	参数类型	描述
fw_instance_id	否	String	防火墙实例id，创建云防火墙后用于标志防火墙由系统自动生成的标志id，可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下，fw_instance_id为空时，返回账号下第一个墙的信息；fw_instance_id非空时，返回与fw_instance_id对应墙的信息。
name	否	String	防火墙名称

请求参数

表 4-38 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）

响应参数

状态码： 200

表 4-39 响应 Body 参数

参数	参数类型	描述
data	GetFirewallInstanceData object	查询防火墙实例数据

表 4-40 GetFirewallInstanceData

参数	参数类型	描述
limit	Integer	每页显示个数，范围为1-1024
offset	Integer	偏移量：指定返回记录的开始位置，必须为数字，取值范围为大于或等于0，默认0
total	Integer	总数

参数	参数类型	描述
records	Array of GetFirewallInstanceResponseRecord objects	查询防火墙实例信息列表

表 4-41 GetFirewallInstanceResponseRecord

参数	参数类型	描述
fw_instance_id	String	防火墙实例id，创建云防火墙后用于标志防火墙由系统自动生成的标志id，可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。
name	String	防火墙名称
ha_type	Integer	集群类型
charge_mode	Integer	计费模式 0: 包年/包月 1: 按需
service_type	Integer	服务类型
engine_type	Integer	引擎类型
flavor	Flavor object	防火墙规格信息
protect_objects	Array of ProtectObjectVO objects	防护对象列表

参数	参数类型	描述
status	Integer	防火墙状态列表，包括-1：等待支付，0：创建中，1，删除中，2：运行中，3：升级中，4：删除完成：5：冻结中，6：创建失败，7：删除失败，8：冻结失败，9：存储中，10：存储失败，11：升级失败 枚举值： <ul style="list-style-type: none">• -1• 0• 1• 2• 3• 4• 5• 6• 7• 8• 9• 10• 11
is_old_firewall_instance	Boolean	是否为旧引擎，true表示是，false表示不是 枚举值： <ul style="list-style-type: none">• true• false
is_available_obs	Boolean	是否支持obs
is_support_threat_tags	Boolean	是否支持威胁标签
support_ipv6	Boolean	是否支持ipv6，true表示是，false表示不是
feature_toggle	Map<String, Boolean>	特性开关，boolean值为true表示是，false表示否
resources	Array of FirewallInstanceResource objects	防火墙资源列表
fw_instance_name	String	防火墙名称
enterprise_project_id	String	企业项目id，用户支持企业项目后，由企业项目生成的id。

参数	参数类型	描述
resource_id	String	资源id
support_url_filtering	Boolean	是否支持应用型，true表示是，false表示不是
tags	String	标签列表

表 4-42 Flavor

参数	参数类型	描述
version	Integer	防火墙版本，0：标准版，1：专业版，2：铂金版，3：基础版，购买时，当防火墙“charge_mode”为“postPaid”时，仅支持专业版。“charge_mode”为“prePaid”时，支持标准版、专业版。 枚举值： <ul style="list-style-type: none">• 0• 1• 2• 3
eip_count	Integer	eip数量 最小值：1
vpc_count	Integer	vpc数量 最小值：1
bandwidth	Integer	带宽 最小值：1
log_storage	Integer	日志存储
session_concurrent	Integer	目前的会话数
session_create	Integer	创建会话数
total_rule_count	Integer	总计规则数
used_rule_count	Integer	已使用规则数
vpc_bandwidth	Integer	vpc间带宽
default_bandwidth	Integer	默认防火墙带宽

参数	参数类型	描述
default_eip_count	Integer	默认eip数
default_log_storage	Integer	默认日志存储
default_vpc_count	Integer	默认vpc数

表 4-43 ProtectObjectVO

参数	参数类型	描述
object_id	String	防护对象id，是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id，可通过调用查询防火墙实例接口获得，注意type为0的为互联网边界防护对象id，type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
object_name	String	防护对象名称
type	Integer	防护对象类型：0 南北向，1 东西向护对象类型枚举值： <ul style="list-style-type: none">• 0• 1

表 4-44 FirewallInstanceResource

参数	参数类型	描述
resource_id	String	资源id
cloud_service_type	String	服务类型，用于CBC使用，特指： hws.service.type.cfw
resource_type	String	资源类型，包括：1、云防火墙:hws.resource.type.cfw 2、EIP:hws.resource.type.cfw.exp.eip 3、带宽:hws.resource.type.cfw.exp.bandwidth 4、VPC:hws.resource.type.cfw.exp.vpc 5、日志存储:hws.resource.type.cfw.exp.logaudit
resource_spec_code	String	库存单位码
resource_size	Integer	资源数量
resource_size_measure_id	Integer	资源单位

请求示例

查询项目id为9d80d070b6d44942af73c9c3d38e0429的项目存在的防火墙列表

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/firewall/exist?  
service_type=0&offset=0&limit=10
```

响应示例

状态码： 200

获取存在防火墙实例响应

```
{  
  "data": {  
    "limit": 10,  
    "offset": 0,  
    "records": [ {  
      "charge_mode": 0,  
      "engine_type": 1,  
      "enterprise_project_id": "default",  
      "feature_toggle": {  
        "is_support_anti_virus": true,  
        "is_support_application": true,  
        "is_support_tcp_proxy": false,  
        "is_support_url_profile": true,  
        "is_support_threat_tags": true,  
        "is_support_flow_associated_host": false,  
        "is_support_predefined": true,  
        "isSupportSession": false,  
        "is_support_acl_region_config": true,  
        "is_support_ips": true,  
        "is_support_ew_create_er_tenant_inspection_mode": false,  
        "ips_rule_list": true,  
        "long_connect": true,  
        "is_support_ew_create_vpc_peering_inspection_mode": true,  
        "alarm_config": true,  
        "is_not_support_resource_reduction": false,  
        "acl_multi_object": true,  
        "is_support_advanced_ips_rule": true,  
        "is_support_multi_account": false,  
        "is_support_capture": true,  
        "is_support_ew_create_er_bearer_inspection_mode": true  
      },  
      "flavor": {  
        "bandwidth": 60,  
        "eip_count": 51,  
        "log_storage": 0,  
        "session_concurrent": 200,  
        "session_create": 200,  
        "total_rule_count": 200,  
        "used_rule_count": 2136,  
        "version": 1,  
        "vpc_bandwidth": 1400,  
        "vpc_count": 8,  
        "default_eip_count": 20,  
        "default_vpc_count": 0,  
        "default_bandwidth": 10,  
        "default_log_storage": 0  
      },  
      "tags": "{\"key1234\":\"1234\",\"key122\":\"2222\"}",  
      "fw_instance_id": "546af3f8-88e9-47f2-a205-2346d7090925",  
      "fw_instance_name": "test",  
      "ha_type": 1,  
      "is_available_obs": false,  
      "is_old_firewall_instance": false,  
      "is_support_threat_tags": false,  
      "name": "1680054140516",  
    }  
  ]  
}
```

```
"protect_objects": [ {
  "object_id": "ae42418e-f077-41a0-9d3b-5b2f5ad9102b",
  "object_name": "1680054141674",
  "type": 0
}, {
  "object_id": "be83d202-df0b-498d-a96e-41589dc85c86",
  "object_name": "ew-1680070626042",
  "type": 1
}],
"resource_id": "546af3f8-88e9-47f2-a205-2346d7090925",
"resources": [ {
  "cloud_service_type": "hws.service.type.cfw",
  "resource_id": "546af3f8-88e9-47f2-a205-2346d7090925",
  "resource_spec_code": "cfw.professional",
  "resource_type": "hws.resource.type.cfw"
}, {
  "cloud_service_type": "hws.service.type.cfw",
  "resource_id": "0acdd5c7-1178-4bea-b5b6-bd55dc5e2669",
  "resource_size": 5,
  "resource_size_measure_id": 14,
  "resource_spec_code": "cfw.expack.vpc.professional",
  "resource_type": "hws.resource.type.cfw.exp.vpc"
}, {
  "cloud_service_type": "hws.service.type.cfw",
  "resource_id": "4002620c-916a-49c7-8042-cbe02fc17e61",
  "resource_size": 5,
  "resource_size_measure_id": 36,
  "resource_spec_code": "cfw.expack.bandwidth.professional",
  "resource_type": "hws.resource.type.cfw.exp.bandwidth"
}, {
  "cloud_service_type": "hws.service.type.cfw",
  "resource_id": "0235c7db-0baa-4c82-8db2-7b8d5108bd86",
  "resource_size": 2,
  "resource_size_measure_id": 14,
  "resource_spec_code": "cfw.expack.eip.professional",
  "resource_type": "hws.resource.type.cfw.exp.eip"
}, {
  "cloud_service_type": "hws.service.type.cfw",
  "resource_id": "079ade46-18cd-4917-b7bb-00d402931097",
  "resource_size": 6,
  "resource_size_measure_id": 14,
  "resource_spec_code": "cfw.expack.vpc.professional",
  "resource_type": "hws.resource.type.cfw.exp.vpc"
}, {
  "cloud_service_type": "hws.service.type.cfw",
  "resource_id": "dd078faa-abfd-4e63-b681-1a93489955b9",
  "resource_size": 1,
  "resource_size_measure_id": 14,
  "resource_spec_code": "cfw.expack.eip.professional",
  "resource_type": "hws.resource.type.cfw.exp.eip"
}, {
  "cloud_service_type": "hws.service.type.cfw",
  "resource_id": "4d78d523-745d-4d54-a9ca-e6d25e555bde",
  "resource_size": 10,
  "resource_size_measure_id": 36,
  "resource_spec_code": "cfw.expack.bandwidth.professional",
  "resource_type": "hws.resource.type.cfw.exp.bandwidth"
}
],
"service_type": 0,
"status": 2,
"support_ipv6": true,
"support_url_filtering": true
}],
"total": 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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class ListFirewallDetailSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListFirewallDetailRequest request = new ListFirewallDetailRequest();
        request.withOffset(<offset>);
        request.withLimit(<limit>);
        request.withServiceType(ListFirewallDetailRequest.ServiceTypeEnum.NUMBER_<service_type>);
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        try {
            ListFirewallDetailResponse response = client.listFirewallDetail(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListFirewallDetailRequest()
        request.offset = <offset>
        request.limit = <limit>
        request.service_type = <service_type>
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        response = client.list_firewall_detail(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListFirewallDetailRequest{}
    request.Offset = int32(<offset>)
    request.Limit = int32(<limit>)
    request.ServiceType = model.GetListFirewallDetailRequestServiceTypeEnum().<SERVICE_TYPE>
    enterpriseProjectIdRequest := "<enterprise_project_id>"
```



```
request.EnterpriseProjectId = &enterpriseProjectIdRequest
fwInstanceIdRequest := "<fw_instance_id>"
request.FwInstanceId = &fwInstanceIdRequest
response, err := client.ListFirewallDetail(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	获取存在防火墙实例响应
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.1.7 获取东西向防火墙信息

功能介绍

获取东西向防火墙信息

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/firewall/east-west

表 4-45 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-46 Query 参数

参数	是否必选	参数类型	描述
limit	是	Integer	每页显示个数，范围为1-1024
offset	是	Integer	偏移量：指定返回记录的开始位置，必须为数字，取值范围为大于或等于0，默认0
enterprise_project_id	否	String	企业项目id，用户支持企业项目后，由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id，创建云防火墙后用于标志防火墙由系统自动生成的标志id，可通过调用查询防火墙实例接口获得。具体可参考API Explorer和帮助中心FAQ。默认情况下，fw_instance_id为空时，返回账号下第一个墙的信息；fw_instance_id非空时，返回与fw_instance_id对应墙的信息。

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）

响应参数

状态码： 200

表 4-48 响应 Body 参数

参数	参数类型	描述
data	GetEastWestFirewallResponseBody object	获取东西向防火墙列表返回data

表 4-49 GetEastWestFirewallResponseBody

参数	参数类型	描述
object_id	String	防护对象id, 是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id, 可通过调用查询防火墙实例接口获得, 注意type为0的为互联网边界防护对象id, type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
project_id	String	租户project_id
status	Integer	防护状态: 0 已开启防护, 1 未开启防护
er_associated_subnet	SubnetInfo object	ER关联子网信息
firewall_associated_subnets	Array of SubnetInfo objects	云防火墙关联子网信息
er	ErInstance object	出方向关联企业路由器信息
inspection_vpc	VpcDetail object	监控vpc信息
protect_infos	Array of EwProtectResourceInfo objects	东西向防护资源信息
total	Integer	防护VPC总数
offset	Integer	偏移量: 指定返回记录的开始位置, 必须为数字, 取值范围为大于或等于0, 默认0
limit	Integer	每页显示个数, 范围为1-1024
mode	String	防护模式
ew_vpc_route_limit	Integer	东西向路由限制

表 4-50 SubnetInfo

参数	参数类型	描述
availability_zone	String	子网id
cidr	String	vpc cidr
name	String	子网名称
id	String	子网id

参数	参数类型	描述
gateway_ip	String	子网网关ip
vpc_id	String	vpc id
status	String	子网的状态
ipv6_enable	Boolean	是否支持ipv6, boolean值为true表示是, false表示否

表 4-51 ErlInstance

参数	参数类型	描述
id	String	ER实例id
name	String	ER名称
state	String	ER状态
enterprise_project_id	String	企业租户id
project_id	String	租户id
enable_ipv6	String	是否开启ipv6

表 4-52 VpcDetail

参数	参数类型	描述
id	String	id
name	String	名称
cidr	String	vpc cidr
status	String	状态

表 4-53 EwProtectResourceInfo

参数	参数类型	描述
protected_resource_type	Integer	防护资源类型: 0 VPC, 1 VGW
protected_resource_name	String	防护资源名称
protected_resource_id	String	防护资源id

参数	参数类型	描述
protected_resource_nat_name	String	防护资源nat网关名称
protected_resource_nat_id	String	防护资源nat网关id
protected_resource_project_id	String	防护资源租户id
protected_resource_mode	String	防护资源模式
status	Integer	防护VPC的防护状态，0表示已关联，1表示未关联。

状态码： 500

表 4-54 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

获取东西向防火墙信息

```
https://{Endpoint}/v1/09bb24e6f280d23d0f9fc0104b901480/firewall/east-west?limit=10&offset=0
```

响应示例

状态码： 200

东西向防火墙信息响应

```
{
  "data": {
    "ew_vpc_route_limit": 5,
    "inspection_vpc": {
      "cidr": "10.90.90.0/24",
      "id": "4471b50f-811a-4f9b-9575-a491e6c81dd0",
      "name": "inspection-vpc"
    }
  }
}
```

```
"limit" : 50,
"mode" : "peer",
"object_id" : "5681ed03-ff3f-4dab-9bb1-daf388121ad1",
"offset" : 0,
"project_id" : "28f403ddd3f141daa6e046e85cb15519",
"protect_infos" : [ {
  "protected_resource_id" : "4c51e814-03b5-4754-87cb-243701f5b4ff",
  "protected_resource_mode" : "peer",
  "protected_resource_name" : "vpc-c247",
  "protected_resource_project_id" : "28f403ddd3f141daa6e046e85cb15519",
  "protected_resource_type" : 0,
  "status" : 0
}, {
  "protected_resource_id" : "af44c693-a13c-46a7-a6d2-eb2cb2570e57",
  "protected_resource_mode" : "peer",
  "protected_resource_name" : "vpc-10.1",
  "protected_resource_project_id" : "28f403ddd3f141daa6e046e85cb15519",
  "protected_resource_type" : 0,
  "status" : 0
}, {
  "protected_resource_id" : "2ae53845-64b8-4cd5-bcbe-efb93a7c9207",
  "protected_resource_mode" : "peer",
  "protected_resource_name" : "vpc-test",
  "protected_resource_project_id" : "28f403ddd3f141daa6e046e85cb15519",
  "protected_resource_type" : 0,
  "status" : 1
}, {
  "protected_resource_id" : "8a51174b-376b-40e5-8ccf-89d33703842b",
  "protected_resource_mode" : "peer",
  "protected_resource_name" : "vpc-a0c1",
  "protected_resource_project_id" : "28f403ddd3f141daa6e046e85cb15519",
  "protected_resource_type" : 0,
  "status" : 1
}, {
  "protected_resource_id" : "9cccb096-31be-4d68-a7ee-156f0c947f0b",
  "protected_resource_mode" : "peer",
  "protected_resource_name" : "vpc-192.168",
  "protected_resource_project_id" : "28f403ddd3f141daa6e046e85cb15519",
  "protected_resource_type" : 0,
  "status" : 1
} ],
"status" : 0,
"total" : 5
}
```

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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class ListEastWestFirewallSolution {

    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);

CfwClient client = CfwClient.newBuilder()
    .withCredential(auth)
    .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
    .build();
ListEastWestFirewallRequest request = new ListEastWestFirewallRequest();
request.withLimit(<limit>);
request.withOffset(<offset>);
request.withEnterpriseProjectId("<enterprise_project_id>");
request.withFwInstanceId("<fw_instance_id>");
try {
    ListEastWestFirewallResponse response = client.listEastWestFirewall(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
```

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

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

```
    try:
```

```
        request = ListEastWestFirewallRequest()
        request.limit = <limit>
        request.offset = <offset>
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        response = client.list_east_west_firewall(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListEastWestFirewallRequest{}
    request.Limit = int32(<limit>)
    request.Offset = int32(<offset>)
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    response, err := client.ListEastWestFirewall(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	东西向防火墙信息响应

状态码	描述
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.1.8 创建东西向防火墙

功能介绍

创建东西向防火墙

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/firewall/east-west

表 4-55 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-56 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。

参数	是否必选	参数类型	描述
fw_instance_id	是	String	防火墙实例id，创建云防火墙后用于标志防火墙由系统自动生成的标志id，可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下，fw_instance_id为空时，返回账号下第一个墙的信息；fw_instance_id非空时，返回与fw_instance_id对应墙的信息。

请求参数

表 4-57 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）

表 4-58 请求 Body 参数

参数	是否必选	参数类型	描述
er_id	否	String	出方向关联ER实例id
inspection_cidr	是	String	inspection cidr
mode	是	String	东西向防火墙模式，填写er

响应参数

状态码： 200

表 4-59 响应 Body 参数

参数	参数类型	描述
data	IdObject object	创建东西向防火墙返回值

表 4-60 IdObject

参数	参数类型	描述
id	String	id值
name	String	名称

请求示例

项目09bb24e6f280d23d0f9fc0104b901480下防火墙55b26ab5-e4b0-40e8-941c-a1778fe2a500创建er模式防火墙inspection_cidr为10.1.0.0/24，er_id为e0b22a23-02cf-4092-ace9-34b39e10dc77。

```
https://{Endpoint}/v1/09bb24e6f280d23d0f9fc0104b901480/firewall/east-west?fw_instance_id=55b26ab5-e4b0-40e8-941c-a1778fe2a500&enterprise_project_id=default
```

```
{
  "inspection_cidr": "10.1.0.0/24",
  "mode": "er",
  "er_id": "e0b22a23-02cf-4092-ace9-34b39e10dc77"
}
```

响应示例

状态码： 200

创建东西向防火墙返回值

```
{
  "data": {
    "id": "b6d4a7d5-388e-4594-b696-fb4bba1d2b9e"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

项目09bb24e6f280d23d0f9fc0104b901480下防火墙55b26ab5-e4b0-40e8-941c-a1778fe2a500创建er模式防火墙inspection_cidr为10.1.0.0/24，er_id为e0b22a23-02cf-4092-ace9-34b39e10dc77。

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class CreateEastWestFirewallSolution {

    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);

CfwClient client = CfwClient.newBuilder()
    .withCredential(auth)
    .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
    .build();
CreateEastWestFirewallRequest request = new CreateEastWestFirewallRequest();
request.withEnterpriseProjectId("<enterprise_project_id>");
request.withFwInstanceId("<fw_instance_id>");
CreateEastWestFirewallRequestBody body = new CreateEastWestFirewallRequestBody();
body.withMode("er");
body.withInspectionCidr("10.1.0.0/24");
body.withErlId("e0b22a23-02cf-4092-ace9-34b39e10dc77");
request.withBody(body);
try {
    CreateEastWestFirewallResponse response = client.createEastWestFirewall(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

项目09bb24e6f280d23d0f9fc0104b901480下防火墙55b26ab5-e4b0-40e8-941c-a1778fe2a500创建er模式防火墙inspection_cidr为10.1.0.0/24，er_id为e0b22a23-02cf-4092-ace9-34b39e10dc77。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CfwClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CfwRegion.value_of("<YOUR REGION>")) \
```

```
.build()

try:
    request = CreateEastWestFirewallRequest()
    request.enterprise_project_id = "<enterprise_project_id>"
    request.fw_instance_id = "<fw_instance_id>"
    request.body = CreateEastWestFirewallRequestBody(
        mode="er",
        inspection_cidr="10.1.0.0/24",
        er_id="e0b22a23-02cf-4092-ace9-34b39e10dc77"
    )
    response = client.create_east_west_firewall(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

项目09bb24e6f280d23d0f9fc0104b901480下防火墙55b26ab5-e4b0-40e8-941c-a1778fe2a500创建er模式防火墙inspection_cidr为10.1.0.0/24，er_id为e0b22a23-02cf-4092-ace9-34b39e10dc77。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateEastWestFirewallRequest{
        enterpriseProjectIdRequest:= "<enterprise_project_id>"
        request.EnterpriseProjectId = &enterpriseProjectIdRequest
        request.FwInstanceId = "<fw_instance_id>"
        erIdCreateEastWestFirewallRequestBody:= "e0b22a23-02cf-4092-ace9-34b39e10dc77"
        request.Body = &model.CreateEastWestFirewallRequestBody{
            Mode: "er",
            InspectionCidr: "10.1.0.0/24",
            ErId: &erIdCreateEastWestFirewallRequestBody,
        }
    }
    response, err := client.CreateEastWestFirewall(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	创建东西向防火墙返回值

错误码

请参见[错误码](#)。

4.1.9 查询防护 VPC 数

功能介绍

查询防护vpc信息

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/vpcs/protection

表 4-61 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-62 Query 参数

参数	是否必选	参数类型	描述
object_id	是	String	防护对象id, 是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id, 可通过调用查询防火墙实例接口获得, 注意type为0的为互联网边界防护对象id, type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

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）

响应参数

状态码： 200

表 4-64 响应 Body 参数

参数	参数类型	描述
trace_id	String	调用链id
data	VPCProtectsV o object	防护vpc返回值

表 4-65 VPCProtectsVo

参数	参数类型	描述
total	Integer	总VPC数
self_total	Integer	本项目防护VPC数
other_total	Integer	其他项目防护VPC数
protect_vpcs	Array of VpcAttachmentDetail objects	防护VPC
self_protect_vpcs	Array of VpcAttachmentDetail objects	本项目防护VPC
other_protect_vpcs	Array of VpcAttachmentDetail objects	其他项目防护VPC
total_assets	Integer	所有资产数量

表 4-66 VpcAttachmentDetail

参数	参数类型	描述
id	String	id
name	String	名称
vpc_id	String	vpc id
virsubnet_id	String	子网id
state	String	状态
created_at	String	创建时间
updated_at	String	更新时间
tags	Array of Tag objects	标签
description	String	描述
project_id	String	项目id
vpc_project_id	String	vpc项目id
enterprise_project_id	String	企业项目id

表 4-67 Tag

参数	参数类型	描述
key	String	键
value	String	值

状态码： 500

表 4-68 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

查询项目id为0b2179bbe180d3762fb0c01a2d5725c7，防护对象id为8839526e-b804-4a15-a082-a2c797dce633的东西向墙防护信息

```
https://{Endpoint}/v1/0b2179bbe180d3762fb0c01a2d5725c7/vpcs/protection?object_id=8839526e-b804-4a15-a082-a2c797dce633
```

响应示例

状态码： 200

查询东西向防护返回值

```
{
  "data": {
    "other_protect_vpcs": [],
    "other_total": 0,
    "protect_vpcs": [],
    "self_protect_vpcs": [],
    "self_total": 0,
    "total": 0,
    "total_assets": 5
  }
}
```

状态码： 400

Bad Request

```
{
  "error_code": "CFW.00109004",
  "error_msg": "HTTP请求错误"
}
```

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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class ListProtectedVpcsSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListProtectedVpcsRequest request = new ListProtectedVpcsRequest();
        request withObjectId("<object_id>");
        request withEnterpriseProjectId("<enterprise_project_id>");
        request withFwInstanceId("<fw_instance_id>");
        try {
            ListProtectedVpcsResponse response = client.listProtectedVpcs(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

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

try:
    request = ListProtectedVpcsRequest()
    request.object_id = "<object_id>"
    request.enterprise_project_id = "<enterprise_project_id>"
    request.fw_instance_id = "<fw_instance_id>"
    response = client.list_protected_vpcs(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListProtectedVpcsRequest{}
    request.ObjectId = "<object_id>"
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    response, err := client.ListProtectedVpcs(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	查询东西向防护返回值
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.2 EIP 管理

4.2.1 弹性 IP 开启关闭

功能介绍

开启关闭EIP，客户购买EIP后首次开启EIP防护前需使用ListEips同步EIP资产，sync字段设置为1。

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/eip/protect

表 4-69 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目ID 最小长度：32 最大长度：32

表 4-70 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id，用户支持企业项目后，由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id，创建云防火墙后用于标志防火墙由系统自动生成的标志id，可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下，fw_instance_id为空时，返回账号下第一个墙的信息；fw_instance_id非空时，返回与fw_instance_id对应墙的信息。

请求参数

表 4-71 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）

表 4-72 请求 Body 参数

参数	是否必选	参数类型	描述
object_id	是	String	防护对象id，是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id，可通过调用查询防火墙实例接口获得，注意type为0的为互联网边界防护对象id，type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。 最小长度：36 最大长度：36
status	是	Integer	EIP状态，0表示防护中，1表示未防护 最小值：0 最大值：1
ip_infos	是	Array of ip_infos objects	EIP信息列表 数组长度：0 - 50

表 4-73 ip_infos

参数	是否必选	参数类型	描述
id	否	String	弹性公网IP数据ID 最小长度：36 最大长度：36
public_ip	否	String	弹性公网IP地址 最小长度：0 最大长度：255
public_ipv6	否	String	弹性公网IP地址IPV6

响应参数

状态码：200

表 4-74 响应 Body 参数

参数	参数类型	描述
data	EIPSwitchStatusVO object	改变EIP防护状态返回值data

表 4-75 EIPSwitchStatusVO

参数	参数类型	描述
object_id	String	防护对象id，是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id，可通过调用查询防火墙实例接口获得，注意type为0的为互联网边界防护对象id，type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
fail_eip_id_list	Array of strings	修改eip防护状态失败列表。
id	String	ID

状态码： 400

表 4-76 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

- 开启项目id为857ddec2-55f2-4503-a93a-fe70021b743c，防护对象id为6d3db4fd-fd58-4d8e-914b-ef91aa268f62弹性公网IP数据id为4a589be0-b40a-4694-94ff-c0710af9a0a2的 EIP (100.85.121.62) 流量防护
<https://{Endpoint}/v1/857ddec2-55f2-4503-a93a-fe70021b743c/eip/protect>

```
{
  "object_id": "6d3db4fd-fd58-4d8e-914b-ef91aa268f62",
  "status": 0,
  "ip_infos": [ {
    "id": "4a589be0-b40a-4694-94ff-c0710af9a0a2",
    "public_ip": "1.2.3.4"
  } ]
}
```

- 关闭项目id为857ddec2-55f2-4503-a93a-fe70021b743c，防护对象id为6d3db4fd-fd58-4d8e-914b-ef91aa268f62弹性公网IP数据id为4a589be0-b40a-4694-94ff-c0710af9a0a2的 EIP (100.85.121.62) 流量防护
[/v1/857ddec2-55f2-4503-a93a-fe70021b743c/eip/protect](https://{Endpoint}/v1/857ddec2-55f2-4503-a93a-fe70021b743c/eip/protect)

```
{
  "object_id": "6d3db4fd-fd58-4d8e-914b-ef91aa268f62",
  "status": 1,
  "ip_infos": [ {
    "id": "4a589be0-b40a-4694-94ff-c0710af9a0a2",
  } ]
}
```

```
"public_ip" : "1.2.3.4"
}]
}
```

响应示例

状态码： 200

开启关闭EIP防护状态返回值

```
{
  "data" : {
    "fail_eip_id_list" : [],
    "object_id" : "ae42418e-f077-41a0-9d3b-5b2f5ad9102b",
    "id" : "b0a2dacc-3886-4805-838e-281653d3cd1f"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

- 开启项目id为857ddec2-55f2-4503-a93a-fe70021b743c，防护对象id为6d3db4fd-fd58-4d8e-914b-ef91aa268f62弹性公网IP数据id为4a589be0-b40a-4694-94ff-c0710af9a0a2的 EIP（100.85.121.62）流量防护

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

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

public class ChangeEipStatusSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();

        ChangeEipStatusRequest request = new ChangeEipStatusRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        EipOperateProtectReq body = new EipOperateProtectReq();
        List<EipOperateProtectReqInfos> listbodyInfos = new ArrayList<>();
```



```
listbodyIpInfos.add(
    new EipOperateProtectReqIpInfos()
        .withId("4a589be0-b40a-4694-94ff-c0710af9a0a2")
        .withPublicIp("1.2.3.4")
);
body.withIpInfos(listbodyIpInfos);
body.withStatus(0);
body.withObjectId("6d3db4fd-fd58-4d8e-914b-ef91aa268f62");
request.withBody(body);
try {
    ChangeEipStatusResponse response = client.changeEipStatus(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());
}
}
```

- 关闭项目id为857ddec2-55f2-4503-a93a-fe70021b743c，防护对象id为6d3db4fd-fd58-4d8e-914b-ef91aa268f62弹性公网IP数据id为4a589be0-b40a-4694-94ff-c0710af9a0a2的 EIP（100.85.121.62）流量防护
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

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

public class ChangeEipStatusSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();

        ChangeEipStatusRequest request = new ChangeEipStatusRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        EipOperateProtectReq body = new EipOperateProtectReq();
        List<EipOperateProtectReqIpInfos> listbodyIpInfos = new ArrayList<>();
        listbodyIpInfos.add(
```

```
        new EipOperateProtectReqIpInfos()
            .withId("4a589be0-b40a-4694-94ff-c0710af9a0a2")
            .withPublicIp("1.2.3.4")
    );
    body.withIpInfos(listbodyIpInfos);
    body.withStatus(1);
    body.withObjectId("6d3db4fd-fd58-4d8e-914b-ef91aa268f62");
    request.withBody(body);
    try {
        ChangeEipStatusResponse response = client.changeEipStatus(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

- 开启项目id为857ddec2-55f2-4503-a93a-fe70021b743c，防护对象id为6d3db4fd-fd58-4d8e-914b-ef91aa268f62弹性公网IP数据id为4a589be0-b40a-4694-94ff-c0710af9a0a2的 EIP (100.85.121.62) 流量防护

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ChangeEipStatusRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        listIpInfosbody = [
            EipOperateProtectReqIpInfos(
                id="4a589be0-b40a-4694-94ff-c0710af9a0a2",
                public_ip="1.2.3.4"
            )
        ]
        request.body = EipOperateProtectReq(
            ip_infos=listIpInfosbody,
            status=0,
            object_id="6d3db4fd-fd58-4d8e-914b-ef91aa268f62"
```

```
)  
response = client.change_eip_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)
```

- 关闭项目id为857ddec2-55f2-4503-a93a-fe70021b743c，防护对象id为6d3db4fd-fd58-4d8e-914b-ef91aa268f62弹性公网IP数据id为4a589be0-b40a-4694-94ff-c0710af9a0a2的 EIP (100.85.121.62) 流量防护

coding: utf-8

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")  
    sk = __import__('os').getenv("CLOUD_SDK_SK")
```

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

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

```
try:
```

```
    request = ChangeEipStatusRequest()  
    request.enterprise_project_id = "<enterprise_project_id>"  
    request.fw_instance_id = "<fw_instance_id>"  
    listIpInfosbody = [  
        EipOperateProtectReqIpInfos(  
            id="4a589be0-b40a-4694-94ff-c0710af9a0a2",  
            public_ip="1.2.3.4"  
        )  
    ]
```

```
    request.body = EipOperateProtectReq(  
        ip_infos=listIpInfosbody,  
        status=1,  
        object_id="6d3db4fd-fd58-4d8e-914b-ef91aa268f62"  
    )
```

```
    response = client.change_eip_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

- 开启项目id为857ddec2-55f2-4503-a93a-fe70021b743c，防护对象id为6d3db4fd-fd58-4d8e-914b-ef91aa268f62弹性公网IP数据id为4a589be0-b40a-4694-94ff-c0710af9a0a2的 EIP (100.85.121.62) 流量防护

```
package main
```

```
import (  
    "fmt"
```

```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ChangeEipStatusRequest{}
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceldRequest := "<fw_instance_id>"
    request.FwInstanceld = &fwInstanceldRequest
    idIpInfos := "4a589be0-b40a-4694-94ff-c0710af9a0a2"
    publicIpInfos := "1.2.3.4"
    var listIpInfosbody = []model.EipOperateProtectReqIpInfos{
        {
            Id: &idIpInfos,
            PublicIp: &publicIpInfos,
        },
    }
    request.Body = &model.EipOperateProtectReq{
        IpInfos: listIpInfosbody,
        Status: int32(0),
        Objectid: "6d3db4fd-fd58-4d8e-914b-ef91aa268f62",
    }
    response, err := client.ChangeEipStatus(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
}
```

- 关闭项目id为857ddec2-55f2-4503-a93a-fe70021b743c，防护对象id为6d3db4fd-fd58-4d8e-914b-ef91aa268f62弹性公网IP数据id为4a589be0-b40a-4694-94ff-c0710af9a0a2的 EIP (100.85.121.62) 流量防护

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ChangeEipStatusRequest{}
enterpriseProjectIdRequest:= "<enterprise_project_id>"
request.EnterpriseProjectId = &enterpriseProjectIdRequest
fwInstanceIdRequest:= "<fw_instance_id>"
request.FwInstanceId = &fwInstanceIdRequest
idIpInfos:= "4a589be0-b40a-4694-94ff-c0710af9a0a2"
publicIpInfos:= "1.2.3.4"
var listIpInfosbody = []model.EipOperateProtectReqIpInfos{
    {
        Id: &idIpInfos,
        PublicIp: &publicIpInfos,
    },
}
request.Body = &model.EipOperateProtectReq{
    IpInfos: listIpInfosbody,
    Status: int32(1),
    Objectid: "6d3db4fd-fd58-4d8e-914b-ef91aa268f62",
}
response, err := client.ChangeEipStatus(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	开启关闭EIP防护状态返回值
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.2.2 查询 Eip 个数

功能介绍

查询Eip个数

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/eip-count/{object_id}

表 4-77 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目ID 最小长度：32 最大长度：32
object_id	是	String	防护对象id，是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id，可通过调用查询防火墙实例接口获得，注意type为0的为互联网边界防护对象id，type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。 最小长度：36 最大长度：36

表 4-78 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id，用户支持企业项目后，由企业项目生成的id。

