

云防火墙

API 参考

文档版本 07
发布日期 2025-02-18



版权所有 © 华为云计算技术有限公司 2025。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 |
| 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 | 51 |
| 4.1.8 创建东西向防火墙 | 59 |
| 4.1.9 查询防护 VPC 数 | 65 |
| 4.1.10 标签创建接口 | 70 |
| 4.1.11 删除标签 | 75 |
| 4.2 EIP 管理 | 80 |
| 4.2.1 弹性 IP 开启关闭 | 80 |
| 4.2.2 查询 Eip 个数 | 90 |
| 4.2.3 弹性 IP 列表查询 | 95 |
| 4.2.4 查看 eip 告警白名单 | 101 |
| 4.2.5 修改 eip 自动防护开关 | 106 |
| 4.2.6 获取 eip 自动防护状态信息 | 112 |
| 4.3 ACL 规则管理 | 116 |
| 4.3.1 创建 ACL 规则 | 116 |
| 4.3.2 删除 ACL 规则 | 131 |
| 4.3.3 批量删除 Acl 规则 | 135 |
| 4.3.4 删除规则击中次数 | 141 |
| 4.3.5 更新 ACL 规则 | 146 |

| | |
|--------------------------|-----|
| 4.3.6 批量更新规则动作..... | 159 |
| 4.3.7 ACL 防护规则优先级设置..... | 164 |
| 4.3.8 查询防护规则..... | 170 |
| 4.3.9 查询规则标签..... | 180 |
| 4.3.10 获取规则击中次数..... | 185 |
| 4.3.11 查看 region 列表..... | 190 |
| 4.3.12 查看导入状态接口..... | 194 |
| 4.4 黑白名单管理..... | 198 |
| 4.4.1 创建黑白名单规则..... | 199 |
| 4.4.2 更新黑白名单列表..... | 205 |
| 4.4.3 删除黑白名单规则..... | 210 |
| 4.4.4 查询黑白名单列表..... | 215 |
| 4.5 地址组管理..... | 221 |
| 4.5.1 添加地址组..... | 221 |
| 4.5.2 添加地址组成员..... | 227 |
| 4.5.3 删除地址组..... | 233 |
| 4.5.4 删除地址组成员..... | 238 |
| 4.5.5 批量删除地址组成员..... | 240 |
| 4.5.6 更新地址组信息..... | 245 |
| 4.5.7 查询地址组列表..... | 251 |
| 4.5.8 查询地址组详细信息..... | 257 |
| 4.5.9 查询地址组成员..... | 262 |
| 4.5.10 更新对象配置描述..... | 267 |
| 4.6 服务组管理..... | 270 |
| 4.6.1 新建服务组..... | 270 |
| 4.6.2 新建服务成员..... | 275 |
| 4.6.3 删除服务组..... | 281 |
| 4.6.4 删除服务成员..... | 284 |
| 4.6.5 批量删除服务组成员信息..... | 289 |
| 4.6.6 修改服务组..... | 294 |
| 4.6.7 获取服务组列表..... | 299 |
| 4.6.8 查询服务组详情..... | 305 |
| 4.6.9 查询服务成员列表..... | 310 |
| 4.7 域名解析及域名组管理..... | 316 |
| 4.7.1 添加域名组..... | 316 |
| 4.7.2 删除域名组..... | 323 |
| 4.7.3 更新域名组..... | 327 |
| 4.7.4 更新 dns 服务器列表..... | 332 |
| 4.7.5 查询域名组列表..... | 338 |
| 4.7.6 查询 dns 服务器列表..... | 344 |
| 4.7.7 查询域名解析 ip 地址..... | 348 |
| 4.7.8 获取域名组下域名列表..... | 353 |

| | |
|--------------------------|------------|
| 4.7.9 添加域名列表..... | 358 |
| 4.7.10 删除域名列表..... | 364 |
| 4.7.11 查看域名组详情..... | 370 |
| 4.7.12 获取域名地址解析结果..... | 374 |
| 4.7.13 批量删除域名组..... | 378 |
| 4.8 IPS 管理..... | 384 |
| 4.8.1 查询 IPS 特性开关状态..... | 384 |
| 4.8.2 IPS 特性开关操作..... | 389 |
| 4.8.3 查询防护模式..... | 394 |
| 4.8.4 切换防护模式..... | 399 |
| 4.8.5 改变 ips 规则模式..... | 404 |
| 4.8.6 创建频率 ips 规则..... | 409 |
| 4.8.7 查询频率 ips 规则信息..... | 415 |
| 4.8.8 获取 ips 规则列表..... | 417 |
| 4.8.9 获取 ips 规则细节..... | 424 |
| 4.8.10 获取自定义 ips 规则..... | 428 |
| 4.9 日志管理..... | 435 |
| 4.9.1 获取日志配置..... | 435 |
| 4.9.2 创建日志配置..... | 438 |
| 4.9.3 更新日志配置..... | 443 |
| 4.9.4 查询流日志..... | 449 |
| 4.9.5 查询访问控制日志..... | 456 |
| 4.9.6 查询攻击日志..... | 463 |
| 4.10 抓包管理..... | 472 |
| 4.10.1 查询抓包任务..... | 472 |
| 4.10.2 创建抓包任务..... | 477 |
| 4.10.3 批量删除抓包任务..... | 484 |
| 4.10.4 获取抓包任务结果..... | 489 |
| 4.10.5 取消抓包任务..... | 494 |
| 4.11 反病毒管理..... | 498 |
| 4.11.1 查看反病毒开关..... | 499 |
| 4.11.2 修改反病毒开关..... | 503 |
| 4.11.3 获取防火墙反病毒规则信息..... | 508 |
| 4.11.4 修改反病毒规则..... | 513 |
| 4.12 告警配置管理..... | 518 |
| 4.12.1 获取告警配置信息..... | 518 |
| 4.12.2 修改告警配置接口..... | 523 |
| 4.13 标签管理..... | 528 |
| 4.13.1 查询标签信息..... | 529 |
| 4.13.2 查询资源标签信息..... | 533 |
| 4.13.3 保存资源标签接口..... | 537 |
| A 附录..... | 543 |

| | |
|--|-----|
| A.1 状态码..... | 543 |
| A.2 错误码..... | 543 |
| A.3 获取项目 ID..... | 550 |
| A.4 获取企业项目 ID..... | 551 |
| A.5 获取防火墙 ID..... | 551 |
| A.6 获取账号、IAM 用户、项目、用户组、区域、委托的名称和 ID..... | 551 |
| A.7 地域信息表..... | 553 |

1 使用前必读

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对防火墙实例进行相关操作，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](#)。

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](#)。

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。

基本概念

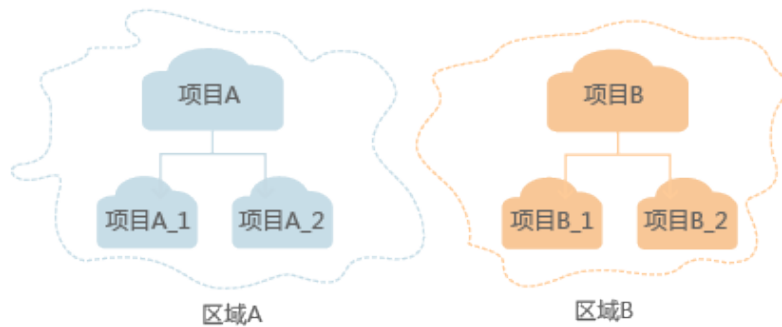
- 账号
用户注册时的账号，账号对其所拥有的资源及云服务具有完全的访问权限，可以重置用户密码、分配用户权限等。由于账号是付费主体，为了确保账号安全，建议您不要直接使用账号进行日常管理工作，而是通过创建用户来进行日常管理工作。
- 用户
由账号在IAM中创建的用户，是云服务的使用人员，具有身份凭证（密码和访问密钥）。
在[我的凭证](#)下，您可以查看账号ID和用户ID。通常在调用API的鉴权过程中，您需要用到账号、用户和密码等信息。
- 区域（Region）
从地理位置和网络时延维度划分，同一个Region内共享弹性计算、块存储、对象存储、弹性公网IP、镜像等公共服务。Region分为通用Region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，您可以完整地使用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，包括查询访问控制日志、查询攻击事件日志、查询流量日志等接口。 |
| 抓包管理 | 管理抓包任务，包括创建、查询、删除抓包任务等接口。 |
| 反病毒管理 | 管理反病毒功能，包括查看、修改反病毒开关，修改反病毒规则等接口。 |
| 告警配置管理 | 管理告警配置，包括获取告警配置信息、修改告警配置接口。 |
| 标签管理 | 管理标签功，包括查询资源标签信息等接口。 |

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}

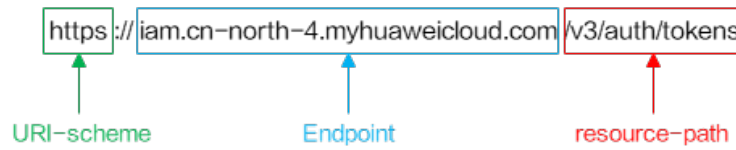
尽管请求URI包含在请求消息头中，但大多数语言或框架都要求您从请求消息中单独传递它，所以在此单独强调。

- **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
→ MIIVXQVJKoZIhvcNAQcCoIIYtJCCGEoCAQEgDTALBglghkgBZQMEAgEwgharBgkqhkiG9w0BBwGgghacBIIVmHsidG9rZW4iOnsiZmVhbnVlc19hdCI6IjwMTktMDitMTNUMC
fj3Kjs6YgKnpVNRbW2eZ5eb785Z0kqJACgkqO1wi4JlGzrpd18LGXK5tdf4q4qHCYb8P4NaY0NYejcAgz/VeFYtLWT1GSO0zxKZmlQHQj82HBqHdglZO9fuEbL5dMhdavj+33wEI
xHRCe9I87o+k9-
j+CMZSEB7bUGd5Uj6eRASXI1jipPEGA270g1FruooL6jaglfkNPQuFSOU8+uSsttVwRtNfsC+qTp22Rkd5MCqFGQ8LcuUxC3a+9CMBnOintWW7oeRUVhVpXk8pxiX1wTEboX-
RzT6MUbvpvGw-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 | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

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 | 是 | String | 用户Token。可通过 如何获取用户Token 获取。 |

| 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。 |

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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |
| tags | 否 | Array of tags objects | 服务资源标签列表，防火墙资源添加标签后，可根据键、值组合查询资源，同时可根据键、值组合进行话单合并统计。 |
| flavor | 是 | flavor object | 防火墙规格信息 |
| 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 | 否 | String | 资源标签键 |
| value | 否 | String | 资源标签值 |

表 4-5 flavor

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------------------|------|---------|---|
| version | 是 | String | 防火墙版本 “charge_mode”为 “prePaid”时，支持标准版、 专业版。 “charge_mode”为 “postPaid”时，仅支持专业 版。 Standard - 标准版 Professional - 专业版 |
| extend_eip_co unt | 否 | Integer | 扩展EIP数量，仅包周期场景下 生效，当用户需要在增加EIP使 用时需要使用此参数。 |
| extend_band width | 否 | Integer | 扩展带宽，步长为5，仅包周期 场景下生效，当用户需要在增加 带宽使用时需要使用此参数。 |
| extend_vpc_c ount | 否 | Integer | 扩展VPC数量，仅包周期场景下 生效，当用户需要增加VPC使用 时需要使用此参数。 |

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： <ul style="list-style-type: none">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。 |
| 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。 <ul style="list-style-type: none">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），自动支付失败会生成、但订单状态为“待支付”，等待客户手动支付（手动支付时，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） |

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 | 创建防火墙请求体 |

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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |
| tags | Array of tags objects | 服务资源标签列表，防火墙资源添加标签后，可根据键、值组合查询资源，同时可根据键、值组合进行话单合并统计。 |
| flavor | flavor object | 防火墙规格信息 |
| 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 | 资源标签键 |
| value | String | 资源标签值 |

表 4-10 flavor

| 参数 | 参数类型 | 描述 |
|---------|--------|---|
| version | String | 防火墙版本 “charge_mode”为“prePaid”时，支持标准版、专业版。 “charge_mode”为“postPaid”时，仅支持专业版。 Standard - 标准版 Professional - 专业版 |

| 参数 | 参数类型 | 描述 |
|------------------|---------|---|
| extend_eip_count | Integer | 扩展EIP数量，仅包周期场景下生效，当用户需要在增加EIP使用时需要使用此参数。 |
| extend_bandwidth | Integer | 扩展带宽，步长为5，仅包周期场景下生效，当用户需要在增加带宽使用时需要使用此参数。 |
| extend_vpc_count | Integer | 扩展VPC数量，仅包周期场景下生效，当用户需要增加VPC使用时需要使用此参数。 |

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： <ul style="list-style-type: none">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。 |

| 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。 <ul style="list-style-type: none">• 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），自动支付失败会生成、但订单状态为“待支付”，等待客户手动支付(手动支付时，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） |

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 |
| error_msg | String | 错误描述 |

请求示例

项目ID为124147da-5b08-471a-93d2-bc82acc290c6的客户开通标准版防火墙，防火墙名称为CFW-Test，企业项目ID为0，资源标签传入一个键值对，键为TagKey，值为TagValue，扩展防护EIP2000个，扩展EIP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。

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

购买防火墙成功返回信息。

```
{  
  "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"  
    }]  
  },  
  "job_id": "CS2403271050ZEM0L"  
}
```

状态码：400

错误返回信息。

```
{  
  "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

项目ID为124147da-5b08-471a-93d2-bc82acc290c6的客户开通标准版防火墙，防火墙名称为CFW-Test，企业项目ID为0，资源标签传入一个键值对，键为TagKey，值为TagValue，扩展防护EIP2000个，扩展EIP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 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");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .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

项目ID为124147da-5b08-471a-93d2-bc82acc290c6的客户开通标准版防火墙，防火墙名称为CFW-Test，企业项目ID为0，资源标签传入一个键值对，键为TagKey，值为TagValue，扩展防护EIP2000个，扩展EIP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。

```
# coding: utf-8

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    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

项目ID为124147da-5b08-471a-93d2-bc82acc290c6的客户开通标准版防火墙，防火墙名称为CFW-Test，企业项目ID为0，资源标签传入一个键值对，键为TagKey，值为TagValue，扩展防护EIP2000个，扩展EIP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")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        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,
    }
    extendEipCountFlavor := int32(2000)
    extendBandwidthFlavor := int32(5000)
    extendVpcCountFlavor := int32(100)
    flavorBody := &model.CreateFirewallReqFlavor{
        Version: model.GetCreateFirewallReqFlavorVersionEnum().STANDARD,
        ExtendEipCount: &extendEipCountFlavor,
        ExtendBandwidth: &extendBandwidthFlavor,
        ExtendVpcCount: &extendVpcCountFlavor,
    }
    keyTags := "TagKey"
    valueTags := "TagVal"
    var listTagsBody = []model.CreateFirewallReqTags{
        {
            Key: &keyTags,
            Value: &valueTags,
        },
    }
    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 | 购买防火墙成功返回信息。 |
| 400 | 错误返回信息。 |

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, 可以从调API处获取，也可以从控制台获取。 项目ID获取方式 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|--------|------|--------|--|
| job_id | 是 | String | 创建按需防火墙返回的任务ID，可通过调用 创建防火墙接口 返回值获得。返回值中job_id即为此处的job_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 | 是 | String | 用户Token。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-15 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|---|--------------|
| data | GetCreateFirewallJobResponseData object | 创建按需防火墙返回数据。 |

表 4-16 GetCreateFirewallJobResponseData

| 参数 | 参数类型 | 描述 |
|------------|--------|---|
| id | String | 创建按需防火墙任务ID |
| status | String | 任务执行状态，用于向客户显示创建防火墙是否成功。 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。 |
| 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。 |

| 参数 | 参数类型 | 描述 |
|----------|--------|---|
| 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下的创建按需防火墙的job ID为的f588ce71-e26c-400d-8981-f854355f6849的任务情况。

```
https://{Endpoint}/v3/09bb24e6fe80d23d2fa2c010b53b418c/jobs/f588ce71-e26c-400d-8981-f854355f6849
```

响应示例

状态码：200

获取创建按需防火墙任务情况接口返回体

```
{
  "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");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
```

```
.withProjectId(projectId)
.withAk(ak)
.withSk(sk);

CfwClient client = CfwClient.newBuilder()
    .withCredential(auth)
    .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
    .build();
ListJobRequest request = new ListJobRequest();
request.withJobId("{job_id}");
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.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListJobRequest()
        request.job_id = "{job_id}"
        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")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := cfw.NewCfwClient(  
        cfw.CfwClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.ListJobRequest{  
        request.JobId = "{job_id}"  
    }  
    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 | 获取创建按需防火墙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 | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |
| resource_id | 是 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

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 | 是 | String | 用户Token。可通过 如何获取用户Token 获取。 |

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 | 删除防火墙时生成的任务的job_id |

请求示例

删除项目id为06217ebc876e427a80a2c05d51264ab1下的按需防火墙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.cfww.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfww.v1.*;
import com.huaweicloud.sdk.cfww.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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteFirewallRequest request = new DeleteFirewallRequest();
        request.withResourceId("{resource_id}");
        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
```

```
import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DeleteFirewallRequest()
        request.resource_id = "{resource_id}"
        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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

```
request.ResourceId = "{resource_id}"
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 | 项目ID, 可以从调API处获取，也可以从控制台获取。 项目ID获取方式 |

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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

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。可通过 如何获取用户Token 获取。 |

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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |
| key_word | 否 | String | 查询关键字，可为防火墙id或防火墙名称的一部分。可通过 防火墙ID获取方式 获取 |
| tags | 否 | Array of TagInfo objects | 标签列表，可通过查询标签服务查询标签接口获得，返回值即为标签列表 |
| 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 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|--------|------|------------------|-------|
| 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 | 是否支持企业项目，true表示是，false表示不是 |
| has_ndr | Boolean | 是否存在ndr，true表示是，false表示不是，NDR为原旁路版防火墙，现已停止售卖。 |
| is_support_postpaid | Boolean | 是否支持按需购买，true表示是，false表示不是 |
| is_support_basic_version | Boolean | 是否支持基础版，true表示是，false表示不是 |
| is_support_buy_professional | Boolean | 是否支持购买专业版，true表示是，false表示不是 |
| 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 | 防火墙总数量 |
| 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记防火墙由系统自动生成的id。 |
| resource_id | String | 资源id, 与防火墙实例id fw_instance_id 相同 |
| name | String | 创建防火墙时的时间戳 |
| 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, 包含主备(0)和集群(1)两种方式, 主备模式包含四个节点, 2个主节点构成集群, 剩余两个节点为主节点的备节点, 集群模式仅拉起两个节点作为集群。 |
| charge_mode | Integer | 计费模式 0: 包年/包月 1: 按需 |
| service_type | Integer | 防火墙防护类型, 目前仅支持0, 互联网防护。 |
| engine_type | Integer | 引擎类型, 0: 自研引擎 1: 山石引擎 3: 天融信引擎 |
| 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, 标签键值map转化的json字符串, 如{"key":"value"} |

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, 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。 |
| eip_count | Integer | eip数量 |

| 参数 | 参数类型 | 描述 |
|---------------------|---------|---|
| vpc_count | Integer | vpc数量 |
| bandwidth | Integer | 带宽, 单位为mbps |
| log_storage | Integer | 日志存储, 单位为byte |
| default_bandwidth | Integer | 默认防火墙带宽, 单位为mbps, 标准版为10, 专业版为50, 按需专业版为200 |
| default_eip_count | Integer | 默认eip数, 标准版为20, 专业版为50, 按需专业版为1000 |
| default_log_storage | Integer | 默认日志存储, 单位为byte, 默认为0 |
| default_vpc_count | Integer | 默认vpc数, 标准版为0, 专业版为2, 按需专业版为5 |

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" : 1,
    "offset" : 0,
    "project_id" : "14181c1245cf4fd786824efe1e2b9388",
    "records" : [ {
      "fw_instance_id" : "ebf891cd-2163-48a0-9963-6309f99dd3c4",
      "resource_id" : "ebf891cd-2163-48a0-9963-6309f99dd3c4",
      "name" : "1709176078374",
      "fw_instance_name" : "test",
      "enterprise_project_id" : "default",
      "tags" : "{\"key_test3\":\"value_test3\"}",
      "ha_type" : 0,
      "charge_mode" : 0,
      "service_type" : 0,
      "engine_type" : 1,
      "flavor" : {
        "version" : 1,
        "eip_count" : 50,
        "vpc_count" : 6,
        "bandwidth" : 50,
        "log_storage" : 0,

```

```
"default_eip_count" : 50,
"default_vpc_count" : 2,
"default_bandwidth" : 50,
"default_log_storage" : 0
},
"status" : 2
}],
"total" : 18
},
"has_ndr" : false,
"is_support_basic_version" : true,
"is_support_buy_professional" : false,
"is_support_postpaid" : true,
"user_support_eps" : 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");
        String projectId = "{project_id}";

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

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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListFirewallListRequest()
        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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListFirewallListRequest{}
    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 | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

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。可通过 如何获取用户Token 获取。 |

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, 通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 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。此处仅取type为1的防护对象id, 可通过 data.records.protect_objects.type (.表示各对象之间层级的区分) 获得。 |
| status | 是 | Integer | 防护状态: 0 开启, 1 关闭 |

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 | ChangeEastWestFirewallStatusResponseData object | 改变东西向防护返回值data |

表 4-34 ChangeEastWestFirewallStatusResponseData

| 参数 | 参数类型 | 描述 |
|----|--------|---|
| id | String | 东西向防护对象ID, 可通过调用 查询防火墙实例接口 获得, 通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 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。此处仅取type为1的防护对象id, type可通过 data.records.protect_objects.type (.表示各对象之间层级的区分) 获得。 |

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 |
| error_msg | String | 错误描述 |

请求示例

将项目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

将项目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");
        String projectId = "{project_id}";