参数	是否必选	参数类型	描述
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

请求参数

表 4-79 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）

响应参数

状态码： 200

表 4-80 响应 Body 参数

参数	参数类型	描述
data	EipCountRespData object	Eip个数数据

表 4-81 EipCountRespData

参数	参数类型	描述
object_id	String	防护对象id, 是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id, 可通过调用查询防火墙实例接口获得, 注意type为0的为互联网边界防护对象id, type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。 最小长度： 36 最大长度： 36

参数	参数类型	描述
eip_total	Integer	EIP总数 最小值：0 缺省值：0
eip_protected	Integer	该账号下所有墙防护EIP总数量 最小值：0 缺省值：0
eip_protected_self	Integer	该当前防火墙防护EIP数量

状态码：400

表 4-82 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度：8 最大长度：36
error_msg	String	错误描述 最小长度：2 最大长度：512

请求示例

查询项目id为9d80d070b6d44942af73c9c3d38e0429，防护对象id为cfebd347-b655-4b84-b938-3c54317599b2的eip个数

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/eip-count/cfebd347-b655-4b84-b938-3c54317599b2
```

响应示例

状态码：200

OK

```
{  
  "data": {  
    "eip_protected": 1,  
    "eip_protected_self": 4,  
    "eip_total": 5,  
    "object_id": "6d3db4fd-fd58-4d8e-914b-ef91aa268f62"  
  }  
}
```

状态码：400

Bad Request

```
{
  "error_code" : "CFW.00200005",
  "error_msg" : "操作内容不存在"
}
```

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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class ListEipCountSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListEipCountRequest request = new ListEipCountRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        try {
            ListEipCountResponse response = client.listEipCount(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListEipCountRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        response = client.list_eip_count(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListEipCountRequest{}
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
```

```
request.FwInstanceId = &fwInstanceIdRequest
response, err := client.ListEipCount(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	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.2.3 弹性 IP 列表查询

功能介绍

弹性IP列表查询

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/eips/protect

表 4-83 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目ID 最小长度：32 最大长度：32

表 4-84 Query 参数

参数	是否必选	参数类型	描述
object_id	是	String	防护对象id，是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id，可通过调用查询防火墙实例接口获得，注意type为0的为互联网边界防护对象id，type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。 最小长度：36 最大长度：36
key_word	否	String	弹性公网ID/弹性公网IP 最小长度：0 最大长度：255
status	否	String	防护状态 null-全部 0-开启防护 1-关闭防护 枚举值： <ul style="list-style-type: none">• null• 0• 1
sync	否	Integer	是否同步租户EIP数据 0-不同步 1-同步 枚举值： <ul style="list-style-type: none">• 0• 1
limit	是	Integer	每页显示个数，范围为1-1024 最小值：0
offset	是	Integer	偏移量：指定返回记录的开始位置，必须为数字，取值范围为大于或等于0，默认0 最小值：0

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
device_key	否	String	设备键
address_type	否	Integer	地址类型0 ipv4,1 ipv6
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考API Explorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。
fw_key_word	否	String	所绑定防火墙id防火墙名称
eps_id	否	String	弹性公网ip的企业项目id
tags	否	String	标签列表信息

请求参数

表 4-85 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）

响应参数

状态码： 200

表 4-86 响应 Body 参数

参数	参数类型	描述
data	EipResponse Data object	查询eip返回data

表 4-87 EipResponseData

参数	参数类型	描述
limit	Integer	每页显示个数，范围为1-1024
offset	Integer	偏移量：指定返回记录的开始位置，必须为数字，取值范围为大于或等于0，默认0
total	Integer	总数
records	Array of EipResource objects	eip资源记录

表 4-88 EipResource

参数	参数类型	描述
id	String	弹性公网ID
public_ip	String	弹性公网IP
status	Integer	EIP防护状态，0表示防护中，1表示未防护 枚举值： <ul style="list-style-type: none">• 0• 1
public_ipv6	String	弹性公网IP,IPV6
enterprise_project_id	String	企业项目id
device_id	String	设备id
device_name	String	设备名称
device_owner	String	设备拥有者
associate_instance_type	String	关联实例类型
fw_instance_name	String	防火墙名称
fw_instance_id	String	防火墙实例id，创建云防火墙后用于标志防火墙由系统自动生成的标志id，可通过调用查询防火墙实例接口获得。具体可参考API Explorer和帮助中心FAQ。
fw_enterprise_project_id	String	Eip绑定的防火墙企业项目id

参数	参数类型	描述
object_id	String	防护对象id，是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id，可通过调用查询防火墙实例接口获得，注意type为0的为互联网边界防护对象id，type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
tags	String	标签列表
domain_id	String	EIP所属用户
owner	String	所属用户的名称
fw_domain_id	String	防火墙所属用户

状态码： 400

表 4-89 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

查询项目id为9d80d070b6d44942af73c9c3d38e0429，防护对象id为cfebd347-b655-4b84-b938-3c54317599b2，非同步的第一页的数据

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/eips/protect?object_id=cfebd347-b655-4b84-b938-3c54317599b2&limit=10&offset=0&sync=0
```

响应示例

状态码： 200

EIP数据查询返回值

```
{
  "data": {
    "limit": 200,
    "offset": 0,
    "records": [ {
      "associate_instance_type": "PORT",
      "device_id": "c87579ab-c76a-4afd-83ce-62e0f531f13e",
      "device_name": "test",
      "device_owner": "compute:cn-north-7c",
```

```
"domain_id": "7d07807209524a4280266db9df63c4fa",
"enterprise_project_id": "0",
"fw_domain_id": "7d07807209524a4280266db9df63c4fa",
"fw_enterprise_project_id": "default",
"fw_instance_id": "546af3f8-88e9-47f2-a205-2346d7090925",
"fw_instance_name": "test",
"id": "465b34fe-e017-4831-a21c-9c6c753bb1f2",
"object_id": "ae42418e-f077-41a0-9d3b-5b2f5ad9102b",
"public_ip": "100.85.223.15",
"status": 0,
"tags": "combined_order_id=CBRC523040615138M2KW912"
}],
"total": 1
}
```

状态码： 400

Bad Request

```
{
  "error_code": "CFW.00109004",
  "error_msg": "HTTP请求错误"
}
```

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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class ListEipsSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();

        ListEipsRequest request = new ListEipsRequest();
        request.withObjectId("<object_id>");
        request.withKeyword("<key_word>");
        request.withStatus(ListEipsRequest.StatusEnum.fromValue("<status>"));
        request.withSync(ListEipsRequest.SyncEnum.NUMBER_<sync>);
        request.withLimit(<limit>);
    }
}
```



```
request.withOffset(<offset>);
request.withEnterpriseProjectId("<enterprise_project_id>");
request.withDeviceKey("<device_key>");
request.withAddressType("<address_type>");
request.withFwInstanceId("<fw_instance_id>");
request.withFwKeyword("<fw_key_word>");
request.withEpsId("<eps_id>");
request.withTags("<tags>");
try {
    ListEipsResponse response = client.listEips(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListEipsRequest()
        request.object_id = "<object_id>"
        request.key_word = "<key_word>"
        request.status = "<status>"
        request.sync = <sync>
        request.limit = <limit>
        request.offset = <offset>
        request.enterprise_project_id = "<enterprise_project_id>"
        request.device_key = "<device_key>"
        request.address_type = <address_type>
        request.fw_instance_id = "<fw_instance_id>"
        request.fw_key_word = "<fw_key_word>"
        request.eps_id = "<eps_id>"
        request.tags = "<tags>"
        response = client.list_eips(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListEipsRequest{}
    request.ObjectId = "<object_id>"
    keywordRequest := "<key_word>"
    request.KeyWord = &keywordRequest
    statusRequest := model.GetListEipsRequestStatusEnum().<STATUS>
    request.Status = &statusRequest
    syncRequest := model.GetListEipsRequestSyncEnum().<SYNC>
    request.Sync = &syncRequest
    request.Limit = int32(<limit>)
    request.Offset = int32(<offset>)
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    deviceKeyRequest := "<device_key>"
    request.DeviceKey = &deviceKeyRequest
    addressTypeRequest := int32(<address_type>)
    request.AddressType = &addressTypeRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    fwKeywordRequest := "<fw_key_word>"
    request.FwKeyWord = &fwKeywordRequest
    epsIdRequest := "<eps_id>"
    request.EpsId = &epsIdRequest
    tagsRequest := "<tags>"
    request.Tags = &tagsRequest
    response, err := client.ListEips(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	EIP数据查询返回值
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.3 ACL 规则管理

4.3.1 创建 ACL 规则

功能介绍

创建ACL规则

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/acl-rule

表 4-90 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-91 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

请求参数

表 4-92 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）

表 4-93 请求 Body 参数

参数	是否必选	参数类型	描述
object_id	是	String	防护对象id, 是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id, 可通过调用查询防火墙实例接口获得, 注意type为0的为互联网边界防护对象id, type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
type	是	Integer	规则type, 0: 互联网规则, 1: vpc规则, 2: nat规则 枚举值: <ul style="list-style-type: none">• 0• 1• 2

参数	是否必选	参数类型	描述
rules	是	Array of rules objects	rules

表 4-94 rules

参数	是否必选	参数类型	描述
name	是	String	规则名称
sequence	是	OrderRuleAcl Dto object	规则序列
address_type	是	Integer	地址类型, 0 ipv4,1 ipv6,2 domain 枚举值: <ul style="list-style-type: none">• 0• 1• 2
action_type	是	Integer	动作0: permit,1: deny
status	是	Integer	规则下发状态 0: 禁用,1: 启用 枚举值: <ul style="list-style-type: none">• 0• 1
applications	否	Array of strings	应用列表
applicationsJsonString	否	String	应用列表转化json字符串
long_connect_time	否	Long	长连接时长
long_connect_time_hour	否	Long	长连接时长小时
long_connect_time_minute	否	Long	长连接时长分钟
long_connect_time_second	否	Long	长连接时长秒
long_connect_enable	是	Integer	是否支持长连接, 0表示不支持长连接, 1表示支持长连接 枚举值: <ul style="list-style-type: none">• 0• 1

参数	是否必选	参数类型	描述
description	否	String	描述
direction	否	Integer	方向：0表示外到内，1表示内到外【说明：规则type=0：互联网规则 2：nat规则时方向值必填】 枚举值： <ul style="list-style-type: none">• 0• 1
profile	否	RuleProfileDto object	域名url信息
source	是	RuleAddressDtoForRequest object	源地址传输对象
destination	是	RuleAddressDtoForRequest object	目的地址传输对象
service	是	RuleServiceDto object	服务对象
tag	否	TagsVO object	标签显示值

表 4-95 OrderRuleAclDto

参数	是否必选	参数类型	描述
dest_rule_id	否	String	目标规则id，添加规则位于此规则之后，非置顶时不能为空，置顶时为空
top	否	Integer	是否置顶，0代表非置顶，1代表置顶
bottom	否	Integer	是否置底，0代表非置底，1代表置底

表 4-96 RuleProfileDto

参数	是否必选	参数类型	描述
url	否	String	域名url

表 4-97 RuleAddressDtoForRequest

参数	是否必选	参数类型	描述
type	是	Integer	源类型0手工输入,1关联IP地址组,2域名,3地理位置,4域名组,5多对象,6域名组-DNS解析,7域名组-应用型。
address_type	否	Integer	源类型0 ipv4,1 ipv6
address	否	String	源IP,手动类型不能为空,自动及domain类型为空
address_set_id	否	String	关联IP地址组ID,自动类型不能为空,手动类型合domain类型为空
address_set_name	否	String	地址组名称
domain_address_name	否	String	域名地址名称,域名类型时不能为空,手动类型及自动类型时为空
region_list_json	否	String	规则region列表json值
region_list	否	Array of IpRegionDto objects	规则region列表
domain_set_id	否	String	域名组id
domain_set_name	否	String	域名组名称
ip_address	否	Array of strings	IP地址列表
address_set_type	否	Integer	地址组类型,0表示自定义地址组,1表示WAF回源IP地址组,2表示DDoS回源IP地址组,3表示NAT64转换地址组
predefined_group	否	Array of strings	预定义地址组列表

表 4-98 IpRegionDto

参数	是否必选	参数类型	描述
region_id	否	String	区域id
description_cn	否	String	中文描述

参数	是否必选	参数类型	描述
description_en	否	String	英文描述
region_type	否	Integer	区域类型，0表示国家，1表示省份，2表示大洲

表 4-99 RuleServiceDto

参数	是否必选	参数类型	描述
type	是	Integer	服务输入类型，0为手动输入类型，1为自动输入类型
protocol	否	Integer	协议类型:TCP为6, UDP为17,ICMP为1,ICMPV6为58,ANY为-1,手动类型不为空，自动类型为空
protocols	否	Array of integers	协议列表，协议类型:TCP为6, UDP为17,ICMP为1,ICMPV6为58,ANY为-1,手动类型不为空，自动类型为空
source_port	否	String	源端口
dest_port	否	String	目的端口
service_set_id	否	String	服务组id，手动类型为空，自动类型为非空
service_set_name	否	String	服务组名称
custom_service	否	Array of ServiceItem objects	自定义服务
predefined_group	否	Array of strings	预定义服务组列表
service_group	否	Array of strings	服务组列表
service_group_names	否	Array of AddressGroupVO objects	服务组名称列表
service_set_type	否	Integer	服务组类型，0表示自定义服务组，1表示常用WEB服务，2表示常用远程登录和PING，3表示常用数据库

表 4-100 ServiceItem

参数	是否必选	参数类型	描述
protocol	否	Integer	协议类型:TCP为6, UDP为17,ICMP为1,ICMPV6为58,ANY为-1,手动类型不为空, 自动类型为空
source_port	否	String	源端口
dest_port	否	String	目的端口
description	否	String	服务成员描述
name	否	String	服务成员名称

表 4-101 AddressGroupVO

参数	是否必选	参数类型	描述
set_id	否	String	地址组id
name	否	String	地址组名称
protocols	否	Array of integers	协议列表, 协议类型:TCP为6, UDP为17,ICMP为1,ICMPV6为58,ANY为-1,手动类型不为空, 自动类型为空
service_set_type	否	Integer	服务组类型, 0表示自定义服务组, 1表示常用WEB服务, 2表示常用远程登录和PING, 3表示常用数据库

表 4-102 TagsVO

参数	是否必选	参数类型	描述
tag_id	否	String	标签id
tag_key	否	String	标签键
tag_value	否	String	标签值

响应参数

状态码: 200

表 4-103 响应 Body 参数

参数	参数类型	描述
data	RuleIdList object	规则id列表

表 4-104 RuleIdList

参数	参数类型	描述
rules	Array of RuleId objects	规则id列表

表 4-105 RuleId

参数	参数类型	描述
id	String	id
name	String	名称

状态码： 400

表 4-106 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

示例为添加一个IPv4类型的外到内的规则，名称为测试规则，源类型为IP地址，地址为1.1.1.1，目的类型为IP地址，目的地址类型为2.2.2.2，服务类型为服务，协议类型为TCP，源端口为0，目的端口为0，不支持长连接，动作为放行，启用状态为启用

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/acl-rule
```

```
{  
  "object_id": "ae42418e-f077-41a0-9d3b-5b2f5ad9102b",  
  "rules": [ {
```

```
"name": "测试规则",
"status": 1,
"action_type": 0,
"description": "",
"source": {
  "type": 0,
  "address": "1.1.1.1"
},
"destination": {
  "type": 0,
  "address": "2.2.2.2"
},
"service": {
  "type": 0,
  "protocol": 6,
  "source_port": "0",
  "dest_port": "0"
},
"address_type": 0,
"tag": {
  "tag_key": "",
  "tag_value": ""
},
"long_connect_enable": 0,
"direction": 0,
"sequence": {
  "top": 1,
  "dest_rule_id": null
}
}],
"type": 0
}
```

响应示例

状态码： 200

添加acl响应

```
{
  "data": {
    "rules": [ [ {
      "id": "0475c516-0e41-4caf-990b-0c504eebd73f"
    } ] ]
  }
}
```

状态码： 400

Bad Request

```
{
  "error_code": "CFW.00900016",
  "error_msg": "导入任务进行中，请任务结束后再操作"
}
```

SDK 代码示例

SDK代码示例如下。

Java

示例为添加一个IPv4类型的外到内的规则，名称为测试规则，源类型为IP地址，地址为1.1.1.1，目的类型为IP地址，目的地址类型为2.2.2.2，服务类型为服务，协议类型为TCP，源端口为0，目的端口为0，不支持长连接，动作为放行，启用状态为启用

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

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

public class AddAclRuleSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        AddAclRuleRequest request = new AddAclRuleRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        AddRuleAclDto body = new AddRuleAclDto();
        TagsVO tagRules = new TagsVO();
        tagRules.withTagKey("")
            .withTagValue("");
        RuleServiceDto serviceRules = new RuleServiceDto();
        serviceRules.withType(0)
            .withProtocol(6)
            .withSourcePort("0")
            .withDestPort("0");
        RuleAddressDto destinationRules = new RuleAddressDto();
        destinationRules.withType(0)
            .withAddress("2.2.2.2");
        RuleAddressDto sourceRules = new RuleAddressDto();
        sourceRules.withType(0)
            .withAddress("1.1.1.1");
        OrderRuleAclDto sequenceRules = new OrderRuleAclDto();
        sequenceRules.withTop(1);
        List<AddRuleAclDtoRules> listbodyRules = new ArrayList<>();
        listbodyRules.add(
            new AddRuleAclDtoRules()
                .withName("测试规则")
                .withSequence(sequenceRules)
                .withAddressType(AddRuleAclDtoRules.AddressTypeEnum.NUMBER_0)
                .withActionType(0)
                .withStatus(AddRuleAclDtoRules.StatusEnum.NUMBER_1)
                .withLongConnectEnable(AddRuleAclDtoRules.LongConnectEnableEnum.NUMBER_0)
                .withDescription("")
                .withDirection(AddRuleAclDtoRules.DirectionEnum.NUMBER_0)
                .withSource(sourceRules)
                .withDestination(destinationRules)
                .withService(serviceRules)
                .withTag(tagRules)
        );
    }
}
```

```
body.withRules(listbodyRules);
body.withType(AddRuleAclDto.TypeEnum.NUMBER_0);
body.withObjectId("ae42418e-f077-41a0-9d3b-5b2f5ad9102b");
request.withBody(body);
try {
    AddAclRuleResponse response = client.addAclRule(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

示例为添加一个IPv4类型的外到内的规则，名称为测试规则，源类型为IP地址，地址为1.1.1.1，目的类型为IP地址，目的地址类型为2.2.2.2，服务类型为服务，协议类型为TCP，源端口为0，目的端口为0，不支持长连接，动作为放行，启用状态为启用

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = AddAclRuleRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        tagRules = TagsVO(
            tag_key="",
            tag_value=""
        )
        serviceRules = RuleServiceDto(
            type=0,
            protocol=6,
            source_port="0",
            dest_port="0"
        )
        destinationRules = RuleAddressDto(
            type=0,
            address="2.2.2.2"
        )
```

```
sourceRules = RuleAddressDto(
    type=0,
    address="1.1.1.1"
)
sequenceRules = OrderRuleAclDto(
    top=1
)
listRulesbody = [
    AddRuleAclDtoRules(
        name="测试规则",
        sequence=sequenceRules,
        address_type=0,
        action_type=0,
        status=1,
        long_connect_enable=0,
        description="",
        direction=0,
        source=sourceRules,
        destination=destinationRules,
        service=serviceRules,
        tag=tagRules
    )
]
request.body = AddRuleAclDto(
    rules=listRulesbody,
    type=0,
    object_id="ae42418e-f077-41a0-9d3b-5b2f5ad9102b"
)
response = client.add_acl_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

示例为添加一个IPv4类型的外到内的规则，名称为测试规则，源类型为IP地址，地址为1.1.1.1，目的类型为IP地址，目的地址类型为2.2.2.2，服务类型为服务，协议类型为TCP，源端口为0，目的端口为0，不支持长连接，动作为放行，启用状态为启用

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
```

```
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build()

request := &model.AddAclRuleRequest{}
enterpriseProjectIdRequest:= "<enterprise_project_id>"
request.EnterpriseProjectId = &enterpriseProjectIdRequest
fwInstanceIdRequest:= "<fw_instance_id>"
request.FwInstanceId = &fwInstanceIdRequest
tagKeyTag:= ""
tagValueTag:= ""
tagRules := &model.TagsVo{
    TagKey: &tagKeyTag,
    TagValue: &tagValueTag,
}
protocolService:= int32(6)
sourcePortService:= "0"
destPortService:= "0"
serviceRules := &model.RuleServiceDto{
    Type: int32(0),
    Protocol: &protocolService,
    SourcePort: &sourcePortService,
    DestPort: &destPortService,
}
addressDestination:= "2.2.2.2"
destinationRules := &model.RuleAddressDto{
    Type: int32(0),
    Address: &addressDestination,
}
addressSource:= "1.1.1.1"
sourceRules := &model.RuleAddressDto{
    Type: int32(0),
    Address: &addressSource,
}
topSequence:= int32(1)
sequenceRules := &model.OrderRuleAclDto{
    Top: &topSequence,
}
descriptionRules:= ""
directionRules:= model.GetAddRuleAclDtoRulesDirectionEnum().E_0
var listRulesbody = []model.AddRuleAclDtoRules{
    {
        Name: "测试规则",
        Sequence: sequenceRules,
        AddressType: model.GetAddRuleAclDtoRulesAddressTypeEnum().E_0,
        ActionType: int32(0),
        Status: model.GetAddRuleAclDtoRulesStatusEnum().E_1,
        LongConnectEnable: model.GetAddRuleAclDtoRulesLongConnectEnableEnum().E_0,
        Description: &descriptionRules,
        Direction: &directionRules,
        Source: sourceRules,
        Destination: destinationRules,
        Service: serviceRules,
        Tag: tagRules,
    },
}
request.Body = &model.AddRuleAclDto{
    Rules: listRulesbody,
    Type: model.GetAddRuleAclDtoTypeEnum().E_0,
    ObjectId: "ae42418e-f077-41a0-9d3b-5b2f5ad9102b",
}
response, err := client.AddAclRule(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	添加acl响应
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.3.2 删除 ACL 规则

功能介绍

删除ACL规则组

调用方法

请参见[如何调用API](#)。

URI

DELETE /v1/{project_id}/acl-rule/{acl_rule_id}

表 4-107 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目Id
acl_rule_id	是	String	规则Id

表 4-108 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考 API Explorer 和帮助中心FAQ。默认情况下, fw_instance_id 为空时, 返回账号下第一个墙的信息; fw_instance_id 非空时, 返回与fw_instance_id 对应墙的信息。

请求参数

表 4-109 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）

响应参数

状态码： 200

表 4-110 响应 Body 参数

参数	参数类型	描述
data	RuleId object	

表 4-111 RuleId

参数	参数类型	描述
id	String	id
name	String	名称

状态码： 400

表 4-112 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

删除项目Id： 9d80d070b6d44942af73c9c3d38e0429的ruleId： ceaa0407-b9c8-4dfd-9eca-b6ead2dfd031的规则

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/acl-rule/ceaa0407-b9c8-4dfd-9eca-b6ead2dfd031
```

响应示例

状态码： 200

OK

```
{
  "data" : {
    "id" : "ceaa0407-b9c8-4dfd-9eca-b6ead2dfd031",
    "name" : "name"
  }
}
```

状态码： 400

Bad Request

```
{
  "error_code" : "CFW.00900016",
  "error_msg" : "导入任务进行中，请任务结束后再操作"
}
```

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.cfw.v1.region.CfwRegion;
```

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

public class DeleteAclRuleSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteAclRuleRequest request = new DeleteAclRuleRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        try {
            DeleteAclRuleResponse response = client.deleteAclRule(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 huaweicloudsdkcfww.v1.region.cfww_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfww.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
```

```
request = DeleteAclRuleRequest()
request.enterprise_project_id = "<enterprise_project_id>"
request.fw_instance_id = "<fw_instance_id>"
response = client.delete_acl_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/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteAclRuleRequest{}
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    response, err := client.DeleteAclRule(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	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.3.3 批量删除 Acl 规则

功能介绍

批量删除Acl规则

调用方法

请参见[如何调用API](#)。

URI

DELETE /v1/{project_id}/acl-rule

表 4-113 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目ID

表 4-114 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。

参数	是否必选	参数类型	描述
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

请求参数

表 4-115 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）

表 4-116 请求 Body 参数

参数	是否必选	参数类型	描述
object_id	是	String	防护对象id, 是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id, 可通过调用查询防火墙实例接口获得, 注意type为0的为互联网边界防护对象id, type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
rule_ids	是	Array of strings	规则ID列表

响应参数

状态码： 200

表 4-117 响应 Body 参数

参数	参数类型	描述
data	BatchDeleteAclRulesResponseData object	批量删除ACL规则返回值data

表 4-118 BatchDeleteAclRulesResponseData

参数	参数类型	描述
responseData	Array of BatchDeleteRuleInfo objects	批量删除规则返回data

表 4-119 BatchDeleteRuleInfo

参数	参数类型	描述
name	String	acl名称
id	String	aclId

请求示例

删除项目id为9d80d070b6d44942af73c9c3d38e0429的项目下的防护对象 id为ae42418e-f077-41a0-9d3b-5b2f5ad9102b的规则，规则id为0475c516-0e41-4caf-990b-0c504eebd73f和8662868e-fe7e-4dfc-bfb1-ca4d73081ca6

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/acl-rule
{
  "rule_ids" : [ "0475c516-0e41-4caf-990b-0c504eebd73f", "8662868e-fe7e-4dfc-bfb1-ca4d73081ca6" ],
  "object_id" : "ae42418e-f077-41a0-9d3b-5b2f5ad9102b"
}
```

响应示例

状态码： 200

批量删除规则返回值

```
{
  "data" : {
    "responseData" : [ {
      "name" : "test",
      "id" : "0475c516-0e41-4caf-990b-0c504eebd73f"
    }, {
      "name" : "test2",
```

```
"id" : "8662868e-fe7e-4dfc-bfb1-ca4d73081ca6"  
  }  
}  
}
```

SDK 代码示例

SDK代码示例如下。

Java

删除项目id为9d80d070b6d44942af73c9c3d38e0429的项目下的防护对象 id为ae42418e-f077-41a0-9d3b-5b2f5ad9102b的规则，规则id为0475c516-0e41-4caf-990b-0c504eebd73f和8662868e-fe7e-4dfc-bfb1-ca4d73081ca6

```
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.cfw.v1.region.CfwRegion;  
import com.huaweicloud.sdk.cfw.v1.*;  
import com.huaweicloud.sdk.cfw.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class BatchDeleteAclRulesSolution {  
  
    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);  
  
        CfwClient client = CfwClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))  
            .build();  
        BatchDeleteAclRulesRequest request = new BatchDeleteAclRulesRequest();  
        request.withEnterpriseProjectId("<enterprise_project_id>");  
        request.withFwInstanceId("<fw_instance_id>");  
        DeleteRuleAclDto body = new DeleteRuleAclDto();  
        List<String> listbodyRuleIds = new ArrayList<>();  
        listbodyRuleIds.add("0475c516-0e41-4caf-990b-0c504eebd73f");  
        listbodyRuleIds.add("8662868e-fe7e-4dfc-bfb1-ca4d73081ca6");  
        body.withRuleIds(listbodyRuleIds);  
        body.withObjectId("ae42418e-f077-41a0-9d3b-5b2f5ad9102b");  
        request.withBody(body);  
        try {  
            BatchDeleteAclRulesResponse response = client.batchDeleteAclRules(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.getStatusCode());
System.out.println(e.getRequestId());
System.out.println(e.getErrorCode());
System.out.println(e.getErrorMsg());
    }
}
}
```

Python

删除项目id为9d80d070b6d44942af73c9c3d38e0429的项目下的防护对象 id为ae42418e-f077-41a0-9d3b-5b2f5ad9102b的规则，规则id为0475c516-0e41-4caf-990b-0c504eebd73f和8662868e-fe7e-4dfc-bfb1-ca4d73081ca6

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = BatchDeleteAclRulesRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        listRuleIdsbody = [
            "0475c516-0e41-4caf-990b-0c504eebd73f",
            "8662868e-fe7e-4dfc-bfb1-ca4d73081ca6"
        ]
        request.body = DeleteRuleAclDto(
            rule_ids=listRuleIdsbody,
            object_id="ae42418e-f077-41a0-9d3b-5b2f5ad9102b"
        )
        response = client.batch_delete_acl_rules(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

删除项目id为9d80d070b6d44942af73c9c3d38e0429的项目下的防护对象 id为ae42418e-f077-41a0-9d3b-5b2f5ad9102b的规则，规则id为0475c516-0e41-4caf-990b-0c504eebd73f和8662868e-fe7e-4dfc-bfb1-ca4d73081ca6

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteAclRulesRequest{}
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    var listRuleIdsbody = []string{
        "0475c516-0e41-4caf-990b-0c504eebd73f",
        "8662868e-fe7e-4dfc-bfb1-ca4d73081ca6",
    }
    request.Body = &model.DeleteRuleAclDto{
        RuleIds: listRuleIdsbody,
        Objectid: "ae42418e-f077-41a0-9d3b-5b2f5ad9102b",
    }
    response, err := client.BatchDeleteAclRules(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	批量删除规则返回值

错误码

请参见[错误码](#)。

4.3.4 删除规则击中次数

功能介绍

清除规则击中次数

调用方法

请参见[如何调用API](#)。

URI

DELETE /v1/{project_id}/acl-rule/count

表 4-120 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	project id

表 4-121 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考API Explorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

请求参数

表 4-122 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）

表 4-123 请求 Body 参数

参数	是否必选	参数类型	描述
rule_ids	是	Array of strings	规则id列表

响应参数

状态码： 400

表 4-124 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

清除项目id为0b2179bbe180d3762fb0c01a2d5725c7中acl规则id为59ff6bd9-0a76-41ec-9650-380086069965的命中次数

```
https://{Endpoint}/v1/0b2179bbe180d3762fb0c01a2d5725c7/acl-rule/count  
  
{  
  "rule_ids" : [ "59ff6bd9-0a76-41ec-9650-380086069965" ]  
}
```

响应示例

状态码： 200

OK

{ }

状态码： 400

Bad Request

```
{
  "error_code" : "CFW.00400006",
  "error_msg" : "清除规则击中次数参数错误"
}
```

SDK 代码示例

SDK代码示例如下。

Java

清除项目id为0b2179bbe180d3762fb0c01a2d5725c7中acl规则id为59ff6bd9-0a76-41ec-9650-380086069965的命中次数

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

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

public class DeleteAclRuleHitCountSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteAclRuleHitCountRequest request = new DeleteAclRuleHitCountRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        ClearAccessLogRuleHitCountsDto body = new ClearAccessLogRuleHitCountsDto();
        List<String> listbodyRuleIds = new ArrayList<>();
        listbodyRuleIds.add("59ff6bd9-0a76-41ec-9650-380086069965");
        body.withRuleIds(listbodyRuleIds);
        request.withBody(body);
        try {
            DeleteAclRuleHitCountResponse response = client.deleteAclRuleHitCount(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

清除项目id为0b2179bbe180d3762fb0c01a2d5725c7中acl规则id为59ff6bd9-0a76-41ec-9650-380086069965的命中次数

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = DeleteAclRuleHitCountRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        listRuleIdsbody = [
            "59ff6bd9-0a76-41ec-9650-380086069965"
        ]
        request.body = ClearAccessLogRuleHitCountsDto(
            rule_ids=listRuleIdsbody
        )
        response = client.delete_acl_rule_hit_count(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

清除项目id为0b2179bbe180d3762fb0c01a2d5725c7中acl规则id为59ff6bd9-0a76-41ec-9650-380086069965的命中次数

```
package main

import (
```

```
"fmt"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteAclRuleHitCountRequest{}
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    var listRuleIdsbody = []string{
        "59ff6bd9-0a76-41ec-9650-380086069965",
    }
    request.Body = &model.ClearAccessLogRuleHitCountsDto{
        RuleIds: listRuleIdsbody,
    }
    response, err := client.DeleteAclRuleHitCount(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	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found

状态码	描述
500	Internal Server Error

错误码

请参见[错误码](#)。

4.3.5 更新 ACL 规则

功能介绍

更新ACL规则

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/acl-rule/{acl_rule_id}

表 4-125 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id
acl_rule_id	是	String	规则Id

表 4-126 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考API Explorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

请求参数

表 4-127 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）

表 4-128 请求 Body 参数

参数	是否必选	参数类型	描述
address_type	否	Integer	地址类型，0 ipv4,1 ipv6 枚举值： <ul style="list-style-type: none">• 0• 1
name	否	String	规则名称
sequence	否	OrderRuleAclDto object	UpdateRuleAclDto
direction	否	Integer	方向：0表示外到内，1表示内到外【说明：规则type=0：互联网规则 2：nat规则时方向值必填】 枚举值： <ul style="list-style-type: none">• 0• 1
action_type	否	Integer	动作0：permit,1：deny 枚举值： <ul style="list-style-type: none">• 0• 1
status	否	Integer	规则下发状态 0：禁用,1：启用
applications	否	Array of strings	应用列表
applicationsJsonString	否	String	应用列表转化为字符串
description	否	String	描述
long_connect_time_hour	否	Long	长连接时长小时

参数	是否必选	参数类型	描述
long_connect_time_minute	否	Long	长连接时长分钟
long_connect_time_second	否	Long	长连接时长秒
long_connect_time	否	Long	长连接时长
long_connect_enable	否	Integer	是否支持长连接，0表示不支持，1表示支持 枚举值： <ul style="list-style-type: none">• 0• 1
profile	否	RuleProfileDto object	域名url信息
source	否	RuleAddressDto object	源地址传输对象
destination	否	RuleAddressDto object	目的地址传输对象
service	否	RuleServiceDto object	服务对象
type	否	Integer	规则type，0：互联网规则，1：vpc规则，2：nat规则 枚举值： <ul style="list-style-type: none">• 0• 1• 2
tag	否	TagsVO object	标签显示值

表 4-129 OrderRuleAclDto

参数	是否必选	参数类型	描述
dest_rule_id	否	String	目标规则id，添加规则位于此规则之后，非置顶时不能为空，置顶时为空
top	否	Integer	是否置顶，0代表非置顶，1代表置顶
bottom	否	Integer	是否置底，0代表非置底，1代表置底

表 4-130 RuleProfileDto

参数	是否必选	参数类型	描述
url	否	String	域名url

表 4-131 RuleAddressDto

参数	是否必选	参数类型	描述
type	是	Integer	源类型0手工输入,1关联IP地址组,2域名,3地理位置,4域名组,5多对象,6域名组-DNS解析,7域名组-应用型。
address_type	否	Integer	源类型0 ipv4,1 ipv6
address	否	String	源IP, 手动类型不能为空, 自动及domain类型为为空
address_set_id	否	String	关联IP地址组ID, 自动类型不能为空, 手动类型合domain类型为为空
address_set_name	否	String	地址组名称
domain_address_name	否	String	域名地址名称, 域名类型时不能为空, 手动类型及自动类型时为为空
region_list_json	否	String	规则region列表json值
region_list	否	Array of IpRegionDto objects	规则region列表
domain_set_id	否	String	域名组id
domain_set_name	否	String	域名组名称
ip_address	否	Array of strings	IP地址列表
address_group	否	Array of strings	地址组列表
address_group_names	否	Array of AddressGroupVO objects	地址组名称列表

参数	是否必选	参数类型	描述
address_set_type	否	Integer	地址组类型，0表示自定义地址组，1表示WAF回源IP地址组，2表示DDoS回源IP地址组，3表示NAT64转换地址组
predefined_group	否	Array of strings	预定义地址组列表

表 4-132 IpRegionDto

参数	是否必选	参数类型	描述
region_id	否	String	区域id
description_cn	否	String	中文描述
description_en	否	String	英文描述
region_type	否	Integer	区域类型，0表示国家，1表示省份，2表示大洲

表 4-133 RuleServiceDto

参数	是否必选	参数类型	描述
type	是	Integer	服务输入类型，0为手动输入类型，1为自动输入类型
protocol	否	Integer	协议类型:TCP为6, UDP为17,ICMP为1,ICMPV6为58,ANY为-1,手动类型不为空，自动类型为空
protocols	否	Array of integers	协议列表，协议类型:TCP为6, UDP为17,ICMP为1,ICMPV6为58,ANY为-1,手动类型不为空，自动类型为空
source_port	否	String	源端口
dest_port	否	String	目的端口
service_set_id	否	String	服务组id，手动类型为空，自动类型为非空
service_set_name	否	String	服务组名称

参数	是否必选	参数类型	描述
custom_service	否	Array of ServiceItem objects	自定义服务
predefined_group	否	Array of strings	预定义服务组列表
service_group	否	Array of strings	服务组列表
service_group_names	否	Array of AddressGroupVO objects	服务组名称列表
service_set_type	否	Integer	服务组类型，0表示自定义服务组，1表示常用WEB服务，2表示常用远程登录和PING，3表示常用数据库

表 4-134 ServiceItem

参数	是否必选	参数类型	描述
protocol	否	Integer	协议类型:TCP为6, UDP为17,ICMP为1,ICMPV6为58,ANY为-1,手动类型不为空, 自动类型为空
source_port	否	String	源端口
dest_port	否	String	目的端口
description	否	String	服务成员描述
name	否	String	服务成员名称

表 4-135 AddressGroupVO

参数	是否必选	参数类型	描述
set_id	否	String	地址组id
name	否	String	地址组名称
protocols	否	Array of integers	协议列表，协议类型:TCP为6, UDP为17,ICMP为1,ICMPV6为58,ANY为-1,手动类型不为空, 自动类型为空

参数	是否必选	参数类型	描述
service_set_type	否	Integer	服务组类型，0表示自定义服务组，1表示常用WEB服务，2表示常用远程登录和PING，3表示常用数据库

表 4-136 TagsVO

参数	是否必选	参数类型	描述
tag_id	否	String	标签id
tag_key	否	String	标签键
tag_value	否	String	标签值

响应参数

状态码： 200

表 4-137 响应 Body 参数

参数	参数类型	描述
data	RuleId object	规则Id

表 4-138 RuleId

参数	参数类型	描述
id	String	id
name	String	名称

状态码： 400

表 4-139 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36

参数	参数类型	描述
error_msg	String	错误描述 最小长度：2 最大长度：512

请求示例

示例为更新一个IPv4类型的外到内的规则，名称为测试规则，源类型为IP地址，地址为1.1.1.1，目的类型为IP地址，目的地址类型为2.2.2.2，服务类型为服务，协议类型为TCP，源端口为0，目的端口为0，不支持长连接，动作为放行，启用状态为启用

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/acl-rule/ceaa0407-b9c8-4dfd-9eca-b6ead2dfd031
```

```
{
  "name": "测试规则",
  "status": 1,
  "action_type": 0,
  "description": "",
  "source": {
    "type": 0,
    "address": "1.1.1.1"
  },
  "destination": {
    "type": 0,
    "address": "2.2.2.2"
  },
  "service": {
    "type": 0,
    "protocol": 6,
    "source_port": "0",
    "dest_port": "0"
  },
  "type": 0,
  "address_type": 0,
  "tag": {
    "tag_key": "",
    "tag_value": ""
  },
  "long_connect_enable": 0,
  "direction": 0
}
```

响应示例

状态码：200

OK

```
{
  "data": {
    "id": "ceaa0407-b9c8-4dfd-9eca-b6ead2dfd031"
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200005",
}
```

```
"error_msg": "操作内容不存在"  
}
```

SDK 代码示例

SDK代码示例如下。

Java

示例为更新一个IPv4类型的外到内的规则，名称为测试规则，源类型为IP地址，地址为1.1.1.1，目的类型为IP地址，目的地址类型为2.2.2.2，服务类型为服务，协议类型为TCP，源端口为0，目的端口为0，不支持长连接，动作为放行，启用状态为启用

```
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.cfw.v1.region.CfwRegion;  
import com.huaweicloud.sdk.cfw.v1.*;  
import com.huaweicloud.sdk.cfw.v1.model.*;  
  
public class UpdateAclRuleSolution {  
  
    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);  
  
        CfwClient client = CfwClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))  
            .build();  
        UpdateAclRuleRequest request = new UpdateAclRuleRequest();  
        request.withEnterpriseProjectId("<enterprise_project_id>");  
        request.withFwInstanceId("<fw_instance_id>");  
        UpdateRuleAclDto body = new UpdateRuleAclDto();  
        TagsVO tagbody = new TagsVO();  
        tagbody.withTagKey("");  
        tagbody.withTagValue("");  
        RuleServiceDto servicebody = new RuleServiceDto();  
        servicebody.withType(0)  
            .withProtocol(6)  
            .withSourcePort("0")  
            .withDestPort("0");  
        RuleAddressDto destinationbody = new RuleAddressDto();  
        destinationbody.withType(0)  
            .withAddress("2.2.2.2");  
        RuleAddressDto sourcebody = new RuleAddressDto();  
        sourcebody.withType(0)  
            .withAddress("1.1.1.1");  
        body.withTag(tagbody);  
        body.withType(UpdateRuleAclDto.TypeEnum.NUMBER_0);  
        body.withService(servicebody);  
        body.withDestination(destinationbody);  
        body.withSource(sourcebody);  
    }  
}
```



```
body.withLongConnectEnable(UpdateRuleAcldto.LongConnectEnableEnum.NUMBER_0);
body.withDescription("");
body.withStatus(1);
body.withActionType(UpdateRuleAcldto.ActionTypeEnum.NUMBER_0);
body.withDirection(UpdateRuleAcldto.DirectionEnum.NUMBER_0);
body.withName("测试规则");
body.withAddressType(UpdateRuleAcldto.AddressTypeEnum.NUMBER_0);
request.withBody(body);
try {
    UpdateAcclRuleResponse response = client.updateAcclRule(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

示例为更新一个IPv4类型的外到内的规则，名称为测试规则，源类型为IP地址，地址为1.1.1.1，目的类型为IP地址，目的地址类型为2.2.2.2，服务类型为服务，协议类型为TCP，源端口为0，目的端口为0，不支持长连接，动作为放行，启用状态为启用

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateAcclRuleRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        tagbody = TagsVO(
            tag_key="",
            tag_value=""
        )
        servicebody = RuleServiceDto(
            type=0,
            protocol=6,
            source_port="0",
            dest_port="0"
        )
```

```
destinationbody = RuleAddressDto(  
    type=0,  
    address="2.2.2.2"  
)  
sourcebody = RuleAddressDto(  
    type=0,  
    address="1.1.1.1"  
)  
request.body = UpdateRuleAclDto(  
    tag=tagbody,  
    type=0,  
    service=servicebody,  
    destination=destinationbody,  
    source=sourcebody,  
    long_connect_enable=0,  
    description="",  
    status=1,  
    action_type=0,  
    direction=0,  
    name="测试规则",  
    address_type=0  
)  
response = client.update_acl_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

示例为更新一个IPv4类型的外到内的规则，名称为测试规则，源类型为IP地址，地址为1.1.1.1，目的类型为IP地址，目的地址类型为2.2.2.2，服务类型为服务，协议类型为TCP，源端口为0，目的端口为0，不支持长连接，动作为放行，启用状态为启用

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(  
        cfw.CfwClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.UpdateAclRuleRequest{}  
    enterpriseProjectIdRequest := "<enterprise_project_id>"
```

```
request.EnterpriseProjectId = &enterpriseProjectIdRequest
fwInstanceIdRequest:= "<fw_instance_id>"
request.FwInstanceId = &fwInstanceIdRequest
tagKeyTag:= ""
tagValueTag:= ""
tagbody := &model.TagsVo{
    TagKey: &tagKeyTag,
    TagValue: &tagValueTag,
}
protocolService:= int32(6)
sourcePortService:= "0"
destPortService:= "0"
servicebody := &model.RuleServiceDto{
    Type: int32(0),
    Protocol: &protocolService,
    SourcePort: &sourcePortService,
    DestPort: &destPortService,
}
addressDestination:= "2.2.2.2"
destinationbody := &model.RuleAddressDto{
    Type: int32(0),
    Address: &addressDestination,
}
addressSource:= "1.1.1.1"
sourcebody := &model.RuleAddressDto{
    Type: int32(0),
    Address: &addressSource,
}
typeUpdateRuleAclDto:= model.GetUpdateRuleAclDtoTypeEnum().E_0
longConnectEnableUpdateRuleAclDto:= model.GetUpdateRuleAclDtoLongConnectEnableEnum().E_0
descriptionUpdateRuleAclDto:= ""
statusUpdateRuleAclDto:= int32(1)
actionTypeUpdateRuleAclDto:= model.GetUpdateRuleAclDtoActionTypeEnum().E_0
directionUpdateRuleAclDto:= model.GetUpdateRuleAclDtoDirectionEnum().E_0
nameUpdateRuleAclDto:= "测试规则"
addressTypeUpdateRuleAclDto:= model.GetUpdateRuleAclDtoAddressTypeEnum().E_0
request.Body = &model.UpdateRuleAclDto{
    Tag: tagbody,
    Type: &typeUpdateRuleAclDto,
    Service: servicebody,
    Destination: destinationbody,
    Source: sourcebody,
    LongConnectEnable: &longConnectEnableUpdateRuleAclDto,
    Description: &descriptionUpdateRuleAclDto,
    Status: &statusUpdateRuleAclDto,
    ActionType: &actionTypeUpdateRuleAclDto,
    Direction: &directionUpdateRuleAclDto,
    Name: &nameUpdateRuleAclDto,
    AddressType: &addressTypeUpdateRuleAclDto,
}
response, err := client.UpdateAclRule(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	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.3.6 批量更新规则动作

功能介绍

批量更新规则动作

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/acl-rule/action

表 4-140 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目ID

表 4-141 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。

请求参数

表 4-142 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）

表 4-143 请求 Body 参数

参数	是否必选	参数类型	描述
object_id	是	String	防护对象id，是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id，可通过调用查询防火墙实例接口获得，注意type为0的为互联网边界防护对象id，type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
action	是	String	动作
rule_ids	是	Array of strings	规则ID列表

响应参数

状态码： 200

表 4-144 响应 Body 参数

参数	参数类型	描述
data	Array of strings	批量更新acl规则id

请求示例

批量更新项目id为14181c1245cf4fd786824efe1e2b9388下防火墙id为546af3f8-88e9-47f2-a205-2346d7090925下的规则id为4e12d889-c1d3-491b-8470-3d1b3dadcd1fd和f798a6a8-c4c5-42b4-838c-c922c9908cb4的规则为启用状态

```
https://{Endpoint}/v1/14181c1245cf4fd786824efe1e2b9388/acl-rule/action?  
fw_instance_id=546af3f8-88e9-47f2-a205-2346d7090925&enterprise_project_id=default
```

```
{
  "action": "enable",
  "rule_ids": [ "4e12d889-c1d3-491b-8470-3d1b3dad1fd", "f798a6a8-c4c5-42b4-838c-c922c9908cb4" ],
  "object_id": "ae42418e-f077-41a0-9d3b-5b2f5ad9102b"
}
```

响应示例

状态码： 200

批量更新acl规则返回值

```
{
  "data": [ "4e12d889-c1d3-491b-8470-3d1b3dad1fd", "f798a6a8-c4c5-42b4-838c-c922c9908cb4" ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

批量更新项目id为14181c1245cf4fd786824efe1e2b9388下防火墙id为546af3f8-88e9-47f2-a205-2346d7090925下的规则id为4e12d889-c1d3-491b-8470-3d1b3dad1fd和f798a6a8-c4c5-42b4-838c-c922c9908cb4的规则为启用状态

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

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

public class BatchUpdateAclRuleActionsSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();

        BatchUpdateAclRuleActionsRequest request = new BatchUpdateAclRuleActionsRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        UpdateSecurityPoliciesActionDto body = new UpdateSecurityPoliciesActionDto();
        List<String> listbodyRuleIds = new ArrayList<>();
        listbodyRuleIds.add("4e12d889-c1d3-491b-8470-3d1b3dad1fd");
```

```
listbodyRuleIds.add("f798a6a8-c4c5-42b4-838c-c922c9908cb4");
body.withRuleIds(listbodyRuleIds);
body.withAction("enable");
body.withObjectId("ae42418e-f077-41a0-9d3b-5b2f5ad9102b");
request.withBody(body);
try {
    BatchUpdateAclRuleActionsResponse response = client.batchUpdateAclRuleActions(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

批量更新项目id为14181c1245cf4fd786824efe1e2b9388下防火墙id为546af3f8-88e9-47f2-a205-2346d7090925下的规则id为4e12d889-c1d3-491b-8470-3d1b3dad1fd和f798a6a8-c4c5-42b4-838c-c922c9908cb4的规则为启用状态

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = BatchUpdateAclRuleActionsRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        listRuleIdsbody = [
            "4e12d889-c1d3-491b-8470-3d1b3dad1fd",
            "f798a6a8-c4c5-42b4-838c-c922c9908cb4"
        ]
        request.body = UpdateSecurityPoliciesActionDto(
            rule_ids=listRuleIdsbody,
            action="enable",
            object_id="ae42418e-f077-41a0-9d3b-5b2f5ad9102b"
        )
        response = client.batch_update_acl_rule_actions(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

批量更新项目id为14181c1245cf4fd786824efe1e2b9388下防火墙id为546af3f8-88e9-47f2-a205-2346d7090925下的规则id为4e12d889-c1d3-491b-8470-3d1b3dad1fd和f798a6a8-c4c5-42b4-838c-c922c9908cb4的规则为启用状态

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchUpdateAclRuleActionsRequest{}
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    var listRuleIidsbody = []string{
        "4e12d889-c1d3-491b-8470-3d1b3dad1fd",
        "f798a6a8-c4c5-42b4-838c-c922c9908cb4",
    }
    request.Body = &model.UpdateSecurityPoliciesActionDto{
        RuleIids: listRuleIidsbody,
        Action: "enable",
        ObjectID: "ae42418e-f077-41a0-9d3b-5b2f5ad9102b",
    }
    response, err := client.BatchUpdateAclRuleActions(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	批量更新acl规则返回值

错误码

请参见[错误码](#)。

4.3.7 ACL 防护规则优先级设置

功能介绍

ACL防护规则优先级设置

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/acl-rule/order/{acl_rule_id}

表 4-145 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id
acl_rule_id	是	String	规则id

表 4-146 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

请求参数

表 4-147 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）

表 4-148 请求 Body 参数

参数	是否必选	参数类型	描述
dest_rule_id	否	String	目标规则id，添加规则位于此规则之后，非置顶时不能为空，置顶时为空
top	否	Integer	是否置顶，0代表非置顶，1代表置顶
bottom	否	Integer	是否置底，0代表非置底，1代表置底

响应参数

状态码： 200

表 4-149 响应 Body 参数

参数	参数类型	描述
data	RuleId object	规则id列表

表 4-150 RuleId

参数	参数类型	描述
id	String	id
name	String	名称

状态码： 400

表 4-151 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度：8 最大长度：36
error_msg	String	错误描述 最小长度：2 最大长度：512

请求示例

设置项目id9d80d070b6d44942af73c9c3d38e0429中规则id为ffe9af47-d893-483b-86e3-ee5242e8cb15的规则移动到规则id为69c32dc5-f801-4294-98ee-978b51f97d35后

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/acl-rule/order/ffe9af47-d893-483b-86e3-ee5242e8cb15
```

```
{
  "top": 0,
  "dest_rule_id": "69c32dc5-f801-4294-98ee-978b51f97d35"
}
```

响应示例

状态码：200

规则排序响应

```
{
  "data": {
    "id": "ffe9af47-d893-483b-86e3-ee5242e8cb15"
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "00200005",
  "error_msg": "操作内容不存在"
}
```

SDK 代码示例

SDK代码示例如下。

Java

设置项目id9d80d070b6d44942af73c9c3d38e0429中规则id为ffe9af47-d893-483b-86e3-ee5242e8cb15的规则移动到规则id为69c32dc5-f801-4294-98ee-978b51f97d35后

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class UpdateAclRuleOrderSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateAclRuleOrderRequest request = new UpdateAclRuleOrderRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        OrderRuleAclDto body = new OrderRuleAclDto();
        body.withTop(0);
        body.withDestRuleId("69c32dc5-f801-4294-98ee-978b51f97d35");
        request.withBody(body);
        try {
            UpdateAclRuleOrderResponse response = client.updateAclRuleOrder(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

设置项目id9d80d070b6d44942af73c9c3d38e0429中规则id为ffe9af47-d893-483b-86e3-ee5242e8cb15的规则移动到规则id为69c32dc5-f801-4294-98ee-978b51f97d35后

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateAclRuleOrderRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        request.body = OrderRuleAclDto(
            top=0,
            dest_rule_id="69c32dc5-f801-4294-98ee-978b51f97d35"
        )
        response = client.update_acl_rule_order(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

设置项目id9d80d070b6d44942af73c9c3d38e0429中规则id为ffe9af47-d893-483b-86e3-ee5242e8cb15的规则移动到规则id为69c32dc5-f801-4294-98ee-978b51f97d35后

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```
request := &model.UpdateAclRuleOrderRequest{}
enterpriseProjectIdRequest:= "<enterprise_project_id>"
request.EnterpriseProjectId = &enterpriseProjectIdRequest
fwInstanceIdRequest:= "<fw_instance_id>"
request.FwInstanceId = &fwInstanceIdRequest
topOrderRuleAclDto:= int32(0)
destRuleIdOrderRuleAclDto:= "69c32dc5-f801-4294-98ee-978b51f97d35"
request.Body = &model.OrderRuleAclDto{
    Top: &topOrderRuleAclDto,
    DestRuleId: &destRuleIdOrderRuleAclDto,
}
response, err := client.UpdateAclRuleOrder(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	规则排序响应
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.3.8 查询防护规则

功能介绍

查询防护规则

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/acl-rules

表 4-152 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-153 Query 参数

参数	是否必选	参数类型	描述
object_id	是	String	防护对象id, 是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id, 可通过调用查询防火墙实例接口获得, 注意type为0的为互联网边界防护对象id, type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
type	否	Integer	规则Type0: 互联网规则,1: vpc规则, 2:nat规则 枚举值: <ul style="list-style-type: none">• 0• 1• 2
protocol	否	Integer	协议类型:TCP为6, UDP为17,ICMP为1,ICMPV6为58,ANY为-1 枚举值: <ul style="list-style-type: none">• 6• 17• 1• 58
ip	否	String	ip地址
name	否	String	名称
direction	否	Integer	方向0: 外到内1: 内到外
status	否	Integer	规则下发状态 0: 禁用,1: 启用 枚举值: <ul style="list-style-type: none">• 0• 1

参数	是否必选	参数类型	描述
action_type	否	Integer	动作0: permit,1: deny 枚举值: <ul style="list-style-type: none">• 0• 1
address_type	否	Integer	地址类型0 ipv4,1 ipv6,2 domain 枚举值: <ul style="list-style-type: none">• 0• 1• 2
limit	是	Integer	每页显示个数, 范围为1-1024
offset	是	Integer	偏移量: 指定返回记录的开始位置, 必须为数字, 取值范围为大于或等于0, 默认0
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。
tags_id	否	String	标签id
source	否	String	源地址
destination	否	String	目的地址
service	否	String	服务端口
application	否	String	应用

请求参数

表 4-154 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）

响应参数

状态码： 200

表 4-155 响应 Body 参数

参数	参数类型	描述
data	data object	data

表 4-156 data

参数	参数类型	描述
offset	Integer	偏移量：指定返回记录的开始位置，必须为数字，取值范围为大于或等于0，默认0
limit	Integer	每页显示个数，范围为1-1024
total	Integer	查询总条数
object_id	String	防护对象id，是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id，可通过调用查询防火墙实例接口获得，注意type为0的为互联网边界防护对象id，type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
records	Array of records objects	records

表 4-157 records

参数	参数类型	描述
rule_id	String	规则id
address_type	Integer	地址类型0 ipv4,1 ipv6

参数	参数类型	描述
name	String	规则名称
sequence	OrderRuleAclDto object	UpdateRuleAclDto
direction	Integer	规则方向0: 外到内1: 内到外 枚举值: <ul style="list-style-type: none">• 0• 1
action_type	Integer	动作0: permit,1: deny
status	Integer	规则下发状态 0: 禁用,1: 启用
description	String	描述
long_connect_time_hour	Long	长连接时长小时
long_connect_time_minute	Long	长连接时长分钟
long_connect_time_second	Long	长连接时长秒
long_connect_time	Long	长连接时长
long_connect_enable	Integer	长连接支持
source	RuleAddressDto object	源地址传输对象
destination	RuleAddressDto object	destination
service	RuleServiceDto object	service
type	Integer	规则type, 0: 互联网规则, 1: vpc规则, 2: nat规则 枚举值: <ul style="list-style-type: none">• 0• 1• 2
created_date	String	创建时间
last_open_time	String	最后开启时间

参数	参数类型	描述
tag	TagsVO object	标签显示信息

表 4-158 OrderRuleAclDto

参数	参数类型	描述
dest_rule_id	String	目标规则id, 添加规则位于此规则之后, 非置顶时不能为空, 置顶时为空
top	Integer	是否置顶, 0代表非置顶, 1代表置顶
bottom	Integer	是否置底, 0代表非置底, 1代表置底

表 4-159 RuleAddressDto

参数	参数类型	描述
type	Integer	源类型0手工输入,1关联IP地址组,2域名, 3地理位置, 4域名组, 5多对象, 6域名组-DNS解析, 7域名组-应用型。
address_type	Integer	源类型0 ipv4,1 ipv6
address	String	源IP, 手动类型不能为空, 自动及domain类型为空
address_set_id	String	关联IP地址组ID, 自动类型不能为空, 手动类型合domain类型为空
address_set_name	String	地址组名称
domain_address_name	String	域名地址名称, 域名类型时不能为空, 手动类型及自动类型时为空
region_list_json	String	规则region列表json值
region_list	Array of IpRegionDto objects	规则region列表
domain_set_id	String	域名组id
domain_set_name	String	域名组名称
ip_address	Array of strings	IP地址列表

参数	参数类型	描述
address_group	Array of strings	地址组列表
address_group_names	Array of AddressGroupVO objects	地址组名称列表
address_set_type	Integer	地址组类型, 0表示自定义地址组, 1表示WAF回源IP地址组, 2表示DDoS回源IP地址组, 3表示NAT64转换地址组
predefined_group	Array of strings	预定义地址组列表

表 4-160 IpRegionDto

参数	参数类型	描述
region_id	String	区域id
description_cn	String	中文描述
description_en	String	英文描述
region_type	Integer	区域类型, 0表示国家, 1表示省份, 2表示大洲

表 4-161 RuleServiceDto

参数	参数类型	描述
type	Integer	服务输入类型, 0为手动输入类型, 1为自动输入类型
protocol	Integer	协议类型:TCP为6, UDP为17,ICMP为1,ICMPV6为58,ANY为-1,手动类型不为空, 自动类型为空
protocols	Array of integers	协议列表, 协议类型:TCP为6, UDP为17,ICMP为1,ICMPV6为58,ANY为-1,手动类型不为空, 自动类型为空
source_port	String	源端口
dest_port	String	目的端口
service_set_id	String	服务组id, 手动类型为空, 自动类型为非空
service_set_name	String	服务组名称

参数	参数类型	描述
custom_service	Array of ServiceItem objects	自定义服务
predefined_group	Array of strings	预定义服务组列表
service_group	Array of strings	服务组列表
service_group_names	Array of AddressGroupVO objects	服务组名称列表
service_set_type	Integer	服务组类型，0表示自定义服务组，1表示常用WEB服务，2表示常用远程登录和PING，3表示常用数据库

表 4-162 ServiceItem

参数	参数类型	描述
protocol	Integer	协议类型:TCP为6, UDP为17,ICMP为1,ICMPV6为58,ANY为-1,手动类型不为空, 自动类型为空
source_port	String	源端口
dest_port	String	目的端口
description	String	服务成员描述
name	String	服务成员名称

表 4-163 AddressGroupVO

参数	参数类型	描述
set_id	String	地址组id
name	String	地址组名称
protocols	Array of integers	协议列表, 协议类型:TCP为6, UDP为17,ICMP为1,ICMPV6为58,ANY为-1,手动类型不为空, 自动类型为空
service_set_type	Integer	服务组类型，0表示自定义服务组，1表示常用WEB服务，2表示常用远程登录和PING，3表示常用数据库

表 4-164 TagsVO

参数	参数类型	描述
tag_id	String	标签id
tag_key	String	标签键
tag_value	String	标签值

状态码： 400

表 4-165 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

查询项目id为9d80d070b6d44942af73c9c3d38e0429，防护对象id为e12bd2cd-ebfc-4af7-ad6f-ebe6da398029的第一页，size为10的数据

```
https://{Endpoint}/cfw/v1/9d80d070b6d44942af73c9c3d38e0429/acl-rules?object_id=e12bd2cd-ebfc-4af7-ad6f-ebe6da398029&limit=10&offset=0
```

响应示例

状态码： 200

OK

```
{
  "data": {
    "limit": 10,
    "object_id": "cfebd347-b655-4b84-b938-3c54317599b2",
    "offset": 0,
    "records": [ {
      "action_type": 0,
      "address_type": 0,
      "destination": {
        "address": "0.0.0.0/0",
        "address_type": 0,
        "type": 0
      },
      "direction": 1,
      "long_connect_enable": 0,
      "created_date": "2024-02-27 04:01:17",
      "last_open_time": "2024-02-27 04:01:17",
    }
  ]
}
```

```
"description" : "description",
"name" : "eip_ipv4_n_w_allow",
"rule_id" : "ffe9af47-d893-483b-86e3-ee5242e8cb15",
"service" : {
  "dest_port" : "0",
  "protocol" : "-1",
  "source_port" : "0",
  "type" : 0
},
"source" : {
  "address_set_id" : "48bf09b-6f3a-4371-8ddb-05d5d7148bcc",
  "address_set_name" : "ip_group",
  "address_type" : 0,
  "type" : 1
},
"status" : 1,
"type" : "0"
}],
"total" : 1
}
}
```

状态码： 400

Bad Request

```
{
  "error_code" : "CFW.0020016",
  "error_msg" : "实例状态错误"
}
```

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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class ListAclRulesSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
    }
}
```

```
ListAclRulesRequest request = new ListAclRulesRequest();
request.withObjectId("<object_id>");
request.withType(ListAclRulesRequest.TypeEnum.NUMBER_<type>);
request.withProtocol(ListAclRulesRequest.ProtocolEnum.NUMBER_<protocol>);
request.withIp("<ip>");
request.withName("<name>");
request.withDirection(<direction>);
request.withStatus(ListAclRulesRequest.StatusEnum.NUMBER_<status>);
request.withActionType(ListAclRulesRequest.ActionTypeEnum.NUMBER_<action_type>);
request.withAddressType(ListAclRulesRequest.AddressTypeEnum.NUMBER_<address_type>);
request.withLimit(<limit>);
request.withOffset(<offset>);
request.withEnterpriseProjectId("<enterprise_project_id>");
request.withFwInstanceId("<fw_instance_id>");
request.withTagsId("<tags_id>");
request.withSource("<source>");
request.withDestination("<destination>");
request.withService("<service>");
try {
    ListAclRulesResponse response = client.listAclRules(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListAclRulesRequest()
        request.object_id = "<object_id>"
        request.type = <type>
        request.protocol = <protocol>
        request.ip = "<ip>"
        request.name = "<name>"
        request.direction = <direction>
        request.status = <status>
```



```
request.action_type = <action_type>
request.address_type = <address_type>
request.limit = <limit>
request.offset = <offset>
request.enterprise_project_id = "<enterprise_project_id>"
request.fw_instance_id = "<fw_instance_id>"
request.tags_id = "<tags_id>"
request.source = "<source>"
request.destination = "<destination>"
request.service = "<service>"
response = client.list_acl_rules(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAclRulesRequest{
        request.ObjectId = "<object_id>"
        typeRequest:= model.GetListAclRulesRequestTypeEnum().<TYPE>
        request.Type = &typeRequest
        protocolRequest:= model.GetListAclRulesRequestProtocolEnum().<PROTOCOL>
        request.Protocol = &protocolRequest
        ipRequest:= "<ip>"
        request.Ip = &ipRequest
        nameRequest:= "<name>"
        request.Name = &nameRequest
        directionRequest:= int32(<direction>)
        request.Direction = &directionRequest
        statusRequest:= model.GetListAclRulesRequestStatusEnum().<STATUS>
        request.Status = &statusRequest
        actionTypeRequest:= model.GetListAclRulesRequestActionTypeEnum().<ACTION_TYPE>
        request.ActionType = &actionTypeRequest
        addressTypeRequest:= model.GetListAclRulesRequestAddressTypeEnum().<ADDRESS_TYPE>
        request.AddressType = &addressTypeRequest
        request.Limit = int32(<limit>)
```

```
request.Offset = int32(<offset>)
enterpriseProjectIdRequest:= "<enterprise_project_id>"
request.EnterpriseProjectId = &enterpriseProjectIdRequest
fwInstanceIdRequest:= "<fw_instance_id>"
request.FwInstanceId = &fwInstanceIdRequest
tagsIdRequest:= "<tags_id>"
request.TagsId = &tagsIdRequest
sourceRequest:= "<source>"
request.Source = &sourceRequest
destinationRequest:= "<destination>"
request.Destination = &destinationRequest
serviceRequest:= "<service>"
request.Service = &serviceRequest
response, err := client.ListAclRules(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	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.3.9 查询规则标签

功能介绍

查询规则标签

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/cfw-acl/tags

表 4-166 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-167 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	是	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。
offset	是	Integer	偏移量: 指定返回记录的开始位置, 必须为数字, 取值范围为大于或等于0, 默认0
limit	是	Integer	每页显示个数, 范围为1-1024

请求参数

表 4-168 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)

响应参数

状态码: 200

表 4-169 响应 Body 参数

参数	参数类型	描述
data	HttpGetAclTagResponseData object	

表 4-170 HttpGetAclTagResponseData

参数	参数类型	描述
offset	Integer	偏移量：指定返回记录的开始位置，必须为数字，取值范围为大于或等于0，默认0
limit	Integer	每页显示个数，范围为1-1024
total	Integer	总数
records	Array of TagsVO objects	标签列表

表 4-171 TagsVO

参数	参数类型	描述
tag_id	String	标签id
tag_key	String	标签键
tag_value	String	标签值

请求示例

查询项目id为14181c1245cf4fd786824efe1e2b9388的防火墙id为546af3f8-88e9-47f2-a205-2346d7090925的已有acl标签。

```
https://{Endpoint}/v2/14181c1245cf4fd786824efe1e2b9388/cfw-acl/tags?limit=1000&offset=0&fw_instance_id=546af3f8-88e9-47f2-a205-2346d7090925&enterprise_project_id=default
```

响应示例

状态码： 200

```
{
  "data": {
    "limit": 1000,
    "offset": 0,
    "records": [ {
      "tag_id": "98fdf013-e7ad-4581-9c71-6de04c76a18f",
      "tag_key": "1",
      "tag_value": "1"
    }
  ]
}
```

```
    }, {  
      "tag_id" : "36e6fbfe-7fcd-48be-872b-4f6074e1e4e8",  
      "tag_key" : "1",  
      "tag_value" : "2"  
    }, {  
      "tag_id" : "0bf41046-6587-42f2-8399-a6864022b504",  
      "tag_key" : "测试",  
      "tag_value" : "测试"  
    }  
  ],  
  "total" : 3  
}  
}
```

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.cfw.v1.region.CfwRegion;  
import com.huaweicloud.sdk.cfw.v1.*;  
import com.huaweicloud.sdk.cfw.v1.model.*;  
  
public class ListRuleAclTagsSolution {  
  
    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);  
  
        CfwClient client = CfwClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListRuleAclTagsRequest request = new ListRuleAclTagsRequest();  
        request.withEnterpriseProjectId("<enterprise_project_id>");  
        request.withFwInstanceId("<fw_instance_id>");  
        request.withOffset("<offset>");  
        request.withLimit("<limit>");  
        try {  
            ListRuleAclTagsResponse response = client.listRuleAclTags(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")  
    sk = __import__('os').getenv("CLOUD_SDK_SK")  
  
    credentials = BasicCredentials(ak, sk) \  
  
    client = CfwClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(CfwRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ListRuleAclTagsRequest()  
        request.enterprise_project_id = "<enterprise_project_id>"  
        request.fw_instance_id = "<fw_instance_id>"  
        request.offset = <offset>  
        request.limit = <limit>  
        response = client.list_rule_acl_tags(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"  
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListRuleAclTagsRequest{
    enterpriseProjectIdRequest:= "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    request.FwInstanceId = "<fw_instance_id>"
    request.Offset = int32(<offset>)
    request.Limit = int32(<limit>)
}
response, err := client.ListRuleAclTags(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	

错误码

请参见[错误码](#)。

4.3.10 获取规则击中次数

功能介绍

获取规则击中次数

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/acl-rule/count

表 4-172 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-173 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

请求参数

表 4-174 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	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）

表 4-175 请求 Body 参数

参数	是否必选	参数类型	描述
rule_ids	是	Array of strings	规则id列表

响应参数

状态码： 200

表 4-176 响应 Body 参数

参数	参数类型	描述
data	RuleHitCountRecords object	规则击中次数记录

表 4-177 RuleHitCountRecords

参数	参数类型	描述
limit	Integer	每页显示个数，范围为1-1024
offset	Integer	偏移量：指定返回记录的开始位置，必须为数字，取值范围为大于或等于0，默认0
total	Integer	总数
records	Array of RuleHitCount Object objects	规则击中次数列表

表 4-178 RuleHitCountObject

参数	参数类型	描述
rule_id	String	规则id
rule_hit_count	Integer	规则击中次数

请求示例

查询项目id为0b2179bbe180d3762fb0c01a2d5725c7中acl规则id为59ff6bd9-0a76-41ec-9650-380086069965命中次数

```
https://{Endpoint}/v1/0b2179bbe180d3762fb0c01a2d5725c7/acl-rule/count
```

```
{  
  "rule_ids" : [ "59ff6bd9-0a76-41ec-9650-380086069965" ]  
}
```

响应示例

状态码： 200

获取规则击中次数响应