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

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

将项目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

import os
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")
    projectId = "{project_id}"
```

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

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

try:
    request = ChangeEastWestFirewallStatusRequest()
    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

将项目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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ChangeEastWestFirewallStatusRequest{
        request.Body = &model.ChangeProtectStatusRequestBody{
            Status: model.GetChangeProtectStatusRequestBodyEnum().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 | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

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，目前仅支持0互联网防护 |
| enterprise_project_id | 否 | String | 企业项目ID，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |
| 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 ，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。可通过 如何获取用户Token 获取。 |

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 | 防火墙总数 |
| 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 ，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 | String | 防火墙名称 |
| ha_type | Integer | 集群类型，包含主备（0）和集群（1）两种方式，主备模式包含四个节点，2个主节点构成集群，剩余两个节点为主节点的备节点，集群模式仅拉起两个节点作为集群。 |
| charge_mode | Integer | 计费模式 0：包年/包月 1：按需 |
| service_type | Integer | 防火墙防护类型，目前仅支持0，互联网防护 |
| engine_type | Integer | 引擎类型，0：自研引擎 1：山石引擎 3：天融信引擎 |
| 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 |
| is_old_firewall_instance | Boolean | 是否为旧引擎, true表示是, false表示不是 |
| is_available_obs | Boolean | 是否支持obs, true表示是, false表示不是 |
| is_support_threat_tags | Boolean | 是否支持威胁情报标签, true表示是, false表示不是 |
| 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| resource_id | String | 防火墙资源id, 同fw_instance_id |
| support_url_filtering | Boolean | 是否支持url过滤, true表示是, false表示不是 |
| tags | String | 标签列表, 标签键值map转化的json字符串, 如{"key":"value"} |

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, 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。 |
| eip_count | Integer | eip数量 |

| 参数 | 参数类型 | 描述 |
|---------------------|---------|---|
| vpc_count | Integer | vpc数量 |
| bandwidth | Integer | 带宽, 单位为mbps |
| log_storage | Integer | 日志存储, 单位为byte |
| default_bandwidth | Integer | 默认防火墙带宽, 单位为mbps, 标准版为10, 专业版为50, 按需专业版为200 |
| default_eip_count | Integer | 默认eip数, 标准版为20, 专业版为50, 按需专业版为1000 |
| default_log_storage | Integer | 默认日志存储, 单位为byte, 默认为0 |
| default_vpc_count | Integer | 默认vpc数, 标准版为0, 专业版为2, 按需专业版为5 |

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。 |
| object_name | String | 防护对象名称 |
| type | Integer | 防护对象类型: 0 南北向, 1 东西向防护对象类型 |

表 4-44 FirewallInstanceResource

| 参数 | 参数类型 | 描述 |
|--------------------|--------|--|
| resource_id | String | 资源id, 包括防火墙资源id, 带宽资源id, eip资源id, vpc资源id, cbc回调后返回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.cfw2、EIP:hws.resource.type.cfw.exp.eip3、带宽:hws.resource.type.cfw.exp.bandwidth4、VPC:hws.resource.type.cfw.exp |

| 参数 | 参数类型 | 描述 |
|--------------------------|---------|--|
| resource_spec_code | String | 库存单位码，包括：1、防火墙标准版cfw.standard 2、防火墙专业版cfw.professional 3、eip标准版cfw.expack.eip.standard 4、eip专业版cfw.expack.eip.professional 5、带宽基础版cfw.expack.bandwidth.standard 6、带宽专业版cfw.expack.bandwidth.professional 7、vpc专业版cfw.expack.vpc.professional |
| 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,
  "version": 1,
  "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": "0acd5c7-1178-4bea-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");
        String projectId = "{project_id}";

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

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

```
        .build();
ListFirewallDetailRequest request = new ListFirewallDetailRequest();
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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListFirewallDetailRequest()
        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")
projectId := "{project_id}"

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

client := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListFirewallDetailRequest{}
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 | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 是 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

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。可通过 如何获取用户Token 获取。 |

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, 通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 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。此处仅取type为1的防护对象id, 可通过 data.records.protect_objects.type (.表示各对象之间层级的区分) 获得。 |
| project_id | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |
| status | Integer | 防护状态: 0 已开启防护, 1 未开启防护 |
| firewall_associated_subnets | Array of SubnetInfo objects | 云防火墙关联子网信息 |
| er | EriInstance object | 出方向关联企业路由器信息 |
| inspection_vpc | VpcDetail object | 引流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, 值为er |

表 4-50 SubnetInfo

| 参数 | 参数类型 | 描述 |
|-------------------|---------|---|
| availability_zone | String | 子网所在的可用区标识，从终端节点获取，参考 终端节点 |
| cidr | String | 功能说明：虚拟私有云下可用子网的范围 取值范围： 10.0.0.0/8~24 172.16.0.0/12~24 192.168.0.0/16~24 不指定cidr时，默认值为空 约束：必须是cidr格式，例如：192.168.0.0/16 |
| name | String | 子网名称 |
| id | String | 子网id |
| gateway_ip | String | 子网的网关，取值范围为子网网段cidr中的ip地址 |
| vpc_id | String | 创建vpc产生的uuid |
| 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 ID，创建ER时产生的ID |
| name | String | ER名称 |
| project_id | String | 项目ID，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 |
| attachment_id | String | 企业路由器连接id，该连接用于连接防火墙和企业路由器，此字段可在通过id在ER界面查询指定er后在管理连接界面查询连接了解连接具体情况。 |

表 4-52 VpcDetail

| 参数 | 参数类型 | 描述 |
|----|--------|------------------|
| id | String | 创建引流VPC产生的随机UUID |

| 参数 | 参数类型 | 描述 |
|------|--------|---|
| name | String | 引流VPC名称 |
| cidr | String | 功能说明：虚拟私有云下可用子网的范围 取值范围： 10.0.0.0/8~24 172.16.0.0/12~24 192.168.0.0/16~24 不指定cidr时，默认值为空 约束：必须是cidr格式，例如:192.168.0.0/16 |

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， 2 VPN， 3 PEERING |
| protected_resource_name | String | 防护资源名称 |
| protected_resource_id | String | 防护资源id |
| protected_resource_nat_name | String | 防护资源nat网关名称，专业版防火墙支持NAT规则，此字段表示防护连接的NAT的名称。 |
| protected_resource_nat_id | String | 防护资源nat网关id，专业版防火墙支持NAT规则，此字段表示防护连接的NAT的id。 |
| protected_resource_project_id | String | 防火墙支持跨账户防护，此处为防护资源租户id |
| protected_resource_mode | String | 防护资源模式，为er |
| status | Integer | 防护资源的防护状态，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 |
| error_msg | String | 错误描述 |

请求示例

获取项目id为09bb24e6f280d23d0f9fc0104b901480下的防火墙id为80e0f2df-24fd-49c2-8398-11f9a0299b3e的东西向防火墙信息。

```
https://{Endpoint}/v1/09bb24e6f280d23d0f9fc0104b901480/firewall/east-west?limit=10&offset=0&fw_instance_id=80e0f2df-24fd-49c2-8398-11f9a0299b3e
```

响应示例

状态码：200

东西向防火墙信息响应

```
{
  "data": {
    "er": {
      "id": "f0f5275a-40aa-4d1e-ac78-2550f7818d43",
      "name": "er",
      "project_id": "5c69cf330cda42369cbd726ee1bc5e76",
      "attachment_id": "e978c028-0ccf-4168-a015-0094ac631e88"
    },
    "inspection_vpc": {
      "cidr": "10.13.13.0/24",
      "id": "3baa2248-03d4-47ac-ae6a-ec6f9a30ed80",
      "name": "inspection-vpc"
    },
    "limit": 50,
    "mode": "er",
    "object_id": "6cc31075-0b03-461d-aa3a-73c4765bf9ac",
    "offset": 0,
    "project_id": "5c69cf330cda42369cbd726ee1bc5e76",
    "protect_infos": [ {
      "protected_resource_id": "09b26b83-4b7c-499f-805b-50820900cd9f",
      "protected_resource_mode": "er",
      "protected_resource_name": "vpc-4",
      "protected_resource_project_id": "5c69cf330cda42369cbd726ee1bc5e76",
      "protected_resource_type": 0,
      "status": 0
    }, {
      "protected_resource_id": "2956f917-af18-44a2-a0f2-f592633d4431",
      "protected_resource_mode": "er",
      "protected_resource_name": "vpc-t3",
      "protected_resource_project_id": "5c69cf330cda42369cbd726ee1bc5e76",
      "protected_resource_type": 0,
      "status": 0
    }, {
      "protected_resource_id": "69d4c81f-80bd-4c28-9074-f26b400b3a34",
      "protected_resource_mode": "er",
      "protected_resource_name": "vpc-t2",
      "protected_resource_project_id": "5c69cf330cda42369cbd726ee1bc5e76",
      "protected_resource_type": 0,
      "status": 0
    }, {
      "protected_resource_id": "e1b18d3e-8f7d-4c4e-a326-37db7bd89a3a",
      "protected_resource_mode": "er",
      "protected_resource_name": "vpc-t1",
      "protected_resource_project_id": "5c69cf330cda42369cbd726ee1bc5e76",
      "protected_resource_type": 0,
      "status": 0
    }, {
      "protected_resource_id": "f91c1747-1325-4b39-9d58-7fac0bd2786f",
      "protected_resource_mode": "er",
      "protected_resource_name": "vpc-5",
      "protected_resource_project_id": "5c69cf330cda42369cbd726ee1bc5e76",
      "protected_resource_type": 0,
      "status": 0
    }
  ], {
  }
}
```

```
"protected_resource_id" : "bd556a7d-447e-49eb-be17-f93af09769f5",
"protected_resource_mode" : "er",
"protected_resource_name" : "inspct-a",
"protected_resource_project_id" : "5c69cf330cda42369cbd726ee1bc5e76",
"protected_resource_type" : 0,
"status" : 1
}],
"status" : 0,
"total" : 6
}
}
```

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");
        String projectId = "{project_id}";

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

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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListEastWestFirewallRequest()
        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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```



```
Build()  
  
request := &model.ListEastWestFirewallRequest{}  
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 |
| 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 | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 是 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

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。可通过 如何获取用户Token 获取。 |

表 4-58 请求 Body 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|-----------------|------|--------|--|
| er_id | 是 | String | 出方向关联ER ID,可通过ER服务查询企业路由器列表接口获得,返回值中instances.id即为erid (.表示各对象之间层级的区分) |
| inspection_cidr | 是 | String | 创建引流VPC时使用的网段 |
| 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 | CreateEWFirewallResp object | 创建东西向防火墙返回值 |

表 4-60 CreateEWFirewallResp

| 参数 | 参数类型 | 描述 |
|----------------|---------------------------------------|------------------------|
| id | String | 东西向防护id, 对应object_id字段 |
| er | ER object | ER信息 |
| inspection_vpc | CreateEWFirewallInspectVpcResp object | 引流vpc信息 |

表 4-61 ER

| 参数 | 参数类型 | 描述 |
|--------------|--------|--|
| er_id | String | ER ID, 创建东西向防护引用的 ID |
| er_attach_id | String | 企业路由器连接id, 该连接用于连接防火墙和企业路由器, 此字段可在通过id在ER界面查询指定er后在管理连接界面查询连接了解连接具体情况。 |

表 4-62 CreateEWFirewallInspectVpcResp

| 参数 | 参数类型 | 描述 |
|------------|------------------|-----------------|
| vpc_id | String | 引流VPC的ID |
| subnet_ids | Array of strings | 创建的引流VPC的子网ID列表 |

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" : "acc86ca7-818b-4c3d-8a9a-3915a2b21651",  
    "er" : {  
      "er_id" : "f0f5275a-40aa-4d1e-ac78-2550f7818d43",  
      "er_attach_id" : "bd62ddd3-5e20-482b-aefa-9e2940e2b1a9"  
    },  
    "inspersion_vpc" : {  
      "vpc_id" : "00672633-0466-4c35-99ef-5e3f5c813a4b",  
      "subnet_ids" : [ "294682a0-1e85-45f5-92c8-e52bee09c204", "a86277bb-35d5-4442-  
bc0b-2e9d4e6a9080", "95829240-14e0-47e6-b9e7-2ac228e7b00f" ]  
    }  
  }  
}
```

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");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        CfwClient client = CfwClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
```

```
        .build();
        CreateEastWestFirewallRequest request = new CreateEastWestFirewallRequest();
        CreateEastWestFirewallRequestBody body = new CreateEastWestFirewallRequestBody();
        body.withMode("er");
        body.withInspectionCidr("10.1.0.0/24");
        body.withErId("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.getStatusCode());
            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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateEastWestFirewallRequest()
        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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateEastWestFirewallRequest{}
    request.Body = &model.CreateEastWestFirewallRequestBody{
        Mode: "er",
        InspectionCidr: "10.1.0.0/24",
        ErId: "e0b22a23-02cf-4092-ace9-34b39e10dc77",
    }
    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 4-63 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-64 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, 通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 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。此处仅取type为1的防护对象id, 可通过 data.records.protect_objects.type (.表示各对象之间层级的区分) 获得。 |
| enterprise_project_id | 否 | String | 企业项目ID, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

请求参数

表 4-65 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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-66 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--------------------------------------|----------|
| data | VPCProtectsVo object | 防护vpc返回值 |

表 4-67 VPCProtectsVo

| 参数 | 参数类型 | 描述 |
|--------------------|--|---|
| total | Integer | 总防护VPC数 |
| self_total | Integer | 防火墙东西向防护可进行跨账号防护VPC, self_total表示本项目防护VPC总数。 |
| other_total | Integer | 防火墙东西向防护可进行跨账号防护VPC, other_total表示其他项目防护VPC数 |
| protect_vpcs | Array of VpcAttachmentDetail objects | 防火墙东西向防护可进行跨账号防护VPC, protect_vpcs指的是总体防护VPC列表 |
| self_protect_vpcs | Array of VpcAttachmentDetail objects | 防火墙东西向防护可进行跨账号防护VPC, self_protect_vpcs指的是本项目防护VPC列表 |
| other_protect_vpcs | Array of VpcAttachmentDetail objects | 防火墙东西向防护可进行跨账号防护VPC, other_protect_vpcs指的是其他项目防护VPC列表 |
| total_assets | Integer | 租户的所有VPC资产数量 |

表 4-68 VpcAttachmentDetail

| 参数 | 参数类型 | 描述 |
|--------|--------|------------------|
| vpc_id | String | 东西向防护添加的防护vpc的id |

状态码: 500

表 4-69 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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");
        String projectId = "{project_id}";

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

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

```
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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListProtectedVpcsRequest()
        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")
projectId := "{project_id}"

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

client := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListProtectedVpcsRequest{}
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](#)。

4.1.10 标签创建接口

功能介绍

创建标签

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/cfw-cfw/{fw_instance_id}/tags/create

表 4-70 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |
| fw_instance_id | 是 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

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。可通过 如何获取用户Token 获取。 |

表 4-72 请求 Body 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------|------|--|-----------|
| tags | 否 | Array of CreateTag objects | 创建防火墙标签列表 |

表 4-73 CreateTag

| 参数 | 是否必选 | 参数类型 | 描述 |
|-------|------|--------|-----|
| key | 否 | String | 标签键 |
| value | 否 | String | 标签值 |

响应参数

状态码：200

表 4-74 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|----|--------|----|
| - | String | |

请求示例

给项目id为0b2179bbe180d3762fb0c01a2d5725c7，防火墙id为5e7eba7f-5de4-4ce9-8f60-11330dfc6565添加标签，标签的键为keytest，值为valuetest。

```
https://{Endpoint}/v2/0b2179bbe180d3762fb0c01a2d5725c7/cfw-cfw/5e7eba7f-5de4-4ce9-8f60-11330dfc6565/tags/create
```

```
{
  "tags": [ {
    "key": "keytest",
    "value": "valuetest"
  } ]
}
```

响应示例

状态码：200

OK

```
{}
```

SDK 代码示例

SDK代码示例如下。

Java

给项目id为0b2179bbe180d3762fb0c01a2d5725c7，防火墙id为5e7eba7f-5de4-4ce9-8f60-11330dfc6565添加标签，标签的键为keytest，值为valuetest。

```
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 CreateTagSolution {

    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");
String projectId = "{project_id}";

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

CfwClient client = CfwClient.newBuilder()
    .withCredential(auth)
    .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
    .build();
CreateTagRequest request = new CreateTagRequest();
request.withFwInstanceId("{fw_instance_id}");
CreateTagsDto body = new CreateTagsDto();
List<CreateTag> listbodyTags = new ArrayList<>();
listbodyTags.add(
    new CreateTag()
        .withKey("keytest")
        .withValue("valuetest")
);
body.withTags(listbodyTags);
request.withBody(body);
try {
    CreateTagResponse response = client.createTag(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0b2179bbe180d3762fb0c01a2d5725c7，防火墙id为5e7eba7f-5de4-4ce9-8f60-11330dfc6565添加标签，标签的键为keytest，值为valuetest。

```
# coding: utf-8

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

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

```
client = CfwClient.new_builder() \  
  .with_credentials(credentials) \  
  .with_region(CfwRegion.value_of("<YOUR REGION>")) \  
  .build()  
  
try:  
  request = CreateTagRequest()  
  request.fw_instance_id = "{fw_instance_id}"  
  listTagsbody = [  
    CreateTag(  
      key="keytest",  
      value="valuetest"  
    )  
  ]  
  request.body = CreateTagsDto(  
    tags=listTagsbody  
  )  
  response = client.create_tag(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0b2179bbe180d3762fb0c01a2d5725c7，防火墙id为5e7eba7f-5de4-4ce9-8f60-11330dfc6565添加标签，标签的键为keytest，值为valuetest。

```
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")  
  projectId := "{project_id}"  
  
  auth := basic.NewCredentialsBuilder().  
    WithAk(ak).  
    WithSk(sk).  
    WithProjectId(projectId).  
    Build()  
  
  client := cfw.NewCfwClient(  
    cfw.CfwClientBuilder().  
      WithRegion(region.ValueOf("<YOUR REGION>")).  
      WithCredential(auth).  
      Build())  
  
  request := &model.CreateTagRequest{}  
  request.FwInstanceId = "{fw_instance_id}"  
  keyTags:= "keytest"  
  valueTags:= "valuetest"
```



```
var listTagsbody = []model.CreateTag{
    {
        Key: &keyTags,
        Value: &valueTags,
    },
}
request.Body = &model.CreateTagsDto{
    Tags: &listTagsbody,
}
response, err := client.CreateTag(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

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

状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

错误码

请参见[错误码](#)。

4.1.11 删除标签

功能介绍

删除标签

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/cfw-cfw/{fw_instance_id}/tags/delete

表 4-75 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|--|
| project_id | 是 | String | 项目ID, 可以从调API处获取，也可以从控制台获取。 项目ID获取方式 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------------|------|--------|---|
| fw_instance_id | 是 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

请求参数

表 4-76 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。可通过 如何获取用户Token 获取。 |

表 4-77 请求 Body 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------|------|--|---------|
| tags | 否 | Array of ResourceTag objects | 防火墙标签列表 |

表 4-78 ResourceTag

| 参数 | 是否必选 | 参数类型 | 描述 |
|-------------|------|--------|--------|
| key | 否 | String | 标签键 |
| value | 否 | String | 标签值 |
| update_time | 否 | String | 标签更新时间 |

响应参数

状态码：200

表 4-79 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|----|--------|----|
| - | String | |

请求示例

删除项目id为0b2179bbe180d3762fb0c01a2d5725c7，防火墙id为5e7eba7f-5de4-4ce9-8f60-11330dfc6565下的防火墙标签，标签key值为keytest，value为valuetest

```
https://{Endpoint}/v2/408972e72dcd4c1a9b033e955802a36b/cfw-cfw/5e7eba7f-5de4-4ce9-8f60-11330dfc6565/tags/delete

{
  "tags" : [ {
    "key" : "keytest",
    "value" : "valuetest"
  } ]
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

删除项目id为0b2179bbe180d3762fb0c01a2d5725c7，防火墙id为5e7eba7f-5de4-4ce9-8f60-11330dfc6565下的防火墙标签，标签key值为keytest，value为valuetest

```
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 DeleteTagSolution {

    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");
        String projectId = "{project_id}";

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

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

```
        .build();
DeleteTagRequest request = new DeleteTagRequest();
request.withFwInstanceId("{fw_instance_id}");
DeleteTagsDto body = new DeleteTagsDto();
List<ResourceTag> listbodyTags = new ArrayList<>();
listbodyTags.add(
    new ResourceTag()
        .withKey("keytest")
        .withValue("valuetest")
);
body.withTags(listbodyTags);
request.withBody(body);
try {
    DeleteTagResponse response = client.deleteTag(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0b2179bbe180d3762fb0c01a2d5725c7，防火墙id为5e7eba7f-5de4-4ce9-8f60-11330dfc6565下的防火墙标签，标签key值为keytest，value为valuetest

```
# coding: utf-8

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DeleteTagRequest()
        request.fw_instance_id = "{fw_instance_id}"
        listTagsbody = [
            ResourceTag(
                key="keytest",
                value="valuetest"
            )
        ]
```

```
]
request.body = DeleteTagsDto(
    tags=listTagsbody
)
response = client.delete_tag(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0b2179bbe180d3762fb0c01a2d5725c7，防火墙id为5e7eba7f-5de4-4ce9-8f60-11330dfc6565下的防火墙标签，标签key值为keytest，value为valuetest

```
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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteTagRequest{
        request.FwInstanceId = "{fw_instance_id}"
        keyTags:= "keytest"
        valueTags:= "valuetest"
        var listTagsbody = []model.ResourceTag{
            {
                Key: &keyTags,
                Value: &valueTags,
            },
        }
        request.Body = &model.DeleteTagsDto{
            Tags: &listTagsbody,
        }
        response, err := client.DeleteTag(request)
        if err == nil {
            fmt.Printf("%+v\n", response)
        } else {
            fmt.Println(err)
        }
    }
```