```
{  
  "data" : {  
    "limit" : 1,  
    "offset" : 1,  
    "records" : [{  
      "rule_hit_count" : 0,  
      "rule_id" : "59ff6bd9-0a76-41ec-9650-380086069965"  
    }],  
    "total" : 1  
  }  
}
```

SDK 代码示例

SDK代码示例如下。

Java

查询项目id为0b2179bbe180d3762fb0c01a2d5725c7中acl规则id为59ff6bd9-0a76-41ec-9650-380086069965命中次数

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

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

public class ListAclRuleHitCountSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListAclRuleHitCountRequest request = new ListAclRuleHitCountRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        ListRuleHitCountDto body = new ListRuleHitCountDto();
        List<String> listbodyRuleIds = new ArrayList<>();
        listbodyRuleIds.add("59ff6bd9-0a76-41ec-9650-380086069965");
        body.withRuleIds(listbodyRuleIds);
        request.withBody(body);
        try {
            ListAclRuleHitCountResponse response = client.listAclRuleHitCount(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.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

查询项目id为0b2179bbe180d3762fb0c01a2d5725c7中acl规则id为59ff6bd9-0a76-41ec-9650-380086069965命中次数

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListAclRuleHitCountRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        listRuleIdsbody = [
            "59ff6bd9-0a76-41ec-9650-380086069965"
        ]
        request.body = ListRuleHitCountDto(
            rule_ids=listRuleIdsbody
        )
        response = client.list_acl_rule_hit_count(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

查询项目id为0b2179bbe180d3762fb0c01a2d5725c7中acl规则id为59ff6bd9-0a76-41ec-9650-380086069965命中次数

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListAclRuleHitCountRequest{
    enterpriseProjectIdRequest:= "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest:= "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    var listRuleIdsbody = []string{
        "59ff6bd9-0a76-41ec-9650-380086069965",
    }
    request.Body = &model.ListRuleHitCountDto{
        RuleIds: listRuleIdsbody,
    }
}
response, err := client.ListAclRuleHitCount(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	获取规则击中次数响应
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.4 黑白名单管理

4.4.1 创建黑白名单规则

功能介绍

创建黑白名单规则

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/black-white-list

表 4-179 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-180 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

请求参数

表 4-181 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）

表 4-182 请求 Body 参数

参数	是否必选	参数类型	描述
object_id	是	String	防护对象id, 是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id, 可通过调用查询防火墙实例接口获得, 注意type为0的为互联网边界防护对象id, type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
list_type	是	Integer	黑白名单类型4: 黑名单, 5: 白名单
direction	是	Integer	地址方向0: 源地址1: 目的地址
address_type	是	Integer	Ip地址类型 0: ipv4,1:ipv6,2:domain
address	是	String	地址类型
protocol	是	Integer	协议类型:TCP为6, UDP为17,ICMP为1,ICMPV6为58,ANY为-1,手动类型不为空, 自动类型为
port	是	String	目的端口
description	否	String	描述

响应参数

状态码: 200

表 4-183 响应 Body 参数

参数	参数类型	描述
data	IdObject object	添加黑白名单响应data

表 4-184 IdObject

参数	参数类型	描述
id	String	id值
name	String	名称

状态码： 400**表 4-185 响应 Body 参数**

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

给项目id为9d80d070b6d44942af73c9c3d38e0429防护对象id为cfebd347-b655-4b84-b938-3c54317599b2的对象添加方向为源地址，地址为1.1.1.1，协议类型为tcp，端口为1的ipv4的tcp的白名单

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/black-white-list
{
  "object_id": "cfebd347-b655-4b84-b938-3c54317599b2",
  "list_type": 5,
  "direction": 0,
  "address": "1.1.1.1",
  "protocol": 6,
  "port": "1",
  "address_type": 0
}
```

响应示例

状态码： 200

添加黑白名单响应

```
{
  "data": {
    "id": "2eee3fe8-0b9b-49ac-8e7f-eaafa321e99a",
    "name": "test"
  }
}
```

状态码： 400

Bad Request

```
{
  "error_code": "CFW.0020016",
  "error_msg": "实例状态错误"
}
```

SDK 代码示例

SDK代码示例如下。

Java

给项目id为9d80d070b6d44942af73c9c3d38e0429防护对象id为cfebd347-b655-4b84-b938-3c54317599b2的对象添加方向为源地址，地址为1.1.1.1，协议类型为tcp，端口为1的ipv4的tcp的白名单

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class AddBlackWhiteListSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        AddBlackWhiteListRequest request = new AddBlackWhiteListRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        AddBlackWhiteListDto body = new AddBlackWhiteListDto();
        body.withPort("1");
        body.withProtocol(6);
        body.withAddress("1.1.1.1");
        body.withAddressType(0);
        body.withDirection(0);
        body.withListType(5);
        body.withObjectId("cfebd347-b655-4b84-b938-3c54317599b2");
        request.withBody(body);
        try {
            AddBlackWhiteListResponse response = client.addBlackWhiteList(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

给项目id为9d80d070b6d44942af73c9c3d38e0429防护对象id为cfebd347-b655-4b84-b938-3c54317599b2的对象添加方向为源地址，地址为1.1.1.1，协议类型为tcp，端口为1的ipv4的tcp的白名单

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = AddBlackWhiteListRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        request.body = AddBlackWhiteListDto(
            port="1",
            protocol=6,
            address="1.1.1.1",
            address_type=0,
            direction=0,
            list_type=5,
            object_id="cfebd347-b655-4b84-b938-3c54317599b2"
        )
        response = client.add_black_white_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

给项目id为9d80d070b6d44942af73c9c3d38e0429防护对象id为cfebd347-b655-4b84-b938-3c54317599b2的对象添加方向为源地址，地址为1.1.1.1，协议类型为tcp，端口为1的ipv4的tcp的白名单

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.AddBlackWhiteListRequest{
    enterpriseProjectIdRequest:= "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest:= "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    request.Body = &model.AddBlackWhiteListDto{
        Port: "1",
        Protocol: int32(6),
        Address: "1.1.1.1",
        AddressType: int32(0),
        Direction: int32(0),
        ListType: int32(5),
        Objectid: "cfebd347-b655-4b84-b938-3c54317599b2",
    }
}
response, err := client.AddBlackWhiteList(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	添加黑白名单响应
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.4.2 更新黑白名单列表

功能介绍

更新黑白名单列表

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/black-white-list/{list_id}

表 4-186 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id
list_id	是	String	黑白名单列表id

表 4-187 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

请求参数

表 4-188 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）

表 4-189 请求 Body 参数

参数	是否必选	参数类型	描述
direction	否	Integer	地址方向0：源地址1：目的地址
address_type	否	Integer	地址类型0： ipv4,1:ipv6,2:domain
address	否	String	ip地址
protocol	否	Integer	协议类型:TCP为6, UDP为17,ICMP为1,ICMPV6为58,ANY为-1
port	否	String	端口
list_type	否	Integer	黑白名单类型4：黑名单，5： 白名单 枚举值： <ul style="list-style-type: none">• 4• 5
object_id	否	String	防护对象id，是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id，可通过调用查询防火墙实例接口获得，注意type为0的为互联网边界防护对象id，type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
description	否	String	描述

响应参数

状态码： 200

表 4-190 响应 Body 参数

参数	参数类型	描述
data	IdObject object	更新黑白名单响应data

表 4-191 IdObject

参数	参数类型	描述
id	String	id值
name	String	名称

状态码： 400

表 4-192 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

更新项目id为9d80d070b6d44942af73c9c3d38e0429防护对象id为cfebd347-b655-4b84-b938-3c54317599b2的白名单为方向为源地址，地址为1.1.1.1，协议类型为tcp，端口为1的ipv4的tcp

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/black-white-list/  
9d80d070b6d44942af73c9c3d38e042b
```

```
{  
  "object_id" : "cfebd347-b655-4b84-b938-3c54317599b2",  
  "list_type" : 5,  
  "direction" : 0,  
  "address" : "1.1.1.1",  
  "protocol" : 6,  
  "port" : "1",  
  "address_type" : 0  
}
```

响应示例

状态码： 200

更新黑白名单响应

```
{
  "data" : {
    "id" : "2eee3fe8-0b9b-49ac-8e7f-eaafa321e99a",
    "name" : "test"
  }
}
```

状态码: 400

Bad Request

```
{
  "error_code" : "CFW.00200005",
  "error_msg" : "操作内容不存在"
}
```

SDK 代码示例

SDK代码示例如下。

Java

更新项目id为9d80d070b6d44942af73c9c3d38e0429防护对象id为cfebd347-b655-4b84-b938-3c54317599b2的白名单为方向为源地址，地址为1.1.1.1，协议类型为tcp，端口为1的ipv4的tcp

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class UpdateBlackWhiteListSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateBlackWhiteListRequest request = new UpdateBlackWhiteListRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        UpdateBlackWhiteListDto body = new UpdateBlackWhiteListDto();
        body.withObjectId("cfebd347-b655-4b84-b938-3c54317599b2");
        body.withListType(UpdateBlackWhiteListDto.ListTypeEnum.NUMBER_5);
        body.withPort("1");
        body.withProtocol(6);
    }
}
```

```
body.withAddress("1.1.1.1");
body.withAddressType(0);
body.withDirection(0);
request.withBody(body);
try {
    UpdateBlackWhiteListResponse response = client.updateBlackWhiteList(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.getErrMsg());
}
}
```

Python

更新项目id为9d80d070b6d44942af73c9c3d38e0429防护对象id为cfebd347-b655-4b84-b938-3c54317599b2的白名单为方向为源地址，地址为1.1.1.1，协议类型为tcp，端口为1的ipv4的tcp

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateBlackWhiteListRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        request.body = UpdateBlackWhiteListDto(
            object_id="cfebd347-b655-4b84-b938-3c54317599b2",
            list_type=5,
            port="1",
            protocol=6,
            address="1.1.1.1",
            address_type=0,
            direction=0
        )
        response = client.update_black_white_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

更新项目id为9d80d070b6d44942af73c9c3d38e0429防护对象id为cfebd347-b655-4b84-b938-3c54317599b2的白名单为方向为源地址，地址为1.1.1.1，协议类型为tcp，端口为1的ipv4的tcp

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateBlackWhiteListRequest{}
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    objectIdUpdateBlackWhiteListDto := "cfebd347-b655-4b84-b938-3c54317599b2"
    listTypeUpdateBlackWhiteListDto := model.GetUpdateBlackWhiteListDtoTypeEnum().E_5
    portUpdateBlackWhiteListDto := "1"
    protocolUpdateBlackWhiteListDto := int32(6)
    addressUpdateBlackWhiteListDto := "1.1.1.1"
    addressTypeUpdateBlackWhiteListDto := int32(0)
    directionUpdateBlackWhiteListDto := int32(0)
    request.Body = &model.UpdateBlackWhiteListDto{
        ObjectId: &objectIdUpdateBlackWhiteListDto,
        ListType: &listTypeUpdateBlackWhiteListDto,
        Port: &portUpdateBlackWhiteListDto,
        Protocol: &protocolUpdateBlackWhiteListDto,
        Address: &addressUpdateBlackWhiteListDto,
        AddressType: &addressTypeUpdateBlackWhiteListDto,
        Direction: &directionUpdateBlackWhiteListDto,
    }
    response, err := client.UpdateBlackWhiteList(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	更新黑白名单响应
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.4.3 删除黑白名单规则

功能介绍

删除黑白名单规则

调用方法

请参见[如何调用API](#)。

URI

DELETE /v1/{project_id}/black-white-list/{list_id}

表 4-193 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id
list_id	是	String	黑白名单id

表 4-194 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

请求参数

表 4-195 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）

响应参数

状态码： 200

表 4-196 响应 Body 参数

参数	参数类型	描述
data	IdObject object	删除黑白名单响应data

表 4-197 IdObject

参数	参数类型	描述
id	String	id值
name	String	名称

状态码： 400

表 4-198 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

删除项目id为9d80d070b6d44942af73c9c3d38e0429下的黑白名单id为2eee3fe8-0b9b-49ac-8e7f-eaafa321e99a的黑白名单

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/black-white-list/2eee3fe8-0b9b-49ac-8e7f-eaafa321e99a
```

响应示例

状态码： 200

删除黑白名单响应

```
{
  "data": {
    "id": "2eee3fe8-0b9b-49ac-8e7f-eaafa321e99a"
  }
}
```

状态码： 400

Bad Request

```
{
  "error_code": "CFW.00200005",
  "error_msg": "操作内容不存在"
}
```

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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
```

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

public class DeleteBlackWhiteListSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteBlackWhiteListRequest request = new DeleteBlackWhiteListRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        try {
            DeleteBlackWhiteListResponse response = client.deleteBlackWhiteList(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = DeleteBlackWhiteListRequest()
```

```
request.enterprise_project_id = "<enterprise_project_id>"
request.fw_instance_id = "<fw_instance_id>"
response = client.delete_black_white_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/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteBlackWhiteListRequest{}
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    response, err := client.DeleteBlackWhiteList(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	删除黑白名单响应

状态码	描述
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.4.4 查询黑白名单列表

功能介绍

查询黑白名单列表

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/black-white-lists

表 4-199 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-200 Query 参数

参数	是否必选	参数类型	描述
object_id	是	String	防护对象id，是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id，可通过调用查询防火墙实例接口获得，注意type为0的为互联网边界防护对象id，type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。

参数	是否必选	参数类型	描述
list_type	是	Integer	黑白名单类型4: 黑名单, 5: 白名单 枚举值: <ul style="list-style-type: none"> • 4 • 5
address_type	否	Integer	IP地址类型0: ipv4,1:ipv6,2:domain 枚举值: <ul style="list-style-type: none"> • 0 • 1 • 2
address	否	String	ip地址
port	否	String	端口
limit	是	Integer	每页显示个数, 范围为1-1024
offset	是	Integer	偏移量: 指定返回记录的开始位置, 必须为数字, 取值范围为大于或等于0, 默认0
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考API Explorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

请求参数

表 4-201 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）

响应参数

状态码： 200

表 4-202 响应 Body 参数

参数	参数类型	描述
data	data object	查询黑白名单返回值

表 4-203 data

参数	参数类型	描述
object_id	String	防护对象id，是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id，可通过调用查询防火墙实例接口获得，注意type为0的为互联网边界防护对象id，type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
offset	Integer	偏移量：指定返回记录的开始位置，必须为数字，取值范围为大于或等于0，默认0
limit	Integer	每页显示个数，范围为1-1024
total	Integer	查询的总条数
records	Array of records objects	黑白名单记录

表 4-204 records

参数	参数类型	描述
list_id	String	黑白名单列表id
direction	Integer	黑白地址方向0：源地址1：目的地址
address_type	Integer	IP地址类型0：ipv4,1:ipv6,2:domain
address	String	ip地址
protocol	Integer	协议类型:TCP为6,UDP为17,ICMP为1,ICMPV6为58,ANY为-1,手动类型不为空，自动类型为空
port	String	端口
description	String	描述

状态码： 400

表 4-205 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度：8 最大长度：36
error_msg	String	错误描述 最小长度：2 最大长度：512

请求示例

查询项目id为9d80d070b6d44942af73c9c3d38e0429防护对象id为cfebd347-b655-4b84-b938-3c54317599b2第一页的白名单，查询条数为5条

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/black-white-lists?object_id=cfebd347-b655-4b84-b938-3c54317599b2&limit=10&offset=0&list_type=5
```

响应示例

状态码：200

查询黑白名单列表返回值

```
{
  "data": {
    "limit": 10,
    "offset": 0,
    "records": [ {
      "address": "1.1.1.1",
      "address_type": 0,
      "description": "",
      "direction": 0,
      "list_id": "1310d401-daf5-44f2-8276-f79e1643984d",
      "protocol": 6
    } ],
    "total": 1
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.0020016",
  "error_msg": "实例状态错误"
}
```

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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class ListBlackWhiteListsSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListBlackWhiteListsRequest request = new ListBlackWhiteListsRequest();
        request.withObjectId("<object_id>");
        request.withListType(ListBlackWhiteListsRequest.ListTypeEnum.NUMBER_<list_type>);
        request.withAddressType(ListBlackWhiteListsRequest.AddressTypeEnum.NUMBER_<address_type>);
        request.withAddress("<address>");
        request.withPort("<port>");
        request.withLimit(<limit>);
        request.withOffset(<offset>);
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        try {
            ListBlackWhiteListsResponse response = client.listBlackWhiteLists(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

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

try:
    request = ListBlackWhiteListsRequest()
    request.object_id = "<object_id>"
    request.list_type = <list_type>
    request.address_type = <address_type>
    request.address = "<address>"
    request.port = "<port>"
    request.limit = <limit>
    request.offset = <offset>
    request.enterprise_project_id = "<enterprise_project_id>"
    request.fw_instance_id = "<fw_instance_id>"
    response = client.list_black_white_lists(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListBlackWhiteListsRequest{}
    request.ObjectId = "<object_id>"
    request.ListType = model.GetListBlackWhiteListsRequestListTypeEnum().<LIST_TYPE>
    addressTypeRequest := model.GetListBlackWhiteListsRequestAddressTypeEnum().<ADDRESS_TYPE>
    request.AddressType = &addressTypeRequest
    addressRequest := "<address>"
```

```
request.Address = &addressRequest
portRequest:= "<port>"
request.Port = &portRequest
request.Limit = int32(<limit>)
request.Offset = int32(<offset>)
enterpriseProjectIdRequest:= "<enterprise_project_id>"
request.EnterpriseProjectId = &enterpriseProjectIdRequest
fwInstanceIdRequest:= "<fw_instance_id>"
request.FwInstanceId = &fwInstanceIdRequest
response, err := client.ListBlackWhiteLists(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	查询黑白名单列表返回值
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.5 地址组管理

4.5.1 添加地址组

功能介绍

添加地址组

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/address-set

表 4-206 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-207 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考 API Explorer 和帮助中心 FAQ。默认情况下, fw_instance_id 为空时, 返回账号下第一个墙的信息; fw_instance_id 非空时, 返回与 fw_instance_id 对应墙的信息。

请求参数

表 4-208 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）

表 4-209 请求 Body 参数

参数	是否必选	参数类型	描述
object_id	是	String	防护对象id，是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id，可通过调用查询防火墙实例接口获得，注意type为0的为互联网边界防护对象id，type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
name	是	String	地址组名称
description	否	String	地址组描述
address_type	否	Integer	地址类型0 ipv4,1 ipv6 枚举值： <ul style="list-style-type: none">• 0• 1

响应参数

状态码： 200

表 4-210 响应 Body 参数

参数	参数类型	描述
data	IdObject object	添加地址组返回数据

表 4-211 IdObject

参数	参数类型	描述
id	String	id值
name	String	名称

状态码： 400

表 4-212 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度：8 最大长度：36
error_msg	String	错误描述 最小长度：2 最大长度：512

请求示例

添加项目id为14181c1245cf4fd786824efe1e2b9388，防护对象 id为1530de8a-522d-4771-9067-9fa4e2f53b48，名称为ceshi的ipV4的地址组。

```
https://{Endpoint}/v1/14181c1245cf4fd786824efe1e2b9388/address-set
{
  "object_id" : "1530de8a-522d-4771-9067-9fa4e2f53b48",
  "name" : "ceshi",
  "description" : "",
  "address_type" : 0
}
```

响应示例

状态码： 200

OK

```
{
  "data" : {
    "id" : "9dffcd62-23bf-4456-83fa-80fa0fee47db"
  }
}
```

状态码： 400

Bad Request

```
{
  "error_code" : "CFW.00900020",
  "error_msg" : "地址组超过最大数量限制"
}
```

SDK 代码示例

SDK代码示例如下。

Java

添加项目id为14181c1245cf4fd786824efe1e2b9388，防护对象 id为1530de8a-522d-4771-9067-9fa4e2f53b48，名称为ceshi的ipV4的地址组。

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class AddAddressSetSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        AddAddressSetRequest request = new AddAddressSetRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        AddAddressSetDto body = new AddAddressSetDto();
        body.withAddressType(AddAddressSetDto.AddressTypeEnum.NUMBER_0);
        body.withDescription("");
        body.withName("ceshi");
        body.withObjectId("1530de8a-522d-4771-9067-9fa4e2f53b48");
        request.withBody(body);
        try {
            AddAddressSetResponse response = client.addAddressSet(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

添加项目id为14181c1245cf4fd786824efe1e2b9388，防护对象 id为1530de8a-522d-4771-9067-9fa4e2f53b48，名称为ceshi的ipV4的地址组。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

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

try:
    request = AddAddressSetRequest()
    request.enterprise_project_id = "<enterprise_project_id>"
    request.fw_instance_id = "<fw_instance_id>"
    request.body = AddAddressSetDto(
        address_type=0,
        description="",
        name="ceshi",
        object_id="1530de8a-522d-4771-9067-9fa4e2f53b48"
    )
    response = client.add_address_set(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

添加项目id为14181c1245cf4fd786824efe1e2b9388，防护对象 id为1530de8a-522d-4771-9067-9fa4e2f53b48，名称为ceshi的ipV4的地址组。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddAddressSetRequest{
        enterpriseProjectIdRequest: "<enterprise_project_id>"
    }
```

```
request.EnterpriseProjectId = &enterpriseProjectIdRequest
fwInstanceIdRequest:= "<fw_instance_id>"
request.FwInstanceId = &fwInstanceIdRequest
addressTypeAddAddressSetDto:= model.GetAddAddressSetDtoAddressTypeEnum().E_0
descriptionAddAddressSetDto:= ""
request.Body = &model.AddAddressSetDto{
    AddressType: &addressTypeAddAddressSetDto,
    Description: &descriptionAddAddressSetDto,
    Name: "ceshi",
    Objectid: "1530de8a-522d-4771-9067-9fa4e2f53b48",
}
response, err := client.AddAddressSet(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	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.5.2 添加地址组成员

功能介绍

添加地址组成员

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/address-items

表 4-213 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-214 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考API Explorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

请求参数

表 4-215 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）

表 4-216 请求 Body 参数

参数	是否必选	参数类型	描述
set_id	否	String	地址组id
address_items	否	Array of address_items objects	地址组成员信息

表 4-217 address_items

参数	是否必选	参数类型	描述
name	是	String	地址名称
address_type	否	Integer	地址类型0 ipv4,1 ipv6
address	否	String	地址组ip信息
description	否	String	地址组成员描述

响应参数

状态码： 200

表 4-218 响应 Body 参数

参数	参数类型	描述
data	AddressItems object	添加地址组成员返回数据

表 4-219 AddressItems

参数	参数类型	描述
items	Array of IdObject objects	地址组成员id列表
covered_ip	Array of CoveredIPVO objects	覆盖ip列表

表 4-220 IdObject

参数	参数类型	描述
id	String	id值
name	String	名称

表 4-221 CoveredIPVO

参数	参数类型	描述
ip	String	ip地址

参数	参数类型	描述
covered_ip	String	覆盖ip地址。

状态码： 400

表 4-222 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

给项目id为9d80d070b6d44942af73c9c3d38e0429的项目下的set_id为8773c082-2a6c-4529-939a-edc28ef1a67c添加ip地址为2.2.2.2，名称为ceshi的地址组成员

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/address-items
```

```
{
  "set_id": "8773c082-2a6c-4529-939a-edc28ef1a67c",
  "address_items": [ {
    "description": "",
    "name": "ceshi",
    "address": "2.2.2.2"
  } ]
}
```

响应示例

状态码： 200

添加地址组成员返回值

```
{
  "data": {
    "covered_ip": [],
    "items": [ {
      "id": "65cb47fc-e666-4af4-8c2c-1fbd2f4b1eae"
    } ]
  }
}
```

状态码： 400

Bad Request

```
{
  "error_code": "CFW.00200001",
}
```

```
"error_msg" : "空参数错误"  
}
```

SDK 代码示例

SDK代码示例如下。

Java

给项目id为9d80d070b6d44942af73c9c3d38e0429的项目下的set_id为8773c082-2a6c-4529-939a-edc28ef1a67c添加ip地址为2.2.2.2，名称为ceshi的地址组成员

```
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.cfw.v1.region.CfwRegion;  
import com.huaweicloud.sdk.cfw.v1.*;  
import com.huaweicloud.sdk.cfw.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class AddAddressItemSolution {  
  
    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);  
  
        CfwClient client = CfwClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))  
            .build();  
        AddAddressItemRequest request = new AddAddressItemRequest();  
        request.withEnterpriseProjectId("<enterprise_project_id>");  
        request.withFwInstanceId("<fw_instance_id>");  
        AddAddressItemsInfoDto body = new AddAddressItemsInfoDto();  
        List<AddAddressItemsInfoDtoAddressItems> listbodyAddressItems = new ArrayList<>();  
        listbodyAddressItems.add(  
            new AddAddressItemsInfoDtoAddressItems()  
                .withName("ceshi")  
                .withAddress("2.2.2.2")  
                .withDescription("")  
        );  
        body.withAddressItems(listbodyAddressItems);  
        body.withSetId("8773c082-2a6c-4529-939a-edc28ef1a67c");  
        request.withBody(body);  
        try {  
            AddAddressItemResponse response = client.addAddressItem(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

给项目id为9d80d070b6d44942af73c9c3d38e0429的项目下的set_id为8773c082-2a6c-4529-939a-edc28ef1a67c添加ip地址为2.2.2.2，名称为ceshi的地址组成员

```
# coding: utf-8  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")  
    sk = __import__('os').getenv("CLOUD_SDK_SK")  
  
    credentials = BasicCredentials(ak, sk) \  
  
    client = CfwClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(CfwRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = AddAddressItemRequest()  
        request.enterprise_project_id = "<enterprise_project_id>"  
        request.fw_instance_id = "<fw_instance_id>"  
        listAddressItemsbody = [  
            AddAddressItemsInfoDtoAddressItems(  
                name="ceshi",  
                address="2.2.2.2",  
                description=""  
            )  
        ]  
        request.body = AddAddressItemsInfoDto(  
            address_items=listAddressItemsbody,  
            set_id="8773c082-2a6c-4529-939a-edc28ef1a67c"  
        )  
        response = client.add_address_item(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

给项目id为9d80d070b6d44942af73c9c3d38e0429的项目下的set_id为8773c082-2a6c-4529-939a-edc28ef1a67c添加ip地址为2.2.2.2，名称为ceshi的地址组成员

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddAddressItemRequest{}
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    addressAddressItems := "2.2.2.2"
    descriptionAddressItems := ""
    var listAddressItemsbody = []model.AddAddressItemsInfoDtoAddressItems{
        {
            Name: "ceshi",
            Address: &addressAddressItems,
            Description: &descriptionAddressItems,
        },
    }
    setIdAddAddressItemsInfoDto := "8773c082-2a6c-4529-939a-edc28ef1a67c"
    request.Body = &model.AddAddressItemsInfoDto{
        AddressItems: &listAddressItemsbody,
        SetId: &setIdAddAddressItemsInfoDto,
    }
    response, err := client.AddAddressItem(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	添加地址组成员返回值
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.5.3 删除地址组

功能介绍

删除地址组

调用方法

请参见[如何调用API](#)。

URI

DELETE /v1/{project_id}/address-sets/{set_id}

表 4-223 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id
set_id	是	String	地址组id

表 4-224 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。

参数	是否必选	参数类型	描述
fw_instance_id	否	String	防火墙实例id，创建云防火墙后用于标志防火墙由系统自动生成的标志id，可通过调用查询防火墙实例接口获得。具体可参考API Explorer和帮助中心FAQ。默认情况下，fw_instance_id为空时，返回账号下第一个墙的信息；fw_instance_id非空时，返回与fw_instance_id对应墙的信息。

请求参数

表 4-225 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）

响应参数

状态码： 200

表 4-226 响应 Body 参数

参数	参数类型	描述
data	JsonObject object	删除地址组返回数据

表 4-227 JsonObject

参数	参数类型	描述
id	String	id值
name	String	名称

状态码： 400

表 4-228 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度：8 最大长度：36
error_msg	String	错误描述 最小长度：2 最大长度：512

请求示例

删除项目id为9d80d070b6d44942af73c9c3d38e0429下的地址组id为cf18f0b1-0ce7-4eb8-83b6-4b33c8448e16的地址组

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/address-sets/  
cf18f0b1-0ce7-4eb8-83b6-4b33c8448e16
```

响应示例

状态码： 200

OK

```
{  
  "data": {  
    "id": "cf18f0b1-0ce7-4eb8-83b6-4b33c8448e16",  
    "name": "test"  
  }  
}
```

状态码： 400

Bad Request

```
{  
  "error_code": "CFW.00200004",  
  "error_msg": "所删除资源被引用，删除失败"  
}
```

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.cfw.v1.region.CfwRegion;  
import com.huaweicloud.sdk.cfw.v1.*;  
import com.huaweicloud.sdk.cfw.v1.model.*;
```

```
public class DeleteAddressSetSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteAddressSetRequest request = new DeleteAddressSetRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        try {
            DeleteAddressSetResponse response = client.deleteAddressSet(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = DeleteAddressSetRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
```

```
response = client.delete_address_set(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteAddressSetRequest{}
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    response, err := client.DeleteAddressSet(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	Bad Request

状态码	描述
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.5.4 删除地址组成员

功能介绍

删除地址组成员

调用方法

请参见[如何调用API](#)。

URI

DELETE /v1/{project_id}/address-items/{item_id}

表 4-229 路径参数

参数	是否必选	参数类型	描述
item_id	是	String	地址组成员id
project_id	是	String	租户项目id

表 4-230 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id，用户支持企业项目后，由企业项目生成的id。

参数	是否必选	参数类型	描述
fw_instance_id	否	String	防火墙实例id，创建云防火墙后用于标志防火墙由系统自动生成的标志id，可通过调用查询防火墙实例接口获得。具体可参考API Explorer和帮助中心FAQ。默认情况下，fw_instance_id为空时，返回账号下第一个墙的信息；fw_instance_id非空时，返回与fw_instance_id对应墙的信息。

请求参数

表 4-231 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）

响应参数

状态码： 200

表 4-232 响应 Body 参数

参数	参数类型	描述
data	IdObject object	删除地址组成员id数据

表 4-233 IdObject

参数	参数类型	描述
id	String	id值
name	String	名称

状态码： 400

表 4-234 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度：8 最大长度：36
error_msg	String	错误描述 最小长度：2 最大长度：512

请求示例

删除项目id为9d80d070b6d44942af73c9c3d38e0429下的地址组成员id为65cb47fc-e666-4af4-8c2c-1fbd2f4b1eae的地址组成员

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/address-items/65cb47fc-e666-4af4-8c2c-1fbd2f4b1eae
```

响应示例

状态码： 200

删除地址组成员返回值

```
{
  "data": {
    "id": "65cb47fc-e666-4af4-8c2c-1fbd2f4b1eae",
    "name": "test"
  }
}
```

状态码： 400

Bad Request

```
{
  "error_code": "CFW.0020016",
  "error_msg": "实例状态错误"
}
```

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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;
```



```
public class DeleteAddressItemSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteAddressItemRequest request = new DeleteAddressItemRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        try {
            DeleteAddressItemResponse response = client.deleteAddressItem(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = DeleteAddressItemRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
```

```
response = client.delete_address_item(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteAddressItemRequest{}
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    response, err := client.DeleteAddressItem(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	删除地址组成员返回值
400	Bad Request

状态码	描述
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.5.5 批量删除地址组成员

功能介绍

批量删除地址组成员

调用方法

请参见[如何调用API](#)。

URI

DELETE /v1/{project_id}/address-items

表 4-235 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-236 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。

参数	是否必选	参数类型	描述
fw_instance_id	否	String	防火墙实例id，创建云防火墙后用于标志防火墙由系统自动生成的标志id，可通过调用查询防火墙实例接口获得。具体可参考API Explorer和帮助中心FAQ。默认情况下，fw_instance_id为空时，返回账号下第一个墙的信息；fw_instance_id非空时，返回与fw_instance_id对应墙的信息。

请求参数

表 4-237 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）

表 4-238 请求 Body 参数

参数	是否必选	参数类型	描述
set_id	是	String	地址组id
address_item_ids	是	Array of strings	地址组成员id列表

响应参数

状态码： 200

表 4-239 响应 Body 参数

参数	参数类型	描述
data	Array of strings	批量删除地址组成员id列表

请求示例

删除项目9d80d070b6d44942af73c9c3d38e0429地址组id为e4884376-7efb-40e7-b98b-13668d6f8b85下的地址组成员id为d072ad2e-033c-40a9-b0b5-751f9c2943a6

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/address-items?
fw_instance_id=7a004e79-0b8b-4679-ab20-267f3946e8ba&enterprise_project_id=default

{
  "set_id" : "e4884376-7efb-40e7-b98b-13668d6f8b85",
  "address_item_ids" : [ "d072ad2e-033c-40a9-b0b5-751f9c2943a6" ]
}
```

响应示例

状态码： 200

```
{
  "data" : [ "d072ad2e-033c-40a9-b0b5-751f9c2943a6" ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

删除项目9d80d070b6d44942af73c9c3d38e0429地址组id为e4884376-7efb-40e7-b98b-13668d6f8b85下的地址组成员id为d072ad2e-033c-40a9-b0b5-751f9c2943a6

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

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

public class BatchDeleteAddressItemsSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchDeleteAddressItemsRequest request = new BatchDeleteAddressItemsRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
```

```
request.withFwInstanceId("<fw_instance_id>");
DeleteAddressItemsInfoDto body = new DeleteAddressItemsInfoDto();
List<String> listbodyAddressItemIds = new ArrayList<>();
listbodyAddressItemIds.add("d072ad2e-033c-40a9-b0b5-751f9c2943a6");
body.withAddressItemIds(listbodyAddressItemIds);
body.withSetId("e4884376-7efb-40e7-b98b-13668d6f8b85");
request.withBody(body);
try {
    BatchDeleteAddressItemsResponse response = client.batchDeleteAddressItems(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.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

删除项目9d80d070b6d44942af73c9c3d38e0429地址组id为e4884376-7efb-40e7-b98b-13668d6f8b85下的地址组成员id为d072ad2e-033c-40a9-b0b5-751f9c2943a6

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = BatchDeleteAddressItemsRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        listAddressItemIdsbody = [
            "d072ad2e-033c-40a9-b0b5-751f9c2943a6"
        ]
        request.body = DeleteAddressItemsInfoDto(
            address_item_ids=listAddressItemIdsbody,
            set_id="e4884376-7efb-40e7-b98b-13668d6f8b85"
        )
        response = client.batch_delete_address_items(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

删除项目9d80d070b6d44942af73c9c3d38e0429地址组id为e4884376-7efb-40e7-b98b-13668d6f8b85下的地址组成员id为d072ad2e-033c-40a9-b0b5-751f9c2943a6

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteAddressItemsRequest{}
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    var listAddressItemIdsbody = []string{
        "d072ad2e-033c-40a9-b0b5-751f9c2943a6",
    }
    request.Body = &model.DeleteAddressItemsInfoDto{
        AddressItemIds: listAddressItemIdsbody,
        SetId: "e4884376-7efb-40e7-b98b-13668d6f8b85",
    }
    response, err := client.BatchDeleteAddressItems(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	

错误码

请参见[错误码](#)。

4.5.6 更新地址组信息

功能介绍

更新地址组信息

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/address-sets/{set_id}

表 4-240 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id
set_id	是	String	地址组id

表 4-241 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

请求参数

表 4-242 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）

表 4-243 请求 Body 参数

参数	是否必选	参数类型	描述
name	否	String	地址组名称
description	否	String	地址组描述
address_type	否	Integer	地址类型0 ipv4,1 ipv6,2 domain 枚举值： <ul style="list-style-type: none">• 0• 1• 2

响应参数

状态码： 200

表 4-244 响应 Body 参数

参数	参数类型	描述
data	UpdateAddressSetResponseData object	更新地址组返回数据

表 4-245 UpdateAddressSetResponseData

参数	参数类型	描述
id	String	Id

状态码： 400

表 4-246 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度：8 最大长度：36
error_msg	String	错误描述 最小长度：2 最大长度：512

请求示例

更新项目id为9d80d070b6d44942af73c9c3d38e0429下的地址组id为cf18f0b1-0ce7-4eb8-83b6-4b33c8448e16的地址组的名称改为ABCD，地址组类型改为 ipV4

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/address-sets/  
cf18f0b1-0ce7-4eb8-83b6-4b33c8448e16
```

```
{  
  "name" : "ABCD",  
  "description" : "",  
  "address_type" : 0  
}
```

响应示例

状态码： 200

OK

```
{  
  "data" : {  
    "id" : "cf18f0b1-0ce7-4eb8-83b6-4b33c8448e16"  
  }  
}
```

状态码： 400

Bad Request

```
{  
  "error_code" : "CFW.00200005",  
  "error_msg" : "操作内容不存在"  
}
```

SDK 代码示例

SDK代码示例如下。

Java

更新项目id为9d80d070b6d44942af73c9c3d38e0429下的地址组id为cf18f0b1-0ce7-4eb8-83b6-4b33c8448e16的地址组的名称改为ABCD，地址组类型改为 ipV4

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class UpdateAddressSetSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateAddressSetRequest request = new UpdateAddressSetRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        UpdateAddressSetDto body = new UpdateAddressSetDto();
        body.withAddressType(UpdateAddressSetDto.AddressTypeEnum.NUMBER_0);
        body.withDescription("");
        body.withName("ABCD");
        request.withBody(body);
        try {
            UpdateAddressSetResponse response = client.updateAddressSet(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

更新项目id为9d80d070b6d44942af73c9c3d38e0429下的地址组id为cf18f0b1-0ce7-4eb8-83b6-4b33c8448e16的地址组的名称改为ABCD，地址组类型改为ipV4

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateAddressSetRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        request.body = UpdateAddressSetDto(
            address_type=0,
            description="",
            name="ABCD"
        )
        response = client.update_address_set(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

更新项目id为9d80d070b6d44942af73c9c3d38e0429下的地址组id为cf18f0b1-0ce7-4eb8-83b6-4b33c8448e16的地址组的名称改为ABCD，地址组类型改为ipV4

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```
Build()  
  
request := &model.UpdateAddressSetRequest{  
    enterpriseProjectIdRequest:= "<enterprise_project_id>"  
    request.EnterpriseProjectId = &enterpriseProjectIdRequest  
    fwInstanceIdRequest:= "<fw_instance_id>"  
    request.FwInstanceId = &fwInstanceIdRequest  
    addressTypeUpdateAddressSetDto:= model.GetUpdateAddressSetDtoAddressTypeEnum().E_0  
    descriptionUpdateAddressSetDto:= ""  
    nameUpdateAddressSetDto:= "ABCD"  
    request.Body = &model.UpdateAddressSetDto{  
        AddressType: &addressTypeUpdateAddressSetDto,  
        Description: &descriptionUpdateAddressSetDto,  
        Name: &nameUpdateAddressSetDto,  
    }  
}  
response, err := client.UpdateAddressSet(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	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.5.7 查询地址组列表

功能介绍

查询地址组列表

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/address-sets

表 4-247 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-248 Query 参数

参数	是否必选	参数类型	描述
object_id	是	String	防护对象id，是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id，可通过调用查询防火墙实例接口获得，注意type为0的为互联网边界防护对象id，type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
key_word	否	String	关键字
limit	是	Integer	每页显示个数，范围为1-1024
offset	是	Integer	偏移量：指定返回记录的开始位置，必须为数字，取值范围为大于或等于0，默认0
address	否	String	ip地址
address_type	否	Integer	地址类型0 ipv4,1 ipv6 枚举值： <ul style="list-style-type: none">• 0• 1
enterprise_project_id	否	String	企业项目id，用户支持企业项目后，由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id，创建云防火墙后用于标志防火墙由系统自动生成的标志id，可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下，fw_instance_id为空时，返回账号下第一个墙的信息；fw_instance_id非空时，返回与fw_instance_id对应墙的信息。

参数	是否必选	参数类型	描述
query_address_set_type	否	Integer	查询地址组类型，0表示自定义地址组，1表示预定义地址组
address_set_type	否	Integer	地址组类型，0表示自定义地址组，1表示WAF回源IP地址组，2表示DDoS回源IP地址组，3表示NAT64转换地址组

请求参数

表 4-249 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）

响应参数

状态码： 200

表 4-250 响应 Body 参数

参数	参数类型	描述
data	data object	查询地址组列表返回数据

表 4-251 data

参数	参数类型	描述
offset	Integer	偏移量：指定返回记录的开始位置，必须为数字，取值范围为大于或等于0，默认0
limit	Integer	每页显示个数，范围为1-1024
total	Integer	总数
records	Array of records objects	地址组列表

表 4-252 records

参数	参数类型	描述
set_id	String	地址组id
ref_count	Integer	引用次数
description	String	描述信息
name	String	地址组名称
address_type	Integer	地址类型0 ipv4,1 ipv6
object_id	String	防护对象id, 是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id, 可通过调用查询防火墙实例接口获得, 注意type为0的为互联网边界防护对象id, type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
address_set_type	Integer	地址组类型, 0表示自定义地址组, 1表示WAF回源IP地址组, 2表示DDoS回源IP地址组, 3表示NAT64转换地址组

状态码: 400

表 4-253 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度: 8 最大长度: 36
error_msg	String	错误描述 最小长度: 2 最大长度: 512

请求示例

查询项目id为5c69cf330cda42369cbd726ee1bc5e76, 防护对象id为8a41d6a5-f215-428a-a76c-dc923b5d599a的第一页的ip地址组信息

```
https://{Endpoint}/v1/5c69cf330cda42369cbd726ee1bc5e76/address-sets?object_id=8a41d6a5-f215-428a-a76c-dc923b5d599a&limit=10&offset=0
```

响应示例

状态码: 200

OK

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



```
"limit" : 10,
"offset" : 0,
"records" : [ {
  "address_set_type" : 0,
  "object_id" : "cf18f0b1-0ce7-4eb8-83b6-4b33c8448e16",
  "address_type" : 0,
  "description" : "",
  "name" : "test",
  "ref_count" : 0,
  "set_id" : "50da1eff-e58d-4380-b899-a78f94137d3b"
} ],
"total" : 1
}
}
```

状态码： 400

Bad Request

```
{
  "error_code" : "CFW.0020016",
  "error_msg" : "实例状态错误"
}
```

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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class ListAddressSetsSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListAddressSetsRequest request = new ListAddressSetsRequest();
        request withObjectId("<object_id>");
        request withKeyword("<key_word>");
        request withLimit("<limit>");
        request withOffset("<offset>");
        request withAddress("<address>");
        request withAddressType(ListAddressSetsRequest.AddressTypeEnum.NUMBER_<address_type>);
    }
}
```

```
request.withEnterpriseProjectId("<enterprise_project_id>");
request.withFwInstanceId("<fw_instance_id>");
request.withQueryAddressSetType("<query_address_set_type>");
try {
    ListAddressSetsResponse response = client.listAddressSets(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListAddressSetsRequest()
        request.object_id = "<object_id>"
        request.key_word = "<key_word>"
        request.limit = <limit>
        request.offset = <offset>
        request.address = "<address>"
        request.address_type = <address_type>
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        request.query_address_set_type = <query_address_set_type>
        response = client.list_address_sets(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"
cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAddressSetsRequest{}
    request.ObjectId = "<object_id>"
    keyWordRequest := "<key_word>"
    request.KeyWord = &keyWordRequest
    request.Limit = int32(<limit>)
    request.Offset = int32(<offset>)
    addressRequest := "<address>"
    request.Address = &addressRequest
    addressTypeRequest := model.GetListAddressSetsRequestAddressTypeEnum().<ADDRESS_TYPE>
    request.AddressType = &addressTypeRequest
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    queryAddressSetTypeRequest := int32(<query_address_set_type>)
    request.QueryAddressSetType = &queryAddressSetTypeRequest
    response, err := client.ListAddressSets(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	Bad Request
401	Unauthorized

状态码	描述
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.5.8 查询地址组详细信息

功能介绍

查询地址组详细

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/address-sets/{set_id}

表 4-254 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id
set_id	是	String	地址组id

表 4-255 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。

参数	是否必选	参数类型	描述
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。
query_address_set_type	否	Integer	查询地址组类型, 0表示自定义地址组, 1表示预定义地址组

请求参数

表 4-256 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）

响应参数

状态码： 200

表 4-257 响应 Body 参数

参数	参数类型	描述
data	data object	查询地址组详情数据

表 4-258 data

参数	参数类型	描述
id	String	地址组id
name	String	地址组名称
description	String	地址组描述

参数	参数类型	描述
address_set_type	Integer	地址组类型，0表示自定义地址组，1表示WAF回源IP地址组，2表示DDoS回源IP地址组，3表示NAT64转换地址组
address_type	Integer	地址类型0 ipv4,1 ipv6 枚举值： <ul style="list-style-type: none">• 0• 1

状态码： 400

表 4-259 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

查询项目id为9d80d070b6d44942af73c9c3d38e0429，地址组id为cf18f0b1-0ce7-4eb8-83b6-4b33c8448e16的地址组的详情

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/address-sets/cf18f0b1-0ce7-4eb8-83b6-4b33c8448e16
```

响应示例

状态码： 200

OK

```
{
  "data": {
    "address_set_type": 0,
    "address_type": 0,
    "description": "",
    "id": "cf18f0b1-0ce7-4eb8-83b6-4b33c8448e16",
    "name": "ABC"
  }
}
```

状态码： 400

Bad Request

```
{
  "error_code" : "CFW.00200005",
  "error_msg" : "操作内容不存在"
}
```

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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class ListAddressSetDetailSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListAddressSetDetailRequest request = new ListAddressSetDetailRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        request.withQueryAddressSetType("<query_address_set_type>");
        try {
            ListAddressSetDetailResponse response = client.listAddressSetDetail(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListAddressSetDetailRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        request.query_address_set_type = <query_address_set_type>
        response = client.list_address_set_detail(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAddressSetDetailRequest{}
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
```



```
fwInstanceIdRequest:= "<fw_instance_id>"
request.FwInstanceId = &fwInstanceIdRequest
queryAddressSetTypeRequest:= int32(<query_address_set_type>)
request.QueryAddressSetType = &queryAddressSetTypeRequest
response, err := client.ListAddressSetDetail(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	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.5.9 查询地址组成员

功能介绍

查询地址组成员

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/address-items

表 4-260 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-261 Query 参数

参数	是否必选	参数类型	描述
set_id	是	String	地址组id
key_word	否	String	关键字
limit	是	Integer	每页显示个数，范围为1-1024
offset	是	Integer	偏移量：指定返回记录的开始位置，必须为数字，取值范围为大于或等于0，默认0
address	否	String	ip地址
enterprise_project_id	否	String	企业项目id，用户支持企业项目后，由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id，创建云防火墙后用于标志防火墙由系统自动生成的标志id，可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下，fw_instance_id为空时，返回账号下第一个墙的信息；fw_instance_id非空时，返回与fw_instance_id对应墙的信息。
query_address_set_type	否	Integer	查询地址组类型，0表示自定义地址组，1表示预定义地址组

请求参数

表 4-262 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）

响应参数

状态码： 200

表 4-263 响应 Body 参数

参数	参数类型	描述
data	data object	查询地址组成员返回数据

表 4-264 data

参数	参数类型	描述
offset	Integer	偏移量：指定返回记录的开始位置，必须为数字，取值范围为大于或等于0，默认0
limit	Integer	每页显示个数，范围为1-1024
total	Integer	总数
set_id	String	地址组id
records	Array of records objects	成员信息

表 4-265 records

参数	参数类型	描述
item_id	String	地址组成员id
name	String	地址组成员name
description	String	描述
address_type	Integer	地址组类型，0 ipv4,1 ipv6
address	String	地址组

状态码： 400

表 4-266 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36

参数	参数类型	描述
error_msg	String	错误描述 最小长度：2 最大长度：512

请求示例

查询项目id为9d80d070b6d44942af73c9c3d38e0429的项目的地址组id为8773c082-2a6c-4529-939a-edc28ef1a67c的地址组成员信息

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/address-items?  
set_id=8773c082-2a6c-4529-939a-edc28ef1a67c&limit=10&offset=0
```

响应示例

状态码：200

查询地址组成员返回值

```
{  
  "data": {  
    "limit": 10,  
    "offset": 0,  
    "records": [ {  
      "address": "1.1.1.1",  
      "address_type": 0,  
      "description": "",  
      "item_id": "294fab71-34bf-4858-a380-8f7530e1c816"  
    } ],  
    "set_id": "8773c082-2a6c-4529-939a-edc28ef1a67c",  
    "total": 1  
  }  
}
```

状态码：400

Bad Request

```
{  
  "error_code": "CFW.00200005",  
  "error_msg": "操作内容不存在"  
}
```

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.cfw.v1.region.CfwRegion;  
import com.huaweicloud.sdk.cfw.v1.*;  
import com.huaweicloud.sdk.cfw.v1.model.*;
```

```
public class ListAddressItemsSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListAddressItemsRequest request = new ListAddressItemsRequest();
        request.withSetId("<set_id>");
        request.withKeyword("<key_word>");
        request.withLimit(<limit>);
        request.withOffset(<offset>);
        request.withAddress("<address>");
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        try {
            ListAddressItemsResponse response = client.listAddressItems(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CfwClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CfwRegion.value_of("<YOUR REGION>")) \
```

```
.build()

try:
    request = ListAddressItemsRequest()
    request.set_id = "<set_id>"
    request.key_word = "<key_word>"
    request.limit = <limit>
    request.offset = <offset>
    request.address = "<address>"
    request.enterprise_project_id = "<enterprise_project_id>"
    request.fw_instance_id = "<fw_instance_id>"
    response = client.list_address_items(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAddressItemsRequest{}
    request.SetId = "<set_id>"
    keyWordRequest := "<key_word>"
    request.KeyWord = &keyWordRequest
    request.Limit = int32(<limit>)
    request.Offset = int32(<offset>)
    addressRequest := "<address>"
    request.Address = &addressRequest
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    response, err := client.ListAddressItems(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	查询地址组成员返回值
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.6 服务组管理

4.6.1 新建服务组

功能介绍

创建服务组

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/service-set

表 4-267 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-268 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

请求参数

表 4-269 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）

表 4-270 请求 Body 参数

参数	是否必选	参数类型	描述
object_id	是	String	防护对象id, 是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id, 可通过调用查询防火墙实例接口获得, 注意type为0的为互联网边界防护对象id, type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
name	是	String	服务组名称 最小长度: 1 最大长度: 255
description	否	String	服务组描述信息 最小长度: 1 最大长度: 255

响应参数

状态码： 200

表 4-271 响应 Body 参数

参数	参数类型	描述
data	IdObject object	新建服务组返回数据

表 4-272 IdObject

参数	参数类型	描述
id	String	id值
name	String	名称

状态码： 400

表 4-273 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

添加项目id为9d80d070b6d44942af73c9c3d38e0429，防护对象为cfebd347-b655-4b84-b938-3c54317599b2，名称为ceshi的服务组。

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/service-set
```

```
{  
  "object_id": "cfebd347-b655-4b84-b938-3c54317599b2",  
  "name": "ceshi",  
  "description": ""  
}
```

响应示例

状态码： 200

新建服务组返回值

```
{
  "data" : {
    "id" : "221cfdca-3abf-4c30-ab0d-516a03c70866"
  }
}
```

状态码： 400

Bad Request

```
{
  "error_code" : "CFW.00200024",
  "error_msg" : "超出最大数量限制"
}
```

SDK 代码示例

SDK代码示例如下。

Java

添加项目id为9d80d070b6d44942af73c9c3d38e0429，防护对象为cfebd347-b655-4b84-b938-3c54317599b2，名称为ceshi的服务组。

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class AddServiceSetSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        AddServiceSetRequest request = new AddServiceSetRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        AddServiceSetUsingPOSTRequestBody body = new AddServiceSetUsingPOSTRequestBody();
        body.withDescription("");
        body.withName("ceshi");
        body.withObjectId("cfebd347-b655-4b84-b938-3c54317599b2");
        request.withBody(body);
        try {
            AddServiceSetResponse response = client.addServiceSet(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

添加项目id为9d80d070b6d44942af73c9c3d38e0429，防护对象为cfebd347-b655-4b84-b938-3c54317599b2，名称为ceshi的服务组。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = AddServiceSetRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        request.body = AddServiceSetUsingPOSTRequestBody(
            description="",
            name="ceshi",
            object_id="cfebd347-b655-4b84-b938-3c54317599b2"
        )
        response = client.add_service_set(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

添加项目id为9d80d070b6d44942af73c9c3d38e0429，防护对象为cfebd347-b655-4b84-b938-3c54317599b2，名称为ceshi的服务组。

```
package main
```

```
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(  
        cfw.CfwClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.AddServiceSetRequest{}  
    enterpriseProjectIdRequest := "<enterprise_project_id>"  
    request.EnterpriseProjectId = &enterpriseProjectIdRequest  
    fwInstanceIdRequest := "<fw_instance_id>"  
    request.FwInstanceId = &fwInstanceIdRequest  
    descriptionAddServiceSetUsingPostRequestBody := ""  
    request.Body = &model.AddServiceSetUsingPostRequestBody{  
        Description: &descriptionAddServiceSetUsingPostRequestBody,  
        Name: "ceshi",  
        Objectid: "cfebd347-b655-4b84-b938-3c54317599b2",  
    }  
    response, err := client.AddServiceSet(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	新建服务组返回值
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found

状态码	描述
500	Internal Server Error

错误码

请参见[错误码](#)。

4.6.2 新建服务成员

功能介绍

批量添加服务组成员

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/service-items

表 4-274 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-275 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

请求参数

表 4-276 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）

表 4-277 请求 Body 参数

参数	是否必选	参数类型	描述
set_id	是	String	服务组id
service_items	是	Array of service_items objects	添加服务组成员

表 4-278 service_items

参数	是否必选	参数类型	描述
item_id	否	String	服务成员id
protocol	是	Integer	协议类型:TCP为6, UDP为17, ICMP为1, ICMPV6为58, ANY为-1, 手动类型不为空, 自动类型为空
source_port	是	String	源端口
dest_port	是	String	目的端口
name	否	String	服务成员名称
description	否	String	服务成员描述

响应参数

状态码： 200

表 4-279 响应 Body 参数

参数	参数类型	描述
data	ServiceItemIds object	新建服务组成员返回数据

表 4-280 ServiceItemIds

参数	参数类型	描述
items	Array of IdObject objects	服务组成员id列表

表 4-281 IdObject

参数	参数类型	描述
id	String	id值
name	String	名称

状态码： 400

表 4-282 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

给项目id为9d80d070b6d44942af73c9c3d38e0429的项目添加名称为ceshi的服务组成员，描述为添加服务组成员

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/service-items
```

```
{
  "set_id": "7cdebed3-af07-494e-a3c2-b88bb8d58b57",
  "service_items": [{
    "description": "添加服务组成员",
```

```
"name" : "ceshi",  
"dest_port" : "1",  
"source_port" : "1",  
"protocol" : 6  
}]  
}
```

响应示例

状态码： 200

添加服务组成员返回值

```
{  
  "data" : {  
    "items" : [ {  
      "id" : "cc41c4af-86e8-4ed2-80ad-87d399aeaed0"  
    } ]  
  }  
}
```

状态码： 400

Bad Request

```
{  
  "error_code" : "CFW.00200001",  
  "error_msg" : "空参数错误"  
}
```

SDK 代码示例

SDK代码示例如下。

Java

给项目id为9d80d070b6d44942af73c9c3d38e0429的项目添加名称为ceshi的服务组成员，描述为添加服务组成员

```
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.cfw.v1.region.CfwRegion;  
import com.huaweicloud.sdk.cfw.v1.*;  
import com.huaweicloud.sdk.cfw.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class AddServiceItemsSolution {  
  
    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);

CfwClient client = CfwClient.newBuilder()
    .withCredential(auth)
    .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
    .build();
AddServiceItemsRequest request = new AddServiceItemsRequest();
request.withEnterpriseProjectId("<enterprise_project_id>");
request.withFwInstanceId("<fw_instance_id>");
AddServiceItemsUsingPOSTRequestBody body = new AddServiceItemsUsingPOSTRequestBody();
List<AddServiceItemsUsingPOSTRequestBodyServiceItems> listbodyServiceItems = new ArrayList<>();
listbodyServiceItems.add(
    new AddServiceItemsUsingPOSTRequestBodyServiceItems()
        .withProtocol(6)
        .withSourcePort("1")
        .withDestPort("1")
        .withName("ceshi")
        .withDescription("添加服务组成员")
);
body.withServiceItems(listbodyServiceItems);
body.withSetId("7cdebed3-af07-494e-a3c2-b88bb8d58b57");
request.withBody(body);
try {
    AddServiceItemsResponse response = client.addServiceItems(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.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

给项目id为9d80d070b6d44942af73c9c3d38e0429的项目添加名称为ceshi的服务组成员，描述为添加服务组成员

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = AddServiceItemsRequest()
```

```
request.enterprise_project_id = "<enterprise_project_id>"
request.fw_instance_id = "<fw_instance_id>"
listServiceItemsbody = [
    AddServiceItemsUsingPOSTRequestBodyServiceItems(
        protocol=6,
        source_port="1",
        dest_port="1",
        name="ceshi",
        description="添加服务组成员"
    )
]
request.body = AddServiceItemsUsingPOSTRequestBody(
    service_items=listServiceItemsbody,
    set_id="7cdebed3-af07-494e-a3c2-b88bb8d58b57"
)
response = client.add_service_items(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

给项目id为9d80d070b6d44942af73c9c3d38e0429的项目添加名称为ceshi的服务组成员，描述为添加服务组成员

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddServiceItemsRequest{}
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    nameServiceItems := "ceshi"
    descriptionServiceItems := "添加服务组成员"
    var listServiceItemsbody = []model.AddServiceItemsUsingPostRequestBodyServiceItems{
        {
            Protocol: int32(6),
            SourcePort: "1",
        }
    }
}
```

```
        DestPort: "1",
        Name: &nameServiceItems,
        Description: &descriptionServiceItems,
    },
}
request.Body = &model.AddServiceItemsUsingPostRequestBody{
    ServiceItems: listServiceItemsbody,
    SetId: "7cdebed3-af07-494e-a3c2-b88bb8d58b57",
}
response, err := client.AddServiceItems(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	添加服务组成员返回值
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.6.3 删除服务组

功能介绍

删除服务组

调用方法

请参见[如何调用API](#)。

URI

DELETE /v1/{project_id}/service-sets/{set_id}

表 4-283 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id
set_id	是	String	服务集合id

表 4-284 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

请求参数

表 4-285 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）

响应参数

状态码： 200

表 4-286 响应 Body 参数

参数	参数类型	描述
data	IdObject object	删除服务组返回数据

表 4-287 IdObject

参数	参数类型	描述
id	String	id值
name	String	名称

状态码： 400

表 4-288 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

删除项目id为9d80d070b6d44942af73c9c3d38e0429，服务组id为221cfdca-3abf-4c30-ab0d-516a03c70866的服务组

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/service-sets/221cfdca-3abf-4c30-ab0d-516a03c70866
```

响应示例

状态码： 200

OK

```
{
  "data": {
    "id": "221cfdca-3abf-4c30-ab0d-516a03c70866",
    "name": "test"
  }
}
```

状态码： 400

Bad Request

```
{
  "error_code": "CFW.00200004",
  "error_msg": "所删除资源被引用，删除失败"
}
```

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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class DeleteServiceSetSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteServiceSetRequest request = new DeleteServiceSetRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        try {
            DeleteServiceSetResponse response = client.deleteServiceSet(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
```

```
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

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

try:
    request = DeleteServiceSetRequest()
    request.enterprise_project_id = "<enterprise_project_id>"
    request.fw_instance_id = "<fw_instance_id>"
    response = client.delete_service_set(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteServiceSetRequest{}
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    response, err := client.DeleteServiceSet(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	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.6.4 删除服务成员

功能介绍

删除服务组成员

调用方法

请参见[如何调用API](#)。

URI

DELETE /v1/{project_id}/service-items/{item_id}

表 4-289 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id
item_id	是	String	服务组成员id

表 4-290 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

请求参数

表 4-291 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）

响应参数

状态码： 200

表 4-292 响应 Body 参数

参数	参数类型	描述
data	IdObject object	删除服务组成员数据

表 4-293 IdObject

参数	参数类型	描述
id	String	id值
name	String	名称

状态码： 400

表 4-294 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

删除项目id为9d80d070b6d44942af73c9c3d38e0429，服务组成员id为6b37ed55-1e21-46a5-a7dc-a59ef418d359的服务组成员

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/service-items/6b37ed55-1e21-46a5-a7dc-a59ef418d359
```

响应示例

状态码： 200

删除服务组成员返回信息

```
{
  "data": {
    "id": "26f562c4-fe11-43d0-9654-f54298d5b12e"
  }
}
```

状态码： 400

Bad Request

```
{
  "error_code": "CFW.0020016",
  "error_msg": "实例状态错误"
}
```

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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
```

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

public class DeleteServiceItemSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteServiceItemRequest request = new DeleteServiceItemRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        try {
            DeleteServiceItemResponse response = client.deleteServiceItem(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = DeleteServiceItemRequest()
```

```
request.enterprise_project_id = "<enterprise_project_id>"
request.fw_instance_id = "<fw_instance_id>"
response = client.delete_service_item(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteServiceItemRequest{}
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    response, err := client.DeleteServiceItem(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	删除服务组成员返回信息

状态码	描述
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.6.5 批量删除服务组成员信息

功能介绍

批量删除服务组成员信息

调用方法

请参见[如何调用API](#)。

URI

DELETE /v1/{project_id}/service-items

表 4-295 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-296 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。

参数	是否必选	参数类型	描述
fw_instance_id	否	String	防火墙实例id，创建云防火墙后用于标志防火墙由系统自动生成的标志id，可通过调用查询防火墙实例接口获得。具体可参考API Explorer和帮助中心FAQ。默认情况下，fw_instance_id为空时，返回账号下第一个墙的信息；fw_instance_id非空时，返回与fw_instance_id对应墙的信息。

请求参数

表 4-297 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）

表 4-298 请求 Body 参数

参数	是否必选	参数类型	描述
set_id	是	String	服务组id
service_item_ids	是	Array of strings	服务组成员id列表

响应参数

状态码： 200

表 4-299 响应 Body 参数

参数	参数类型	描述
data	Array of strings	批量删除服务组成员id列表

请求示例

删除项目id为9d80d070b6d44942af73c9c3d38e0429的服务组id为688faf62-20fc-4ca6-b9f9-6fbc518df5ae下的服务组成员f837f7ae-22c9-449d-a99c-4be24533e243

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/service-items?
fw_instance_id=7a004e79-0b8b-4679-ab20-267f3946e8ba&enterprise_project_id=default

{
  "set_id" : "688faf62-20fc-4ca6-b9f9-6fbc518df5ae",
  "service_item_ids" : [ "f837f7ae-22c9-449d-a99c-4be24533e243" ]
}
```

响应示例

状态码： 200

批量删除服务组成员返回值

```
{
  "data" : [ "f837f7ae-22c9-449d-a99c-4be24533e243" ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

删除项目id为9d80d070b6d44942af73c9c3d38e0429的服务组id为688faf62-20fc-4ca6-b9f9-6fbc518df5ae下的服务组成员f837f7ae-22c9-449d-a99c-4be24533e243

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

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

public class BatchDeleteServiceItemsSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
```

```
.withCredential(auth)
.withRegion(CfwRegion.valueOf("<YOUR REGION>"))
.build();
BatchDeleteServiceItemsRequest request = new BatchDeleteServiceItemsRequest();
request.withEnterpriseProjectId("<enterprise_project_id>");
request.withFwInstanceId("<fw_instance_id>");
DeleteServiceItemDto body = new DeleteServiceItemDto();
List<String> listbodyServiceItemIds = new ArrayList<>();
listbodyServiceItemIds.add("f837f7ae-22c9-449d-a99c-4be24533e243");
body.withServiceItemIds(listbodyServiceItemIds);
body.withSetId("688faf62-20fc-4ca6-b9f9-6fbc518df5ae");
request.withBody(body);
try {
    BatchDeleteServiceItemsResponse response = client.batchDeleteServiceItems(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

删除项目id为9d80d070b6d44942af73c9c3d38e0429的服务组id为688faf62-20fc-4ca6-b9f9-6fbc518df5ae下的服务组成员f837f7ae-22c9-449d-a99c-4be24533e243

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CfwClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CfwRegion.valueOf("<YOUR REGION>")) \
        .build()

    try:
        request = BatchDeleteServiceItemsRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        listServiceItemIdsbody = [
            "f837f7ae-22c9-449d-a99c-4be24533e243"
        ]
        request.body = DeleteServiceItemDto(
            service_item_ids=listServiceItemIdsbody,
            set_id="688faf62-20fc-4ca6-b9f9-6fbc518df5ae"
        )
```



```
)
response = client.batch_delete_service_items(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

删除项目id为9d80d070b6d44942af73c9c3d38e0429的服务组id为688faf62-20fc-4ca6-b9f9-6fbc518df5ae下的服务组成员f837f7ae-22c9-449d-a99c-4be24533e243

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteServiceItemsRequest{}
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    var listServiceItemIdsbody = []string{
        "f837f7ae-22c9-449d-a99c-4be24533e243",
    }
    request.Body = &model.DeleteServiceItemDto{
        ServiceItemIds: listServiceItemIdsbody,
        SetId: "688faf62-20fc-4ca6-b9f9-6fbc518df5ae",
    }
    response, err := client.BatchDeleteServiceItems(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	批量删除服务组成员返回值

错误码

请参见[错误码](#)。

4.6.6 修改服务组

功能介绍

更新服务组

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/service-sets/{set_id}

表 4-300 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id
set_id	是	String	服务组id

表 4-301 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id，用户支持企业项目后，由企业项目生成的id。

参数	是否必选	参数类型	描述
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

请求参数

表 4-302 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）

表 4-303 请求 Body 参数

参数	是否必选	参数类型	描述
name	否	String	服务组名称 最小长度：1 最大长度：255
description	否	String	服务组描述信息 最小长度：1 最大长度：255

响应参数

状态码：200

表 4-304 响应 Body 参数

参数	参数类型	描述
data	IdObject object	更新服务组返回数据

表 4-305 IdObject

参数	参数类型	描述
id	String	id值
name	String	名称

状态码： 400

表 4-306 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

更新项目id为9d80d070b6d44942af73c9c3d38e0429的服务组id为221cfdca-3abf-4c30-ab0d-516a03c70866的名称改为ceshi2，描述改为描述

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/service-sets/221cfdca-3abf-4c30-ab0d-516a03c70866
```

```
{  
  "name" : "ceshi2",  
  "description" : "描述"  
}
```

响应示例

状态码： 200

OK

```
{  
  "data" : {  
    "id" : "221cfdca-3abf-4c30-ab0d-516a03c70866"  
  }  
}
```

状态码： 400

Bad Request

```
{  
  "error_code" : "CFW.00200005",  
  "error_msg" : "操作内容不存在"  
}
```

SDK 代码示例

SDK代码示例如下。

Java

更新项目id为9d80d070b6d44942af73c9c3d38e0429的服务组id为221cfdca-3abf-4c30-ab0d-516a03c70866的名称改为ceshi2，描述改为描述

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class UpdateServiceSetSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateServiceSetRequest request = new UpdateServiceSetRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        UpdateServiceSetUsingPUTRequestBody body = new UpdateServiceSetUsingPUTRequestBody();
        body.withDescription("描述");
        body.withName("ceshi2");
        request.withBody(body);
        try {
            UpdateServiceSetResponse response = client.updateServiceSet(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

更新项目id为9d80d070b6d44942af73c9c3d38e0429的服务组id为221cfdca-3abf-4c30-ab0d-516a03c70866的名称改为ceshi2，描述改为描述

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateServiceSetRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        request.body = UpdateServiceSetUsingPUTRequestBody(
            description="描述",
            name="ceshi2"
        )
        response = client.update_service_set(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

更新项目id为9d80d070b6d44942af73c9c3d38e0429的服务组id为221cfdca-3abf-4c30-ab0d-516a03c70866的名称改为ceshi2，描述改为描述

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateServiceSetRequest{}
enterpriseProjectIdRequest:= "<enterprise_project_id>"
request.EnterpriseProjectId = &enterpriseProjectIdRequest
fwInstanceIdRequest:= "<fw_instance_id>"
request.FwInstanceId = &fwInstanceIdRequest
descriptionUpdateServiceSetUsingPutRequestBody:= "描述"
nameUpdateServiceSetUsingPutRequestBody:= "ceshi2"
request.Body = &model.UpdateServiceSetUsingPutRequestBody{
    Description: &descriptionUpdateServiceSetUsingPutRequestBody,
    Name: &nameUpdateServiceSetUsingPutRequestBody,
}
response, err := client.UpdateServiceSet(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	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.6.7 获取服务组列表

功能介绍

获取服务组列表

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/service-sets

表 4-307 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-308 Query 参数

参数	是否必选	参数类型	描述
object_id	是	String	防护对象id, 是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id, 可通过调用查询防火墙实例接口获得, 注意type为0的为互联网边界防护对象id, type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
key_word	否	String	关键字
limit	是	Integer	每页查询个数, 范围为1-1024 最小值: 1 最大值: 1024
offset	是	Integer	偏移量: 指定返回记录的开始位置, 必须为数字, 取值范围为大于或等于0, 默认0 最小值: 0
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

参数	是否必选	参数类型	描述
query_service_set_type	否	Integer	查询服务组类型，0表示自定义服务组，1表示预定义服务组

请求参数

表 4-309 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	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）

响应参数

状态码： 200

表 4-310 响应 Body 参数

参数	参数类型	描述
data	ServiceSetRecords object	查询服务组列表返回值

表 4-311 ServiceSetRecords

参数	参数类型	描述
offset	Integer	偏移量：指定返回记录的开始位置，必须为数字，取值范围为大于或等于0，默认0
limit	Integer	每页显示个数，范围为1-1024
total	Integer	查询总数
records	Array of ServiceSet objects	服务组列表

表 4-312 ServiceSet

参数	参数类型	描述
set_id	String	服务组id
name	String	名称
description	String	描述
service_set_type	Integer	服务组类型，0表示自定义服务组，1表示常用WEB服务，2表示常用远程登录和PING，3表示常用数据库
ref_count	Integer	引用次数
status	String	状态
project_id	String	项目id
protocols	Array of integers	协议类型列表

状态码： 400

表 4-313 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

查询项目id为2349ba469daf4b7daf268bb0261d18b0的项目的防护对象id为a37bb4eb-c49e-4e88-bf77-944a75b0ce8a的第一页服务组列表信息

```
https://{Endpoint}/v1/2349ba469daf4b7daf268bb0261d18b0/service-sets?object_id=a37bb4eb-c49e-4e88-bf77-944a75b0ce8a&limit=10&offset=0
```

响应示例

状态码： 200

查询服务组信息响应

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