```
}  
}
```

更多

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

状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

错误码

请参见[错误码](#)。

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 4-80 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|--|
| project_id | 是 | String | 项目ID, 可以从调API处获取，也可以从控制台获取。 项目ID获取方式 |

表 4-81 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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id，可通过 防火墙ID获取方式 获取 |

请求参数

表 4-82 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。可通过 如何获取用户Token 获取。 |

表 4-83 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，通过返回值中的data.records.protect_objects.object_id（.表示各对象之间层级的区分）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。此处仅取type为0的防护对象id，可通过data.records.protect_objects.type（.表示各对象之间层级的区分）获得。 |
| status | 是 | Integer | EIP切换的状态，0表示防护中，1表示未防护 |
| ip_infos | 是 | Array of ip_infos objects | 切换防护状态的EIP信息列表 |

表 4-84 ip_infos

| 参数 | 是否必选 | 参数类型 | 描述 |
|-------------|------|--------|---|
| id | 否 | String | 弹性公网IP ID，可通过调用弹性IP列表查询接口获得，通过返回值中的data.records.id（.表示各对象之间层级的区分）获得。 |
| public_ip | 否 | String | 弹性公网IP IPv4地址，可通过调用弹性IP列表查询接口获得，通过返回值中的data.records.public_ip（.表示各对象之间层级的区分）获得。 |
| public_ipv6 | 否 | String | 弹性公网IP IPv6地址，可通过调用弹性IP列表查询接口获得，通过返回值中的data.records.public_ipv6（.表示各对象之间层级的区分）获得。 |

响应参数

状态码：200

表 4-85 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 4-86 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，通过返回值中的data.records.protect_objects.object_id（.表示各对象之间层级的区分）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。此处仅取type为0的防护对象id，可通过data.records.protect_objects.type（.表示各对象之间层级的区分）获得 |
| fail_eip_id_list | Array of strings | 修改eip防护状态失败状态列表，状态包括成功"successful"，失败“fail” |

| 参数 | 参数类型 | 描述 |
|---------------|--|---|
| fail_eip_list | Array of FailedEipInfo objects | 修改eip防护状态失败信息列表 |
| id | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

表 4-87 FailedEipInfo

| 参数 | 参数类型 | 描述 |
|---------------|--------|---------------|
| id | String | 修改状态失败的eipId。 |
| error_message | String | 修改状态失败的错误码。 |

状态码: 400

表 4-88 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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IPid为4a589be0-b40a-4694-94ff-c0710af9a0a2的 EIP (1.2.3.4) 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IPid为4a589be0-b40a-4694-94ff-c0710af9a0a2的 EIP (1.2.3.4) 流量防护

```
/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" : [],
    "fail_eip_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IPid为4a589be0-b40a-4694-94ff-c0710af9a0a2的 EIP（1.2.3.4）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");
        String projectId = "{project_id}";

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

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

```
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(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IPid为4a589be0-b40a-4694-94ff-c0710af9a0a2的 EIP（1.2.3.4）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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ChangeEipStatusRequest request = new ChangeEipStatusRequest();
        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IPid为4a589be0-b40a-4694-94ff-c0710af9a0a2的 EIP（1.2.3.4）流量防护

```
# coding: utf-8

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ChangeEipStatusRequest()
        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IPid为4a589be0-b40a-4694-94ff-c0710af9a0a2的 EIP (1.2.3.4) 流量防护

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

```
import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ChangeEipStatusRequest()
        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IPid为4a589be0-b40a-4694-94ff-c0710af9a0a2的 EIP (1.2.3.4) 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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ChangeEipStatusRequest{}
    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IPid为4a589be0-b40a-4694-94ff-c0710af9a0a2的 EIP (1.2.3.4) 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")
projectId := "{project_id}"

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

client := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ChangeEipStatusRequest{}
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 4-89 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|--|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |
| 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, 通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 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。此处仅取type为0的防护对象id, 可通过 data.records.protect_objects.type (.表示各对象之间层级的区分) 获得。 |

表 4-90 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------------|------|--------|---|
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

请求参数

表 4-91 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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-92 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|---|---------|
| data | EipCountRespData object | Eip计数数据 |

表 4-93 EipCountRespData

| 参数 | 参数类型 | 描述 |
|--------------------|---------|-----------------|
| eip_total | Integer | 总体EIP数 |
| eip_protected | Integer | 该账号下所有墙防护EIP总数量 |
| eip_protected_self | Integer | 当前防火墙防护EIP数量 |

状态码：400

表 4-94 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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,
  }
}
```

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");
        String projectId = "{project_id}";

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

```
CfwClient client = CfwClient.newBuilder()
    .withCredential(auth)
    .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
    .build();
ListEipCountRequest request = new ListEipCountRequest();
request withObjectId("{object_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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListEipCountRequest()
        request.object_id = "{object_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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListEipCountRequest{}
    request.ObjectId = "{object_id}"
    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 4-95 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-96 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, 通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 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。此处仅取type为0的防护对象id, 可通过 data.records.protect_objects.type (.表示各对象之间层级的区分) 获得。 |
| key_word | 否 | String | 查询防护EIP列表关键字, 可选用弹性公网ID和弹性公网IP |
| status | 否 | String | 防护状态 null-全部 0-开启防护 1-关闭防护 |
| sync | 否 | Integer | 是否同步租户EIP数据 0-不同步 1-同步 |
| limit | 是 | Integer | 每页显示个数, 范围为1-1024 |

| 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 |
| enterprise_project_id | 否 | String | 企业项目ID，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |
| device_key | 否 | String | 设备关键字，是eip绑定的资产的名称或id |
| address_type | 否 | Integer | 地址类型0 ipv4，1 ipv6 |
| fw_instance_id | 否 | String | 防火墙id，可通过 防火墙ID获取方式 获取 |
| fw_key_word | 否 | String | 防火墙关键字，可使用防火墙id或名称查询，可通过 防火墙ID获取方式 |
| eps_id | 否 | String | 弹性公网ip的企业项目id，可通过 如何获取企业项目ID 获取，租户未开启企业项目时为0 |
| tags | 否 | String | 标签列表信息可通过查询EIP服务界面列表标签页签获得 |

请求参数

表 4-97 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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-98 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--|-------------|
| data | EipResponseData object | 查询eip返回data |

表 4-99 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 | 查询获得EIP列表总数 |
| records | Array of EipResource objects | eip资源记录 |

表 4-100 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 |
| public_ipv6 | String | 弹性公网IP,IPV6类型 |
| enterprise_project_id | String | Eip所在账户企业项目id |
| device_id | String | EIP绑定设备（如ecs，nat）id |
| device_name | String | EIP绑定设备（如ecs，nat）名称 |
| device_owner | String | EIP绑定设备（如ecs，nat）拥有者 |
| associate_instance_type | String | 关联实例类型，包括：NATGW，ELB，PORT等。 |
| 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 。 |
| 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，通过返回值中的data.records.protect_objects.object_id（.表示各对象之间层级的区分）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。此处仅取type为0的防护对象id，可通过data.records.protect_objects.type（.表示各对象之间层级的区分）获得。 |
| tags | String | 标签列表 |
| domain_id | String | EIP所属用户id，可通过 获取账号、IAM用户、项目、用户组、区域、委托的名称和ID 获取。 |
| fw_domain_id | String | 防火墙所属用户，可通过 获取账号、IAM用户、项目、用户组、区域、委托的名称和ID 获取。 |

状态码：400

表 4-101 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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:test",
"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=CBRCS23040615138M2KW912"
}],
"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.cf.w.v1.region.CfwRegion;
import com.huaweicloud.sdk.cf.w.v1.*;
import com.huaweicloud.sdk.cf.w.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");
        String projectId = "{project_id}";

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

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

```
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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListEipsRequest()
        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")
projectId := "{project_id}"

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

client := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListEipsRequest{}
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](#)。

4.2.4 查看 eip 告警白名单

功能介绍

[查看eip告警白名单](#)

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/eip/alarm-whitelist/{fw_instance_id}

表 4-102 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------------|------|--------|---|
| fw_instance_id | 是 | String | 防火墙ID，可通过 防火墙ID获取方式 获取 |
| project_id | 是 | String | 项目ID，可以从调API处获取，也可以从控制台获取。可通过 项目ID获取方式 获取 |

表 4-103 Query 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|-----------------------|------|---------|---|
| ip_address | 否 | String | IP地址 |
| 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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

请求参数

表 4-104 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，可通过 如何获取用户Token 获取 |

响应参数

状态码：200

表 4-105 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|---|--------------|
| data | HttpListAlarmWhiteListResponseData object | 查询告警白名单返回值数据 |

表 4-106 HttpListAlarmWhiteListResponseData

| 参数 | 参数类型 | 描述 |
|-------|--|--------------|
| list | Array of EipInfo objects | 查询告警白名单返回值数据 |
| pages | Integer | 目前页数 |
| size | Integer | 每页个数 |
| total | Integer | 总数 |

表 4-107 EipInfo

| 参数 | 参数类型 | 描述 |
|----------------|---------|----------------------------------|
| device_name | String | 设备名称 |
| eip_id | String | EIP id |
| fw_instance_id | String | 防火墙id |
| object_id | String | 防护对象id |
| public_ip | String | ip v4地址 |
| public_ipv6 | String | ip v6地址 |
| type | Integer | EIP白名单标志，0表示是EIP白名单，1表示不是EIP白名单。 |

请求示例

查看项目id为408972e72dcd4c1a9b033e955802a36b的EIP告警白名单，防火墙id为e743cfaf-8164-4807-aa13-d893d83313cf，企业项目id为fb55459c-41b3-47fc-885d-540946fddda4。查询结果限制为1000条，偏移量为0。

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/eip/alarm-whitelist/e743cfaf-8164-4807-aa13-d893d83313cf?fw_instance_id=e743cfaf-8164-4807-aa13-d893d83313cf&enterprise_project_id=fb55459c-41b3-47fc-885d-540946fddda4&limit=1000&offset=0
```

响应示例

状态码：200

查询告警白名单返回值

```
{
  "data": {
    "list": [ {
      "device_name": "client_ipv6",
      "eip_id": "a40f3306-b7f3-428b-afe7-5ae1e7a5c594",
      "object_id": "",
      "public_ip": "100.100.100.100",
      "type": 1
    }, {
      "device_name": "ecs-172",
      "eip_id": "5e9c7249-2c05-4c0e-a9ed-192af2c1d129",
      "fw_instance_id": "e743cfaf-8164-4807-aa13-d893d83313cf",
      "object_id": "1b90f031-0c7b-4f25-95e2-b6d9940d269e",
      "public_ip": "100.200.200.200",
      "type": 0
    } ],
    "pages": 1,
    "size": 2,
    "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 ListAlarmWhitelistSolution {

    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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListAlarmWhitelistRequest request = new ListAlarmWhitelistRequest();
        request.withFwInstanceId("{fw_instance_id}");
        try {
            ListAlarmWhitelistResponse response = client.listAlarmWhitelist(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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListAlarmWhitelistRequest()
        request.fw_instance_id = "{fw_instance_id}"
        response = client.list_alarm_whitelist(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")
projectId := "{project_id}"

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

client := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListAlarmWhitelistRequest{}
request.FwInstanceId = "{fw_instance_id}"
response, err := client.ListAlarmWhitelist(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 | 查询告警白名单返回值 |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |

错误码

请参见[错误码](#)。

4.2.5 修改 eip 自动防护开关

功能介绍

修改eip自动防护开关

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/eip/auto-protect-status/switch

表 4-108 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID，可以从调API处获取，也可以从控制台获取。可通过 项目ID获取方式 获取 |

表 4-109 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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

请求参数

表 4-110 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，可通过 如何获取用户Token 获取 |

表 4-111 请求 Body 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------------|------|--------|--|
| fw_instance_id | 否 | String | 防火墙id，可通过 防火墙ID获取方式 获取 |

| 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, 通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 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, type可通过 data.records.protect_objects.type (.表示各对象之间层级的区分) 获得 |
| status | 否 | Integer | 是否开启新增eip自动防护, 1; 是, 0: 否 |

响应参数

状态码: 200

表 4-112 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--------|----|
| data | String | |

请求示例

切换项目id为408972e72dcd4c1a9b033e955802a36b的EIP自动保护状态, 防火墙id为e743cfaf-8164-4807-aa13-d893d83313cf, 目标防护对象id为1b90f031-0c7b-4f25-95e2-b6d9940d269e, 将开关状态切换为0

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/eip/auto-protect-status/switch  
  
{  
  "object_id": "1b90f031-0c7b-4f25-95e2-b6d9940d269e",  
  "fw_instance_id": "e743cfaf-8164-4807-aa13-d893d83313cf",  
  "status": 0  
}
```

响应示例

状态码: 200

修改EIP自动防护状态返回值

```
{  
  "data": "1b90f031-0c7b-4f25-95e2-b6d9940d269e"  
}
```

SDK 代码示例

SDK代码示例如下。

Java

切换项目id为408972e72dcd4c1a9b033e955802a36b的EIP自动保护状态，防火墙id为e743cfaf-8164-4807-aa13-d893d83313cf，目标防护对象id为1b90f031-0c7b-4f25-95e2-b6d9940d269e，将开关状态切换为0

```
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 SwitchAutoProtectStatusSolution {

    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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        SwitchAutoProtectStatusRequest request = new SwitchAutoProtectStatusRequest();
        SwitchEipStatusDto body = new SwitchEipStatusDto();
        body.withStatus(0);
        body.withObjectId("1b90f031-0c7b-4f25-95e2-b6d9940d269e");
        body.withFwInstanceId("e743cfaf-8164-4807-aa13-d893d83313cf");
        request.withBody(body);
        try {
            SwitchAutoProtectStatusResponse response = client.switchAutoProtectStatus(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

切换项目id为408972e72dcd4c1a9b033e955802a36b的EIP自动保护状态，防火墙id为e743cfaf-8164-4807-aa13-d893d83313cf，目标防护对象id为1b90f031-0c7b-4f25-95e2-b6d9940d269e，将开关状态切换为0

```
# coding: utf-8

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = SwitchAutoProtectStatusRequest()
        request.body = SwitchEipStatusDto(
            status=0,
            object_id="1b90f031-0c7b-4f25-95e2-b6d9940d269e",
            fw_instance_id="e743cfaf-8164-4807-aa13-d893d83313cf"
        )
        response = client.switch_auto_protect_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

切换项目id为408972e72dcd4c1a9b033e955802a36b的EIP自动保护状态，防火墙id为e743cfaf-8164-4807-aa13-d893d83313cf，目标防护对象id为1b90f031-0c7b-4f25-95e2-b6d9940d269e，将开关状态切换为0

```
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")
projectId := "{project_id}"

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

client := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.SwitchAutoProtectStatusRequest{
    statusSwitchEipStatusDto:= int32(0)
    objectIdSwitchEipStatusDto:= "1b90f031-0c7b-4f25-95e2-b6d9940d269e"
    fwInstanceIdSwitchEipStatusDto:= "e743cfaf-8164-4807-aa13-d893d83313cf"
    request.Body = &model.SwitchEipStatusDto{
        Status: &statusSwitchEipStatusDto,
        ObjectId: &objectIdSwitchEipStatusDto,
        FwInstanceId: &fwInstanceIdSwitchEipStatusDto,
    }
}
response, err := client.SwitchAutoProtectStatus(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 | 修改EIP自动防护状态返回值 |
| 201 | Created |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |

错误码

请参见[错误码](#)。

4.2.6 获取 eip 自动防护状态信息

功能介绍

获取eip自动防护状态信息

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/eip/auto-protect-status/{object_id}

表 4-113 路径参数

| 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，通过返回值中的 data.records.protect_objects.object_id（.表示各对象之间层级的区分）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。此处仅取type为1的防护对象id，可通过 data.records.protect_objects.type（.表示各对象之间层级的区分）获得。 |
| project_id | 是 | String | 项目ID，可以从调API处获取，也可以从控制台获取。可通过 项目ID获取方式 获取 |

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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

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，可通过 如何获取用户Token 获取 |

响应参数

状态码：200

表 4-116 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|-------------------------|---------|
| data | FirewallStatusVO object | 防火墙状态对象 |

表 4-117 FirewallStatusVO

| 参数 | 参数类型 | 描述 |
|---------------------|---------|-----------------------|
| available_eip_count | Integer | 可防护eip数量 |
| beyond_max_count | Boolean | 是否超出eip数量限制 |
| eip_protected_self | Integer | 已防护eip数量 |
| eip_total | Integer | eip总数 |
| eip_un_protected | Integer | 未防护eip数量 |
| object_id | String | 防护对象id |
| status | Integer | 是否开启新增eip自动防护，1；是，0：否 |

请求示例

获取项目id为408972e72dcd4c1a9b033e955802a36b的EIP自动防护状态信息，防火墙id为e743cfaf-8164-4807-aa13-d893d83313cf，企业项目id为fb55459c-41b3-47fc-885d-540946fddda4，目标对象id为1b90f031-0c7b-4f25-95e2-b6d9940d269e。

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/eip/auto-protect-status/1b90f031-0c7b-4f25-95e2-b6d9940d269e?fw_instance_id=e743cfaf-8164-4807-aa13-
```

```
d893d83313cf&enterprise_project_id=fb55459c-41b3-47fc-885d-540946fddda4&object_id=1b90f031-0c7b-4f25-95e2-b6d9940d269e&project_id=408972e72dcd4c1a9b033e955802a36b
```

响应示例

状态码：200

OK

```
{
  "data" : {
    "available_eip_count" : 50,
    "beyond_max_count" : false,
    "eip_protected_self" : 15,
    "eip_total" : 24,
    "eip_un_protected" : 9,
    "object_id" : "1b90f031-0c7b-4f25-95e2-b6d9940d269e",
    "status" : 1
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.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 ShowAutoProtectStatusSolution {

    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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowAutoProtectStatusRequest request = new ShowAutoProtectStatusRequest();
        request.withObjectId("{object_id}");
        try {
            ShowAutoProtectStatusResponse response = client.showAutoProtectStatus(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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowAutoProtectStatusRequest()
        request.object_id = "{object_id}"
        response = client.show_auto_protect_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")
projectId := "{project_id}"

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

client := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowAutoProtectStatusRequest{}
request.ObjectId = "{object_id}"
response, err := client.ShowAutoProtectStatus(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 |

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 4-118 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-119 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

请求参数

表 4-120 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。可通过 如何获取用户Token 获取。 |

表 4-121 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, 通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 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, type可通过 data.records.protect_objects.type (.表示各对象之间层级的区分) 获得 |
| type | 是 | Integer | 规则类型, 0: 互联网边界规则, 1: vpc间规则, 2: nat规则, 当type取0时, 规则源和目的地址需要为公网ip或域名, vpc间规则需要源和目的地址为私有ip, nat规则需要源地址为私有ip, 目的地址为公网ip或域名。 |
| rules | 是 | Array of rules objects | 添加规则请求规则列表 |

表 4-122 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 | 修改规则顺序请求体 |
| address_type | 是 | Integer | 地址类型, 0表示ipv4, 1表示ipv6 |
| action_type | 是 | Integer | 规则动作, 0表示允许通行 (permit), 1表示拒绝通行 (deny) |
| status | 是 | Integer | 规则启用状态, 0表示禁用, 1表示启用 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|--------------------------|------|---|---|
| applications | 否 | Array of strings | 规则应用列表，规则应用类型包括：“HTTP”，“HTTPS”，“TLS1”，“DNS”，“SSH”，“MYSQL”，“SMTP”，“RDP”，“RDPS”，“VNC”，“POP3”，“IMAP4”，“SMTPS”，“POP3S”，“FTPS”，“ANY”，“BGP”等。 |
| long_connect_time | 否 | Long | 长连接时长 |
| long_connect_time_hour | 否 | Long | 长连接时长对应小时 |
| long_connect_time_minute | 否 | Long | 长连接时长对应分钟 |
| 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 |
| description | 否 | String | 描述 |
| direction | 否 | Integer | 方向：0表示外到内，1表示内到外，规则type=0（互联网规则）或者type=2（nat规则）时方向值必填 |
| 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 | 规则附带标签对象 |

表 4-123 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, 目标规则id可以通过 查询防护规则接口 获得, 通过返回值中的data.records.rule_id (.表示各对象之间层级的区分) 获得。 |
| 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 4-124 RuleAddressDtoForRequest

| 参数 | 是否必选 | 参数类型 | 描述 |
|---------------------|------|---------|--|
| type | 是 | Integer | 输入类型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-网络型, 7域名组-应用型。 |
| address_type | 否 | Integer | 地址类型0 ipv4, 1 ipv6, 当type为0手动输入类型时不能为空 |
| address | 否 | String | IP地址信息, 当type为0手动输入类型时不能为空 |
| address_set_id | 否 | String | 关联IP地址组ID, 当type为1关联IP地址组类型时不能为空, 可通过 查询地址组列表接口 查询获得, 通过返回值中的data.records.set_id (.表示各对象之间层级的区分) 获得。 |
| address_set_name | 否 | String | 关联IP地址组名称, 当type为1关联IP地址组类型时不能为空, 可通过 查询地址组列表接口 查询获得, 通过返回值中的data.records.name (.表示各对象之间层级的区分) 获得。 |
| domain_address_name | 否 | String | type为2 (域名) 和7 (应用域名组) 具体内容根据type中7修改后的类型名称 |
| region_list_json | 否 | String | 规则地域列表json值 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------------|------|--|---|
| region_list | 否 | Array of IpRegionDto objects | 规则地域列表 |
| domain_set_id | 否 | String | 域名组id, type为4 (域名组) 或7 (域名组-应用型) 时不能为空。可通过 查询域名组列表接口 查询获得, 通过返回值中的 data.records.set_id (.表示各对象之间层级的区分) 获得。 |
| domain_set_name | 否 | String | 域名组名称, type为4 (域名组) 或7 (域名组-应用型) 时不能为空。可通过 查询域名组列表接口 查询获得, 通过返回值中的 data.records.name (.表示各对象之间层级的区分) 获得。 |
| ip_address | 否 | Array of strings | IP地址列表, 当type为5 (多对象) 时不能为空。 |
| address_set_type | 否 | Integer | 地址组类型, 当type为1 (关联IP地址组) 时不能为空。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id列表, 当type为5 (多对象) 时不能为空。地址组id可通过 查询地址组列表接口 查询获得, 通过返回值中的 data.records.set_id (.表示各对象之间层级的区分) 获得。查询条件中query_address_set_type 需要设置为1预定义地址组。 |
| address_group | 否 | Array of strings | 地址组id列表, 当type为5 (多对象) 时不能为空。地址组id可通过 查询地址组列表接口 查询获得, 通过返回值中的 data.records.set_id (.表示各对象之间层级的区分) 获得。查询条件中query_address_set_type 需要设置为0自定义地址组。 |