```
"limit" : 50,
"offset" : 0,
"records" : [ {
  "name" : "test",
  "project_id" : "2349ba469daf4b7daf268bb0261d18b0",
  "protocols" : [ 6 ],
  "ref_count" : 2,
  "service_set_type" : 0,
  "set_id" : "6f475bad-5d33-45d1-98f8-c79f2f308d5a"
}],
"total" : 1
}
}
```

状态码： 400

Bad Request

```
{
  "error_code" : "CFW.0020016",
  "error_msg" : "实例状态错误"
}
```

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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class ListServiceSetsSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListServiceSetsRequest request = new ListServiceSetsRequest();
        request.withObjectId("<object_id>");
        request.withKeyWord("<key_word>");
        request.withLimit(<limit>);
        request.withOffset(<offset>);
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        try {
```

```
ListServiceSetsResponse response = client.listServiceSets(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.getStatusCode());
    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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListServiceSetsRequest()
        request.object_id = "<object_id>"
        request.key_word = "<key_word>"
        request.limit = <limit>
        request.offset = <offset>
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        response = client.list_service_sets(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListServiceSetsRequest{}
    request.ObjectId = "<object_id>"
    keywordRequest := "<key_word>"
    request.KeyWord = &keywordRequest
    request.Limit = int32(<limit>)
    request.Offset = int32(<offset>)
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    response, err := client.ListServiceSets(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	查询服务组信息响应
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.6.8 查询服务组详情

功能介绍

查询服务组细节

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/service-sets/{set_id}

表 4-314 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id
set_id	是	String	服务组id

表 4-315 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。
query_service_set_type	否	Integer	查询服务组类型, 0表示自定义服务组, 1表示预定义服务组

请求参数

表 4-316 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）

响应参数

状态码：200

表 4-317 响应 Body 参数

参数	参数类型	描述
data	ServiceSetDetailResponseDto object	服务组详情数据

表 4-318 ServiceSetDetailResponseDto

参数	参数类型	描述
id	String	服务组id
name	String	服务组名称 最小长度：1 最大长度：255
description	String	服务组描述信息 最小长度：1 最大长度：255
service_set_type	Integer	服务组类型，0表示自定义服务组，1表示常用WEB服务，2表示常用远程登录和PING，3表示常用数据库

状态码：400

表 4-319 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度：8 最大长度：36
error_msg	String	错误描述 最小长度：2 最大长度：512

请求示例

查询项目id为9d80d070b6d44942af73c9c3d38e0429，服务组id为221cfdca-3abf-4c30-ab0d-516a03c70866的服务组的细节

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/service-sets/221cfdca-3abf-4c30-ab0d-516a03c70866
```

响应示例

状态码： 200

查询服务组成员详情响应

```
{
  "data": {
    "service_set_type": 0,
    "id": "221cfdca-3abf-4c30-ab0d-516a03c70866",
    "name": "ceshi2"
  }
}
```

状态码： 400

Bad Request

```
{
  "error_code": "CFW.00200005",
  "error_msg": "操作内容不存在"
}
```

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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;
```



```
public class ListServiceSetDetailSolution {
    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListServiceSetDetailRequest request = new ListServiceSetDetailRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        try {
            ListServiceSetDetailResponse response = client.listServiceSetDetail(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListServiceSetDetailRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
```

```
request.fw_instance_id = "<fw_instance_id>"
response = client.list_service_set_detail(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListServiceSetDetailRequest{}
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    response, err := client.ListServiceSetDetail(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	查询服务组成员详情响应

状态码	描述
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.6.9 查询服务成员列表

功能介绍

查询服务组成员列表

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/service-items

表 4-320 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-321 Query 参数

参数	是否必选	参数类型	描述
set_id	是	String	服务组id
key_word	否	String	查询字段
limit	是	Integer	每页显示个数，范围为1-1024
offset	是	Integer	偏移量：指定返回记录的开始位置，必须为数字，取值范围为大于或等于0，默认0

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考API Explorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。
query_service_set_type	否	Integer	查询服务组类型, 0表示自定义服务组, 1表示预定义服务组

请求参数

表 4-322 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）

响应参数

状态码：200

表 4-323 响应 Body 参数

参数	参数类型	描述
data	data object	服务组成员列表

表 4-324 data

参数	参数类型	描述
offset	Integer	偏移量：指定返回记录的开始位置，必须为数字，取值范围为大于或等于0，默认0
limit	Integer	每页显示个数，范围为1-1024

参数	参数类型	描述
total	Integer	记录的总数
set_id	String	服务组id
records	Array of records objects	记录

表 4-325 records

参数	参数类型	描述
item_id	String	服务成员id
protocol	Integer	协议类型:TCP为6,UDP为17,ICMP为1,ICMPV6为58,ANY为-1,手动类型不为空,自动类型为空
source_port	String	源端口
dest_port	String	目的端口
name	String	服务成员名称
description	String	服务成员描述

状态码: 400

表 4-326 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度: 8 最大长度: 36
error_msg	String	错误描述 最小长度: 2 最大长度: 512

请求示例

查询项目id为9d80d070b6d44942af73c9c3d38e0429, 服务组id为7cdebed3-af07-494e-a3c2-b88bb8d58b57的服务组成员列表

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/service-items?set_id=7cdebed3-af07-494e-a3c2-b88bb8d58b57&limit=10&offset=0
```

响应示例

状态码： 200

服务组成员列表返回值

```
{
  "data" : {
    "limit" : 10,
    "offset" : 0,
    "records" : [ {
      "dest_port" : "0",
      "item_id" : "805b711d-c558-41e3-aab1-a4b8c3f1f90b",
      "description" : "",
      "protocol" : 1,
      "source_port" : "0"
    } ],
    "set_id" : "7cdebed3-af07-494e-a3c2-b88bb8d58b57",
    "total" : 1
  }
}
```

状态码： 400

Bad Request

```
{
  "error_code" : "CFW.00200005",
  "error_msg" : "操作内容不存在"
}
```

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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class ListServiceItemsSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
    }
}
```

```
        .build();
        ListServiceItemsRequest request = new ListServiceItemsRequest();
        request.withSetId("<set_id>");
        request.withKeyword("<key_word>");
        request.withLimit(<limit>);
        request.withOffset(<offset>);
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        try {
            ListServiceItemsResponse response = client.listServiceItems(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.getStatusCode());
            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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListServiceItemsRequest()
        request.set_id = "<set_id>"
        request.key_word = "<key_word>"
        request.limit = <limit>
        request.offset = <offset>
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        response = client.list_service_items(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListServiceItemsRequest{}
    request.SetId = "<set_id>"
    keyWordRequest := "<key_word>"
    request.KeyWord = &keyWordRequest
    request.Limit = int32(<limit>)
    request.Offset = int32(<offset>)
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    response, err := client.ListServiceItems(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	服务组成员列表返回值
400	Bad Request
401	Unauthorized

状态码	描述
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.7 域名解析及域名组管理

4.7.1 添加域名组

功能介绍

添加域名组

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/domain-set

表 4-327 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-328 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。

参数	是否必选	参数类型	描述
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

请求参数

表 4-329 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）

表 4-330 请求 Body 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。
object_id	是	String	防护对象id, 是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id, 可通过调用查询防火墙实例接口获得, 注意type为0的为互联网边界防护对象id, type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
name	是	String	域名组名称

参数	是否必选	参数类型	描述
description	否	String	描述
domain_names	否	Array of DomainSetInfoDto objects	域名信息列表
domain_set_type	是	Integer	域名组类型，0表示应用域名组，1表示网络域名组

表 4-331 DomainSetInfoDto

参数	是否必选	参数类型	描述
domain_name	否	String	域名
description	否	String	描述

响应参数

状态码：200

表 4-332 响应 Body 参数

参数	参数类型	描述
data	DomainSetResponseData object	添加域名组返回data

表 4-333 DomainSetResponseData

参数	参数类型	描述
id	String	域名组id
name	String	域名组名称

请求示例

向项目id为9d80d070b6d44942af73c9c3d38e0429，防火墙id为546af3f8-88e9-47f2-a205-2346d7090925添加域名组，域名组名称为test，域名组内域名为www.aaa.com，防护对象id为fde07429-2e02-45c0-a85f-4f1cacea24d2，域名组类型为应用型。

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/domain-set?fw_instance_id=546af3f8-88e9-47f2-a205-2346d7090925&enterprise_project_id=default
```

```
{
  "name" : "test",
  "domain_set_type" : 0,
  "description" : "",
  "domain_names" : [ {
    "domain_name" : "www.aaa.com",
    "description" : ""
  } ],
  "fw_instance_id" : "546af3f8-88e9-47f2-a205-2346d7090925",
  "object_id" : "fde07429-2e02-45c0-a85f-4f1cacea24d2"
}
```

响应示例

状态码： 200

添加域名组返回值

```
{
  "data" : {
    "id" : "e43db369-a863-45ed-8850-58d6b571b1ab",
    "name" : "test"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

向项目id为9d80d070b6d44942af73c9c3d38e0429，防火墙id为546af3f8-88e9-47f2-a205-2346d7090925添加域名组，域名组名称为test，域名组内域名为www.aaa.com，防护对象id为fde07429-2e02-45c0-a85f-4f1cacea24d2，域名组类型为应用型。

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

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

public class AddDomainSetSolution {

    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);
```

```
CfwClient client = CfwClient.newBuilder()
    .withCredential(auth)
    .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
    .build();
AddDomainSetRequest request = new AddDomainSetRequest();
request.withEnterpriseProjectId("<enterprise_project_id>");
request.withFwInstanceId("<fw_instance_id>");
AddDomainSetInfoDto body = new AddDomainSetInfoDto();
List<DomainSetInfoDto> listbodyDomainNames = new ArrayList<>();
listbodyDomainNames.add(
    new DomainSetInfoDto()
        .withDomainName("www.aaa.com")
        .withDescription("")
);
body.withDomainSetType(0);
body.withDomainNames(listbodyDomainNames);
body.withDescription("");
body.withName("test");
body.withObjectId("fde07429-2e02-45c0-a85f-4f1cacea24d2");
body.withFwInstanceId("546af3f8-88e9-47f2-a205-2346d7090925");
request.withBody(body);
try {
    AddDomainSetResponse response = client.addDomainSet(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

向项目id为9d80d070b6d44942af73c9c3d38e0429，防火墙id为546af3f8-88e9-47f2-a205-2346d7090925添加域名组，域名组名称为test，域名组内域名为www.aaa.com，防护对象id为fde07429-2e02-45c0-a85f-4f1cacea24d2，域名组类型为应用型。

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
```

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

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

```
try:
    request = AddDomainSetRequest()
    request.enterprise_project_id = "<enterprise_project_id>"
    request.fw_instance_id = "<fw_instance_id>"
    listDomainNamesbody = [
        DomainSetInfoDto(
            domain_name="www.aaa.com",
            description=""
        )
    ]
    request.body = AddDomainSetInfoDto(
        domain_set_type=0,
        domain_names=listDomainNamesbody,
        description="",
        name="test",
        object_id="fde07429-2e02-45c0-a85f-4f1cacea24d2",
        fw_instance_id="546af3f8-88e9-47f2-a205-2346d7090925"
    )
    response = client.add_domain_set(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

向项目id为9d80d070b6d44942af73c9c3d38e0429，防火墙id为546af3f8-88e9-47f2-a205-2346d7090925添加域名组，域名组名称为test，域名组内域名为www.aaa.com，防护对象id为fde07429-2e02-45c0-a85f-4f1cacea24d2，域名组类型为应用型。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddDomainSetRequest{
        enterpriseProjectIdRequest:= "<enterprise_project_id>"
        request.EnterpriseProjectId = &enterpriseProjectIdRequest
        fwInstanceidRequest:= "<fw_instance_id>"
        request.FwInstanceid = &fwInstanceidRequest
    }
```

```
domainNameDomainNames:= "www.aaa.com"
descriptionDomainNames:= ""
var listDomainNamesbody = []model.DomainSetInfoDto{
    {
        DomainName: &domainNameDomainNames,
        Description: &descriptionDomainNames,
    },
}
descriptionAddDomainSetInfoDto:= ""
request.Body = &model.AddDomainSetInfoDto{
    DomainSetType: int32(0),
    DomainNames: &listDomainNamesbody,
    Description: &descriptionAddDomainSetInfoDto,
    Name: "test",
    Objectid: "fde07429-2e02-45c0-a85f-4f1cacea24d2",
    FwInstanceid: "546af3f8-88e9-47f2-a205-2346d7090925",
}
response, err := client.AddDomainSet(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	添加域名组返回值

错误码

请参见[错误码](#)。

4.7.2 删除域名组

功能介绍

删除域名组

调用方法

请参见[如何调用API](#)。

URI

DELETE /v1/{project_id}/domain-set/{set_id}

表 4-334 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id
set_id	是	String	域名组id

表 4-335 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	是	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考 API Explorer 和帮助中心 FAQ。默认情况下, fw_instance_id 为空时, 返回账号下第一个墙的信息; fw_instance_id 非空时, 返回与 fw_instance_id 对应墙的信息。

请求参数

表 4-336 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）

响应参数

状态码：200

表 4-337 响应 Body 参数

参数	参数类型	描述
data	DomainSetResponseData object	删除域名组返回值data

表 4-338 DomainSetResponseData

参数	参数类型	描述
id	String	域名组id
name	String	域名组名称

请求示例

删除项目id为9d80d070b6d44942af73c9c3d38e0429，防火墙id为7a004e79-0b8b-4679-ab20-267f3946e8ba下的域名组，域名组id为89bce6a4-9b59-4d7a-b5f9-cac5ac16d88a

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/domain-set/89bce6a4-9b59-4d7a-b5f9-cac5ac16d88a?fw_instance_id=7a004e79-0b8b-4679-ab20-267f3946e8ba&enterprise_project_id=default
```

响应示例

状态码： 200

删除域名组返回值

```
{
  "data": {
    "id": "89bce6a4-9b59-4d7a-b5f9-cac5ac16d88a",
    "name": "test"
  }
}
```

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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class DeleteDomainSetSolution {

    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);

    CfwClient client = CfwClient.newBuilder()
        .withCredential(auth)
        .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
        .build();
    DeleteDomainSetRequest request = new DeleteDomainSetRequest();
    request.withEnterpriseProjectId("<enterprise_project_id>");
    request.withFwInstanceId("<fw_instance_id>");
    try {
        DeleteDomainSetResponse response = client.deleteDomainSet(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.getStatusCode());
        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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = DeleteDomainSetRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        response = client.delete_domain_set(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"
cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteDomainSetRequest{}
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    request.FwInstanceId = "<fw_instance_id>"
    response, err := client.DeleteDomainSet(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	删除域名组返回值

错误码

请参见[错误码](#)。

4.7.3 更新域名组

功能介绍

更新域名组

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/domain-set/{set_id}

表 4-339 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id
set_id	是	String	防护对象id，是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id，可通过调用查询防火墙实例接口获得，注意type为0的为互联网边界防护对象id，type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。

表 4-340 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id，用户支持企业项目后，由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id，创建云防火墙后用于标志防火墙由系统自动生成的标志id，可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下，fw_instance_id为空时，返回账号下第一个墙的信息；fw_instance_id非空时，返回与fw_instance_id对应墙的信息。

请求参数

表 4-341 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）

表 4-342 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	域名组名称UUID
description	否	String	描述
set_id	否	String	域名组id
domain_set_type	否	Integer	域名组类型，0表示应用域名组，1表示网络域名组

响应参数

状态码：200

表 4-343 响应 Body 参数

参数	参数类型	描述
data	DomainSetResponseData object	更新的域名组返回值data

表 4-344 DomainSetResponseData

参数	参数类型	描述
id	String	域名组id
name	String	域名组名称

请求示例

更新项目id为9d80d070b6d44942af73c9c3d38e0429下的防火墙id为7a004e79-0b8b-4679-ab20-267f3946e8ba的域名组id为94da194d-24b2-4f60-919e-cf0bc76c75b3，修改名称为test。

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/domain-set/94da194d-24b2-4f60-919e-cf0bc76c75b3?fw_instance_id=7a004e79-0b8b-4679-ab20-267f3946e8ba&enterprise_project_id=default

{
  "name" : "test",
  "description" : ""
}
```

响应示例

状态码： 200

更新域名组返回值

```
{
  "data" : {
    "id" : "94da194d-24b2-4f60-919e-cf0bc76c75b3",
    "name" : "test"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

更新项目id为9d80d070b6d44942af73c9c3d38e0429下的防火墙id为7a004e79-0b8b-4679-ab20-267f3946e8ba的域名组id为94da194d-24b2-4f60-919e-cf0bc76c75b3，修改名称为test。

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class UpdateDomainSetSolution {

    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);
```

```
CfwClient client = CfwClient.newBuilder()
    .withCredential(auth)
    .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateDomainSetRequest request = new UpdateDomainSetRequest();
request.withEnterpriseProjectId("<enterprise_project_id>");
request.withFwInstanceId("<fw_instance_id>");
UpdateDomainSetInfoDto body = new UpdateDomainSetInfoDto();
body.withDescription("");
body.withName("test");
request.withBody(body);
try {
    UpdateDomainSetResponse response = client.updateDomainSet(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

更新项目id为9d80d070b6d44942af73c9c3d38e0429下的防火墙id为7a004e79-0b8b-4679-ab20-267f3946e8ba的域名组id为94da194d-24b2-4f60-919e-cf0bc76c75b3，修改名称为test。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateDomainSetRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        request.body = UpdateDomainSetInfoDto(
            description="",
            name="test"
        )
        response = client.update_domain_set(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

更新项目id为9d80d070b6d44942af73c9c3d38e0429下的防火墙id为7a004e79-0b8b-4679-ab20-267f3946e8ba的域名组id为94da194d-24b2-4f60-919e-cf0bc76c75b3，修改名称为test。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateDomainSetRequest{
        enterpriseProjectIdRequest:= "<enterprise_project_id>"
        request.EnterpriseProjectId = &enterpriseProjectIdRequest
        fwInstanceIdRequest:= "<fw_instance_id>"
        request.FwInstanceId = &fwInstanceIdRequest
        descriptionUpdateDomainSetInfoDto:= ""
        request.Body = &model.UpdateDomainSetInfoDto{
            Description: &descriptionUpdateDomainSetInfoDto,
            Name: "test",
        }
    }
    response, err := client.UpdateDomainSet(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	更新域名组返回值

错误码

请参见[错误码](#)。

4.7.4 更新 dns 服务器列表

功能介绍

更新dns服务器列表

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/dns/servers

表 4-345 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-346 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考 API Explorer 和帮助中心 FAQ。默认情况下, fw_instance_id 为空时, 返回账号下第一个墙的信息; fw_instance_id 非空时, 返回与 fw_instance_id 对应墙的信息。若 object_id 非空, 默认返回 object_id 对应墙的信息; 填写时 object_id 需要属于 fw_instance_id 对应的墙。

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。

请求参数

表 4-347 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）

表 4-348 请求 Body 参数

参数	是否必选	参数类型	描述
dns_server	否	Array of dns_server objects	DNS服务器
health_check_domain_name	否	String	健康检查域名

表 4-349 dns_server

参数	是否必选	参数类型	描述
server_ip	否	String	DNS服务器IP
is_customized	否	Integer	是否是用户自定义的dns服务器, 0否 1是
is_applied	否	Integer	是否应用, 0否 1是

响应参数

状态码： 200

表 4-350 响应 Body 参数

参数	参数类型	描述
data	Array of strings	域名服务器列表

状态码： 400

表 4-351 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

更新项目id为2349ba469daf4b7daf268bb0261d18b0的dns解析器的设置，服务器ip为8.8.8.8，默认服务器，更新为使用，服务器ip为192.168.0.2，非默认服务器，更新为非使用

```
https://{Endpoint}/v1/2349ba469daf4b7daf268bb0261d18b0/dns/servers
```

```
{
  "dns_server": [ {
    "server_ip": "8.8.8.8",
    "is_customized": 0,
    "is_applied": 1
  }, {
    "server_ip": "192.168.0.2",
    "is_customized": 1,
    "is_applied": 0
  } ]
}
```

响应示例

状态码： 200

更新Dns服务器响应

```
{
  "data": [ "100.95.150.83", "114.114.114.114", "223.5.5.5", "223.6.6.6", "119.29.29.29", "8.8.8.8",
    "100.79.1.250", "100.79.1.240" ]
}
```

状态码： 400

Bad Request

```
{
  "error_code" : "CFW.01000001",
  "error_msg" : "DNS服务器IP重复"
}
```

SDK 代码示例

SDK代码示例如下。

Java

更新项目id为2349ba469daf4b7daf268bb0261d18b0的dns解析器的设置，服务器ip为8.8.8.8，默认服务器，更新为使用，服务器ip为192.168.0.2，非默认服务器，更新为非使用

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

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

public class UpdateDnsServersSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();

        UpdateDnsServersRequest request = new UpdateDnsServersRequest();
        request.withFwInstanceId("<fw_instance_id>");
        request.withEnterpriseProjectId("<enterprise_project_id>");
        UpdateDnsServersRequestBody body = new UpdateDnsServersRequestBody();
        List<UpdateDnsServersRequestBodyDnsServer> listbodyDnsServer = new ArrayList<>();
        listbodyDnsServer.add(
            new UpdateDnsServersRequestBodyDnsServer()
                .withServerIp("8.8.8.8")
                .withIsCustomized(0)
                .withIsApplied(1)
        );
        listbodyDnsServer.add(
            new UpdateDnsServersRequestBodyDnsServer()
                .withServerIp("192.168.0.2")
                .withIsCustomized(1)
                .withIsApplied(0)
        );
        body.withDnsServer(listbodyDnsServer);
        request.withBody(body);
    }
}
```

```
try {
    UpdateDnsServersResponse response = client.updateDnsServers(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

更新项目id为2349ba469daf4b7daf268bb0261d18b0的dns解析器的设置，服务器ip为8.8.8.8，默认服务器，更新为使用，服务器ip为192.168.0.2，非默认服务器，更新为非使用

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateDnsServersRequest()
        request.fw_instance_id = "<fw_instance_id>"
        request.enterprise_project_id = "<enterprise_project_id>"
        listDnsServerbody = [
            UpdateDnsServersRequestBodyDnsServer(
                server_ip="8.8.8.8",
                is_customized=0,
                is_applied=1
            ),
            UpdateDnsServersRequestBodyDnsServer(
                server_ip="192.168.0.2",
                is_customized=1,
                is_applied=0
            )
        ]
        request.body = UpdateDnsServersRequestBody(
            dns_server=listDnsServerbody
        )
        response = client.update_dns_servers(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

更新项目id为2349ba469daf4b7daf268bb0261d18b0的dns解析器的设置，服务器ip为8.8.8.8，默认服务器，更新为使用，服务器ip为192.168.0.2，非默认服务器，更新为非使用

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateDnsServersRequest{}
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    serverIpDnsServer := "8.8.8.8"
    isCustomizedDnsServer := int32(0)
    isAppliedDnsServer := int32(1)
    serverIpDnsServer1 := "192.168.0.2"
    isCustomizedDnsServer1 := int32(1)
    isAppliedDnsServer1 := int32(0)
    var listDnsServerbody = []model.UpdateDnsServersRequestBodyDnsServer{
        {
            ServerIp: &serverIpDnsServer,
            IsCustomized: &isCustomizedDnsServer,
            IsApplied: &isAppliedDnsServer,
        },
        {
            ServerIp: &serverIpDnsServer1,
            IsCustomized: &isCustomizedDnsServer1,
            IsApplied: &isAppliedDnsServer1,
        },
    }
    request.Body = &model.UpdateDnsServersRequestBody{
        DnsServer: &listDnsServerbody,
    }
    response, err := client.UpdateDnsServers(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	更新Dns服务器响应
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.7.5 查询域名组列表

功能介绍

查询域名组列表

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/domain-sets

表 4-352 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id 租户项目id

表 4-353 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	是	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。
limit	是	Integer	每页显示个数, 范围为1-1024
offset	是	Integer	偏移量: 指定返回记录的开始位置, 必须为数字, 取值范围为大于或等于0, 默认0
object_id	是	String	防护对象id, 是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id, 可通过调用查询防火墙实例接口获得, 注意type为0的为互联网边界防护对象id, type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
key_word	否	String	关键字
domain_set_type	否	Integer	域名组类型, 0表示应用域名组, 1表示网络域名组
config_status	否	Integer	配置状态

请求参数

表 4-354 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)

响应参数

状态码： 200

表 4-355 响应 Body 参数

参数	参数类型	描述
data	ListDomainsetsResponseData object	查询域名组返回值列表data

表 4-356 ListDomainsetsResponseData

参数	参数类型	描述
limit	Integer	每页显示个数，范围为1-1024
offset	Integer	偏移量：指定返回记录的开始位置，必须为数字，取值范围为大于或等于0，默认0
total	Integer	总数
records	Array of DomainSetVo objects	域名组列表

表 4-357 DomainSetVo

参数	参数类型	描述
set_id	String	域名组id
name	String	域名组名称
description	String	描述
ref_count	Integer	引用次数
domain_set_type	Integer	域名组类型，0表示应用域名组，1表示网络域名组
config_status	Integer	配置状态
message	String	异常信息
rules	Array of UseRuleVO objects	使用规则id列表

表 4-358 UseRuleVO

参数	参数类型	描述
id	String	规则id
name	String	规则名称

请求示例

查询项目id为9d80d070b6d44942af73c9c3d38e0429下的防火墙 id为546af3f8-88e9-47f2-a205-2346d7090925下的域名组列表，防护对象id为ae42418e-f077-41a0-9d3b-5b2f5ad9102b

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/domain-sets?fw_instance_id=546af3f8-88e9-47f2-a205-2346d7090925&enterprise_project_id=default&limit=50&offset=0&object_id=ae42418e-f077-41a0-9d3b-5b2f5ad9102b
```

响应示例

状态码： 200

查询域名组返回值

```
{
  "data": {
    "limit": 50,
    "offset": 0,
    "records": [ {
      "config_status": 3,
      "description": "",
      "domain_set_type": 0,
      "name": "ccdd",
      "ref_count": 0,
      "set_id": "e43db369-a863-45ed-8850-58d6b571b1ab"
    } ],
    "total": 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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class ListDomainSetsSolution {
```

```
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);

    CfwClient client = CfwClient.newBuilder()
        .withCredential(auth)
        .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
        .build();
    ListDomainSetsRequest request = new ListDomainSetsRequest();
    request.withEnterpriseProjectId("<enterprise_project_id>");
    request.withFwInstanceId("<fw_instance_id>");
    request.withLimit(<limit>);
    request.withOffset(<offset>);
    request.withObjectId("<object_id>");
    request.withKeyword("<key_word>");
    request.withDomainSetType(<domain_set_type>);
    request.withConfigStatus(<config_status>);
    try {
        ListDomainSetsResponse response = client.listDomainSets(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.getStatusCode());
        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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
```

```
request = ListDomainSetsRequest()
request.enterprise_project_id = "<enterprise_project_id>"
request.fw_instance_id = "<fw_instance_id>"
request.limit = <limit>
request.offset = <offset>
request.object_id = "<object_id>"
request.key_word = "<key_word>"
request.domain_set_type = <domain_set_type>
request.config_status = <config_status>
response = client.list_domain_sets(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDomainSetsRequest{
        enterpriseProjectIdRequest:= "<enterprise_project_id>"
        request.EnterpriseProjectId = &enterpriseProjectIdRequest
        request.FwInstanceId = "<fw_instance_id>"
        request.Limit = int32(<limit>)
        request.Offset = int32(<offset>)
        request.ObjectId = "<object_id>"
        keyWordRequest:= "<key_word>"
        request.KeyWord = &keyWordRequest
        domainSetTypeRequest:= int32(<domain_set_type>)
        request.DomainSetType = &domainSetTypeRequest
        configStatusRequest:= int32(<config_status>)
        request.ConfigStatus = &configStatusRequest
        response, err := client.ListDomainSets(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	查询域名组返回值

错误码

请参见[错误码](#)。

4.7.6 查询 dns 服务器列表

功能介绍

查询dns服务器列表

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/dns/servers

表 4-359 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-360 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页显示个数，范围为1-1024
offset	否	Integer	偏移量：指定返回记录的开始位置，必须为数字，取值范围为大于或等于0，默认0

参数	是否必选	参数类型	描述
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。若object_id非空, 默认返回object_id对应墙的信息; 填写时object_id需要属于fw_instance_id对应的墙。
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。

请求参数

表 4-361 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）

响应参数

状态码： 200

表 4-362 响应 Body 参数

参数	参数类型	描述
data	Array of DnsServersResponseDTO objects	dns服务器列表
total	Integer	dns服务器总数

表 4-363 DnsServersResponseDTO

参数	参数类型	描述
id	Integer	id
is_applied	Integer	是否应用, 0否 1是
is_customized	Integer	是否是用户自定义的dns服务器, 0否 1是
server_ip	String	DNS服务器IP
health_check_domain_name	String	健康检查域名

请求示例

获取项目id为2349ba469daf4b7daf268bb0261d18b0的dns服务器列表

```
https://{Endpoint}/v1/2349ba469daf4b7daf268bb0261d18b0/dns/servers
```

响应示例

状态码： 200

获取dns服务器响应

```
{
  "data": [ {
    "health_check_domain_name": "sslstatic.xiaoyusan.com",
    "id": 20165,
    "is_applied": 0,
    "is_customized": 1,
    "server_ip": "0.0.0.0"
  }, {
    "health_check_domain_name": "sslstatic.xiaoyusan.com",
    "id": 14190,
    "is_applied": 1,
    "is_customized": 0,
    "server_ip": "100.79.1.240"
  } ],
  "total": 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.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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;
```

```
public class ListDnsServersSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListDnsServersRequest request = new ListDnsServersRequest();
        request.withLimit(<limit>);
        request.withOffset(<offset>);
        request.withFwInstanceId("<fw_instance_id>");
        request.withEnterpriseProjectId("<enterprise_project_id>");
        try {
            ListDnsServersResponse response = client.listDnsServers(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListDnsServersRequest()
```



```
request.limit = <limit>
request.offset = <offset>
request.fw_instance_id = "<fw_instance_id>"
request.enterprise_project_id = "<enterprise_project_id>"
response = client.list_dns_servers(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDnsServersRequest{}
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    offsetRequest := int32(<offset>)
    request.Offset = &offsetRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    response, err := client.ListDnsServers(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	获取dns服务器响应
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.7.7 查询域名解析 ip 地址

功能介绍

测试域名有效性

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/domain/parse/{domain_name}

表 4-364 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id
domain_name	是	String	域名

表 4-365 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。

参数	是否必选	参数类型	描述
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。
address_type	否	String	地址类型, 0 ipv4,1 ipv6,2 domain

请求参数

表 4-366 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）

响应参数

状态码： 200

表 4-367 响应 Body 参数

参数	参数类型	描述
data	Array of strings	域名id列表

状态码： 400

表 4-368 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度：8 最大长度：36
error_msg	String	错误描述 最小长度：2 最大长度：512

请求示例

测试项目id为5c69cf330cda42369cbd726ee1bc5e76的项目下的ceshi.com域名是否有效。

```
https://{Endpoint}/v1/5c69cf330cda42369cbd726ee1bc5e76/domain/parse/ceshi.com
```

响应示例

状态码： 200

查询域名有效性返回值

```
{  
  "data" : [ "192.168.88.85", "192.168.88.50", "192.168.88.22", "192.168.88.87", "192.168.88.86",  
    "192.168.5.1", "192.168.88.88", "192.168.88.90", "192.168.88.83", "192.168.88.84" ]  
}
```

状态码： 400

Bad Request

```
{  
  "error_code" : "CFW.00109004",  
  "error_msg" : "HTTP请求错误"  
}
```

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.cfw.v1.region.CfwRegion;  
import com.huaweicloud.sdk.cfw.v1.*;  
import com.huaweicloud.sdk.cfw.v1.model.*;  
  
public class ListDomainParseDetailSolution {
```

```
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);

    CfwClient client = CfwClient.newBuilder()
        .withCredential(auth)
        .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
        .build();
    ListDomainParseDetailRequest request = new ListDomainParseDetailRequest();
    request.withEnterpriseProjectId("<enterprise_project_id>");
    request.withFwInstanceId("<fw_instance_id>");
    request.withAddressType("<address_type>");
    try {
        ListDomainParseDetailResponse response = client.listDomainParseDetail(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListDomainParseDetailRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        request.address_type = "<address_type>"
        response = client.list_domain_parse_detail(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDomainParseDetailRequest{}
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    addressTypeRequest := "<address_type>"
    request.AddressType = &addressTypeRequest
    response, err := client.ListDomainParseDetail(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	查询域名有效性返回值

状态码	描述
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.7.8 获取域名组下域名列表

功能介绍

获取域名组下域名列表

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/domain-set/domains/{domain_set_id}

表 4-369 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目ID
domain_set_id	是	String	域名组id

表 4-370 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id，用户支持企业项目后，由企业项目生成的id。

参数	是否必选	参数类型	描述
fw_instance_id	是	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。
limit	是	Integer	每页显示个数, 范围为1-1024
offset	是	Integer	偏移量: 指定返回记录的开始位置, 必须为数字, 取值范围为大于或等于0, 默认0
domain_name	否	String	域名名称
description	否	String	描述
set_id	否	String	域名组id
object_id	否	String	防护对象id,是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id,可通过调用查询防火墙实例接口获得,注意type为0的为互联网边界防护对象id,type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。

请求参数

表 4-371 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	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）

响应参数

状态码： 200

表 4-372 响应 Body 参数

参数	参数类型	描述
data	ListDomainResponseData object	查询域名返回值data

表 4-373 ListDomainResponseData

参数	参数类型	描述
limit	Integer	每页显示个数，范围为1-1024
offset	Integer	偏移量：指定返回记录的开始位置，必须为数字，取值范围为大于或等于0，默认0
project_id	String	租户项目id
records	Array of DomainInfo objects	域名信息列表
set_id	String	域名组id
total	Integer	总数

表 4-374 DomainInfo

参数	参数类型	描述
domain_address_id	String	域名地址id
domain_name	String	域名
description	String	描述
dns_ips	Array of strings	域名服务器列表

请求示例

查询项目id为14181c1245cf4fd786824efe1e2b9388下的域名列表，域名组id为78719348-6d79-477e-acec-676a29842ab2，防火墙id为546af3f8-88e9-47f2-a205-2346d7090925。

```
https://{Endpoint}/v1/14181c1245cf4fd786824efe1e2b9388/domain-set/domains/78719348-6d79-477e-acec-676a29842ab2?fw_instance_id=546af3f8-88e9-47f2-a205-2346d7090925&enterprise_project_id=default&limit=200&offset=0
```

响应示例

状态码： 200

查询域名返回值

```
{
  "data": {
    "limit": 200,
    "offset": 0,
    "project_id": "14181c1245cf4fd786824efe1e2b9388",
    "records": [ {
      "description": "",
      "domain_address_id": "6718279e-9761-4623-a48d-b16957b19e1b",
      "domain_name": "www.test.com"
    } ],
    "set_id": "78719348-6d79-477e-acec-676a29842ab2",
    "total": 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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class ListDomainsSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListDomainsRequest request = new ListDomainsRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        request.withLimit(<limit>);
        request.withOffset(<offset>);
        request.withDomainName("<domain_name>");
        request.withDescription("<description>");
        request.withSetId("<set_id>");
        request.withObjectId("<object_id>");
    }
}
```