表 4-125 IpRegionDto

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------------|------|---------|--|
| region_id | 否 | String | 区域id, 可通过 获取账号、IAM 用户、项目、用户组、区域、委托的名称和ID 获取。 |
| description_cn | 否 | String | 区域中文描述, 仅当区域为中国区域时使用, 可通过 地域信息表 获取。 |
| description_en | 否 | String | 区域英文描述, 仅当区域为非中国区域时使用, 可通过 地域信息表 获取。 |
| region_type | 否 | Integer | 区域类型, 0表示国家, 1表示省份, 2表示大洲, 可通过 地域信息表 获取。 |

表 4-126 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,type为0手动类型时不能为空。 |
| 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,type为0手动类型时不能为空。 |
| source_port | 否 | String | 源端口 |
| dest_port | 否 | String | 目的端口 |
| service_set_id | 否 | String | 服务组id, 当type为1 (关联IP地址组) 时不能为空, 可通过 获取服务组列表接口 查询获得, 通过返回值中的 data.records.set_id (.表示各对象之间层级的区分) 获得。 |
| service_set_name | 否 | String | 服务组名称, 当type为1 (关联IP地址组) 时不能为空, 可通过 获取服务组列表接口 查询获得, 通过返回值中的 data.records.name (.表示各对象之间层级的区分) 获得。 |

| 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id列表，服务组id可通过 获取服务组列表接口 查询获得，通过返回值中的 data.records.set_id (.表示各对象之间层级的区分) 获得。查询条件中query_service_set_type 需要设置为1预定义服务组。 |
| service_group | 否 | Array of strings | 服务组id列表，服务组id可通过 获取服务组列表接口 查询获得，通过返回值中的 data.records.set_id (.表示各对象之间层级的区分) 获得。查询条件中query_service_set_type 需要设置为0自定义服务组。 |
| service_group_names | 否 | Array of ServiceGroupVO 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 4-127 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,RuleServiceDto.type 为0时不能为空。 |
| 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 4-128 ServiceGroupVO

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------------|------|-------------------|---|
| name | 否 | String | 服务组名称 |
| 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 |
| service_set_type | 否 | Integer | 服务组类型，0表示自定义服务组，1表示预定义服务组 |
| set_id | 否 | String | 服务组id，可通过 获取服务组列表接口 查询获得，通过返回值中的data.records.set_id（.表示各对象之间层级的区分）获得。 |

表 4-129 TagsVO

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

响应参数

状态码：200

表 4-130 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|-----------------------------------|---------------|
| data | RuleIdList object | 创建规则返回值data数据 |

表 4-131 RuleIdList

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

表 4-132 RuleId

| 参数 | 参数类型 | 描述 |
|------|--------|------|
| id | String | 规则id |
| name | String | 规则名称 |

状态码：400

表 4-133 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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",  
      "name" : "testName"  
    } ]  
  }  
}
```

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");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)
```

```
.withAk(ak)
.withSk(sk);

CfwClient client = CfwClient.newBuilder()
    .withCredential(auth)
    .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
    .build();
AddAclRuleRequest request = new AddAclRuleRequest();
AddRuleAclDto body = new AddRuleAclDto();
TagsVO tagRules = new TagsVO();
tagRules.withTagKey("")
    .withTagValue("");
RuleServiceDto serviceRules = new RuleServiceDto();
serviceRules.withType(0)
    .withProtocol(6)
    .withSourcePort("0")
    .withDestPort("0");
RuleAddressDtoForRequest destinationRules = new RuleAddressDtoForRequest();
destinationRules.withType(0)
    .withAddress("2.2.2.2");
RuleAddressDtoForRequest sourceRules = new RuleAddressDtoForRequest();
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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = AddAclRuleRequest()
        tagRules = TagsVO(
            tag_key="",
            tag_value=""
        )
        serviceRules = RuleServiceDto(
            type=0,
            protocol=6,
            source_port="0",
            dest_port="0"
        )
        destinationRules = RuleAddressDtoForRequest(
            type=0,
            address="2.2.2.2"
        )
        sourceRules = RuleAddressDtoForRequest(
            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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddAclRuleRequest{}
    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.RuleAddressDtoForRequest{
        Type: int32(0),
        Address: &addressDestination,
    }
    addressSource := "1.1.1.1"
```

```
sourceRules := &model.RuleAddressDtoForRequest{
    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 4-134 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|-------------|------|--------|--|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |
| acl_rule_id | 是 | String | 规则id, 可通过 查询防护规则接口 查询获得, 通过返回值中的 data.records.rule_id (表示各对象之间层级的区分) 获得。 |

表 4-135 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

请求参数

表 4-136 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。可通过 如何获取用户Token 获取。 |

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数据 |

表 4-138 RuleId

| 参数 | 参数类型 | 描述 |
|------|--------|------|
| id | String | 规则id |
| name | String | 规则名称 |

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 |
| error_msg | String | 错误描述 |

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

删除规则返回数据

```
{
  "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.cfw.v1.*;
import com.huaweicloud.sdk.cfw.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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteAclRuleRequest request = new DeleteAclRuleRequest();
        request.withAclRuleId("{acl_rule_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.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DeleteAclRuleRequest()
        request.acl_rule_id = "{acl_rule_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")
    projectId := "{project_id}"

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

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

client := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.DeleteAclRuleRequest{}
request.AclRuleId = "{acl_rule_id}"
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 | 删除规则返回数据 |
| 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 4-140 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

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。可通过 如何获取用户Token 获取。 |

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，通过返回值中的data.records.protect_objects.object_id（.表示各对象之间层级的区分）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，type可通过data.records.protect_objects.type（.表示各对象之间层级的区分）获得。 |
| rule_ids | 是 | Array of strings | 规则id列表，批量删除规则时传入的id列表。id值可通过 查询防护规则接口 查询获得，通过返回值中的data.records.rule_id（.表示各对象之间层级的区分）获得。 |

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 | BatchDeleteAclRulesResponseData object | 批量删除ACL规则返回值data |

表 4-145 BatchDeleteAclRulesResponseData

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

表 4-146 BatchDeleteRuleInfo

| 参数 | 参数类型 | 描述 |
|------|--------|-------------|
| name | String | 批量删除的acl的名称 |
| id | String | 批量删除的acl的id |

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": {
    "responseDatas": [ {
      "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");
        String projectId = "{project_id}";

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

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

import os
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.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

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

try:
    request = BatchDeleteAclRulesRequest()
    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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
```

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

request := &model.BatchDeleteAclRulesRequest{}
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 4-147 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-148 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

请求参数

表 4-149 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。可通过 如何获取用户Token 获取。 |

表 4-150 请求 Body 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------|------|------------------|---|
| rule_ids | 是 | Array of strings | 删除规则击中次数请求的规则列表, 规则id可通过 查询防护规则接口 查询获得, 通过返回值中的data.records.rule_id (.表示各对象之间层级的区分) 获得。 |

响应参数

状态码：200

表 4-151 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|----|--------|----|
| - | String | |

状态码：400

表 4-152 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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" : "清除规则命中次数参数错误"  
}
```

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");
        String projectId = "{project_id}";

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

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

import os
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.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

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

try:
    request = DeleteAclRuleHitCountRequest()
    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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```
request := &model.DeleteAclRuleHitCountRequest{}
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 |
| 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 4-153 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|-------------|------|--------|--|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |
| acl_rule_id | 是 | String | 规则id, 可通过 查询防护规则接口 查询获得, 通过返回值中的 data.records.rule_id (表示各对象之间层级的区分) 获得。 |

表 4-154 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

请求参数

表 4-155 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。可通过 如何获取用户Token 获取。 |

表 4-156 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表示ipv4, 1表示ipv6 |
| name | 否 | String | 规则名称 |
| direction | 否 | Integer | 方向: 0表示外到内, 1表示内到外,规则type=0 (互联网规则) 或者type= 2 (nat规则) 时方向值必填 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|--------------------------|------|------------------------------|--|
| action_type | 否 | Integer | 规则动作, 0表示允许通行 (permit), 1表示拒绝通行 (deny) |
| status | 否 | Integer | 规则启用状态, 0表示禁用, 1表示启用 |
| applications | 否 | Array of strings | 规则应用列表, 规则应用类型包括: “HTTP”, “HTTPS”, “TLS1”, “DNS”, “SSH”, “MYSQL”, “SMTP”, “RDP”, “RDPS”, “VNC”, “POP3”, “IMAP4”, “SMTPS”, “POP3S”, “FTPS”, “ANY”, “BGP” 等。 |
| description | 否 | String | 规则描述 |
| long_connect_time_hour | 否 | Long | 长连接时长对应小时 |
| long_connect_time_minute | 否 | Long | 长连接时长对应分钟 |
| 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 |
| 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 | 规则类型, 0: 互联网规则, 1: vpc规则, 2: nat规则 |
| tag | 否 | TagsVO object | 规则附带标签对象 |

表 4-157 RuleAddressDto

| 参数 | 是否必选 | 参数类型 | 描述 |
|---------------------|------|------------------------------|---|
| type | 是 | Integer | 地址类型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-网络型，7域名组-应用型。 |
| address_type | 否 | Integer | 地址类型0 ipv4，1 ipv6，当type为0手动输入类型时不能为空 |
| address | 否 | String | IP地址信息，当type为0手动输入类型时不能为空 |
| address_set_id | 否 | String | 关联IP地址组ID，当type为1关联IP地址组类型时不能为空，可通过 查询地址组列表接口 查询获得，通过返回值中的data.records.set_id（.表示各对象之间层级的区分）获得。 |
| address_set_name | 否 | String | 关联IP地址组名称，当type为1关联IP地址组类型时不能为空，可通过 查询地址组列表接口 查询获得，通过返回值中的data.records.name（.表示各对象之间层级的区分）获得。 |
| domain_address_name | 否 | String | type为2（域名）和7（应用域名组）具体内容根据type中7修改后的类型名称 |
| region_list_json | 否 | String | 规则地域列表json值 |
| region_list | 否 | Array of IpRegionDto objects | 规则地域列表 |
| domain_set_id | 否 | String | 域名组id，type为4（域名组）或7（域名组-应用型）时不能为空。可通过 查询域名组列表接口 查询获得，通过返回值中的data.records.set_id（.表示各对象之间层级的区分）获得。 |
| domain_set_name | 否 | String | 域名组名称，type为4（域名组）或7（域名组-应用型）时不能为空。可通过 查询域名组列表接口 查询获得，通过返回值中的data.records.name（.表示各对象之间层级的区分）获得。 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|---------------------|------|---------------------------------|---|
| ip_address | 否 | Array of strings | IP地址列表，当type为5（多对象）时不能为空。 |
| address_group | 否 | Array of strings | 地址组id列表，当type为5（多对象）时不能为空。地址组id可通过 查询地址组列表接口 查询获得，通过返回值中的data.records.set_id（.表示各对象之间层级的区分）获得。查询条件中query_address_set_type需要设置为0自定义地址组。 |
| address_group_names | 否 | Array of AddressGroupVO objects | 地址组名称列表 |
| address_set_type | 否 | Integer | 地址组类型，当type为1（关联IP地址组）时不能为空。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id列表，当type为5（多对象）时不能为空。地址组id可通过 查询地址组列表接口 查询获得，通过返回值中的data.records.set_id（.表示各对象之间层级的区分）获得。查询条件中query_address_set_type需要设置为1预定义地址组。 |

表 4-158 IpRegionDto

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------------|------|---------|--|
| region_id | 否 | String | 区域id，可通过 获取账号、IAM用户、项目、用户组、区域、委托的名称和ID 获取。 |
| description_cn | 否 | String | 区域中文描述，仅当区域为中国区域时使用，可通过 地域信息表 获取。 |
| description_en | 否 | String | 区域英文描述，仅当区域为非中国区域时使用，可通过 地域信息表 获取。 |
| region_type | 否 | Integer | 区域类型，0表示国家，1表示省份，2表示大洲，可通过 地域信息表 获取。 |

表 4-159 AddressGroupVO

| 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 |
| name | 否 | String | 关联IP地址组名称，可通过 查询地址组列表接口 查询获得，通过返回值中的data.records.name（.表示各对象之间层级的区分）获得。 |
| set_id | 否 | String | 关联IP地址组ID，可通过 查询地址组列表接口 查询获得，通过返回值中的data.records.set_id（.表示各对象之间层级的区分）获得。 |

表 4-160 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,type为0手动类型时不能为空。 |
| 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,type为0手动类型时不能为空。 |
| source_port | 否 | String | 源端口 |
| dest_port | 否 | String | 目的端口 |
| service_set_id | 否 | String | 服务组id，当type为1（关联IP地址组）时不能为空，可通过 获取服务组列表接口 查询获得，通过返回值中的data.records.set_id（.表示各对象之间层级的区分）获得。 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|---------------------|------|---|---|
| service_set_name | 否 | String | 服务组名称,当type为1（关联IP地址组）时不能为空，可通过 获取服务组列表接口 查询获得，通过返回值中的data.records.name（.表示各对象之间层级的区分）获得。 |
| custom_service | 否 | Array of ServiceItem objects | 自定义服务 |
| predefined_group | 否 | Array of strings | 预定义服务组id列表，服务组id可通过 获取服务组列表接口 查询获得，通过返回值中的data.records.set_id（.表示各对象之间层级的区分）获得。查询条件中query_service_set_type需要设置为1预定义服务组。 |
| service_group | 否 | Array of strings | 服务组id列表，服务组id可通过 获取服务组列表接口 查询获得，通过返回值中的data.records.set_id（.表示各对象之间层级的区分）获得。查询条件中query_service_set_type需要设置为0自定义服务组。 |
| service_group_names | 否 | Array of ServiceGroupVO 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 4-161 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,RuleServiceDto.type为0时不能为空。 |
| 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 4-162 ServiceGroupVO

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------------|------|-------------------|--|
| name | 否 | String | 服务组名称 |
| 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 |
| service_set_type | 否 | Integer | 服务组类型，0表示自定义服务组，1表示预定义服务组 |
| set_id | 否 | String | 服务组id，可通过 获取服务组列表接口 查询获得，通过返回值中的data.records.set_id（表示各对象之间层级的区分）获得。 |

表 4-163 TagsVO

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

响应参数

状态码：200

表 4-164 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|-------------------------------|------|
| data | RuleId object | 规则数据 |

表 4-165 RuleId

| 参数 | 参数类型 | 描述 |
|------|--------|------|
| id | String | 规则id |
| name | String | 规则名称 |

状态码：400**表 4-166** 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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");
        String projectId = "{project_id}";

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

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

        UpdateAclRuleRequest request = new UpdateAclRuleRequest();
        request.withAclRuleId("{acl_rule_id}");
        UpdateRuleAclDto body = new UpdateRuleAclDto();
        TagsVO tagbody = new TagsVO();
        tagbody.withTagKey("")
            .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 {
    UpdateAclRuleResponse response = client.updateAclRule(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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateAclRuleRequest()
        request.acl_rule_id = "{acl_rule_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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
```

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

request := &model.UpdateAclRuleRequest{
request.AclRuleId = "{acl_rule_id}"
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 4-167 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-168 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------------|------|--------|---|
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

请求参数

表 4-169 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。可通过 如何获取用户Token 获取。 |

表 4-170 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, 通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 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, type可通过 data.records.protect_objects.type (.表示各对象之间层级的区分) 获得。 |
| action | 是 | String | 规则动作, enable表示允许通行 (permit), disable表示拒绝通行 (deny) |
| rule_ids | 是 | Array of strings | 规则id列表, 规则id可通过 查询防护规则接口 查询获得, 通过返回值中的data.records.rule_id (.表示各对象之间层级的区分) 获得。 |

响应参数

状态码: 200

表 4-171 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, 为请求体中传入的规则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-8470-3d1b3dad1fd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");
    String projectId = "{project_id}";

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

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

import os
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")
```



```
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

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

try:
    request = BatchUpdateAclRuleActionsRequest()
    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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

```
var listRuleIdsbody = []string{
    "4e12d889-c1d3-491b-8470-3d1b3dad1fd",
    "f798a6a8-c4c5-42b4-838c-c922c9908cb4",
}
request.Body = &model.UpdateSecurityPoliciesActionDto{
    RuleIds: listRuleIdsbody,
    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 4-172 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|--|
| project_id | 是 | String | 项目ID, 可以从调API处获取，也可以从控制台获取。 项目ID获取方式 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|-------------|------|--------|--|
| acl_rule_id | 是 | String | 规则id, 可通过 查询防护规则接口 查询获得, 通过返回值中的data.records.rule_id (.表示各对象之间层级的区分) 获得。 |

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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

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 | 是 | String | 用户Token。可通过 如何获取用户Token 获取。 |

表 4-175 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, 目标规则id可以通过 查询防护规则接口 获得, 通过返回值中的data.records.rule_id (.表示各对象之间层级的区分) 获得。 |
| 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 4-176 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|-----------------------|--------|
| data | OrderRuleId object | 规则id数据 |

表 4-177 OrderRuleId

| 参数 | 参数类型 | 描述 |
|----|--------|------|
| id | String | 规则id |

状态码：400

表 4-178 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateAclRuleOrderRequest request = new UpdateAclRuleOrderRequest();
        request.withAclRuleId("{acl_rule_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.getStatusCode());
        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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateAclRuleOrderRequest()
        request.acl_rule_id = "{acl_rule_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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateAclRuleOrderRequest{}
    request.AclRuleId = "{acl_rule_id}"
    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 4-179 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 |
|-----------|------|---------|---|
| 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, 通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 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, type可通过 data.records.protect_objects.type (.表示各对象之间层级的区分) 获得 |
| type | 否 | Integer | 规则类型, 0: 互联网规则, 1: vpc规则, 2: nat规则 |
| ip | 否 | String | ip地址 |
| name | 否 | String | 规则名称 |
| direction | 否 | Integer | 方向0: 外到内1: 内到外 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|-----------------------|------|---------|--|
| status | 否 | Integer | 规则下发状态 0: 禁用, 1: 启用 |
| action_type | 否 | Integer | 动作0: permit, 1: deny |
| address_type | 否 | Integer | 地址类型, 0表示ipv4, 1表示ipv6 |
| 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |
| tags_id | 否 | String | 规则标签id, 创建规则时产生。 |
| source | 否 | String | 源地址 |
| destination | 否 | String | 目的地址 |
| service | 否 | String | 服务端口 |
| application | 否 | String | 规则应用类型包括: “HTTP”, “HTTPS”, “TLS1”, “DNS”, “SSH”, “MYSQL”, “SMTP”, “RDP”, “RDPS”, “VNC”, “POP3”, “IMAP4”, “SMTPS”, “POP3S”, “FTPS”, “ANY”, “BGP” 等。 |

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） |

响应参数

状态码：200

表 4-182 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--------------------|-------------|
| data | data object | 查询规则列表返回值数据 |

表 4-183 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 | 查询规则列表总条数 |
| 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 ，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。 |
| records | Array of records objects | 查询规则列表记录 |

表 4-184 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 |
| name | String | 规则名称 |
| direction | Integer | 规则方向0：外到内1：内到外 |
| 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 | Long | 长连接时长 |

| 参数 | 参数类型 | 描述 |
|--------------------------|--|-------------------------------------|
| long_connect_enable | Integer | 长连接支持 |
| long_connect_time_hour | Long | 长连接时长对应小时 |
| long_connect_time_minute | Long | 长连接时长对应分钟 |
| long_connect_time_second | Long | 长连接时长秒 |
| source | RuleAddressDtoForResponse object | 源地址对象 |
| destination | RuleAddressDtoForResponse object | 目的地址对象 |
| service | RuleServiceDtoForResponse object | 服务对象 |
| type | Integer | 规则类型, 0: 互联网规则, 1: vpc规则, 2: nat规则 |
| created_date | String | 规则创建时间, 例如: "2024-08-12 08:40:00" |
| last_open_time | String | 规则最后开启时间, 例如: "2024-08-12 08:40:00" |
| tag | TagsVO object | 规则附带标签对象 |

表 4-185 RuleAddressDtoForResponse

| 参数 | 参数类型 | 描述 |
|---------------------|---------|--|
| type | Integer | 地址类型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 | 地址类型0 ipv4, 1 ipv6, 当type为0手工输入类型时不能为空 |
| address | String | IP地址信息 |
| address_set_id | String | 关联IP地址组ID |
| address_set_name | String | 地址组名称 |
| domain_address_name | String | 域名地址名称 |

| 参数 | 参数类型 | 描述 |
|---------------------|--|--|
| region_list_json | String | 规则地域列表json值 |
| region_list | Array of IpRegionDto objects | 规则地域列表 |
| 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 |
| address_group | Array of strings | 地址组id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 |

表 4-186 IpRegionDto

| 参数 | 参数类型 | 描述 |
|----------------|---------|--|
| region_id | String | 区域id，可通过 获取账号、IAM用户、项目、用户组、区域、委托的名称和ID 获取。 |
| description_cn | String | 区域中文描述，仅当区域为中国区域时使用，可通过 地域信息表 获取。 |
| description_en | String | 区域英文描述，仅当区域为非中国区域时使用，可通过 地域信息表 获取。 |
| region_type | Integer | 区域类型，0表示国家，1表示省份，2表示大洲，可通过 地域信息表 获取。 |

表 4-187 AddressGroupVO

| 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 |
| name | String | 关联IP地址组名称，可通过 查询地址组列表接口 查询获得，通过返回值中的data.records.name（.表示各对象之间层级的区分）获得。 |

| 参数 | 参数类型 | 描述 |
|--------|--------|---|
| set_id | String | 关联IP地址组ID，可通过 查询地址组列表接口 查询获得，通过返回值中的data.records.set_id（.表示各对象之间层级的区分）获得。 |