```
try {
    ListDomainsResponse response = client.listDomains(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListDomainsRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        request.limit = <limit>
        request.offset = <offset>
        request.domain_name = "<domain_name>"
        request.description = "<description>"
        request.set_id = "<set_id>"
        request.object_id = "<object_id>"
        response = client.list_domains(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
```

```
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDomainsRequest{}
    enterpriseProjectIdRequest:= "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    request.FwInstanceId = "<fw_instance_id>"
    request.Limit = int32(<limit>)
    request.Offset = int32(<offset>)
    domainNameRequest:= "<domain_name>"
    request.DomainName = &domainNameRequest
    descriptionRequest:= "<description>"
    request.Description = &descriptionRequest
    setIdRequest:= "<set_id>"
    request.SetId = &setIdRequest
    objectIdRequest:= "<object_id>"
    request.ObjectId = &objectIdRequest
    response, err := client.ListDomains(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	查询域名返回值

错误码

请参见[错误码](#)。

4.7.9 添加域名列表

功能介绍

添加域名列表

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/domain-set/domains/{set_id}

表 4-375 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id
set_id	是	String	域名组id

表 4-376 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

请求参数

表 4-377 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）

表 4-378 请求 Body 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	防火墙实例id，创建云防火墙后用于标志防火墙由系统自动生成的标志id，可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下，fw_instance_id为空时，返回账号下第一个墙的信息；fw_instance_id非空时，返回与fw_instance_id对应墙的信息。
object_id	是	String	防护对象id，是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id，可通过调用查询防火墙实例接口获得，注意type为0的为互联网边界防护对象id，type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
domain_names	是	Array of DomainSetInfoDto objects	域名列表

表 4-379 DomainSetInfoDto

参数	是否必选	参数类型	描述
domain_name	否	String	域名
description	否	String	描述

响应参数

状态码： 200

表 4-380 响应 Body 参数

参数	参数类型	描述
data	DomainSetResponseData object	域名组返回值数据

表 4-381 DomainSetResponseData

参数	参数类型	描述
id	String	域名组id
name	String	域名组名称

请求示例

给项目id为14181c1245cf4fd786824efe1e2b9388下的域名组添加域名，防火墙id为546af3f8-88e9-47f2-a205-2346d7090925，防护对象id为ae42418e-f077-41a0-9d3b-5b2f5ad9102b，域名组id为78719348-6d79-477e-acec-676a29842ab2，域名为www.bnm.com和www.vbc.com

```
https://{Endpoint}v1/14181c1245cf4fd786824efe1e2b9388/domain-set/domains/78719348-6d79-477e-acec-676a29842ab2?fw_instance_id=546af3f8-88e9-47f2-a205-2346d7090925&enterprise_project_id=default
```

```
{
  "domain_names": [ {
    "description": "",
    "domain_name": "www.bnm.com"
  }, {
    "description": "",
    "domain_name": "www.vbc.com"
  } ],
  "fw_instance_id": "546af3f8-88e9-47f2-a205-2346d7090925",
  "object_id": "ae42418e-f077-41a0-9d3b-5b2f5ad9102b"
}
```

响应示例

状态码： 200

添加域名返回值

```
{
  "data": {
    "id": "78719348-6d79-477e-acec-676a29842ab2",
    "name": "test26"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

给项目id为14181c1245cf4fd786824efe1e2b9388下的域名组添加域名，防火墙id为546af3f8-88e9-47f2-a205-2346d7090925，防护对象id为ae42418e-f077-41a0-9d3b-5b2f5ad9102b，域名组id为78719348-6d79-477e-acec-676a29842ab2，域名为www.bnm.com和www.vbc.com

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

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

public class AddDomainsSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        AddDomainsRequest request = new AddDomainsRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        AddDomainListDto body = new AddDomainListDto();
        List<DomainSetInfoDto> listbodyDomainNames = new ArrayList<>();
        listbodyDomainNames.add(
            new DomainSetInfoDto()
                .withDomainName("www.bnm.com")
                .withDescription("")
        );
        listbodyDomainNames.add(
            new DomainSetInfoDto()
                .withDomainName("www.vbc.com")
                .withDescription("")
        );
        body.withDomainNames(listbodyDomainNames);
        body.withObjectId("ae42418e-f077-41a0-9d3b-5b2f5ad9102b");
        body.withFwInstanceId("546af3f8-88e9-47f2-a205-2346d7090925");
        request.withBody(body);
        try {
            AddDomainsResponse response = client.addDomains(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.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

给项目id为14181c1245cf4fd786824efe1e2b9388下的域名组添加域名，防火墙id为546af3f8-88e9-47f2-a205-2346d7090925，防护对象id为ae42418e-f077-41a0-9d3b-5b2f5ad9102b，域名组id为78719348-6d79-477e-acec-676a29842ab2，域名为www.bnm.com和www.vbc.com

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = AddDomainsRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        listDomainNamesbody = [
            DomainSetInfoDto(
                domain_name="www.bnm.com",
                description=""
            ),
            DomainSetInfoDto(
                domain_name="www.vbc.com",
                description=""
            )
        ]
        request.body = AddDomainListDto(
            domain_names=listDomainNamesbody,
            object_id="ae42418e-f077-41a0-9d3b-5b2f5ad9102b",
            fw_instance_id="546af3f8-88e9-47f2-a205-2346d7090925"
        )
        response = client.add_domains(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

给项目id为14181c1245cf4fd786824efe1e2b9388下的域名组添加域名，防火墙id为546af3f8-88e9-47f2-a205-2346d7090925，防护对象id为ae42418e-f077-41a0-9d3b-5b2f5ad9102b，域名组id为78719348-6d79-477e-acec-676a29842ab2，域名为www.bnm.com和www.vbc.com

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddDomainsRequest{}
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    domainNameDomainNames := "www.bnm.com"
    descriptionDomainNames := ""
    domainNameDomainNames1 := "www.vbc.com"
    descriptionDomainNames1 := ""
    var listDomainNamesbody = []model.DomainSetInfoDto{
        {
            DomainName: &domainNameDomainNames,
            Description: &descriptionDomainNames,
        },
        {
            DomainName: &domainNameDomainNames1,
            Description: &descriptionDomainNames1,
        },
    }
    request.Body = &model.AddDomainListDto{
        DomainNames: listDomainNamesbody,
        Objectid: "ae42418e-f077-41a0-9d3b-5b2f5ad9102b",
        FwInstanceId: "546af3f8-88e9-47f2-a205-2346d7090925",
    }
    response, err := client.AddDomains(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	添加域名返回值

错误码

请参见[错误码](#)。

4.7.10 删除域名列表

功能介绍

删除域名列表

调用方法

请参见[如何调用API](#)。

URI

DELETE /v1/{project_id}/domain-set/domains/{set_id}

表 4-382 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id
set_id	是	String	域名组id

表 4-383 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id，用户支持企业项目后，由企业项目生成的id。

请求参数

表 4-384 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）

表 4-385 请求 Body 参数

参数	是否必选	参数类型	描述
object_id	是	String	防护对象id，是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id，可通过调用查询防火墙实例接口获得，注意type为0的为互联网边界防护对象id，type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
domain_addresses_ids	是	Array of strings	域名地址组

响应参数

状态码： 200

表 4-386 响应 Body 参数

参数	参数类型	描述
data	DomainSetResponseData object	删除域名列表返回值

表 4-387 DomainSetResponseData

参数	参数类型	描述
id	String	域名组id
name	String	域名组名称

请求示例

项目id为14181c1245cf4fd786824efe1e2b9388的项目，删除防火墙id为546af3f8-88e9-47f2-a205-2346d7090925的防火墙下的域名组域名，防护对象id为ae42418e-f077-41a0-9d3b-5b2f5ad9102b，域名组id为78719348-6d79-477e-acec-676a29842ab2，域名列表为"b9c23ad8-16d2-4f14-894f-29250c5d27e5"，"c36f9462-467b-4303-9734-f9abc38ddb95"

```
https://{Endpoint}/v1/14181c1245cf4fd786824efe1e2b9388/domain-set/domains/78719348-6d79-477e-acec-676a29842ab2?fw_instance_id=546af3f8-88e9-47f2-a205-2346d7090925&enterprise_project_id=default

{
  "domain_address_ids": [ "b9c23ad8-16d2-4f14-894f-29250c5d27e5", "c36f9462-467b-4303-9734-f9abc38ddb95" ],
  "object_id": "ae42418e-f077-41a0-9d3b-5b2f5ad9102b"
}
```

响应示例

状态码： 200

删除域名列表返回值

```
{
  "data": {
    "id": "78719348-6d79-477e-acec-676a29842ab2",
    "name": "test26"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

项目id为14181c1245cf4fd786824efe1e2b9388的项目，删除防火墙id为546af3f8-88e9-47f2-a205-2346d7090925的防火墙下的域名组域名，防护对象id为ae42418e-f077-41a0-9d3b-5b2f5ad9102b，域名组id为78719348-6d79-477e-acec-676a29842ab2，域名列表为"b9c23ad8-16d2-4f14-894f-29250c5d27e5"，"c36f9462-467b-4303-9734-f9abc38ddb95"

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

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

public class DeleteDomainsSolution {

    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);

CfwClient client = CfwClient.newBuilder()
    .withCredential(auth)
    .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
    .build();

DeleteDomainsRequest request = new DeleteDomainsRequest();
request.withEnterpriseProjectId("<enterprise_project_id>");
DeleteDomainDto body = new DeleteDomainDto();
List<String> listbodyDomainAddressIds = new ArrayList<>();
listbodyDomainAddressIds.add("b9c23ad8-16d2-4f14-894f-29250c5d27e5");
listbodyDomainAddressIds.add("c36f9462-467b-4303-9734-f9abc38ddb95");
body.withDomainAddressIds(listbodyDomainAddressIds);
body.withObjectId("ae42418e-f077-41a0-9d3b-5b2f5ad9102b");
request.withBody(body);
try {
    DeleteDomainsResponse response = client.deleteDomains(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

项目id为14181c1245cf4fd786824efe1e2b9388的项目，删除防火墙id为546af3f8-88e9-47f2-a205-2346d7090925的防火墙下的域名组域名，防护对象id为ae42418e-f077-41a0-9d3b-5b2f5ad9102b，域名组id为78719348-6d79-477e-acec-676a29842ab2，域名列表为"b9c23ad8-16d2-4f14-894f-29250c5d27e5", "c36f9462-467b-4303-9734-f9abc38ddb95"

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

```
try:
    request = DeleteDomainsRequest()
    request.enterprise_project_id = "<enterprise_project_id>"
    listDomainAddressIdsbody = [
        "b9c23ad8-16d2-4f14-894f-29250c5d27e5",
        "c36f9462-467b-4303-9734-f9abc38ddb95"
    ]
    request.body = DeleteDomainDto(
        domain_address_ids=listDomainAddressIdsbody,
        object_id="ae42418e-f077-41a0-9d3b-5b2f5ad9102b"
    )
    response = client.delete_domains(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

项目id为14181c1245cf4fd786824efe1e2b9388的项目，删除防火墙id为546af3f8-88e9-47f2-a205-2346d7090925的防火墙下的域名组域名，防护对象id为ae42418e-f077-41a0-9d3b-5b2f5ad9102b，域名组id为78719348-6d79-477e-acec-676a29842ab2，域名列表为"b9c23ad8-16d2-4f14-894f-29250c5d27e5"，"c36f9462-467b-4303-9734-f9abc38ddb95"

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteDomainsRequest{
        enterpriseProjectIdRequest: "<enterprise_project_id>"
        request.EnterpriseProjectId = &enterpriseProjectIdRequest
        var listDomainAddressIdsbody = []string{
            "b9c23ad8-16d2-4f14-894f-29250c5d27e5",
            "c36f9462-467b-4303-9734-f9abc38ddb95",
        }
    }
    request.Body = &model.DeleteDomainDto{
        DomainAddressIds: listDomainAddressIdsbody,
        ObjectID: "ae42418e-f077-41a0-9d3b-5b2f5ad9102b",
    }
```

```
}  
response, err := client.DeleteDomains(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	删除域名列表返回值

错误码

请参见[错误码](#)。

4.8 IPS 管理

4.8.1 查询 IPS 特性开关状态

功能介绍

查询IPS特性开关状态

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/ips/switch

表 4-388 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目ID

表 4-389 Query 参数

参数	是否必选	参数类型	描述
object_id	是	String	防护对象id，是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id，可通过调用查询防火墙实例接口获得，注意type为0的为互联网边界防护对象id，type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
enterprise_project_id	否	String	企业项目id，用户支持企业项目后，由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id，创建云防火墙后用于标志防火墙由系统自动生成的标志id，可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下，fw_instance_id为空时，返回账号下第一个墙的信息；fw_instance_id非空时，返回与fw_instance_id对应墙的信息。

请求参数

表 4-390 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）

响应参数

状态码： 200

表 4-391 响应 Body 参数

参数	参数类型	描述
data	IpsSwitchResponseDTO object	ips开关data

表 4-392 IpsSwitchResponseDTO

参数	参数类型	描述
id	String	ips开关id
basic_defense_status	Integer	基础防御状态
virtual_patches_status	Integer	虚拟补丁状态

请求示例

通过租户id14181c1245cf4fd786824efe1e2b9388查询当前租户的补丁开启状态，用来加载入侵防御页面的虚拟补丁的开关状态。

```
https://{Endpoint}/v1/14181c1245cf4fd786824efe1e2b9388/ips/switch?fw_instance_id=546af3f8-88e9-47f2-a205-2346d7090925&enterprise_project_id=default&object_id=cfebd347-b655-4b84-b938-3c54317599b2
```

响应示例

状态码： 200

OK

```
{
  "data" : {
    "basic_defense_status" : 1,
    "id" : "cefe80aa-83e4-4308-99aa-f9b6c816de00",
    "virtual_patches_status" : 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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class ListIpsSwitchStatusSolution {

    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);

CfwClient client = CfwClient.newBuilder()
    .withCredential(auth)
    .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
    .build();
ListIpsSwitchStatusRequest request = new ListIpsSwitchStatusRequest();
request.withObjectId("<object_id>");
request.withEnterpriseProjectId("<enterprise_project_id>");
request.withFwInstanceId("<fw_instance_id>");
try {
    ListIpsSwitchStatusResponse response = client.listIpsSwitchStatus(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListIpsSwitchStatusRequest()
        request.object_id = "<object_id>"
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        response = client.list_ips_switch_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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListIpsSwitchStatusRequest{}
    request.ObjectId = "<object_id>"
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest := "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    response, err := client.ListIpsSwitchStatus(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
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.8.2 IPS 特性开关操作

功能介绍

切换开关状态

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/ips/switch

表 4-393 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	project_id

表 4-394 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

请求参数

表 4-395 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）
X-Language	否	String	语言头部，默认为zh-cn，如需使用英文，请选择en-us

表 4-396 请求 Body 参数

参数	是否必选	参数类型	描述
object_id	是	String	防护对象id，是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id，可通过调用查询防火墙实例接口获得，注意type为0的为互联网边界防护对象id，type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
ips_type	是	Integer	补丁类型，仅支持虚拟补丁，值为2。 枚举值： <ul style="list-style-type: none">• 1• 2
status	是	Integer	ips特性开关状态

响应参数

状态码： 200

表 4-397 响应 Body 参数

参数	参数类型	描述
trace_id	String	trace_id
data	data object	object

表 4-398 data

参数	参数类型	描述
id	String	防护对象id

请求示例

ips开关接口，该接口是租户界面通过开关控制引擎的基础补丁和虚拟补丁的开闭状态。以下示例为开启项目id为14181c1245cf4fd786824efe1e2b9388，防护对象id为1530de8a-522d-4771-9067-9fa4e2f53b48的基础补丁开关。

```
https://{Endpoint}/v1/14181c1245cf4fd786824efe1e2b9388/ips/switch?fw_instance_id=546af3f8-88e9-47f2-a205-2346d7090925&enterprise_project_id=default

{
  "ips_type": 1,
  "object_id": "1530de8a-522d-4771-9067-9fa4e2f53b48",
  "status": 1
}
```

响应示例

状态码： 200

OK

```
{
  "data": {
    "id": "1530de8a-522d-4771-9067-9fa4e2f53b48"
  },
  "trace_id": "358144a9885ff55100aa63cb0d0e1039"
}
```

SDK 代码示例

SDK代码示例如下。

Java

ips开关接口，该接口是租户界面通过开关控制引擎的基础补丁和虚拟补丁的开闭状态。以下示例为开启项目id为14181c1245cf4fd786824efe1e2b9388，防护对象id为1530de8a-522d-4771-9067-9fa4e2f53b48的基础补丁开关。

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class ChangelpsSwitchStatusSolution {

    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);

CfwClient client = CfwClient.newBuilder()
    .withCredential(auth)
    .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
    .build();
ChangeIpsSwitchStatusRequest request = new ChangeIpsSwitchStatusRequest();
request.withEnterpriseProjectId("<enterprise_project_id>");
request.withFwInstanceId("<fw_instance_id>");
IpsSwitchDTO body = new IpsSwitchDTO();
body.withStatus(1);
body.withIpsType(IpsSwitchDTO.IpsTypeEnum.NUMBER_1);
body.withObjectId("1530de8a-522d-4771-9067-9fa4e2f53b48");
request.withBody(body);
try {
    ChangeIpsSwitchStatusResponse response = client.changeIpsSwitchStatus(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

ips开关接口，该接口是租户界面通过开关控制引擎的基础补丁和虚拟补丁的开闭状态。以下示例为开启项目id为14181c1245cf4fd786824efe1e2b9388，防护对象id为1530de8a-522d-4771-9067-9fa4e2f53b48的基础补丁开关。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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



```
try:
    request = ChangepsSwitchStatusRequest()
    request.enterprise_project_id = "<enterprise_project_id>"
    request.fw_instance_id = "<fw_instance_id>"
    request.body = IpsSwitchDTO(
        status=1,
        ips_type=1,
        object_id="1530de8a-522d-4771-9067-9fa4e2f53b48"
    )
    response = client.change_ips_switch_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

ips开关接口，该接口是租户界面通过开关控制引擎的基础补丁和虚拟补丁的开闭状态。以下示例为开启项目id为14181c1245cf4fd786824efe1e2b9388，防护对象id为1530de8a-522d-4771-9067-9fa4e2f53b48的基础补丁开关。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ChangepsSwitchStatusRequest{
        enterpriseProjectIdRequest:= "<enterprise_project_id>"
        request.EnterpriseProjectId = &enterpriseProjectIdRequest
        fwInstanceIdRequest:= "<fw_instance_id>"
        request.FwInstanceId = &fwInstanceIdRequest
        request.Body = &model.IpsSwitchDto{
            Status: int32(1),
            IpsType: model.GetIpsSwitchDtolpsTypeEnum().E_1,
            ObjectId: "1530de8a-522d-4771-9067-9fa4e2f53b48",
        }
    }
    response, err := client.ChangepsSwitchStatus(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
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.8.3 查询防护模式

功能介绍

查询防护模式

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/ips/protect

表 4-399 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-400 Query 参数

参数	是否必选	参数类型	描述
object_id	是	String	防护对象id，是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id，可通过调用查询防火墙实例接口获得，注意type为0的为互联网边界防护对象id，type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
enterprise_project_id	否	String	企业项目id，用户支持企业项目后，由企业项目生成的id。
fw_instance_id	否	String	防火墙实例id，创建云防火墙后用于标志防火墙由系统自动生成的标志id，可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下，fw_instance_id为空时，返回账号下第一个墙的信息；fw_instance_id非空时，返回与fw_instance_id对应墙的信息。

请求参数

表 4-401 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）

响应参数

状态码： 200

表 4-402 响应 Body 参数

参数	参数类型	描述
data	IpsProtectModeObject object	ips防护模式数据

表 4-403 IpsProtectModeObject

参数	参数类型	描述
id	String	ips防护模式id
mode	Integer	ips防护模式，0：观察模式，1：严格模式，2：中等模式，3：宽松模式

状态码： 400

表 4-404 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

查询项目id为9d80d070b6d44942af73c9c3d38e0429的ips防护模式。

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/ips/protect?  
fw_instance_id=546af3f8-88e9-47f2-  
a205-2346d7090925&enterprise_project_id=default&object_id=cfebd347-b655-4b84-b938-3c54317599b2
```

响应示例

状态码： 400

Bad Request

```
{  
  "error_code": "CFW.0020016",  
  "error_msg": "实例状态错误"  
}
```

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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class ListIpsProtectModeSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListIpsProtectModeRequest request = new ListIpsProtectModeRequest();
        request.withObjectId("<object_id>");
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        try {
            ListIpsProtectModeResponse response = client.listIpsProtectMode(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CfwClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CfwRegion.value_of("<YOUR REGION>")) \
```

```
.build()

try:
    request = ListIpsProtectModeRequest()
    request.object_id = "<object_id>"
    request.enterprise_project_id = "<enterprise_project_id>"
    request.fw_instance_id = "<fw_instance_id>"
    response = client.list_ips_protect_mode(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListIpsProtectModeRequest{}
    request.ObjectId = "<object_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest := &enterpriseProjectIdRequest
    request.FwInstanceId = &fwInstanceIdRequest := &fwInstanceIdRequest
    response, err := client.ListIpsProtectMode(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	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.8.4 切换防护模式

功能介绍

切换防护模式

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/ips/protect

表 4-405 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-406 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目id，用户支持企业项目后，由企业项目生成的id。

参数	是否必选	参数类型	描述
fw_instance_id	否	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。默认情况下, fw_instance_id为空时, 返回账号下第一个墙的信息; fw_instance_id非空时, 返回与fw_instance_id对应墙的信息。

请求参数

表 4-407 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）

表 4-408 请求 Body 参数

参数	是否必选	参数类型	描述
object_id	否	String	防护对象id, 是创建云防火墙后用于区分互联网边界防护和VPC边界防护的标志id, 可通过调用查询防火墙实例接口获得, 注意type为0的为互联网边界防护对象id, type为1的为VPC边界防护对象id。具体可参考APIExplorer和帮助中心FAQ。
mode	否	Integer	ips防护模式, 0: 观察模式, 1: 严格模式, 2: 中等模式, 3: 宽松模式

响应参数

状态码： 200

表 4-409 响应 Body 参数

参数	参数类型	描述
data	IdObject object	更新ips防护模式数据

表 4-410 IdObject

参数	参数类型	描述
id	String	id值
name	String	名称

状态码： 400

表 4-411 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

给项目id为9d80d070b6d44942af73c9c3d38e0429的项目下的防护id为cfebd347-b655-4b84-b938-3c54317599b2的防护对象下发拦截模式为严格的防护模式。

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/ips/protect
{
  "object_id" : "cfebd347-b655-4b84-b938-3c54317599b2",
  "mode" : 1
}
```

响应示例

状态码： 200

OK

```
{
  "data" : {
    "id" : "cfebd347-b655-4b84-b938-3c54317599b2"
  }
}
```

状态码： 400

Bad Request

```
{
  "error_code" : "CFW.0020016",
  "error_msg" : "实例状态错误"
}
```

SDK 代码示例

SDK代码示例如下。

Java

给项目id为9d80d070b6d44942af73c9c3d38e0429的项目下的防护id为cfebd347-b655-4b84-b938-3c54317599b2的防护对象下发拦截模式为严格的防护模式。

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class ChangelpsProtectModeSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ChangelpsProtectModeRequest request = new ChangelpsProtectModeRequest();
        request.withEnterpriseProjectId("<enterprise_project_id>");
        request.withFwInstanceId("<fw_instance_id>");
        IpsProtectDTO body = new IpsProtectDTO();
        body.withMode(1);
        body.withObjectId("cfebd347-b655-4b84-b938-3c54317599b2");
        request.withBody(body);
        try {
            ChangelpsProtectModeResponse response = client.changelpsProtectMode(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

给项目id为9d80d070b6d44942af73c9c3d38e0429的项目下的防护id为cfebd347-b655-4b84-b938-3c54317599b2的防护对象下发拦截模式为严格的防护模式。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ChangepsProtectModeRequest()
        request.enterprise_project_id = "<enterprise_project_id>"
        request.fw_instance_id = "<fw_instance_id>"
        request.body = IpsProtectDTO(
            mode=1,
            object_id="cfebd347-b655-4b84-b938-3c54317599b2"
        )
        response = client.change_ips_protect_mode(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

给项目id为9d80d070b6d44942af73c9c3d38e0429的项目下的防护id为cfebd347-b655-4b84-b938-3c54317599b2的防护对象下发拦截模式为严格的防护模式。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ChangelpsProtectModeRequest{
    enterpriseProjectIdRequest:= "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    fwInstanceIdRequest:= "<fw_instance_id>"
    request.FwInstanceId = &fwInstanceIdRequest
    modelpsProtectDto:= int32(1)
    objectIdIpsProtectDto:= "cfebd347-b655-4b84-b938-3c54317599b2"
    request.Body = &model.IpsProtectDto{
        Mode: &modelpsProtectDto,
        ObjectId: &objectIdIpsProtectDto,
    }
}
response, err := client.ChangelpsProtectMode(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	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.9 日志管理

4.9.1 获取日志配置

功能介绍

获取日志配置

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/cfw/logs/configuration

表 4-412 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-413 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。

请求参数

表 4-414 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）

响应参数

状态码： 200

表 4-415 响应 Body 参数

参数	参数类型	描述
data	LogConfigDt o object	日志配置数据

表 4-416 LogConfigDto

参数	参数类型	描述
fw_instance_id	String	防火墙id
lts_enable	Integer	是否开启LTS
lts_log_group_id	String	LTS日志分组id
lts_attack_log_stream_id	String	攻击日志流id
lts_attack_log_stream_enable	Integer	是否开启攻击日志流
lts_access_log_stream_id	String	访问控制日志流id
lts_access_log_stream_enable	Integer	是否开启访问控制流
lts_flow_log_stream_id	String	流量日志id
lts_flow_log_stream_enable	Integer	是否开启流量日志

请求示例

查询项目id为408972e72dcd4c1a9b033e955802a36b下的防火墙id为4e113415-7811-4bb3-bf5e-eb835953f7d4的防火墙的日志配置。

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/cfw/logs/configuration?fw_instance_id=4e113415-7811-4bb3-bf5e-eb835953f7d4&enterprise_project_id=default
```

响应示例

状态码： 200

日志配置DTO

```
{
  "data" : {
    "fw_instance_id" : "4df2bcd1-6299-4fba-8e71-8d50ea807090",
    "lts_access_log_stream_enable" : 0,
    "lts_attack_log_stream_enable" : 0,
    "lts_enable" : 0,
    "lts_flow_log_stream_enable" : 0,
    "lts_log_group_id" : "d783ce42-7f56-4c2d-9a96-b1043d016f5a"
  }
}
```

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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class ListLogConfigSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListLogConfigRequest request = new ListLogConfigRequest();
        request.withFwInstanceId("<fw_instance_id>");
        request.withEnterpriseProjectId("<enterprise_project_id>");
        try {
            ListLogConfigResponse response = client.listLogConfig(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.getStatusCode());
            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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")  
    sk = __import__('os').getenv("CLOUD_SDK_SK")  
  
    credentials = BasicCredentials(ak, sk) \  
  
    client = CfwClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(CfwRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ListLogConfigRequest()  
        request.fw_instance_id = "<fw_instance_id>"  
        request.enterprise_project_id = "<enterprise_project_id>"  
        response = client.list_log_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"  
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(  
        cfw.CfwClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).
```



```
WithCredential(auth).
Build()

request := &model.ListLogConfigRequest{}
request.FwInstanceId = "<fw_instance_id>"
enterpriseProjectIdRequest := "<enterprise_project_id>"
request.EnterpriseProjectId = &enterpriseProjectIdRequest
response, err := client.ListLogConfig(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	日志配置DTO

错误码

请参见[错误码](#)。

4.9.2 创建日志配置

功能介绍

创建日志配置

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/cfw/logs/configuration

表 4-417 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-418 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考 API Explorer 和帮助中心 FAQ。
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。

请求参数

表 4-419 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）

表 4-420 请求 Body 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	防火墙id
lts_enable	是	Integer	是否开启LTS
lts_log_group_id	是	String	LTS日志分组id
lts_attack_log_stream_id	否	String	攻击日志流id
lts_attack_log_stream_enable	是	Integer	是否开启攻击日志流
lts_access_log_stream_id	否	String	访问控制日志流id
lts_access_log_stream_enable	是	Integer	是否开启访问控制流
lts_flow_log_stream_id	否	String	流量日志id

参数	是否必选	参数类型	描述
lts_flow_log_stream_enable	是	Integer	是否开启流量日志

响应参数

状态码： 200

表 4-421 响应 Body 参数

参数	参数类型	描述
data	String	添加日志配置返回值

请求示例

给项目id为408972e72dcd4c1a9b033e955802a36b下的防火墙4d6c860a-0338-49e8-ac64-fcaeb4182ba5添加日志流配置，lts分组id为20282428-a8f9-4e75-8246-165e64cf8ba8，访问控制日志流关闭，流量日志流关闭，攻击日志流关闭，lts关闭。

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/cfw/logs/configuration?  
fw_instance_id=4d6c860a-0338-49e8-ac64-fcaeb4182ba5&enterprise_project_id=default
```

```
{  
  "fw_instance_id": "4d6c860a-0338-49e8-ac64-fcaeb4182ba5",  
  "lts_enable": 0,  
  "lts_log_group_id": "20282428-a8f9-4e75-8246-165e64cf8ba8",  
  "lts_attack_log_stream_enable": 0,  
  "lts_access_log_stream_enable": 0,  
  "lts_flow_log_stream_enable": 0  
}
```

响应示例

状态码： 200

添加日志配置返回值

```
{  
  "data": "4d6c860a-0338-49e8-ac64-fcaeb4182ba5"  
}
```

SDK 代码示例

SDK代码示例如下。

Java

给项目id为408972e72dcd4c1a9b033e955802a36b下的防火墙4d6c860a-0338-49e8-ac64-fcaeb4182ba5添加日志流配置，lts分组id为20282428-a8f9-4e75-8246-165e64cf8ba8，访问控制日志流关闭，流量日志流关闭，攻击日志流关闭，lts关闭。

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class AddLogConfigSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        AddLogConfigRequest request = new AddLogConfigRequest();
        request.withFwInstanceId("<fw_instance_id>");
        request.withEnterpriseProjectId("<enterprise_project_id>");
        LogConfigDto body = new LogConfigDto();
        body.withLtsFlowLogStreamEnable(0);
        body.withLtsAccessLogStreamEnable(0);
        body.withLtsAttackLogStreamEnable(0);
        body.withLtsLogGroupId("20282428-a8f9-4e75-8246-165e64cf8ba8");
        body.withLtsEnable(0);
        body.withFwInstanceId("4d6c860a-0338-49e8-ac64-fcaeb4182ba5");
        request.withBody(body);
        try {
            AddLogConfigResponse response = client.addLogConfig(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.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

给项目id为408972e72dcd4c1a9b033e955802a36b下的防火墙4d6c860a-0338-49e8-ac64-fcaeb4182ba5添加日志流配置，lts分组id为20282428-a8f9-4e75-8246-165e64cf8ba8，访问控制日志流关闭，流量日志流关闭，攻击日志流关闭，lts关闭。

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = AddLogConfigRequest()
        request.fw_instance_id = "<fw_instance_id>"
        request.enterprise_project_id = "<enterprise_project_id>"
        request.body = LogConfigDto(
            lts_flow_log_stream_enable=0,
            lts_access_log_stream_enable=0,
            lts_attack_log_stream_enable=0,
            lts_log_group_id="20282428-a8f9-4e75-8246-165e64cf8ba8",
            lts_enable=0,
            fw_instance_id="4d6c860a-0338-49e8-ac64-fcaeb4182ba5"
        )
        response = client.add_log_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

给项目id为408972e72dcd4c1a9b033e955802a36b下的防火墙4d6c860a-0338-49e8-ac64-fcaeb4182ba5添加日志流配置，lts分组id为20282428-a8f9-4e75-8246-165e64cf8ba8，访问控制日志流关闭，流量日志流关闭，攻击日志流关闭，lts关闭。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.AddLogConfigRequest{}
request.FwInstanceId = "<fw_instance_id>"
enterpriseProjectIdRequest := "<enterprise_project_id>"
request.EnterpriseProjectId = &enterpriseProjectIdRequest
request.Body = &model.LogConfigDto{
    LtsFlowLogStreamEnable: int32(0),
    LtsAccessLogStreamEnable: int32(0),
    LtsAttackLogStreamEnable: int32(0),
    LtsLogGroupId: "20282428-a8f9-4e75-8246-165e64cf8ba8",
    LtsEnable: int32(0),
    FwInstanceId: "4d6c860a-0338-49e8-ac64-fcaeb4182ba5",
}
response, err := client.AddLogConfig(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	添加日志配置返回值

错误码

请参见[错误码](#)。

4.9.3 更新日志配置

功能介绍

更新日志配置

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/cfw/logs/configuration

表 4-422 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-423 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。

请求参数

表 4-424 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）

表 4-425 请求 Body 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	防火墙id
lts_enable	是	Integer	是否开启LTS
lts_log_group_id	是	String	LTS日志分组id
lts_attack_log_stream_id	否	String	攻击日志流id
lts_attack_log_stream_enable	是	Integer	是否开启攻击日志流

参数	是否必选	参数类型	描述
lts_access_log_stream_id	否	String	访问控制日志流id
lts_access_log_stream_enable	是	Integer	是否开启访问控制流
lts_flow_log_stream_id	否	String	流量日志id
lts_flow_log_stream_enable	是	Integer	是否开启流量日志

响应参数

状态码： 200

表 4-426 响应 Body 参数

参数	参数类型	描述
data	String	日志配置id

请求示例

更新项目id为408972e72dcd4c1a9b033e955802a36b的防火墙id为22c4a5db-504c-471f-8187-5192bc11de0b的防火墙的日志配置，lts日志配置为关闭，流日志、访问控制日志、攻击日志设置为关闭。

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/cfw/logs/configuration?fw_instance_id=22c4a5db-504c-471f-8187-5192bc11de0b&enterprise_project_id=default
```

```
{
  "fw_instance_id": "22c4a5db-504c-471f-8187-5192bc11de0b",
  "lts_enable": 0,
  "lts_log_group_id": "20282428-a8f9-4e75-8246-165e64cf8ba8",
  "lts_attack_log_stream_enable": 0,
  "lts_access_log_stream_enable": 0,
  "lts_flow_log_stream_enable": 0
}
```

响应示例

状态码： 200

更新日志配置返回值

```
{
  "data": "4e113415-7811-4bb3-bf5e-eb835953f7d4"
}
```


SDK 代码示例

SDK代码示例如下。

Java

更新项目id为408972e72dcd4c1a9b033e955802a36b的防火墙id为22c4a5db-504c-471f-8187-5192bc11de0b的防火墙的日志配置，lts日志配置为关闭，流日志、访问控制日志、攻击日志设置为关闭。

```
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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class UpdateLogConfigSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateLogConfigRequest request = new UpdateLogConfigRequest();
        request.withFwInstanceId("<fw_instance_id>");
        request.withEnterpriseProjectId("<enterprise_project_id>");
        LogConfigDto body = new LogConfigDto();
        body.withLtsFlowLogStreamEnable(0);
        body.withLtsAccessLogStreamEnable(0);
        body.withLtsAttackLogStreamEnable(0);
        body.withLtsLogGroupId("20282428-a8f9-4e75-8246-165e64cf8ba8");
        body.withLtsEnable(0);
        body.withFwInstanceId("22c4a5db-504c-471f-8187-5192bc11de0b");
        request.withBody(body);
        try {
            UpdateLogConfigResponse response = client.updateLogConfig(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

更新项目id为408972e72dcd4c1a9b033e955802a36b的防火墙id为22c4a5db-504c-471f-8187-5192bc11de0b的防火墙的日志配置，lts日志配置为关闭，流日志、访问控制日志、攻击日志设置为关闭。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateLogConfigRequest()
        request.fw_instance_id = "<fw_instance_id>"
        request.enterprise_project_id = "<enterprise_project_id>"
        request.body = LogConfigDto(
            lts_flow_log_stream_enable=0,
            lts_access_log_stream_enable=0,
            lts_attack_log_stream_enable=0,
            lts_log_group_id="20282428-a8f9-4e75-8246-165e64cf8ba8",
            lts_enable=0,
            fw_instance_id="22c4a5db-504c-471f-8187-5192bc11de0b"
        )
        response = client.update_log_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

更新项目id为408972e72dcd4c1a9b033e955802a36b的防火墙id为22c4a5db-504c-471f-8187-5192bc11de0b的防火墙的日志配置，lts日志配置为关闭，流日志、访问控制日志、攻击日志设置为关闭。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateLogConfigRequest{}
request.FwInstanceId = "<fw_instance_id>"
enterpriseProjectIdRequest := "<enterprise_project_id>"
request.EnterpriseProjectId = &enterpriseProjectIdRequest
request.Body = &model.LogConfigDto{
    LtsFlowLogStreamEnable: int32(0),
    LtsAccessLogStreamEnable: int32(0),
    LtsAttackLogStreamEnable: int32(0),
    LtsLogGroupId: "20282428-a8f9-4e75-8246-165e64cf8ba8",
    LtsEnable: int32(0),
    FwInstanceId: "22c4a5db-504c-471f-8187-5192bc11de0b",
}
response, err := client.UpdateLogConfig(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	更新日志配置返回值

错误码

请参见[错误码](#)。

4.9.4 查询流日志

功能介绍

查询流日志

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/cfw/logs/flow

表 4-427 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-428 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考APIExplorer和帮助中心FAQ。
direction	否	String	方向
log_type	否	String	日志类型 枚举值: <ul style="list-style-type: none">• internet• vpc• nat
start_time	是	Long	开始时间
end_time	是	Long	结束时间
src_ip	否	String	源IP
src_port	否	Integer	源端口 最小值: 0 最大值: 65535
dst_ip	否	String	目的IP
dst_port	否	Integer	目的端口 最小值: 0 最大值: 65535

参数	是否必选	参数类型	描述
protocol	否	String	协议类型:TCP为6, UDP为17,ICMP为1,ICMPV6为58,ANY为-1,手动类型不为空, 自动类型为 枚举值: <ul style="list-style-type: none">• 6• 17• 1• 58
app	否	String	应用协议
log_id	否	String	文档ID, 首页时为null, 非首页时不为null
next_date	否	Long	日期, 首页时为null, 非首页时不为null
offset	否	Integer	偏移量: 指定返回记录的开始位置, 必须为数字, 取值范围为大于或等于0, 默认0
limit	是	Integer	每页显示个数, 范围为1-1024 最小值: 1 最大值: 1024
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
dst_host	否	String	目的主机
src_region_name	否	String	源region名称
dst_region_name	否	String	目的region名称

请求参数

表 4-429 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）

响应参数

状态码： 200

表 4-430 响应 Body 参数

参数	参数类型	描述
data	data object	查询流日志返回值

表 4-431 data

参数	参数类型	描述
total	Integer	返回数量
limit	Integer	每页显示个数，范围为1-1024
records	Array of records objects	记录

表 4-432 records

参数	参数类型	描述
bytes	Double	字节
direction	String	方向，有内到外和外到内两种枚举值： <ul style="list-style-type: none">• out2in• in2out
packets	Integer	包
start_time	Long	开始时间
end_time	Long	结束时间
log_id	String	文档ID
src_ip	String	源IP
src_port	Integer	源端口
dst_ip	String	目的IP
app	String	应用协议
dst_port	Integer	目的端口
protocol	String	协议类型:TCP为6,UDP为17,ICMP为1,ICMPV6为58,ANY为-1,手动类型不为空，自动类型为空

参数	参数类型	描述
dst_host	String	目标主机
dst_region_id	String	目的地域id
dst_region_name	String	目的地域名称
src_region_id	String	源地域id
src_region_name	String	源地域名称

状态码： 400

表 4-433 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

查询项目id为9d80d070b6d44942af73c9c3d38e0429防火墙id为2af58b7c-893c-4453-a984-bdd9b1bd6318，起点时间为1663555012000，终点时间为1664159798000的首页的流日志。

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/cfw/logs/flow?  
fw_instance_id=2af58b7c-893c-4453-a984-  
bdd9b1bd6318&start_time=1663555012000&end_time=1664159798000&limit=10
```

响应示例

状态码： 200

OK

```
{  
  "data": {  
    "limit": 10,  
    "records": [ {  
      "app": "SSH",  
      "bytes": 34.5,  
      "direction": "out2in",  
      "dst_ip": "100.95.148.49",  
      "dst_port": 22,  
      "end_time": 1664155493000,
```

```
"log_id" : "76354",
"packets" : 25,
"protocol" : "TCP",
"src_ip" : "100.93.27.17",
"src_port" : 49634,
"start_time" : 1664155428000
}],
"total" : 1
}
```

状态码： 400

Bad Request

```
{
  "error_code" : "CFW.00500002",
  "error_msg" : "时间间距错误"
}
```

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.cfw.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfw.v1.*;
import com.huaweicloud.sdk.cfw.v1.model.*;

public class ListFlowLogsSolution {

    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);

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListFlowLogsRequest request = new ListFlowLogsRequest();
        request.withFwInstanceId("<fw_instance_id>");
        request.withDirection("<direction>");
        request.withLogType(ListFlowLogsRequest.LogTypeEnum.fromValue("<log_type>"));
        request.withStartTime("<start_time>L");
        request.withEndTime("<end_time>L");
        request.withSrcIp("<src_ip>");
        request.withSrcPort("<src_port>");
        request.withDstIp("<dst_ip>");
        request.withDstPort("<dst_port>");
        request.withProtocol(ListFlowLogsRequest.ProtocolEnum.fromValue("<protocol>"));
    }
}
```



```
request.withApp("<app>");
request.withLogId("<log_id>");
request.withNextDate("<next_date>L");
request.withOffset("<offset>");
request.withLimit("<limit>");
request.withEnterpriseProjectId("<enterprise_project_id>");
request.withDstHost("<dst_host>");
try {
    ListFlowLogsResponse response = client.listFlowLogs(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.getStatusCode());
    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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListFlowLogsRequest()
        request.fw_instance_id = "<fw_instance_id>"
        request.direction = "<direction>"
        request.log_type = "<log_type>"
        request.start_time = <start_time>
        request.end_time = <end_time>
        request.src_ip = "<src_ip>"
        request.src_port = <src_port>
        request.dst_ip = "<dst_ip>"
        request.dst_port = <dst_port>
        request.protocol = "<protocol>"
        request.app = "<app>"
        request.log_id = "<log_id>"
        request.next_date = <next_date>
        request.offset = <offset>
        request.limit = <limit>
        request.enterprise_project_id = "<enterprise_project_id>"
        request.dst_host = "<dst_host>"
        response = client.list_flow_logs(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListFlowLogsRequest{}
    request.FwInstanceId = "<fw_instance_id>"
    directionRequest := "<direction>"
    request.Direction = &directionRequest
    logTypeRequest := model.GetListFlowLogsRequestLogTypeEnum().<LOG_TYPE>
    request.LogType = &logTypeRequest
    request.StartTime = int64(<start_time>)
    request.EndTime = int64(<end_time>)
    srcIpRequest := "<src_ip>"
    request.SrcIp = &srcIpRequest
    srcPortRequest := int32(<src_port>)
    request.SrcPort = &srcPortRequest
    dstIpRequest := "<dst_ip>"
    request.DstIp = &dstIpRequest
    dstPortRequest := int32(<dst_port>)
    request.DstPort = &dstPortRequest
    protocolRequest := model.GetListFlowLogsRequestProtocolEnum().<PROTOCOL>
    request.Protocol = &protocolRequest
    appRequest := "<app>"
    request.App = &appRequest
    logIdRequest := "<log_id>"
    request.LogId = &logIdRequest
    nextDateRequest := int64(<next_date>)
    request.NextDate = &nextDateRequest
    offsetRequest := int32(<offset>)
    request.Offset = &offsetRequest
    request.Limit = int32(<limit>)
    enterpriseProjectIdRequest := "<enterprise_project_id>"
    request.EnterpriseProjectId = &enterpriseProjectIdRequest
    dstHostRequest := "<dst_host>"
```

```
request.DstHost = &dstHostRequest
response, err := client.ListFlowLogs(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	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.9.5 查询访问控制日志

功能介绍

查询访问控制日志

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/cfw/logs/access-control

表 4-434 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-435 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考 API Explorer 和帮助中心 FAQ。
rule_id	否	String	规则ID
start_time	是	Long	开始时间
end_time	是	Long	结束时间
src_ip	否	String	源IP
src_port	否	Integer	源端口
dst_ip	否	String	目的IP
dst_port	否	Integer	目的端口
protocol	否	String	协议
app	否	String	应用协议
log_id	否	String	文档ID, 第一页为空, 其他页不为空
next_date	否	Integer	日期, 第一页为空, 其他页不为空
offset	否	Integer	偏移量: 指定返回记录的开始位置, 必须为数字, 取值范围为大于或等于0, 默认0
limit	是	Integer	每页显示个数, 范围为1-1024
log_type	否	String	日志类型 枚举值: <ul style="list-style-type: none">• internet• nat• vpc
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
dst_host	否	String	目标主机
rule_name	否	String	规则名称
action	否	String	动作0: permit, 1: deny
src_region_name	否	String	源region名称

参数	是否必选	参数类型	描述
dst_region_name	否	String	目的region名称

请求参数

表 4-436 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）

响应参数

状态码： 200

表 4-437 响应 Body 参数

参数	参数类型	描述
data	data object	查询访问控制日志返回数据

表 4-438 data

参数	参数类型	描述
total	Integer	返回数量
limit	Integer	每页显示个数，范围为1-1024
records	Array of records objects	记录

表 4-439 records

参数	参数类型	描述
action	String	动作0: permit,1: deny
rule_name	String	规则名称
rule_id	String	规则ID

参数	参数类型	描述
hit_time	Long	命中时间
src_region_id	String	源区域id
src_region_name	String	源区域name
dst_region_id	String	目的区域id
dst_region_name	String	目的区域name
log_id	String	文档ID
src_ip	String	源IP
src_port	Integer	源端口
dst_ip	String	目的IP
dst_port	Integer	目的端口
protocol	String	协议类型:TCP为6,UDP为17,ICMP为1,ICMPV6为58,ANY为-1,手动类型不为空,自动类型为空
app	String	应用协议
dst_host	String	目标主机

状态码： 400

表 4-440 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

查询项目id为9d80d070b6d44942af73c9c3d38e0429, 防火墙id为2af58b7c-893c-4453-a984-bdd9b1bd6318, 开始时间为1664159069544, 结束时间为1664162669544, 初始位置为0的第一页的数据

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/cfw/logs/access-control?  
fw_instance_id=2af58b7c-893c-4453-a984-  
bdd9b1bd6318&start_time=1664159069544&end_time=1664162669544&limit=10
```

响应示例

状态码： 200

OK

```
{  
  "data" : {  
    "limit" : 10,  
    "records" : [ {  
      "action" : "deny",  
      "app" : "PING",  
      "dst_ip" : "100.85.216.211",  
      "dst_port" : 59,  
      "hit_time" : 1664164255000,  
      "log_id" : "46032",  
      "protocol" : "ICMP: ECHO_REQUEST",  
      "rule_id" : "c755be1c-4b92-4ae7-a15e-c2d02b152538",  
      "rule_name" : "eip_ipv4_w_n_default_deny",  
      "src_ip" : "100.95.148.49",  
      "src_port" : 24954  
    } ],  
    "total" : 1  
  }  
}
```

状态码： 400

Bad Request

```
{  
  "error_code" : "CFW.00500002",  
  "error_msg" : "时间间距错误"  
}
```

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.cfw.v1.region.CfwRegion;  
import com.huaweicloud.sdk.cfw.v1.*;  
import com.huaweicloud.sdk.cfw.v1.model.*;  
  
public class ListAccessControlLogsSolution {  
  
    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);

CfwClient client = CfwClient.newBuilder()
    .withCredential(auth)
    .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
    .build();
ListAccessControlLogsRequest request = new ListAccessControlLogsRequest();
request.withFwInstanceId("<fw_instance_id>");
request.withRuleId("<rule_id>");
request.withStartTime(<start_time>L);
request.withEndTime(<end_time>L);
request.withSrcIp("<src_ip>");
request.withSrcPort(<src_port>);
request.withDstIp("<dst_ip>");
request.withDstPort(<dst_port>);
request.withProtocol("<protocol>");
request.withApp("<app>");
request.withLogId("<log_id>");
request.withNextDate(<next_date>);
request.withOffset(<offset>);
request.withLimit(<limit>);
request.withLogType(ListAccessControlLogsRequest.LogTypeEnum.fromValue("<log_type>"));
request.withEnterpriseProjectId("<enterprise_project_id>");
request.withDstHost("<dst_host>");
request.withRuleName("<rule_name>");
request.withAction("<action>");
try {
    ListAccessControlLogsResponse response = client.listAccessControlLogs(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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CfwClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CfwRegion.value_of("<YOUR REGION>")) \
```



```
.build()

try:
    request = ListAccessControlLogsRequest()
    request.fw_instance_id = "<fw_instance_id>"
    request.rule_id = "<rule_id>"
    request.start_time = <start_time>
    request.end_time = <end_time>
    request.src_ip = "<src_ip>"
    request.src_port = <src_port>
    request.dst_ip = "<dst_ip>"
    request.dst_port = <dst_port>
    request.protocol = "<protocol>"
    request.app = "<app>"
    request.log_id = "<log_id>"
    request.next_date = <next_date>
    request.offset = <offset>
    request.limit = <limit>
    request.log_type = "<log_type>"
    request.enterprise_project_id = "<enterprise_project_id>"
    request.dst_host = "<dst_host>"
    request.rule_name = "<rule_name>"
    request.action = "<action>"
    response = client.list_access_control_logs(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAccessControlLogsRequest{
        request.FwInstanceId = "<fw_instance_id>"
        request.RuleIdRequest = "<rule_id>"
        request.RuleId = &ruleIdRequest
        request.StartTime = int64(<start_time>)
        request.EndTime = int64(<end_time>)
```

```
srcIpRequest:= "<src_ip>"
request.SrcIp = &srcIpRequest
srcPortRequest:= int32(<src_port>)
request.SrcPort = &srcPortRequest
dstIpRequest:= "<dst_ip>"
request.DstIp = &dstIpRequest
dstPortRequest:= int32(<dst_port>)
request.DstPort = &dstPortRequest
protocolRequest:= "<protocol>"
request.Protocol = &protocolRequest
appRequest:= "<app>"
request.App = &appRequest
logIdRequest:= "<log_id>"
request.LogId = &logIdRequest
nextDateRequest:= int32(<next_date>)
request.NextDate = &nextDateRequest
offsetRequest:= int32(<offset>)
request.Offset = &offsetRequest
request.Limit = int32(<limit>)
logTypeRequest:= model.GetListAccessControlLogsRequestLogTypeEnum().<LOG_TYPE>
request.LogType = &logTypeRequest
enterpriseProjectIdRequest:= "<enterprise_project_id>"
request.EnterpriseProjectId = &enterpriseProjectIdRequest
dstHostRequest:= "<dst_host>"
request.DstHost = &dstHostRequest
ruleNameRequest:= "<rule_name>"
request.RuleName = &ruleNameRequest
actionRequest:= "<action>"
request.Action = &actionRequest
response, err := client.ListAccessControlLogs(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	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

4.9.6 查询攻击日志

功能介绍

查询攻击日志

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/cfw/logs/attack

表 4-441 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目id

表 4-442 Query 参数

参数	是否必选	参数类型	描述
start_time	是	Long	开始时间
end_time	是	Long	结束时间
src_ip	否	String	源IP
src_port	否	Integer	源端口号 最小值：0 最大值：65535
dst_ip	否	String	目的IP
dst_port	否	Integer	目的端口号 最小值：0 最大值：65535
protocol	否	String	协议类型:TCP为6, UDP为17,ICMP为1,ICMPV6为58,ANY为-1,手动类型不为空, 自动类型为 枚举值: <ul style="list-style-type: none">• 6• 17• 1• 58

参数	是否必选	参数类型	描述
app	否	String	应用协议
log_id	否	String	日志ID, 当是第一页时为空, 不是第一页时不为空
next_date	否	Long	下个日期, 当是第一页时为空, 不是第一页时不为空
offset	否	Integer	偏移量: 指定返回记录的开始位置, 必须为数字, 取值范围为大于或等于0, 默认0
limit	是	Integer	每页显示个数, 范围为1-1024
fw_instance_id	是	String	防火墙实例id, 创建云防火墙后用于标志防火墙由系统自动生成的标志id, 可通过调用查询防火墙实例接口获得。具体可参考API Explorer和帮助中心FAQ。
action	否	String	动作0: permit, 1: deny 枚举值: <ul style="list-style-type: none">• 0• 1
direction	否	String	方向0: 外到内, 1: 内到外 枚举值: <ul style="list-style-type: none">• 0• 1
attack_type	否	String	入侵事件类型
attack_rule	否	String	入侵事件规则
level	否	String	威胁等级 <ul style="list-style-type: none">• CRITICAL: 严重• HIGH: 高• MEDIUM: 中• LOW: 低
source	否	String	判断来源
enterprise_project_id	否	String	企业项目id, 用户支持企业项目后, 由企业项目生成的id。
dst_host	否	String	目标主机

参数	是否必选	参数类型	描述
log_type	否	String	日志类型 枚举值： <ul style="list-style-type: none">• internet• nat• vpc
attack_rule_id	否	String	入侵事件id
src_region_name	否	String	源region名称
dst_region_name	否	String	目的region名称

请求参数

表 4-443 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）

响应参数

状态码： 200

表 4-444 响应 Body 参数

参数	参数类型	描述
data	data object	查询攻击日志返回值

表 4-445 data

参数	参数类型	描述
total	Integer	返回数量
limit	Integer	每页显示个数，范围为1-1024

参数	参数类型	描述
records	Array of records objects	记录

表 4-446 records

参数	参数类型	描述
direction	String	方向，有内到外和外到内两种枚举值： <ul style="list-style-type: none">• out2in• in2out
action	String	动作
event_time	Long	事件时间
attack_type	String	攻击类型
attack_rule	String	攻击规则
level	String	威胁等级
source	String	来源
packet_length	Long	报文长度
attack_rule_id	String	攻击规则id
hit_time	Integer	命中时间
log_id	String	日志ID
src_ip	String	源IP
src_port	Integer	源端口 最小值： 0 最大值： 65535
dst_ip	String	目的IP
dst_port	Integer	目的端口 最小值： 0 最大值： 65535
protocol	String	协议
packet	String	攻击日志报文
app	String	应用协议

参数	参数类型	描述
packetMessages	Array of PacketMessage objects	攻击报文信息
dst_host	String	目标主机
src_region_id	String	源区域id
src_region_name	String	源区域名称
dst_region_id	String	目的区域id
dst_region_name	String	目的区域名称

表 4-447 PacketMessage

参数	参数类型	描述
hex_index	String	16进制index
hexs	Array of strings	16进制数列
utf8_String	String	utf_8字符串

状态码： 400

表 4-448 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码 最小长度： 8 最大长度： 36
error_msg	String	错误描述 最小长度： 2 最大长度： 512

请求示例

查询项目id为9d80d070b6d44942af73c9c3d38e0429防火墙id为2af58b7c-893c-4453-a984-bdd9b1bd6318初始时间为1663567058000，结束时间为1664171765000的第一页数据，查询条数为10条

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/cfw/logs/attack?  
fw_instance_id=2af58b7c-893c-4453-a984-  
bdd9b1bd6318&start_time=1663567058000&end_time=1664171765000&limit=10
```

响应示例

状态码： 200

OK

```
{  
  "data": {  
    "limit": 10,  
    "records": [ {  
      "action": "deny",  
      "app": "HTTP",  
      "attack_rule": "Tool Nmap Web Server Probe Detected",  
      "attack_rule_id": "336154",  
      "attack_type": "Web Attack",  
      "direction": "out2in",  
      "dst_ip": "100.95.148.49",  
      "dst_port": 8080,  
      "event_time": 1664146216000,  
      "level": "MEDIUM",  
      "log_id": "15591",  
      "packet": "+hZUZMhV+hY/AaHMCABFKABpXPNAADAGof1kVe6QZF  
+UMcTQH5B0wdaz888+uoAYAOVyNQAAAQEIcJrmikVb9JLCR0VUIC9uaWNUtIwcG9ydHMIMkMvVHJpJTZFaX  
R5LnR4dCUyZWJhayBIVFRQLzEuMA0KDQo=",  
      "packetMessages": [ {  
        "hex_index": "00000000",  
        "hexs": [ "fa", "16", "54", "64", "c8", "55", "fa", "16", "3f", "01", "a1", "cc", "08", "00", "45", "28" ],  
        "utf8_String": ".\u0016Td.U.\u0016?...E("   
      }, {  
        "hex_index": "00000010",  
        "hexs": [ "00", "69", "5c", "f3", "40", "00", "30", "06", "a1", "fd", "64", "55", "ee", "90", "64", "5f" ],  
        "utf8_String": ".i\.\@.0...dU.d_"   
      }, {  
        "hex_index": "00000020",  
        "hexs": [ "94", "31", "c4", "d0", "1f", "90", "74", "c1", "d6", "b3", "f3", "cf", "3e", "ba", "80", "18" ],  
        "utf8_String": ".1..\u001Ft.;>..."   
      }, {  
        "hex_index": "00000030",  
        "hexs": [ "00", "e5", "72", "35", "00", "00", "01", "01", "08", "0a", "3a", "e6", "8a", "45", "5b", "f4" ],  
        "utf8_String": "..r5.....:E["   
      }, {  
        "hex_index": "00000040",  
        "hexs": [ "92", "c2", "47", "45", "54", "20", "2f", "6e", "69", "63", "65", "25", "32", "30", "70", "6f" ],  
        "utf8_String": "..GET /nice%20po"   
      }, {  
        "hex_index": "00000050",  
        "hexs": [ "72", "74", "73", "25", "32", "43", "2f", "54", "72", "69", "25", "36", "45", "69", "74", "79" ],  
        "utf8_String": "rts%2C/Tri%6Eity"   
      }, {  
        "hex_index": "00000060",  
        "hexs": [ "2e", "74", "78", "74", "25", "32", "65", "62", "61", "6b", "20", "48", "54", "54", "50", "2f" ],  
        "utf8_String": ".txt%2ebak HTTP/"   
      }, {  
        "hex_index": "00000070",  
        "hexs": [ "31", "2e", "30", "0d", "0a", "0d", "0a" ],  
        "utf8_String": "1.0\r\r:"   
      } ],  
      "packet_length": 119,  
      "protocol": "TCP",  
      "source": "0",  
      "src_ip": "100.85.238.144",  
      "src_port": 50384  
    } ],  
    "total": 1  
  }  
}
```



```
}  
}
```

状态码： 400

Bad Request

```
{  
  "error_code" : "00500002",  
  "error_msg" : "时间间距错误"  
}
```

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.cfw.v1.region.CfwRegion;  
import com.huaweicloud.sdk.cfw.v1.*;  
import com.huaweicloud.sdk.cfw.v1.model.*;  
  
public class ListAttackLogsSolution {  
    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);  
  
        CfwClient client = CfwClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListAttackLogsRequest request = new ListAttackLogsRequest();  
        request.withStartTime(<start_time>L);  
        request.withEndTime(<end_time>L);  
        request.withSrcIp("<src_ip>");  
        request.withSrcPort(<src_port>);  
        request.withDstIp("<dst_ip>");  
        request.withDstPort(<dst_port>);  
        request.withProtocol(ListAttackLogsRequest.ProtocolEnum.fromValue("<protocol>"));  
        request.withApp("<app>");  
        request.withLogId("<log_id>");  
        request.withNextDate(<next_date>L);  
        request.withOffset(<offset>);  
        request.withLimit(<limit>);  
        request.withFwInstanceId("<fw_instance_id>");  
        request.withAction(ListAttackLogsRequest.ActionEnum.fromValue("<action>"));  
        request.withDirection(ListAttackLogsRequest.DirectionEnum.fromValue("<direction>"));  
        request.withAttackType("<attack_type>");  
        request.withAttackRule("<attack_rule>");  
        request.withLevel("<level>");  
    }  
}
```

```
request.withSource("<source>");
request.withEnterpriseProjectId("<enterprise_project_id>");
request.withDstHost("<dst_host>");
request.withLogType(ListAttackLogsRequest.LogTypeEnum.fromValue("<log_type>"));
try {
    ListAttackLogsResponse response = client.listAttackLogs(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.getStatusCode());
    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 huaweicloudsdkcfw.v1.region.cfw_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfw.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 = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListAttackLogsRequest()
        request.start_time = <start_time>
        request.end_time = <end_time>
        request.src_ip = "<src_ip>"
        request.src_port = <src_port>
        request.dst_ip = "<dst_ip>"
        request.dst_port = <dst_port>
        request.protocol = "<protocol>"
        request.app = "<app>"
        request.log_id = "<log_id>"
        request.next_date = <next_date>
        request.offset = <offset>
        request.limit = <limit>
        request.fw_instance_id = "<fw_instance_id>"
        request.action = "<action>"
        request.direction = "<direction>"
        request.attack_type = "<attack_type>"
        request.attack_rule = "<attack_rule>"
        request.level = "<level>"
        request.source = "<source>"
        request.enterprise_project_id = "<enterprise_project_id>"
        request.dst_host = "<dst_host>"
```

```
request.log_type = "<log_type>"
response = client.list_attack_logs(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"
    cfw "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/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 := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAttackLogsRequest{}
    request.StartTime = int64(<start_time>)
    request.EndTime = int64(<end_time>)
    srcIpRequest := "<src_ip>"
    request.SrcIp = &srcIpRequest
    srcPortRequest := int32(<src_port>)
    request.SrcPort = &srcPortRequest
    dstIpRequest := "<dst_ip>"
    request.DstIp = &dstIpRequest
    dstPortRequest := int32(<dst_port>)
    request.DstPort = &dstPortRequest
    protocolRequest := model.GetListAttackLogsRequestProtocolEnum().<PROTOCOL>
    request.Protocol = &protocolRequest
    appRequest := "<app>"
    request.App = &appRequest
    logIdRequest := "<log_id>"
    request.LogId = &logIdRequest
    nextDateRequest := int64(<next_date>)
    request.NextDate = &nextDateRequest
    offsetRequest := int32(<offset>)
    request.Offset = &offsetRequest
    request.Limit = int32(<limit>)
    request.FwInstanceId = "<fw_instance_id>"
    actionRequest := model.GetListAttackLogsRequestActionEnum().<ACTION>
    request.Action = &actionRequest
    directionRequest := model.GetListAttackLogsRequestDirectionEnum().<DIRECTION>
    request.Direction = &directionRequest
    attackTypeRequest := "<attack_type>"
```

```
request.AttackType = &attackTypeRequest
attackRuleRequest:= "<attack_rule>"
request.AttackRule = &attackRuleRequest
levelRequest:= "<level>"
request.Level = &levelRequest
sourceRequest:= "<source>"
request.Source = &sourceRequest
enterpriseProjectIdRequest:= "<enterprise_project_id>"
request.EnterpriseProjectId = &enterpriseProjectIdRequest
dstHostRequest:= "<dst_host>"
request.DstHost = &dstHostRequest
logTypeRequest:= model.GetListAttackLogsRequestLogTypeEnum().<LOG_TYPE>
request.LogType = &logTypeRequest
response, err := client.ListAttackLogs(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	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

错误码

请参见[错误码](#)。

A 附录

A.1 状态码

- 正常

状态码	描述	说明
200	OK	请求成功。

- 异常

状态码	描述	说明
400	Bad Request	错误的请求。
401	Unauthorized	请求未授权。
403	Forbidden	禁止访问。
404	Not Found	网页未找到。
500	Internal Server Error	系统内部错误。

A.2 错误码

当您调用API时，如果遇到“APIGW”开头的错误码，请参见[API网关错误码](#)进行处理。

状态码	错误码	错误信息	描述	处理措施
400	CFW.0010900 4	HTTP请求错误	HTTP请求错误	请稍后重试或联系技术支持
400	CFW.0020000 1	空参数错误	空参数错误	请联系技术支持。

状态码	错误码	错误信息	描述	处理措施
400	CFW.00200004	所删除资源被引用，删除失败	所删除资源被引用，删除失败	请联系技术支持。
400	CFW.00200005	操作内容不存在	操作内容不存在	请联系技术支持
400	CFW.00200007	Name冲突	Name冲突	请重新命名。
400	CFW.00200009	已存在相同的参数	已存在相同的参数	请联系技术支持。
400	CFW.00200010	配置类型错误	配置类型错误	请联系技术支持。
400	CFW.00200011	不支持批量操作	不支持批量操作	请联系技术支持。
400	CFW.00200013	组装URI错误	组装URI错误	请联系技术支持。
400	CFW.00200020	增加的ACL规则数量不能超过20个	增加的ACL规则数量不能超过20个	请减少添加acl规则数量。
400	CFW.00200022	黑白名单禁止配置全部IP段	黑白名单禁止配置全部IP段	请明确黑白名单ip地址段。
400	CFW.00200023	后台升级任务超过限制，请稍后再试	后台升级任务超过限制，请稍后再试	请联系技术支持。
400	CFW.00200024	超出最大数量限制	超出最大数量限制	请联系技术支持。
400	CFW.00200025	长连接时间超出范围	长连接时间超出范围	请确保长连接规则时长从一秒到一千天。
400	CFW.00200026	长连接规则数目到达限制	长连接规则数目到达限制	请删除一些长连接规则。
400	CFW.00200027	acl地址错误	acl地址错误	请确保acl规则地址符合规范。
400	CFW.00200028	地址类型不一致	地址类型不一致	请确保地址类型一致。
400	CFW.00200030	地址类型错误	地址类型错误	请联系技术支持。
400	CFW.00200032	引擎不支持IPv6	引擎不支持IPv6	请联系技术支持

状态码	错误码	错误信息	描述	处理措施
400	CFW.00200036	不允许更改为私网网段	不允许更改为私网网段	请联系技术支持。
400	CFW.00200041	地址参数为null	地址参数为null	请添加地址类型参数。
400	CFW.00200016	实例状态错误	实例状态错误	请联系技术支持。
400	CFW.00200110	不能操作基础防御	不能操作基础防御	请联系技术支持。
400	CFW.00300001	命令解析异常	命令解析异常	请联系技术支持。
400	CFW.00400002	无需操作	无需操作	请联系技术支持。
400	CFW.00400004	成员已存在	成员已存在	请删除一些服务组成员。
400	CFW.00400006	清除规则击中次数参数错误	清除规则击中次数参数错误	请检查并确认参数值是否合法。
400	CFW.00400007	添加规则类型不一致	添加规则类型不一致	请确保添加规则类型一致。
400	CFW.00400008	防护对象不存在	防护对象不存在	请联系技术支持。
400	CFW.00400009	泛域名防护规则超过数量限制	泛域名防护规则超过数量限制	请删除一些泛域名防护规则。
400	CFW.00400010	长连接不支持的协议	长连接不支持的协议	请确保规则协议属于TCP,UDP。
400	CFW.00400011	黑白名单信息重复	黑白名单信息重复	请确保添加不同的黑白名单。
400	CFW.00400012	东西向防护不存在,不能下发私网ip黑白名单	东西向防护不存在,不能下发私网ip黑白名单	请添加东西向防护。
400	CFW.00400013	黑白名单超过最大数量2000条	黑白名单超过最大数量2000条	请删除一些黑白名单。
400	CFW.00500002	时间间距错误	时间间距错误	请联系技术支持。
400	CFW.00500004	页码跳转间距过大	页码跳转间距过大	请联系技术支持。

状态码	错误码	错误信息	描述	处理措施
400	CFW.00600003	HTTP响应状态码异常	HTTP响应状态码异常	请联系技术支持。
400	CFW.00700001	关联ER不存在	关联ER不存在	请检查er是否存在。
400	CFW.00700002	关联VPC不存在	关联VPC不存在	请检查vpc是否存在。
400	CFW.00700003	关联子网网段冲突	关联子网网段冲突。	请确保创建子网与现有vpc下子网网段不重合。
400	CFW.00700004	子网创建失败	子网创建失败。	请联系技术支持。
400	CFW.00700007	ER创建VPC连接失败	ER创建VPC连接失败	请联系技术支持
400	CFW.00700012	修改路由失败	修改路由失败	请联系技术支持。
400	CFW.00700015	查询VPC配额信息失败	查询VPC配额信息失败。	请联系技术支持。
400	CFW.00700016	VPC可创建路由表配额不足	VPC可创建路由表配额不足	请删除vpc下已有路由表。
400	CFW.00800001	查询ETCD异常	查询ETCD异常	请联系技术支持。
400	CFW.00800002	删除ETCD异常	删除ETCD异常	请联系技术支持。
400	CFW.00800003	插入ETCD异常	插入ETCD异常	请联系技术支持。
400	CFW.00900016	导入任务进行中，请任务结束后再操作	导入任务进行中，请任务结束后再操作	请等待一些时间，直至导入任务结束。
400	CFW.00900020	地址组超过最大数量限制	地址组超过最大数量限制	请删除一些地址组。
400	CFW.00900030	整体服务数量到达限制	整体服务数量到达限制	请删除一些服务组成员。
400	CFW.01100008	集群扩容中不允许下发配置	集群扩容中不允许下发配置	请联系技术支持。

A.3 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": "xxxxxxx",
      "description": "",
      "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f897d6b99"
      },
      "id": "a4a5d4098fb4474fa22cd05f897d6b99",
      "enabled": true
    }
  ],
  "links": {
    "next": null,
    "previous": null,
    "self": "https://www.example.com/v3/projects"
  }
}
```

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 修订记录

发布日期	修改说明
2023-12-15	第三次正式发布： 新增以下接口： <ul style="list-style-type: none">• 创建防火墙• 获取CFW任务执行状态• 删除防火墙• 获取日志配置• 创建日志配置• 更新日志配置
2023-10-11	第二次正式发布。 新增以下接口： <ul style="list-style-type: none">• 查询防火墙列表• 批量更新规则动作• 查询规则标签• “域名解析及域名组管理”章节下全部接口。
2022-11-30	第一次正式发布。