表 4-188 RuleServiceDtoForResponse

| 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 |
| service_set_name | String | 服务组名称 |
| custom_service | Array of ServiceItem objects | 自定义服务 |
| service_group | Array of strings | 服务组id列表 |
| service_group_names | Array of ServiceGroupVO 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 4-189 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,RuleServiceDto.type为0时不能为空。 |

| 参数 | 参数类型 | 描述 |
|-------------|--------|--------|
| 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 4-190 ServiceGroupVO

| 参数 | 参数类型 | 描述 |
|------------------|-------------------|--|
| name | String | 服务组名称 |
| 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 |
| service_set_type | Integer | 服务组类型，0表示自定义服务组，1表示预定义服务组 |
| set_id | String | 服务组id，可通过 获取服务组列表接口 查询获得，通过返回值中的 data.records.set_id（.表示各对象之间层级的区分）获得。 |

表 4-191 TagsVO

| 参数 | 参数类型 | 描述 |
|-----------|--------|-------|
| tag_id | String | 规则id |
| tag_key | String | 规则标签键 |
| tag_value | String | 规则标签值 |

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 |
| error_msg | String | 错误描述 |

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，limit为10的数据

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

响应示例

状态码：200

查询规则列表返回值

```
{
  "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": "48fbf09b-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");
        String projectId = "{project_id}";

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

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

import os
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.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

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

try:
    request = ListAclRulesRequest()
    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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAclRulesRequest{}
    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 | 查询规则列表返回值 |
| 400 | Bad Request |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |
| 500 | Internal Server Error |

错误码

请参见[错误码](#)。

4.3.9 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 4-193 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------------|------|---------|---|
| fw_instance_id | 是 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |
| 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 4-195 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。可通过 如何获取用户Token 获取。 |

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 | HttpGetAclTagResponseData object | 获取规则标签数据 |

表 4-197 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 | 规则标签总数 |
| records | Array of TagsVO objects | 规则标签列表 |

表 4-198 TagsVO

| 参数 | 参数类型 | 描述 |
|-----------|--------|-------|
| tag_id | String | 规则id |
| tag_key | String | 规则标签键 |
| tag_value | String | 规则标签值 |

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

查询规则标签返回值

```
{
  "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");
        String projectId = "{project_id}";

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

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

import os
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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
```

```
request = ListRuleAclTagsRequest()
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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListRuleAclTagsRequest{}
    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 | 查询规则标签返回值 |

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 4-199 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-200 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

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。可通过 如何获取用户Token 获取。 |

表 4-202 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，规则id，可通过 查询防护规则接口 查询获得，通过返回值中的data.records.rule_id（.表示各对象之间层级的区分）获得。 |

响应参数

状态码：200

表 4-203 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--|--------------|
| data | RuleHitCountRecords object | 获取规则击中次数响应数据 |

表 4-204 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 | 获取规则击中次数总条数 |
| records | Array of RuleHitCountObject objects | 规则击中次数信息列表 |

表 4-205 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，当acl规则被触发时，对应规则id的击中次数会增加一次。 |

请求示例

查询项目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

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 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");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
```

```
.withProjectId(projectId)
.withAk(ak)
.withSk(sk);

CfwClient client = CfwClient.newBuilder()
    .withCredential(auth)
    .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
    .build();
ListAclRuleHitCountRequest request = new ListAclRuleHitCountRequest();
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.getHttpStatusCode());
    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的命中次数

```
# coding: utf-8

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListAclRuleHitCountRequest()
        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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAclRuleHitCountRequest{}
    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](#)。

4.3.11 查看 region 列表

功能介绍

查看region列表

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/regions

表 4-206 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID，可以从调API处获取，也可以从控制台获取。可通过 项目ID获取方式 获取 |

表 4-207 Query 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------------|------|--------|--|
| fw_instance_id | 是 | String | 防火墙ID，可通过 防火墙ID获取方式 获取 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|-----------------------|------|--------|---|
| enterprise_project_id | 否 | String | 企业项目ID，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

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，可通过 如何获取用户Token 获取 |

响应参数

状态码：200

表 4-209 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--------|----------|
| data | Object | region列表 |

请求示例

查询region列表，项目id为408972e72dcd4c1a9b033e955802a36b，防火墙id为fcd7179-64a4-4438-bc6d-b510a2da2f64。

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/regions?fw_instance_id=fcd7179-64a4-4438-bc6d-b510a2da2f64
```

响应示例

状态码：200

查看region列表接口响应数据

```
{
  "data": [ {
    "region_id": "RW",
    "description_cn": "卢旺达",
    "description_en": "Rwanda",
    "region_type": 0,
    "superior_region_id": "AF"
  }, {
    "region_id": "SO",
    "description_cn": "索马里",
```

```
"description_en" : "Somalia",
"region_type" : 0,
"superior_region_id" : "AF"
}, {
"region_id" : "YE",
"description_cn" : "也门",
"description_en" : "Yemen",
"region_type" : 0,
"superior_region_id" : "AS"
}]
}
```

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 ListRegionsSolution {

    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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListRegionsRequest request = new ListRegionsRequest();
        try {
            ListRegionsResponse response = client.listRegions(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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListRegionsRequest()
        response = client.list_regions(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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```
Build()  
  
request := &model.ListRegionsRequest{}  
response, err := client.ListRegions(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 | 查看region列表接口响应数据 |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |

错误码

请参见[错误码](#)。

4.3.12 查看导入状态接口

功能介绍

查看导入状态接口

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/acl-rule/import-status

表 4-210 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID，可以从调API处获取，也可以从控制台获取。可通过 项目ID获取方式 获取 |

表 4-211 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，通过返回值中的 data.records.protect_objects.object_id（.表示各对象之间层级的区分）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。此处仅取type为1的防护对象id，可通过 data.records.protect_objects.type（.表示各对象之间层级的区分）获得。 |
| enterprise_project_id | 否 | String | 企业项目ID，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

请求参数

表 4-212 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，可通过 如何获取用户Token 获取 |

响应参数

状态码：200

表 4-213 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--|----------|
| data | ShowImportStat usId object | 查看导入状态数据 |

表 4-214 ShowImportStatusId

| 参数 | 参数类型 | 描述 |
|--------|---------|--------------------------------------|
| id | String | 防护对象id |
| status | Integer | 导入状态，-1表示未配置，0表示配置失败，1表示配置成功，2表示配置中。 |

请求示例

查询项目id为176d761dc156471c9c9b227376594160的ACL规则导入状态，目标防护对象id为cfb245cf-28b0-4c23-8164-dac013fdcf0c。

```
https://{Endpoint}/v1/176d761dc156471c9c9b227376594160/acl-rule/import-status?object_id=cfb245cf-28b0-4c23-8164-dac013fdcf0c
```

响应示例

状态码：200

查看ACL规则导入状态的接口响应

```
{
  "data" : {
    "id" : "cfb245cf-28b0-4c23-8164-dac013fdcf0c",
    "status" : 4
  }
}
```

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 ShowImportStatusSolution {

    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");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
```

```
.withProjectId(projectId)
.withAk(ak)
.withSk(sk);

CfwClient client = CfwClient.newBuilder()
    .withCredential(auth)
    .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
    .build();
ShowImportStatusRequest request = new ShowImportStatusRequest();
try {
    ShowImportStatusResponse response = client.showImportStatus(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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowImportStatusRequest()
        response = client.show_import_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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowImportStatusRequest{}
    response, err := client.ShowImportStatus(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 | 查看ACL规则导入状态的接口响应 |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |

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 4-215 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-216 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

请求参数

表 4-217 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。可通过 如何获取用户Token 获取。 |

表 4-218 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, 通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 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, type可通过 data.records.protect_objects.type (.表示各对象之间层级的区分) 获得 |
| 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 |
| address | 是 | String | ip地址 |
| 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 |
| port | 是 | String | 目的端口 |
| description | 否 | String | 描述 |

响应参数

状态码: 200

表 4-219 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|-----------------------------------|--------------|
| data | BlackWhiteListId object | 添加黑白名单响应data |

表 4-220 BlackWhiteListId

| 参数 | 参数类型 | 描述 |
|------|--------|-----------------|
| id | String | 黑白名单id |
| name | String | 黑白名单名称，为黑白名单的地址 |

状态码：400

表 4-221 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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" : "6e91797b-05bd-4c69-9454-6af905178729",
    "name" : "10.10.1.3"
  }
}
```

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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        AddBlackWhiteListRequest request = new AddBlackWhiteListRequest();
        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.getStatusCode());
            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的白名单

```
# coding: utf-8  
  
import os  
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.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = CfwClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(CfwRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = AddBlackWhiteListRequest()  
        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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddBlackWhiteListRequest{}
    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 4-222 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|--|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |
| list_id | 是 | String | 黑白名单列表id, 可通过 查询黑白名单列表接口 查询获得, 通过返回值中的data.records.list_id (.表示各对象之间层级的区分) 获得。 |

表 4-223 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

请求参数

表 4-224 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。可通过 如何获取用户Token 获取。 |

表 4-225 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 |
| address | 是 | 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 |
| description | 否 | String | 描述 |

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 | BlackWhiteListId object | 更新黑白名单响应数据 |

表 4-227 BlackWhiteListId

| 参数 | 参数类型 | 描述 |
|------|--------|------------------|
| id | String | 黑白名单id |
| name | String | 黑白名单名称, 为黑白名单的地址 |

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 |
| error_msg | String | 错误描述 |

请求示例

更新项目id为9d80d070b6d44942af73c9c3d38e0429防护对象id为cfefd347-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  
  
{  
  "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": "5d37afe6-c5b4-400d-8ff3-a8d6396d7ace",  
    "name": "10.1.1.10"  
  }  
}
```

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cfefd347-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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateBlackWhiteListRequest request = new UpdateBlackWhiteListRequest();
        request.withListId("{list_id}");
        UpdateBlackWhiteListDto body = new UpdateBlackWhiteListDto();
        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.getErrorMsg());
        }
    }
}
```

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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateBlackWhiteListRequest()
        request.list_id = "{list_id}"
        request.body = UpdateBlackWhiteListDto(
            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")
    projectId := "{project_id}"

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

```
client := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateBlackWhiteListRequest{
    request.ListId = "{list_id}"
    portUpdateBlackWhiteListDto:= "1"
    protocolUpdateBlackWhiteListDto:= int32(6)
    addressTypeUpdateBlackWhiteListDto:= int32(0)
    directionUpdateBlackWhiteListDto:= int32(0)
    request.Body = &model.UpdateBlackWhiteListDto{
        Port: &portUpdateBlackWhiteListDto,
        Protocol: &protocolUpdateBlackWhiteListDto,
        Address: "1.1.1.1",
        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 4-229 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|--|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |
| list_id | 是 | String | 黑白名单列表id, 可通过 查询黑白名单列表接口 查询获得, 通过返回值中的data.records.list_id (.表示各对象之间层级的区分) 获得。 |

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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

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。可通过 如何获取用户Token 获取。 |

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 | BlackWhiteListId object | 删除黑白名单响应数据 |

表 4-233 BlackWhiteListId

| 参数 | 参数类型 | 描述 |
|------|--------|-----------------|
| id | String | 黑白名单id |
| name | String | 黑白名单名称，为黑白名单的地址 |

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 |
| error_msg | String | 错误描述 |

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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteBlackWhiteListRequest request = new DeleteBlackWhiteListRequest();
        request.withListId("{list_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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DeleteBlackWhiteListRequest()
        request.list_id = "{list_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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteBlackWhiteListRequest{}
    request.ListId = "{list_id}"
    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 |
| 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 4-235 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-236 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, 通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 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, type可通过 data.records.protect_objects.type (.表示各对象之间层级的区分) 获得 |
| list_type | 是 | Integer | 黑白名单类型4: 黑名单, 5: 白名单 |
| address_type | 否 | Integer | ip地址类型0: ipv4, 1:ipv6 |
| 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-238 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数据 |

表 4-239 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 | 查询黑白名单总条数 |
| records | Array of records objects | 黑白名单列表 |

表 4-240 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 |
| 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 4-241 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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");
        String projectId = "{project_id}";

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

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

import os
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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

```
try:
    request = ListBlackWhiteListsRequest()
    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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListBlackWhiteListsRequest{}
    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 |

| 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 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 4-242 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-243 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------------|------|--------|---|
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

请求参数

表 4-244 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。可通过 如何获取用户Token 获取。 |

表 4-245 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, 通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 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, type可通过 data.records.protect_objects.type (.表示各对象之间层级的区分) 获得 |
| 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 |

响应参数

状态码: 200

表 4-246 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|------------------------|-----------|
| data | AddressSetId object | 添加地址组返回数据 |

表 4-247 AddressSetId

| 参数 | 参数类型 | 描述 |
|------|--------|-------|
| id | String | 地址组id |
| name | String | 地址组名称 |

状态码：400

表 4-248 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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

添加地址组返回值

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

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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        AddAddressSetRequest request = new AddAddressSetRequest();
        AddAddressSetDto body = new AddAddressSetDto();
        body.withAddressType(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  
  
import os  
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.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = CfwClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(CfwRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = AddAddressSetRequest()  
        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")
projectId := "{project_id}"

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

client := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.AddAddressSetRequest{
    addressTypeAddAddressSetDto:= int32(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 | 添加地址组返回值 |
| 400 | Bad Request |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |
| 500 | Internal Server Error |

错误码

请参见[错误码](#)。

4.5.2 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 4-249 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-250 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

请求参数

表 4-251 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。可通过 如何获取用户Token 获取。 |

表 4-252 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, 可通过 查询地址组列表接口 查询获得, 通过返回值中的data.records.set_id (表示各对象之间层级的区分) 获得。 |
| address_items | 否 | Array of address_items objects | 地址组成员列表 |

表 4-253 address_items

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

响应参数

状态码: 200

表 4-254 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 4-255 AddressItems

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

表 4-256 AddressItemIdWithoutName

| 参数 | 参数类型 | 描述 |
|----|--------|---------|
| id | String | 地址组成员id |

表 4-257 CoveredIPVO

| 参数 | 参数类型 | 描述 |
|------------|--------|---------|
| ip | String | ip地址 |
| covered_ip | String | 覆盖ip地址。 |

状态码：400

表 4-258 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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": "",
    "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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        AddAddressItemRequest request = new AddAddressItemRequest();
        AddAddressItemsInfoDto body = new AddAddressItemsInfoDto();
        List<AddAddressItemsInfoDtoAddressItems> listbodyAddressItems = new ArrayList<>();
        listbodyAddressItems.add(
            new AddAddressItemsInfoDtoAddressItems()
                .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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = AddAddressItemRequest()
        listAddressItemsbody = [
            AddAddressItemsInfoDtoToAddressItems(
                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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddAddressItemRequest{}
    descriptionAddressItems := ""
    var listAddressItemsbody = []model.AddAddressItemsInfoDtoAddressItems{
        {
            Address: "2.2.2.2",
            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 4-259 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|--|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |
| set_id | 是 | String | 地址组id, 可通过 查询地址组列表接口 查询获得, 通过返回值中的data.records.set_id (表示各对象之间层级的区分) 获得。 |

表 4-260 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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id，可通过 防火墙ID获取方式 获取 |

请求参数

表 4-261 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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-262 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|-------------------------------------|-----------|
| data | AddressSetId object | 删除地址组返回数据 |

表 4-263 AddressSetId

| 参数 | 参数类型 | 描述 |
|------|--------|-------|
| id | String | 地址组id |
| name | String | 地址组名称 |

状态码：400

表 4-264 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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

删除地址组返回值

```
{  
  "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");
String projectId = "{project_id}";

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

CfwClient client = CfwClient.newBuilder()
    .withCredential(auth)
    .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
    .build();
DeleteAddressSetRequest request = new DeleteAddressSetRequest();
request.withSetId("{set_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.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DeleteAddressSetRequest()
        request.set_id = "{set_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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteAddressSetRequest{}
    request.SetId = "{set_id}"
    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 | 删除地址组返回值 |
| 400 | Bad Request |
| 401 | Unauthorized |
| 403 | Forbidden |

| 状态码 | 描述 |
|-----|-----------------------|
| 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 4-265 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, 可通过 查询地址组成员接口 查询获得, 通过返回值中的data.records.item_id (. 表示各对象之间层级的区分) 获得。 |
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-266 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------------|------|--------|---|
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

请求参数

表 4-267 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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-268 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--------------------------------------|-------------|
| data | AddressItemId object | 删除地址组成员id数据 |

表 4-269 AddressItemId

| 参数 | 参数类型 | 描述 |
|------|--------|-----------|
| id | String | 地址组成员id |
| name | String | 地址组成员ip地址 |

状态码：400

表 4-270 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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"
}
```

状态码

| 状态码 | 描述 |
|-----|-----------------------|
| 200 | 删除地址组成员返回值 |
| 400 | Bad Request |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |
| 500 | Internal Server Error |

错误码

请参见[错误码](#)。

4.5.5 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 4-271 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-272 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

请求参数

表 4-273 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。可通过 如何获取用户Token 获取。 |

表 4-274 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, 可通过 查询地址组列表接口 查询获得, 通过返回值中的data.records.set_id (.表示各对象之间层级的区分) 获得。 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------------|------|------------------|---|
| address_item_ids | 是 | Array of strings | 地址组成员id列表，地址组成员id可通过 查询地址组成员接口 查询获得，通过返回值中的data.records.item_id（.表示各对象之间层级的区分）获得。 |

响应参数

状态码：200

表 4-275 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（id为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

批量删除地址组成员返回值

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

SDK 代码示例

SDK代码示例如下。

Java

删除项目（id为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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchDeleteAddressItemsRequest request = new BatchDeleteAddressItemsRequest();
        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.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

删除项目（id为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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = BatchDeleteAddressItemsRequest()
        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（id为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")
    projectId := "{project_id}"

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

```
client := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.BatchDeleteAddressItemsRequest{}
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 | 批量删除地址组成员返回值 |

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 4-276 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |
| set_id | 是 | String | 地址组id, 可通过 查询地址组列表接口 查询获得, 通过返回值中的data.records.set_id (.表示各对象之间层级的区分) 获得。 |

表 4-277 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

请求参数

表 4-278 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。可通过 如何获取用户Token 获取。 |

表 4-279 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 |

响应参数

状态码：200

表 4-280 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 4-281 UpdateAddressSetResponseData

| 参数 | 参数类型 | 描述 |
|------|--------|-------|
| id | String | 地址组id |
| name | String | 地址组名称 |

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 |
| error_msg | String | 错误描述 |

请求示例

将项目id为9d80d070b6d44942af73c9c3d38e0429下的地址组（id为cf18f0b1-0ce7-4eb8-83b6-4b33c8448e16）的名称改为ABCD。

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

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

响应示例

状态码：200

更新地址组返回值

```
{  
  "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

将项目id为9d80d070b6d44942af73c9c3d38e0429下的地址组（id为cf18f0b1-0ce7-4eb8-83b6-4b33c8448e16）的名称改为ABCD。

```
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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateAddressSetRequest request = new UpdateAddressSetRequest();
        request.withSetId("{set_id}");
        UpdateAddressSetDto body = new UpdateAddressSetDto();
        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.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

```
}  
}
```

Python

将项目id为9d80d070b6d44942af73c9c3d38e0429下的地址组（id为cf18f0b1-0ce7-4eb8-83b6-4b33c8448e16）的名称改为ABCD。

```
# coding: utf-8  
  
import os  
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.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = CfwClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(CfwRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = UpdateAddressSetRequest()  
        request.set_id = "{set_id}"  
        request.body = UpdateAddressSetDto(  
            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

将项目id为9d80d070b6d44942af73c9c3d38e0429下的地址组（id为cf18f0b1-0ce7-4eb8-83b6-4b33c8448e16）的名称改为ABCD。

```
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")
projectId := "{project_id}"

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

client := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateAddressSetRequest{
    request.SetId = "{set_id}"
    descriptionUpdateAddressSetDto:= ""
    nameUpdateAddressSetDto:= "ABCD"
    request.Body = &model.UpdateAddressSetDto{
        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 | 更新地址组返回值 |
| 400 | Bad Request |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |
| 500 | Internal Server Error |

错误码

请参见[错误码](#)。

4.5.7 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 4-283 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-284 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, 通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 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, type可通过 data.records.protect_objects.type (.表示各对象之间层级的区分) 获得 |
| key_word | 否 | String | 关键字, 包括地址组名称或描述的一部分 |
| 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 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------------------|------|---------|---|
| address_type | 否 | Integer | 地址类型0 ipv4, 1 ipv6 |
| enterprise_project_id | 否 | String | 企业项目ID, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |
| 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, 当address_set_type不为0时, query_address_set_type为1时才可以生效。 |
| 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 4-285 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。可通过 如何获取用户Token 获取。 |

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 | data object | 查询地址组列表返回数据 |

表 4-287 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 | 地址组总数 |
| records | Array of records objects | 地址组列表 |

表 4-288 records

| 参数 | 参数类型 | 描述 |
|------------------|---------|---|
| set_id | String | 地址组id |
| ref_count | Integer | 地址组被规则引用次数 |
| description | String | 描述信息 |
| 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，通过返回值中的 data.records.protect_objects.object_id（.表示各对象之间层级的区分）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，type可通过 data.records.protect_objects.type（.表示各对象之间层级的区分）获得 |
| 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 |
| name | String | 地址组名称 |

状态码：400

表 4-289 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|-----|
| error_code | String | 错误码 |

| 参数 | 参数类型 | 描述 |
|-----------|--------|------|
| error_msg | String | 错误描述 |

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

查询地址组列表返回值

```
{
  "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");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        CfwClient client = CfwClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListAddressSetsRequest request = new ListAddressSetsRequest();  
        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.getStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

Python

```
# coding: utf-8  
  
import os  
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")  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = CfwClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(CfwRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ListAddressSetsRequest()  
        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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAddressSetsRequest{}
    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 | 查询地址组列表返回值 |
| 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](#)。

4.5.8 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 4-290 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |
| set_id | 是 | String | 地址组id, 可通过 查询地址组列表接口 查询获得, 通过返回值中的data.records.set_id (.表示各对象之间层级的区分) 获得。 |

表 4-291 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------------------|------|---------|---|
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |
| 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 4-292 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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-293 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 4-294 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 |
| 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 |

状态码：400

表 4-295 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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

查询地址组细节返回体

```
{  
  "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");
String projectId = "{project_id}";

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

CfwClient client = CfwClient.newBuilder()
    .withCredential(auth)
    .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
    .build();
ListAddressSetDetailRequest request = new ListAddressSetDetailRequest();
request.withSetId("{set_id}");
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

import os
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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListAddressSetDetailRequest()
        request.set_id = "{set_id}"
        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")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := cfw.NewCfwClient(  
        cfw.CfwClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.ListAddressSetDetailRequest{}  
    request.SetId = "{set_id}"  
    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 | 查询地址组细节返回体 |
| 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](#)。

4.5.9 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 4-296 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-297 Query 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------|------|---------|--|
| set_id | 是 | String | 地址组id, 可通过 查询地址组列表接口 查询获得, 通过返回值中的data.records.set_id (表示各对象之间层级的区分) 获得。 |
| key_word | 否 | String | 关键字, 包括地址组成员名称或描述的一部分 |
| limit | 是 | Integer | 每页显示个数, 范围为1-1024 |

| 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 |
| address | 否 | String | ip地址 |
| enterprise_project_id | 否 | String | 企业项目ID，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id，可通过 防火墙ID获取方式 获取 |
| 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 4-298 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。可通过 如何获取用户Token 获取。 |

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 | data object | 查询地址组成员返回数据 |

表 4-300 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 | 地址组成员总数 |
| set_id | String | 地址组id |
| records | Array of records objects | 地址组成员记录列表 |

表 4-301 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 | 地址信息 |

状态码: 400

表 4-302 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        CfwClient client = CfwClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListAddressItemsRequest request = new ListAddressItemsRequest();  
        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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListAddressItemsRequest()
        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")
    projectId := "{project_id}"

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



```
WithProjectId(projectId).
Build()

client := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListAddressItemsRequest{}
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](#)。

4.5.10 更新对象配置描述

功能介绍

更新对象配置描述

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/object-config/desc

表 4-303 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

请求参数

表 4-304 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。可通过 如何获取用户Token 获取。 |

表 4-305 请求 Body 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------------|------|--------|--|
| description | 否 | String | 成员描述 |
| fw_instance_id | 否 | String | 防火墙id, 防火墙id, 可通过 防火墙ID获取方式 获取 |
| item_id | 否 | String | 成员id, 地址组成员id, 可通过 查询地址组成员接口 查询获得, 通过返回值中的 data.records.item_id (.表示各对象之间层级的区分) 获得。服务组成员id, 可通过 查询服务成员列表接口 查询获得, 通过返回值中的 data.records.item_id (.表示各对象之间层级的区分) 获得。域名id可通过 获取域名组下域名列表接口 查询获得, 通过返回值中的 data.records.domain_address_id (.表示各对象之间层级的区分) 获得。 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|--------|------|--------|---|
| set_id | 否 | String | 组id，地址组id，可通过 查询地址组列表接口 查询获得，通过返回值中的data.records.set_id（.表示各对象之间层级的区分）获得。服务组id，可通过 获取服务组列表接口 查询获得，通过返回值中的data.records.set_id（.表示各对象之间层级的区分）获得。域名组id，可通过 查询域名组列表接口 查询获得，通过返回值中的data.records.set_id（.表示各对象之间层级的区分）获得。 |
| type | 否 | String | 组类型，包含：地址组 ADDR_SET，服务组 SERV_SET，域名组 DOMAIN_SET，URL组URL_SET |

响应参数

状态码：200

表 4-306 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--------|------|
| data | String | 成员id |

请求示例

更新项目id为408972e72dcd4c1a9b033e955802a36b下的地址组成员描述，防火墙id为b273e6c6-59f3-44cc-9a0d-89824814b474，地址组id为4fc0f061-3f3f-4a16-821c-30d7f6fb895d，地址组成员id为0eac72c9-da3e-4386-b2f2-eb7dbc37d25f，描述为test

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/object-config/desc
{
  "set_id": "4fc0f061-3f3f-4a16-821c-30d7f6fb895d",
  "type": "ADDR_SET",
  "item_id": "0eac72c9-da3e-4386-b2f2-eb7dbc37d25f",
  "description": "test",
  "fw_instance_id": "b273e6c6-59f3-44cc-9a0d-89824814b474"
}
```

响应示例

状态码：200

更新对象配置描述返回值

```
{  
  "data": "0eac72c9-da3e-4386-b2f2-eb7dbc37d25f"  
}
```

状态码

| 状态码 | 描述 |
|-----|-------------|
| 200 | 更新对象配置描述返回值 |

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 4-307 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-308 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------------|------|--------|---|
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

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 | 是 | String | 用户Token。可通过 如何获取用户Token 获取。 |

表 4-310 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, 通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 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, type可通过 data.records.protect_objects.type (.表示各对象之间层级的区分) 获得 |
| name | 是 | String | 服务组名称 |
| description | 否 | String | 服务组描述信息 |

响应参数

状态码: 200

表 4-311 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|-------------------------------------|-----------|
| data | ServiceSetId object | 创建服务组返回数据 |

表 4-312 ServiceSetId

| 参数 | 参数类型 | 描述 |
|------|--------|-------|
| id | String | 服务组Id |
| name | String | 服务组名称 |

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 |
| error_msg | String | 错误描述 |

请求示例

添加项目id为9d80d070b6d44942af73c9c3d38e0429，防护对象为cfefd347-b655-4b84-b938-3c54317599b2，名称为ceshi的服务组。

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/service-set  
  
{  
  "object_id": "cfefd347-b655-4b84-b938-3c54317599b2",  
  "name": "ceshi",  
  "description": ""  
}
```

响应示例

状态码：200

创建服务组返回值

```
{  
  "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");
        String projectId = "{project_id}";

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

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

```
import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = AddServiceSetRequest()
        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")
    projectId := "{project_id}"

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



```
client := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.AddServiceSetRequest{}
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 | 创建服务组返回值 |
| 400 | Bad Request |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |
| 500 | Internal Server Error |

错误码

请参见[错误码](#)。

4.6.2 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 4-314 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

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。可通过 如何获取用户Token 获取。 |

表 4-317 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, 可通过 获取服务组列表接口 查询获得, 通过返回值中的data.records.set_id (表示各对象之间层级的区分) 获得。 |
| service_items | 是 | Array of service_items objects | 服务组成员列表 |

表 4-318 service_items

| 参数 | 是否必选 | 参数类型 | 描述 |
|-------------|------|---------|---|
| 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 |
| source_port | 是 | String | 源端口 |
| dest_port | 是 | String | 目的端口 |
| description | 否 | String | 服务成员描述 |

响应参数

状态码: 200

表 4-319 响应 Body 参数

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

表 4-320 ServiceItemIds

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

表 4-321 items

| 参数 | 参数类型 | 描述 |
|----|--------|---------|
| id | String | 服务组成员id |

状态码: 400

表 4-322 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|-----|
| error_code | String | 错误码 |

| 参数 | 参数类型 | 描述 |
|-----------|--------|------|
| error_msg | String | 错误描述 |

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",
    "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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        AddServiceItemsRequest request = new AddServiceItemsRequest();
        AddServiceItemsUsingPOSTRequestBody body = new AddServiceItemsUsingPOSTRequestBody();
        List<AddServiceItemsUsingPOSTRequestBodyServiceItems> listbodyServiceItems = new ArrayList<>();
        listbodyServiceItems.add(
            new AddServiceItemsUsingPOSTRequestBodyServiceItems()
                .withProtocol(6)
                .withSourcePort("1")
                .withDestPort("1")
                .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.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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = AddServiceItemsRequest()
        listServiceItemsbody = [
            AddServiceItemsUsingPOSTRequestBodyServiceItems(
                protocol=6,
                source_port="1",
                dest_port="1",
                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")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
```

```
Build()

client := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.AddServiceItemsRequest{
    descriptionServiceItems:= "添加服务组成员"
    var listServiceItemsbody = []model.AddServiceItemsUsingPostRequestBodyServiceItems{
        {
            Protocol: int32(6),
            SourcePort: "1",
            DestPort: "1",
            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 4-323 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |
| set_id | 是 | String | 服务组id, 可通过 获取服务组列表接口 查询获得, 通过返回值中的data.records.set_id (.表示各对象之间层级的区分) 获得。 |

表 4-324 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

请求参数

表 4-325 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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-326 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|---------------------|-----------|
| data | ServiceSetId object | 删除服务组返回数据 |

表 4-327 ServiceSetId

| 参数 | 参数类型 | 描述 |
|------|--------|-------|
| id | String | 服务组Id |
| name | String | 服务组名称 |

状态码：400

表 4-328 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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

删除服务组返回值

```
{
  "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"
}
```

状态码

| 状态码 | 描述 |
|-----|-----------------------|
| 200 | 删除服务组返回值 |
| 400 | Bad Request |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |
| 500 | Internal Server Error |

错误码

请参见[错误码](#)。

4.6.4 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 4-329 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |
| item_id | 是 | String | 服务组成员id, 可通过 查询服务成员列表接口 查询获得, 通过返回值中的data.records.item_id (.表示各对象之间层级的区分) 获得。 |

表 4-330 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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id，可通过 防火墙ID获取方式 获取 |

请求参数

表 4-331 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。可通过 如何获取用户Token 获取。 |

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 | DeleteServiceItemResponseBodyData object | 删除服务组成员数据 |

表 4-333 DeleteServiceItemResponseBodyData

| 参数 | 参数类型 | 描述 |
|------|--------|-----------------|
| id | String | 服务组成员id |
| name | String | 服务组成员名称，为源和目的端口 |

状态码：400

表 4-334 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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",
    "name": "服务组成员"
  }
}
```

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");
String projectId = "{project_id}";

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

CfwClient client = CfwClient.newBuilder()
    .withCredential(auth)
    .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
    .build();
DeleteServiceItemRequest request = new DeleteServiceItemRequest();
request.withItemId("{item_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.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrMsg());
}
}
```

Python

```
# coding: utf-8

import os
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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DeleteServiceItemRequest()
        request.item_id = "{item_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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteServiceItemRequest{}
    request.ItemId = "{item_id}"
    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 |
| 400 | Bad Request |
| 401 | Unauthorized |
| 403 | Forbidden |

| 状态码 | 描述 |
|-----|-----------------------|
| 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 4-335 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-336 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

请求参数

表 4-337 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。可通过 如何获取用户Token 获取。 |

表 4-338 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，可通过 获取服务组列表接口 查询获得，通过返回值中的data.records.set_id（.表示各对象之间层级的区分）获得。 |
| service_item_ids | 是 | Array of strings | 服务组成员id列表，服务组成员id可通过 查询服务成员列表接口 查询获得，通过返回值中的data.records.item_id（.表示各对象之间层级的区分）获得。 |

响应参数

状态码：200

表 4-339 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 |

状态码：400

表 4-340 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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" ]  
}
```

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

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");
String projectId = "{project_id}";

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

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

import os
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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

```
try:
    request = BatchDeleteServiceItemsRequest()
    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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteServiceItemsRequest{}
    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 |
| 400 | Bad Request |

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 4-341 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|--|
| project_id | 是 | String | 项目ID, 可以从调API处获取，也可以从控制台获取。 项目ID获取方式 |
| set_id | 是 | String | 服务组id, 可通过 获取服务组列表接口 查询获得，通过返回值中的data.records.set_id（.表示各对象之间层级的区分）获得。 |

表 4-342 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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id，可通过 防火墙ID获取方式 获取 |

请求参数

表 4-343 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。可通过 如何获取用户Token 获取。 |

表 4-344 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 |
| description | 否 | String | 服务组描述信息 |

响应参数

状态码：200

表 4-345 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|-------------------------------------|---------|
| data | ServiceSetId object | 更新服务组数据 |

表 4-346 ServiceSetId

| 参数 | 参数类型 | 描述 |
|----|--------|-------|
| id | String | 服务组Id |

| 参数 | 参数类型 | 描述 |
|------|--------|-------|
| name | String | 服务组名称 |

状态码：400

表 4-347 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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

更新服务组返回值

```
{  
  "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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateServiceSetRequest request = new UpdateServiceSetRequest();
        request.withSetId("{set_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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateServiceSetRequest()
        request.set_id = "{set_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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```



```
request := &model.UpdateServiceSetRequest{}
request.SetId = "{set_id}"
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 | 更新服务组返回值 |
| 400 | Bad Request |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |
| 500 | Internal Server Error |

错误码

请参见[错误码](#)。

4.6.7 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 4-348 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-349 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, 通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 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, type可通过 data.records.protect_objects.type (.表示各对象之间层级的区分) 获得 |
| key_word | 否 | String | 关键字, 可使用功能服务组名称和服务组描述的一部分 |
| limit | 是 | Integer | 每页查询个数, 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |
| 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 4-350 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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-351 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--|-----------|
| data | ServiceSetRecords object | 查询服务组列表记录 |

表 4-352 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 | 查询服务组总数 |
| records | Array of ServiceSet objects | 服务组列表 |

表 4-353 ServiceSet

| 参数 | 参数类型 | 描述 |
|------------------|---------|--|
| set_id | String | 服务组id |
| name | String | 服务组名称 |
| description | String | 服务组描述 |
| 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 | 服务组被规则引用次数 |

| 参数 | 参数类型 | 描述 |
|------------|-------------------|---|
| project_id | String | 项目ID |
| 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,type为0手动类型时不能为空。 |

状态码: 400

表 4-354 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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

查询服务组列表返回值

```
{
  "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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListServiceSetsRequest request = new ListServiceSetsRequest();
        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.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
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.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

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

try:
    request = ListServiceSetsRequest()
    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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListServiceSetsRequest{}
    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](#)。

4.6.8 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 4-355 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|--|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |
| set_id | 是 | String | 服务组id, 可通过 获取服务组列表接口 查询获得, 通过返回值中的data.records.set_id (表示各对象之间层级的区分) 获得。 |

表 4-356 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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id，可通过 防火墙ID获取方式 获取 |
| 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 4-357 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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-358 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--|---------|
| data | ServiceSetDetailResponseDto object | 服务组详情数据 |

表 4-359 ServiceSetDetailResponseDto

| 参数 | 参数类型 | 描述 |
|-------------|--------|---------|
| id | String | 服务组id |
| name | String | 服务组名称 |
| description | String | 服务组描述信息 |

| 参数 | 参数类型 | 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 |

状态码：400

表 4-360 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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

查询服务组详情返回值

```
{
  "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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListServiceSetDetailRequest request = new ListServiceSetDetailRequest();
        request.withSetId("{set_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

import os
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")
    projectId = "{project_id}"
```

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

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

try:
    request = ListServiceSetDetailRequest()
    request.set_id = "{set_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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListServiceSetDetailRequest{}
    request.SetId = "{set_id}"
    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 | 查询服务组详情返回值 |
| 400 | Bad Request |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |
| 500 | Internal Server Error |

错误码

请参见[错误码](#)。

4.6.9 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 4-361 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-362 Query 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|--------|------|--------|--|
| set_id | 是 | String | 服务组id, 可通过 获取服务组列表接口 查询获得, 通过返回值中的data.records.set_id (表示各对象之间层级的区分) 获得。 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------------------|------|---------|---|
| key_word | 否 | String | 查询字段，可为服务组成员名称或服务组成员描述的一部分。 |
| 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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id，可通过 防火墙ID获取方式 获取 |
| 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。仅当set_id为预定义服务组id时生效 |

请求参数

表 4-363 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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-364 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--------------------|---------|
| data | data object | 服务组成员列表 |

表 4-365 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 | 服务组成员总数 |
| set_id | String | 服务组id |
| records | Array of records objects | 记录 |

表 4-366 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 |
| description | String | 服务成员描述 |

状态码：400

表 4-367 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

状态码：401

表 4-368 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|----|--------|----|
| - | String | |

状态码：403**表 4-369 响应 Body 参数**

| 参数 | 参数类型 | 描述 |
|----|--------|----|
| - | String | |

状态码：404**表 4-370 响应 Body 参数**

| 参数 | 参数类型 | 描述 |
|----|--------|----|
| - | String | |

状态码：500**表 4-371 响应 Body 参数**

| 参数 | 参数类型 | 描述 |
|----|--------|----|
| - | String | |

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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListServiceItemsRequest request = new ListServiceItemsRequest();
        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.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```


Python

```
# coding: utf-8

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListServiceItemsRequest()
        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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```
Build()  
  
request := &model.ListServiceItemsRequest{}  
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 |
| 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 4-372 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-373 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 是 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

请求参数

表 4-374 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。可通过 如何获取用户Token 获取。 |

表 4-375 请求 Body 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------------|------|--------|---|
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

| 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, 通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 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, type可通过 data.records.protect_objects.type (.表示各对象之间层级的区分) 获得 |
| name | 是 | String | 域名组名称 |
| description | 否 | String | 域名组描述 |
| domain_names | 是 | Array of DomainSetInfoDto objects | 域名信息列表 |
| 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 |

表 4-376 DomainSetInfoDto

| 参数 | 是否必选 | 参数类型 | 描述 |
|-------------|------|--------|----------------------|
| domain_name | 是 | String | 域名, 如www.example.com |
| description | 否 | String | 域名描述 |

响应参数

状态码: 200

表 4-377 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--|-----------|
| data | DomainSetResponseData object | 添加域名组返回数据 |

表 4-378 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，域名组类型为应用域名组。

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/domain-set?fw_instance_id=546af3f8-88e9-47f2-a205-2346d7090925&enterprise_project_id=default

{
  "name": "test",
  "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，域名组类型为应用域名组。

```
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.*;

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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        AddDomainSetRequest request = new AddDomainSetRequest();
        AddDomainSetInfoDto body = new AddDomainSetInfoDto();
        List<DomainSetInfoDto> listbodyDomainNames = new ArrayList<>();
        listbodyDomainNames.add(
            new DomainSetInfoDto()
                .withDomainName("www.aaa.com")
                .withDescription("")
        );
        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，域名组类型为应用域名组。

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

```
import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = AddDomainSetRequest()
        listDomainNamesbody = [
            DomainSetInfoDto(
                domain_name="www.aaa.com",
                description=""
            )
        ]
        request.body = AddDomainSetInfoDto(
            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，域名组类型为应用域名组。

```
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")
projectId := "{project_id}"

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

client := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.AddDomainSetRequest{
    descriptionDomainNames:= ""
}
var listDomainNamesbody = []model.DomainSetInfoDto{
    {
        DomainName: "www.aaa.com",
        Description: &descriptionDomainNames,
    },
}
descriptionAddDomainSetInfoDto:= ""
fwInstanceIdAddDomainSetInfoDto:= "546af3f8-88e9-47f2-a205-2346d7090925"
request.Body = &model.AddDomainSetInfoDto{
    DomainNames: listDomainNamesbody,
    Description: &descriptionAddDomainSetInfoDto,
    Name: "test",
    Objectid: "fde07429-2e02-45c0-a85f-4f1cacea24d2",
    FwInstanceId: &fwInstanceIdAddDomainSetInfoDto,
}
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 4-379 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|--|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |
| set_id | 是 | String | 域名组id, 可通过 查询域名组列表接口 查询获得, 通过返回值中的data.records.set_id (表示各对象之间层级的区分) 获得。 |

表 4-380 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 是 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

请求参数

表 4-381 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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-382 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 4-383 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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteDomainSetRequest request = new DeleteDomainSetRequest();
        request.withSetId("{set_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.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

```
try:
    request = DeleteDomainSetRequest()
    request.set_id = "{set_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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteDomainSetRequest{}
    request.SetId = "{set_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 4-384 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|--|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |
| set_id | 是 | String | 域名组id, 可通过 查询域名组列表接口 查询获得, 通过返回值中的data.records.set_id (表示各对象之间层级的区分) 获得。 |

表 4-385 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |

| 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 。 |

请求参数

表 4-386 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。可通过 如何获取用户Token 获取。 |

表 4-387 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 |
| description | 否 | String | 域名组描述 |

响应参数

状态码：200

表 4-388 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 4-389 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");
        String projectId = "{project_id}";

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

```
.withSk(sk);

CfwClient client = CfwClient.newBuilder()
    .withCredential(auth)
    .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateDomainSetRequest request = new UpdateDomainSetRequest();
request.withSetId("{set_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.getStatusCode());
    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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateDomainSetRequest()
        request.set_id = "{set_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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateDomainSetRequest{}
    request.SetId = "{set_id}"
    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 4-390 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-391 Query 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|-----------------------|------|--------|---|
| fw_instance_id | 是 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |
| enterprise_project_id | 否 | String | 企业项目ID, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |

请求参数

表 4-392 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。可通过 如何获取用户Token 获取。 |

表 4-393 请求 Body 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|--------------------------|------|---|--|
| dns_server | 是 | Array of dns_server objects | DNS服务器列表 |
| health_check_domain_name | 否 | String | 健康检查域名，可通过 查询dns服务器列表接口 查询获得，通过返回值中的 data.health_check_domain_name（.表示各对象之间层级的区分）获得。 |

表 4-394 dns_server

| 参数 | 是否必选 | 参数类型 | 描述 |
|---------------|------|---------|---|
| server_ip | 是 | String | DNS服务器IP，可通过 查询dns服务器列表接口 查询获得，通过返回值中的 data.server_ip（.表示各对象之间层级的区分）获得。 |
| is_customized | 是 | Integer | 是否是用户自定义的dns服务器，0否 1是 |
| is_applied | 是 | Integer | 是否应用，0否 1是 |

响应参数

状态码：200

表 4-395 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 4-396 响应 Body 参数**

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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?  
fw_instance_id=80e0f2df-24fd-49c2-8398-11f9a0299b3e
```

```
{  
  "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.00109003",  
  "error_msg" : "{ \"error_code\": \"00000012\" }"  
}
```

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");
        String projectId = "{project_id}";

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

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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateDnsServersRequest()
        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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateDnsServersRequest{}
    var listDnsServerbody = []model.UpdateDnsServersRequestBodyDnsServer{
        {
            ServerIp: "8.8.8.8",
            IsCustomized: int32(0),
            IsApplied: int32(1),
        },
        {
            ServerIp: "192.168.0.2",
            IsCustomized: int32(1),
            IsApplied: int32(0),
        },
    }
    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 |

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

错误码

请参见[错误码](#)。

4.7.5 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 4-397 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-398 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 是 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |
| limit | 是 | Integer | 每页显示个数, 范围为1-1024 |

| 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 |
| 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，通过返回值中的data.records.protect_objects.object_id（.表示各对象之间层级的区分）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，type可通过data.records.protect_objects.type（.表示各对象之间层级的区分）获得 |
| key_word | 否 | String | 关键字，可使用域名组名称或描述 |
| 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，-1表示未配置态，0表示配置失败，1表示配置成功，2表示配置中，3表示正常，4表示配置异常 |

请求参数

表 4-399 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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-400 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|-----------------------------------|--------------|
| data | ListDomainsetsResponseData object | 查询域名组列表返回值数据 |

表 4-401 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 | 域名组总数 |
| records | Array of DomainSetVo objects | 域名组列表 |

表 4-402 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 | 域名组描述 |
| ref_count | Integer | 域名组被规则引用次数 |
| 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，-1表示未配置态，0表示配置失败，1表示配置成功，2表示配置中，3表示正常，4表示配置异常 |
| rules | Array of UseRuleVO objects | 使用规则id列表 |

表 4-403 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

查询域名组列表返回值

```
{
  "data": {
    "limit": 50,
    "offset": 0,
    "records": [ {
      "config_status": 3,
      "description": "",
      "domain_set_type": 0,
      "name": "ccdd",
      "ref_count": 0,
      "rules": [ ],
      "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");
    String projectId = "{project_id}";

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

    CfwClient client = CfwClient.newBuilder()
        .withCredential(auth)
        .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
        .build();
    ListDomainSetsRequest request = new ListDomainSetsRequest();
    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.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

import os
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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListDomainSetsRequest()
        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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDomainSetsRequest{}
    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 | 查询域名组列表返回值 |

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 4-404 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-405 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 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |
| enterprise_project_id | 否 | String | 企业项目ID, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |

请求参数

表 4-406 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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-407 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 4-408 DnsServersResponseDTO

| 参数 | 参数类型 | 描述 |
|--------------------------|---------|----------------------------|
| id | Integer | 域名服务器id |
| is_applied | Integer | 域名服务器是否应用，0否 1是 |
| is_customized | Integer | 域名服务器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?  
fw_instance_id=80e0f2df-24fd-49c2-8398-11f9a0299b3e
```

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");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        CfwClient client = CfwClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListDnsServersRequest request = new ListDnsServersRequest();  
        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  
  
import os  
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListDnsServersRequest()
        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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDnsServersRequest{}
    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 4-409 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|-------------|------|--------|--|
| project_id | 是 | String | 项目ID, 可以从调API处获取，也可以从控制台获取。 项目ID获取方式 |
| domain_name | 是 | String | 域名，如www.example.com |

表 4-410 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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id，可通过 防火墙ID获取方式 获取 |
| address_type | 否 | Integer | 地址类型，0 ipv4,1 ipv6 |

请求参数

表 4-411 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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-412 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|------------------|----------|
| data | Array of strings | 域名解析ip列表 |

状态码：400

表 4-413 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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");
        String projectId = "{project_id}";

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

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

```
ListDomainParseDetailRequest request = new ListDomainParseDetailRequest();
request.withDomainName("{domain_name}");
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.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListDomainParseDetailRequest()
        request.domain_name = "{domain_name}"
        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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDomainParseDetailRequest{}
    request.DomainName = "{domain_name}"
    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 |
| 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 4-414 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|---------------|------|--------|--|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |
| domain_set_id | 是 | String | 域名组id, 可通过 查询域名组列表接口 查询获得, 通过返回值中的data.records.set_id (表示各对象之间层级的区分) 获得。 |

表 4-415 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 是 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |
| 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, 如 www.example.com |

| 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，通过返回值中的data.records.protect_objects.object_id（.表示各对象之间层级的区分）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，type可通过data.records.protect_objects.type（.表示各对象之间层级的区分）获得 |

请求参数

表 4-416 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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-417 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|---|-------------|
| data | ListDomainResponseData object | 查询域名列表返回值数据 |

表 4-418 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 | 项目ID |

| 参数 | 参数类型 | 描述 |
|---------|------------------------------------|--------|
| records | Array of DomainInfo objects | 域名信息列表 |
| set_id | String | 域名组id |
| total | Integer | 域名总数 |

表 4-419 DomainInfo

| 参数 | 参数类型 | 描述 |
|-------------------|--------|---------------------|
| domain_address_id | String | 域名地址id |
| domain_name | String | 域名，如www.example.com |
| description | String | 域名描述 |

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

查询域名列表返回值

```
{
  "data": {
    "limit": 200,
    "offset": 0,
    "project_id": "14181c1245cf4fd786824efe1e2b9388",
    "records": [ {
      "description": "",
      "domain_address_id": "6718279e-9761-4623-a48d-b16957b19e1b",
      "domain_name": "www.example.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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListDomainsRequest request = new ListDomainsRequest();
        request.withDomainSetId("{domain_set_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

import os
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.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

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

try:
    request = ListDomainsRequest()
    request.domain_set_id = "{domain_set_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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDomainsRequest{}
    request.DomainSetId = "{domain_set_id}"
    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 | 查询域名列表返回值 |

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 4-420 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取，也可以从控制台获取。 项目ID获取方式 |
| set_id | 是 | String | 域名组id，可通过 查询域名组列表接口 查询获得，通过返回值中的data.records.set_id（.表示各对象之间层级的区分）获得。 |

表 4-421 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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id，可通过 防火墙ID获取方式 获取 |

请求参数

表 4-422 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。可通过 如何获取用户Token 获取。 |

表 4-423 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 。 |
| 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，通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 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，type可通过 data.records.protect_objects.type (.表示各对象之间层级的区分) 获得 |
| domain_names | 是 | Array of DomainSetInfoDto objects | 域名列表 |

表 4-424 DomainSetInfoDto

| 参数 | 是否必选 | 参数类型 | 描述 |
|-------------|------|--------|----------------------|
| domain_name | 是 | String | 域名, 如www.example.com |
| description | 否 | String | 域名描述 |

响应参数

状态码: 200

表 4-425 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 4-426 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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        AddDomainsRequest request = new AddDomainsRequest();
        request.withSetId("{set_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.getErrMsg());
}
}
```

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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = AddDomainsRequest()
        request.set_id = "{set_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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddDomainsRequest{}
    request.SetId = "{set_id}"
    descriptionDomainNames:= ""
    descriptionDomainNames1:= ""
    var listDomainNamesbody = []model.DomainSetInfoDto{
        {
            DomainName: "www.bnm.com",
            Description: &descriptionDomainNames,
        },
        {
            DomainName: "www.vbc.com",
            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 4-427 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取，也可以从控制台获取。 项目ID获取方式 |
| set_id | 是 | String | 域名组id, 可通过 查询域名组列表接口 查询获得，通过返回值中的data.records.set_id（表示各对象之间层级的区分）获得。 |

表 4-428 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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id，可通过 防火墙ID获取方式 获取 |

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。可通过 如何获取用户Token 获取。 |

表 4-430 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，通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 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，type可通过 data.records.protect_objects.type (.表示各对象之间层级的区分) 获得 |
| domain_address_ids | 是 | Array of strings | 域名id列表,域名id可通过 获取域名组下域名列表接口 查询获得，通过返回值中的 data.records.domain_address_id (.表示各对象之间层级的区分) 获得。 |

响应参数

状态码：200

表 4-431 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数据 |

表 4-432 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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteDomainsRequest request = new DeleteDomainsRequest();
        request.withSetId("{set_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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DeleteDomainsRequest()
        request.set_id = "{set_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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteDomainsRequest{}
    request.SetId = "{set_id}"
    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](#)。

4.7.11 查看域名组详情

功能介绍

查看域名组详情

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/domain-set/{domain_set_id}

表 4-433 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|---------------|------|--------|--|
| project_id | 是 | String | 项目ID，可以从调API处获取，也可以从控制台获取。可通过 项目ID获取方式 获取 |
| domain_set_id | 是 | String | 域名组ID，可通过 查询域名组列表接口 查询获得，通过返回值中的data.records.set_id（.表示各对象之间层级的区分）获取 |

表 4-434 Query 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|-----------------------|------|--------|---|
| fw_instance_id | 是 | String | 防火墙ID，可通过 防火墙ID获取方式 获取 |
| enterprise_project_id | 否 | String | 企业项目ID，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

请求参数

表 4-435 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，可通过 如何获取用户Token 获取 |

响应参数

状态码：200

表 4-436 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|------------------------------|---------|
| data | DomainSetVo object | 域名组详细信息 |

表 4-437 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 | 域名组描述 |
| ref_count | Integer | 域名组被规则引用次数 |
| 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，-1表示未配置态，0表示配置失败，1表示配置成功，2表示配置中，3表示正常，4表示配置异常 |
| rules | Array of UseRuleVO objects | 使用规则id列表 |

表 4-438 UseRuleVO

| 参数 | 参数类型 | 描述 |
|------|--------|------|
| id | String | 规则id |
| name | String | 规则名称 |

请求示例

查询域名组id为7891eb51-009a-4e4e-ab15-e4290d9360b2的详细信息，防火墙id为fced7179-64a4-4438-bc6d-b510a2da2f64，项目id为408972e72dcd4c1a9b033e955802a36b

https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/domain-set/7891eb51-009a-4e4e-ab15-e4290d9360b2?fw_instance_id=fced7179-64a4-4438-bc6d-b510a2da2f64

响应示例

状态码：200

域名组详细信息响应值

```
{
  "data" : {
    "config_status" : 3,
    "domain_set_type" : 0,
    "name" : "应用组2",
    "set_id" : "7891eb51-009a-4e4e-ab15-e4290d9360b2"
  }
}
```

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 ShowDomainSetDetailSolution {

    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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowDomainSetDetailRequest request = new ShowDomainSetDetailRequest();
        request.withDomainSetId("{domain_set_id}");
        try {
            ShowDomainSetDetailResponse response = client.showDomainSetDetail(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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowDomainSetDetailRequest()
        request.domain_set_id = "{domain_set_id}"
        response = client.show_domain_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")
    projectId := "{project_id}"

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

```
WithProjectId(projectId).
Build()

client := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowDomainSetDetailRequest{}
request.DomainSetId = "{domain_set_id}"
response, err := client.ShowDomainSetDetail(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 | 域名组详细信息响应值 |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |

错误码

请参见[错误码](#)。

4.7.12 获取域名地址解析结果

功能介绍

获取域名地址解析结果

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/domain/parse-ip-list/{domain_address_id}

表 4-439 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|-------------------|------|--------|--|
| domain_address_id | 是 | String | 域名id, 域名id可通过 获取域名组下域名列表接口 查询获得, 通过返回值中的 data.records.domain_address_id (.表示各对象之间层级的区分) 获取 |
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。可通过 项目ID获取方式 获取 |

表 4-440 Query 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 |
| domain_set_id | 是 | String | 域名组ID, 可通过 查询域名组列表接口 查询获得, 通过返回值中的 data.records.set_id (.表示各对象之间层级的区分) 获取 |
| fw_instance_id | 是 | String | 防火墙ID, 可通过 防火墙ID获取方式 获取 |
| enterprise_project_id | 否 | String | 企业项目ID, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |

请求参数

表 4-441 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, 可通过 如何获取用户Token 获取 |

响应参数

状态码: 200

表 4-442 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|------------------|-----------------|
| data | Array of strings | 域名解析数据，包括域名ip列表 |

请求示例

查询域名id为a43a844e-26cf-4319-87c4-5c714888dbc5的IP列表，防火墙id为fced7179-64a4-4438-bc6d-b510a2da2f64，域名组id为2cd83033-67d0-4515-9d72-c40caa11e8c6，项目id为408972e72dcd4c1a9b033e955802a36b

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/domain/parse-ip-list/  
a43a844e-26cf-4319-87c4-5c714888dbc5?fw_instance_id=fced7179-64a4-4438-bc6d-  
b510a2da2f64&domain_set_id=2cd83033-67d0-4515-9d72-  
c40caa11e8c6&domain_address_id=a43a844e-26cf-4319-87c4-5c714888dbc5
```

响应示例

状态码：200

域名的ip解析数据

```
{ }
```

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 ListDomainParseIpSolution {  
  
    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");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        CfwClient client = CfwClient.newBuilder()  
            .withCredential(auth)
```

```
        .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
        .build();
ListDomainParseIpRequest request = new ListDomainParseIpRequest();
request.withDomainAddressId("{domain_address_id}");
try {
    ListDomainParseIpResponse response = client.listDomainParseIp(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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListDomainParseIpRequest()
        request.domain_address_id = "{domain_address_id}"
        response = client.list_domain_parse_ip(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")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := cfw.NewCfwClient(  
        cfw.CfwClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.ListDomainParseIpRequest{}  
    request.DomainAddressId = "{domain_address_id}"  
    response, err := client.ListDomainParseIp(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 | 域名的ip解析数据 |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |

错误码

请参见[错误码](#)。

4.7.13 批量删除域名组

功能介绍

批量删除域名组

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/domain-sets/batch-delete

表 4-443 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID，可以从调API处获取，也可以从控制台获取。可通过 项目ID获取方式 获取 |

表 4-444 Query 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|-----------------------|------|--------|---|
| fw_instance_id | 否 | String | 防火墙ID，可通过 防火墙ID获取方式 获取 |
| enterprise_project_id | 否 | String | 企业项目ID，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

请求参数

表 4-445 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，可通过 如何获取用户Token 获取 |

表 4-446 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, 通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 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, type可通过 data.records.protect_objects.type (.表示各对象之间层级的区分) 获得 |
| set_ids | 否 | Array of strings | 域名组id列表, 域名组id, 可通过 查询域名组列表接口 查询获得, 通过返回值中的 data.records.set_id (.表示各对象之间层级的区分) 获得。 |

响应参数

状态码: 200

表 4-447 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|---|----|
| data | HttpBatchDeleteDomainSetResponseData s object | |

表 4-448 HttpBatchDeleteDomainSetResponseData

| 参数 | 参数类型 | 描述 |
|--------------|--|---------------|
| responseData | Array of DomainSetId objects | 批量删除域名组返回数据列表 |

表 4-449 DomainSetId

| 参数 | 参数类型 | 描述 |
|------|--------|-------|
| name | String | 域名组名称 |
| id | String | 域名组id |

请求示例

批量删除域名组，防火墙id为fced7179-64a4-4438-bc6d-b510a2da2f64，目标防护对象id为b71922fd-3539-421c-9317-c54ebc1bc4ea，待删除的域名组id为7d772766-6c44-4fa3-a140-8ddc7eac0fc5和e17e24d2-bd7d-4e25-993c-38559d44a571，项目id为408972e72dcd4c1a9b033e955802a36b。

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/domain-sets/batch-delete
```

```
{
  "object_id": "b71922fd-3539-421c-9317-c54ebc1bc4ea",
  "set_ids": [ "7d772766-6c44-4fa3-a140-8ddc7eac0fc5", "e17e24d2-bd7d-4e25-993c-38559d44a571" ]
}
```

响应示例

状态码：200

批量删除域名组请求的响应数据

```
{
  "data": {
    "responseDatas": [ {
      "name": "9",
      "id": "7d772766-6c44-4fa3-a140-8ddc7eac0fc5"
    }, {
      "name": "8",
      "id": "e17e24d2-bd7d-4e25-993c-38559d44a571"
    } ]
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

批量删除域名组，防火墙id为fced7179-64a4-4438-bc6d-b510a2da2f64，目标防护对象id为b71922fd-3539-421c-9317-c54ebc1bc4ea，待删除的域名组id为7d772766-6c44-4fa3-a140-8ddc7eac0fc5和e17e24d2-bd7d-4e25-993c-38559d44a571，项目id为408972e72dcd4c1a9b033e955802a36b。

```
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 BatchDeleteDomainSetSolution {

    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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchDeleteDomainSetRequest request = new BatchDeleteDomainSetRequest();
        BatchDeleteDomainSetsDto body = new BatchDeleteDomainSetsDto();
        List<String> listbodySetIds = new ArrayList<>();
        listbodySetIds.add("7d772766-6c44-4fa3-a140-8ddc7eac0fc5");
        listbodySetIds.add("e17e24d2-bd7d-4e25-993c-38559d44a571");
        body.withSetIds(listbodySetIds);
        body.withObjectId("b71922fd-3539-421c-9317-c54ebc1bc4ea");
        request.withBody(body);
        try {
            BatchDeleteDomainSetResponse response = client.batchDeleteDomainSet(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

批量删除域名组，防火墙id为fced7179-64a4-4438-bc6d-b510a2da2f64，目标防护对象id为b71922fd-3539-421c-9317-c54ebc1bc4ea，待删除的域名组id为7d772766-6c44-4fa3-a140-8ddc7eac0fc5和e17e24d2-bd7d-4e25-993c-38559d44a571，项目id为408972e72dcd4c1a9b033e955802a36b。

```
# coding: utf-8

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = BatchDeleteDomainSetRequest()
        listSetIdsbody = [
            "7d772766-6c44-4fa3-a140-8ddc7eac0fc5",
            "e17e24d2-bd7d-4e25-993c-38559d44a571"
        ]
        request.body = BatchDeleteDomainSetsDto(
            set_ids=listSetIdsbody,
            object_id="b71922fd-3539-421c-9317-c54ebc1bc4ea"
        )
        response = client.batch_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

批量删除域名组，防火墙id为fced7179-64a4-4438-bc6d-b510a2da2f64，目标防护对象id为b71922fd-3539-421c-9317-c54ebc1bc4ea，待删除的域名组id为7d772766-6c44-4fa3-a140-8ddc7eac0fc5和e17e24d2-bd7d-4e25-993c-38559d44a571，项目id为408972e72dcd4c1a9b033e955802a36b。

```
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")
    projectId := "{project_id}"

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

```
WithProjectId(projectId).
Build()

client := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.BatchDeleteDomainSetRequest{
var listSetIdsbody = []string{
    "7d772766-6c44-4fa3-a140-8ddc7eac0fc5",
    "e17e24d2-bd7d-4e25-993c-38559d44a571",
}
objectIdBatchDeleteDomainSetsDto:= "b71922fd-3539-421c-9317-c54ebc1bc4ea"
request.Body = &model.BatchDeleteDomainSetsDto{
    SetIds: &listSetIdsbody,
    ObjectId: &objectIdBatchDeleteDomainSetsDto,
}
response, err := client.BatchDeleteDomainSet(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 | 批量删除域名组请求的响应数据 |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |

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 4-450 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-451 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, 通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 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。此处仅取type为0的防护对象id, 可通过 data.records.protect_objects.type (.表示各对象之间层级的区分) 获得。 |
| enterprise_project_id | 否 | String | 企业项目ID, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

请求参数

表 4-452 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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-453 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|-----------------------------|--------------|
| data | IpsSwitchResponseDTO object | 查询ips开关返回值数据 |

表 4-454 IpsSwitchResponseDTO

| 参数 | 参数类型 | 描述 |
|------------------------|---------|--|
| id | String | ips开关id，此处为互联网边界防护对象id，可通过调用 查询防火墙实例接口 获得，通过返回值中的data.records.protect_objects.object_id（.表示各对象之间层级的区分）获得 |
| basic_defense_status | Integer | 基础防御状态，0表示关闭，1表示开启 |
| virtual_patches_status | Integer | 虚拟补丁状态，0表示关闭，1表示开启 |

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

查询ips开关返回值

```
{  
  "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");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        CfwClient client = CfwClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListIpsSwitchStatusRequest request = new ListIpsSwitchStatusRequest();  
        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
```

```
import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListIpsSwitchStatusRequest()
        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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListIpsSwitchStatusRequest{}
    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 | 查询ips开关返回值 |
| 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 4-455 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|--|
| project_id | 是 | String | 项目ID, 可以从调API处获取，也可以从控制台获取。 项目ID获取方式 |

表 4-456 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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id，可通过 防火墙ID获取方式 获取 |

请求参数

表 4-457 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。可通过 如何获取用户Token 获取。 |

表 4-458 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，通过返回值中的data.records.protect_objects.object_id（.表示各对象之间层级的区分）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。此处仅取type为0的防护对象id，可通过data.records.protect_objects.type（.表示各对象之间层级的区分）获得。 |
| ips_type | 是 | Integer | 补丁类型，仅支持虚拟补丁，值为2。 |
| status | 是 | Integer | ips特性开关状态，0表示关闭，1表示开启 |

响应参数

状态码：200

表 4-459 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--------------------|----------------|
| data | data object | 修改ips防护模式返回值数据 |

表 4-460 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

修改IPS开关返回值

```
{
  "data": {
    "id": "1530de8a-522d-4771-9067-9fa4e2f53b48"
  }
}
```

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");
        String projectId = "{project_id}";

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

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

        ChangelpsSwitchStatusRequest request = new ChangelpsSwitchStatusRequest();
        IpsSwitchDTO body = new IpsSwitchDTO();
        body.withStatus(1);
        body.withIpsType(IpsSwitchDTO.IpsTypeEnum.NUMBER_1);
        body.withObjectId("1530de8a-522d-4771-9067-9fa4e2f53b48");
        request.withBody(body);
        try {
            ChangelpsSwitchStatusResponse response = client.changelpsSwitchStatus(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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ChangeIpsSwitchStatusRequest()
        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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```
Build()  
  
request := &model.ChangepsSwitchStatusRequest{  
    request.Body = &model.IpsSwitchDto{  
        Status: int32(1),  
        IpsType: model.GetIpsSwitchDtoIpsTypeEnum().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 | 修改IPS开关返回值 |
| 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 4-461 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-462 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, 通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 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。此处仅取type为0的防护对象id, 可通过 data.records.protect_objects.type (.表示各对象之间层级的区分) 获得。 |
| enterprise_project_id | 否 | String | 企业项目ID, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

请求参数

表 4-463 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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-464 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|-----------------------------|----------------|
| data | IpsProtectModeObject object | 查询ips防护模式返回值数据 |

表 4-465 IpsProtectModeObject

| 参数 | 参数类型 | 描述 |
|------|---------|---|
| id | String | ips防护模式id, 此处为防护对象id, 可通过调用 查询防火墙实例接口 获得, 通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 获得 |
| 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 4-466 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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: 200

查询防护模式返回值

```
{
  "data" : {
    "id" : "d5b75aba-dfca-40e4-99dd-ed56578e8e48",
    "mode" : 0
  }
}
```

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");
        String projectId = "{project_id}";

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

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

```
import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListIpsProtectModeRequest()
        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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListIpsProtectModeRequest{}
    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 | 查询防护模式返回值 |
| 400 | Bad Request |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |
| 500 | Internal Server Error |

错误码

请参见[错误码](#)。

4.8.4 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 4-467 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|--|
| project_id | 是 | String | 项目ID, 可以从调API处获取，也可以从控制台获取。 项目ID获取方式 |

表 4-468 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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |
| fw_instance_id | 否 | String | 防火墙id，可通过 防火墙ID获取方式 获取 |

请求参数

表 4-469 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。可通过 如何获取用户Token 获取。 |

表 4-470 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 ，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。 |
| mode | 是 | 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 4-471 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--------------------|----------------|
| data | data object | 修改ips防护模式返回值数据 |

表 4-472 data

| 参数 | 参数类型 | 描述 |
|----|--------|--------|
| id | String | 防护对象id |

状态码：400

表 4-473 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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

修改ips防护模式请求体

```
{
  "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");
        String projectId = "{project_id}";

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

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

import os
```



```
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ChangeIpsProtectModeRequest()
        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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
```

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

request := &model.ChangepsProtectModeRequest{
request.Body = &model.IpsProtectDto{
    Mode: int32(1),
    Objectid: "cfebd347-b655-4b84-b938-3c54317599b2",
}
response, err := client.ChangepsProtectMode(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 | 修改ips防护模式请求体 |
| 400 | Bad Request |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |
| 500 | Internal Server Error |

错误码

请参见[错误码](#)。

4.8.5 改变 ips 规则模式

功能介绍

改变ips规则模式

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/ips-rule/mode

表 4-474 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID，可以从调API处获取，也可以从控制台获取。可通过 项目ID获取方式 获取 |

表 4-475 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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

请求参数

表 4-476 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，可通过 如何获取用户Token 获取 |

表 4-477 请求 Body 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|-----------|------|------------------|--|
| ips_ids | 否 | Array of strings | ips的id列表，Ips规则id，可通过 获取ips规则列表 查询获得，通过返回值中的data.records.ips_id（.表示各对象之间层级的区分）获得。 |
| object_id | 否 | String | 防护对象id |
| status | 否 | String | ips规则状态 |

响应参数

状态码：200

表 4-478 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|------------------------------|----|
| data | IpsRuleChangeRespBody object | |

表 4-479 IpsRuleChangeRespBody

| 参数 | 参数类型 | 描述 |
|------------|------------------|----------|
| error_code | String | 错误代码 |
| error_msg | String | 错误信息 |
| group_id | String | 分组id |
| id | String | 防火墙id |
| ips_ids | Array of strings | ips的id列表 |
| result | Boolean | 修改结果 |

请求示例

更改ips规则模式为"OBSERVE"

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/ips-rule/mode?fw_instance_id=e743cfaf-8164-4807-aa13-d893d83313cf&enterprise_project_id=fb55459c-41b3-47fc-885d-540946fddda4
```

```
{
  "object_id": "1b90f031-0c7b-4f25-95e2-b6d9940d269e",
  "status": "OBSERVE",
  "ips_ids": [ "340724" ]
}
```

响应示例

状态码：200

OK

```
{
  "data": {
    "group_id": "e743cfaf-8164-4807-aa13-d893d83313cf",
    "id": "e743cfaf-8164-4807-aa13-d893d83313cf",
    "ips_ids": [ "340724" ],
    "result": true
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

更改ips规则模式为"OBSERVE"

```
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 ChangepsRuleModeSolution {

    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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ChangepsRuleModeRequest request = new ChangepsRuleModeRequest();
        IpsRuleChangeDto body = new IpsRuleChangeDto();
        List<String> listbodyIpsIds = new ArrayList<>();
        listbodyIpsIds.add("340724");
        body.withStatus(IpsRuleChangeDto.StatusEnum.fromValue("OBSERVE"));
        body.withObjectId("1b90f031-0c7b-4f25-95e2-b6d9940d269e");
        body.withIpsIds(listbodyIpsIds);
        request.withBody(body);
        try {
            ChangepsRuleModeResponse response = client.changepsRuleMode(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

更改ips规则模式为"OBSERVE"

```
# coding: utf-8

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ChangeIpsRuleModeRequest()
        listIpsIdsbody = [
            "340724"
        ]
        request.body = IpsRuleChangeDto(
            status="OBSERVE",
            object_id="1b90f031-0c7b-4f25-95e2-b6d9940d269e",
            ips_ids=listIpsIdsbody
        )
        response = client.change_ips_rule_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

更改ips规则模式为"OBSERVE"

```
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")
    projectId := "{project_id}"

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

```
WithSk(sk).
WithProjectId(projectId).
Build()

client := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ChangepsRuleModeRequest{}
var listIpsIdsbody = []string{
    "340724",
}
statusIpsRuleChangeDto:= model.GetIpsRuleChangeDtoStatusEnum().OBSERVE
objectIdIpsRuleChangeDto:= "1b90f031-0c7b-4f25-95e2-b6d9940d269e"
request.Body = &model.IpsRuleChangeDto{
    Status: &statusIpsRuleChangeDto,
    Objectid: &objectIdIpsRuleChangeDto,
    Ipsids: &listIpsIdsbody,
}
response, err := client.ChangepsRuleMode(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 |
| 201 | Created |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |

错误码

请参见[错误码](#)。

4.8.6 创建频率 ips 规则

功能介绍

创建频率ips规则

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/advanced-ips-rule

表 4-480 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID，可以从调API处获取，也可以从控制台获取。可通过 项目ID获取方式 获取 |

表 4-481 Query 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|-----------------------|------|--------|---|
| fw_instance_id | 否 | String | 防火墙ID，可通过 防火墙ID获取方式 获取 |
| enterprise_project_id | 否 | String | 企业项目ID，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

请求参数

表 4-482 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，可通过 如何获取用户Token 获取 |

表 4-483 请求 Body 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|-------------|------|---------|----------------------------------|
| action | 否 | Integer | 动作：0表示仅记录日志、1表示拦截session、2表示拦截ip |
| ips_rule_id | 否 | String | 高级ips规则id |

| 参数 | 是否必选 | 参数类型 | 描述 |
|---------------|------|---------|-------------------------------|
| ips_rule_type | 否 | Integer | ips规则类型：0表示敏感目录扫描、1表示反弹xshell |
| object_id | 否 | String | 防护对象id |
| param | 否 | String | 包含特殊参数的JSON字符串 |
| status | 否 | Integer | |

响应参数

状态码：200

表 4-484 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|---------------------|-------|
| data | ResponseData object | 通用返回体 |

表 4-485 ResponseData

| 参数 | 参数类型 | 描述 |
|----|--------|------|
| id | String | 数据id |

请求示例

更改反弹Shell检测防御的配置为：观察模式|低误报

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/advanced-ips-rule?fw_instance_id=e743cfaf-8164-4807-aa13-d893d83313cf&enterprise_project_id=fb55459c-41b3-47fc-885d-540946fddda4

{
  "ips_rule_id": "7d453586-524f-4a1d-a827-34d80389378f",
  "ips_rule_type": 0,
  "action": 0,
  "param": "{\"mode\":\"0\"}",
  "status": 1,
  "object_id": "1b90f031-0c7b-4f25-95e2-b6d9940d269e"
}
```

响应示例

状态码：200

OK

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

```
"id" : "7d453586-524f-4a1d-a827-34d80389378f"  
}  
}
```

SDK 代码示例

SDK代码示例如下。

Java

更改反弹Shell检测防御的配置为：观察模式|低误报

```
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 UpdateAdvancedIpsRuleSolution {  
  
    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");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        CfwClient client = CfwClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))  
            .build();  
        UpdateAdvancedIpsRuleRequest request = new UpdateAdvancedIpsRuleRequest();  
        AdvancedIpsRuleDto body = new AdvancedIpsRuleDto();  
        body.withStatus(1);  
        body.withParam("{\"mode\":0}");  
        body.withObjectId("1b90f031-0c7b-4f25-95e2-b6d9940d269e");  
        body.withIpsRuleType(0);  
        body.withIpsRuleId("7d453586-524f-4a1d-a827-34d80389378f");  
        body.withAction(0);  
        request.withBody(body);  
        try {  
            UpdateAdvancedIpsRuleResponse response = client.updateAdvancedIpsRule(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

更改反弹Shell检测防御的配置为：观察模式|低误报

```
# coding: utf-8  
  
import os  
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.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = CfwClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(CfwRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = UpdateAdvancedIpsRuleRequest()  
        request.body = AdvancedIpsRuleDto(  
            status=1,  
            param="{\"mode\":0}",  
            object_id="1b90f031-0c7b-4f25-95e2-b6d9940d269e",  
            ips_rule_type=0,  
            ips_rule_id="7d453586-524f-4a1d-a827-34d80389378f",  
            action=0  
        )  
        response = client.update_advanced_ips_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

更改反弹Shell检测防御的配置为：观察模式|低误报

```
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")
projectId := "{project_id}"

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

client := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateAdvancedIpsRuleRequest{
    statusAdvancedIpsRuleDto:= int32(1)
    paramAdvancedIpsRuleDto:= "{\"mode\":0}"
    objectIdAdvancedIpsRuleDto:= "1b90f031-0c7b-4f25-95e2-b6d9940d269e"
    ipsRuleTypeAdvancedIpsRuleDto:= int32(0)
    ipsRuleIdAdvancedIpsRuleDto:= "7d453586-524f-4a1d-a827-34d80389378f"
    actionAdvancedIpsRuleDto:= int32(0)
    request.Body = &model.AdvancedIpsRuleDto{
        Status: &statusAdvancedIpsRuleDto,
        Param: &paramAdvancedIpsRuleDto,
        ObjectId: &objectIdAdvancedIpsRuleDto,
        IpsRuleType: &ipsRuleTypeAdvancedIpsRuleDto,
        IpsRuleId: &ipsRuleIdAdvancedIpsRuleDto,
        Action: &actionAdvancedIpsRuleDto,
    }
}
response, err := client.UpdateAdvancedIpsRule(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 |
| 201 | Created |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |

错误码

请参见[错误码](#)。

4.8.7 查询频率 ips 规则信息

功能介绍

查询频率ips规则信息

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/advanced-ips-rules

表 4-486 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID，可以从调API处获取，也可以从控制台获取。可通过 项目ID获取方式 获取 |

表 4-487 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，通过返回值中的 data.records.protect_objects.object_id (.表示各对象之间层级的区分) 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。此处仅取type为1的防护对象id，可通过 data.records.protect_objects.type (.表示各对象之间层级的区分) 获得。 |
| enterprise_project_id | 否 | String | 企业项目ID，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

请求参数

表 4-488 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, 可通过 如何获取用户Token 获取 |

响应参数

状态码: 200

表 4-489 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--|-----------------|
| data | AdvancedIpsRuleListVo object | 查询频率ips规则列表返回数据 |

表 4-490 AdvancedIpsRuleListVo

| 参数 | 参数类型 | 描述 |
|--------------------|--|-----------|
| advanced_ips_rules | Array of AdvancedIpsRuleVo objects | 频率ips规则列表 |
| total | Integer | 频率ips规则总数 |

表 4-491 AdvancedIpsRuleVo

| 参数 | 参数类型 | 描述 |
|---------------|---------|-----------------------------------|
| action | Integer | 动作: 0表示仅记录日志、1表示拦截session、2表示拦截ip |
| ips_rule_id | String | 频率ips规则id |
| ips_rule_type | Integer | ips规则类型: 0表示敏感目录扫描、1表示反弹shell |
| param | String | 频率ips定义JSON字符串 |
| status | Integer | 频率ips规则状态, 0表示关闭, 1表示开启 |

请求示例

查询项目id为408972e72dcd4c1a9b033e955802a36b的频率IPS规则信息，防火墙id为e743cfaf-8164-4807-aa13-d893d83313cf，企业项目id为fb55459c-41b3-47fc-885d-540946fddda4，目标对象id为1b90f031-0c7b-4f25-95e2-b6d9940d269e。

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/advanced-ips-rules?fw_instance_id=e743cfaf-8164-4807-aa13-d893d83313cf&enterprise_project_id=fb55459c-41b3-47fc-885d-540946fddda4&object_id=1b90f031-0c7b-4f25-95e2-b6d9940d269e
```

响应示例

状态码：200

查询频率ips规则返回值

```
{
  "data": {
    "advanced_ips_rules": [ {
      "action": 0,
      "ips_rule_id": "7d453586-524f-4a1d-a827-34d80389378f",
      "ips_rule_type": 0,
      "param": "{\"mode\":\"0\"}",
      "status": 1
    }, {
      "action": 0,
      "ips_rule_id": "ed184f33-b4e2-4c42-9562-313563e60b7c",
      "ips_rule_type": 1,
      "param": "{\"threshold\":\"10\"}",
      "status": 1
    }
  ],
  "total": 2
}
```

状态码

| 状态码 | 描述 |
|-----|--------------|
| 200 | 查询频率ips规则返回值 |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |

错误码

请参见[错误码](#)。

4.8.8 获取 ips 规则列表

功能介绍

获取ips规则列表

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/ips-rule

表 4-492 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID，可以从调API处获取，也可以从控制台获取。可通过 项目ID获取方式 获取 |

表 4-493 Query 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|---------------------------|------|---------|---|
| affected_application_like | 否 | Integer | 受影响对象查询关键字，可包含如下：Others、Sun、Apache、IBM、VMware、WordPress、Adobe、Oracle、Google Chrome等 |
| create_time | 否 | Integer | ips规则创建的年份 |
| fw_instance_id | 否 | String | 防火墙ID，可通过 防火墙ID获取方式 获取 |
| ips_cve_like | 否 | Integer | cve id查询关键字，cve id为cve漏洞库中存储的漏洞id |
| ips_group | 否 | Integer | ips组，使用ips规则拦截模式区分，包含，0：观察模式，1：严格模式，2：中等模式，3：宽松模式 |
| ips_id | 否 | String | ips规则id |
| ips_level | 否 | Integer | ips严重等级，包含CRITICAL、HIGH、MEDIUM、LOW |
| ips_name_like | 否 | String | ips规则名称查询关键字 |
| ips_rules_type_like | 否 | Integer | ips规则类型，包括漏洞扫描、黑客工具、特洛伊木马等 |
| ips_status | 否 | String | ips规则状态，包含观察：OBSERVE、拦截：ENABLE、禁用：CLOSE、恢复默认：DEFAULT、全局恢复默认：ALL_DEFAULT |

| 参数 | 是否必选 | 参数类型 | 描述 |
|-----------------------------|------|---------|--|
| is_updated_ips_rule_queried | 否 | Boolean | 是否查询虚拟补丁相对基础防御更新规则，是表示true,否表示false |
| limit | 是 | Integer | 每页显示个数，范围为1-1024 |
| 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，通过返回值中的data.records.protect_objects.object_id（.表示各对象之间层级的区分）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。此处仅取type为1的防护对象id，可通过data.records.protect_objects.type（.表示各对象之间层级的区分）获得。 |
| 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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

请求参数

表 4-494 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，可通过 如何获取用户Token 获取 |

响应参数

状态码：200

表 4-495 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|-------------------------|-------------|
| data | IpsRuleListVO object | 查询规则列表返回值数据 |

表 4-496 IpsRuleListVO

| 参数 | 参数类型 | 描述 |
|----------------|----------------------------------|---------------------------------------|
| fw_instance_id | String | 防火墙id |
| limit | Integer | 每页显示个数，范围为1-1024 |
| object_id | String | 防护对象id |
| 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 |
| records | Array of IpsRuleVO objects | 查询ips规则列表 |
| total | Integer | 查询ips规则总数 |

表 4-497 IpsRuleVO

| 参数 | 参数类型 | 描述 |
|----------------------|--------|--|
| affected_application | String | 受影响对象，可包含如下：Others、Sun、Apache、IBM、VMware、WordPress、Adobe、Oracle、Google Chrome等 |
| create_time | String | ips规则创建的年份 |
| default_status | String | 默认状态 |
| ips_cve | String | cve id |
| ips_group | String | ips组，使用ips规则拦截模式区分，包含，0：观察模式，1：严格模式，2：中等模式，3：宽松模式 |
| ips_id | String | ips规则id |
| ips_level | String | ips严重等级， ips严重等级，包含CRITICAL、HIGH、MEDIUM、LOW |
| ips_name | String | ips规则名称 |

| 参数 | 参数类型 | 描述 |
|----------------|--------|---|
| ips_rules_type | String | ips规则类型，包括漏洞扫描、黑客工具、特洛伊木马等 |
| ips_status | String | ips规则状态，包含观察：OBSERVE、拦截：ENABLE、禁用：CLOSE、恢复默认：DEFAULT、全局恢复默认：ALL_DEFAULT |

请求示例

获取项目id为408972e72dcd4c1a9b033e955802a36b的IPS规则列表，防火墙id为e743cfaf-8164-4807-aa13-d893d83313cf，企业项目id为fb55459c-41b3-47fc-885d-540946fddda4，目标对象id为1b90f031-0c7b-4f25-95e2-b6d9940d269e。查询结果限制为1000条，偏移量为0。

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/ips-rule?fw_instance_id=e743cfaf-8164-4807-aa13-d893d83313cf&enterprise_project_id=fb55459c-41b3-47fc-885d-540946fddda4&project_id=408972e72dcd4c1a9b033e955802a36b&object_id=1b90f031-0c7b-4f25-95e2-b6d9940d269e&limit=1000&offset=0
```

响应示例

状态码：200

查询规则列表返回值

```
{
  "data": {
    "fw_instance_id": "e743cfaf-8164-4807-aa13-d893d83313cf",
    "limit": 1000,
    "offset": 1,
    "records": [ {
      "affected_application": "Others",
      "create_time": "2015",
      "default_status": "CLOSE",
      "ips_group": "STRICTLY",
      "ips_id": "340710",
      "ips_level": "MEDIUM",
      "ips_name": "WEBC2-QBP登录响应1 - 嵌入式CnC APT1相关",
      "ips_rules_type": "特洛伊木马",
      "ips_status": "CLOSE"
    }, {
      "affected_application": "Others",
      "create_time": "2015",
      "default_status": "CLOSE",
      "ips_group": "STRICTLY",
      "ips_id": "340922",
      "ips_level": "MEDIUM",
      "ips_name": "Win32/Fujacks活动",
      "ips_rules_type": "特洛伊木马",
      "ips_status": "CLOSE"
    }
  ],
  "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 ListIpsRulesSolution {

    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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListIpsRulesRequest request = new ListIpsRulesRequest();
        try {
            ListIpsRulesResponse response = client.listIpsRules(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

import os
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.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

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

try:
    request = ListIpsRulesRequest()
    response = client.list_ips_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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListIpsRulesRequest{}
    response, err := client.ListIpsRules(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 | 查询规则列表返回值 |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |

错误码

请参见[错误码](#)。

4.8.9 获取 ips 规则细节

功能介绍

获取ips规则细节

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/ips-rule/detail

表 4-498 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID，可以从调API处获取，也可以从控制台获取。可通过 项目ID获取方式 获取 |

表 4-499 Query 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|-----------------------|------|--------|---|
| fw_instance_id | 是 | String | 防火墙ID，可通过 防火墙ID获取方式 获取 |
| enterprise_project_id | 否 | String | 企业项目ID，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

请求参数

表 4-500 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，可通过 如何获取用户Token 获取 |

响应参数

状态码：200

表 4-501 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--|---------------|
| data | Array of IpsRuleUpdateTimeVO objects | 查询ips规则细节时间数据 |

表 4-502 IpsRuleUpdateTimeVO

| 参数 | 参数类型 | 描述 |
|-------------|---------|-----------------------|
| ips_type | Integer | ips类型，0表示基础防御，1表示虚拟补丁 |
| ips_version | String | ips规则版本 |
| update_time | Long | ips更新时间戳 |

请求示例

获取项目id为408972e72dcd4c1a9b033e955802a36b的IPS规则细节，防火墙id为e743cfaf-8164-4807-aa13-d893d83313cf，企业项目id为fb55459c-41b3-47fc-885d-540946fddda4

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/ips-rule/detail?fw_instance_id=e743cfaf-8164-4807-aa13-d893d83313cf&enterprise_project_id=fb55459c-41b3-47fc-885d-540946fddda4
```

响应示例

状态码：200

查询ips规则细节返回值

```
{  
  "data": [{  
    "ips_type": 0,
```

```
"ips_version" : "3.0.235",
"update_time" : 1733816977000
}, {
"ips_type" : 1,
"ips_version" : "3.0.220",
"update_time" : 1733816960000
}]
}
```

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 ShowIpsUpdateTimeSolution {

    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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowIpsUpdateTimeRequest request = new ShowIpsUpdateTimeRequest();
        try {
            ShowIpsUpdateTimeResponse response = client.showIpsUpdateTime(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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowIpsUpdateTimeRequest()
        response = client.show_ips_update_time(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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```
Build()  
  
request := &model.ShowIpsUpdateTimeRequest{}  
response, err := client.ShowIpsUpdateTime(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ips规则细节返回值 |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |

错误码

请参见[错误码](#)。

4.8.10 获取自定义 ips 规则

功能介绍

获取自定义ips规则

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/ips/custom-rule

表 4-503 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID，可以从调API处获取，也可以从控制台获取。可通过 项目ID获取方式 获取 |

表 4-504 Query 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------------|------|---------|--|
| action_type | 否 | Integer | 动作（0：只记录日志，1：重置/拦截） |
| affected_os | 否 | Integer | 影响操作系统,包括：0 any、1 Windows、2 Linux、3 FreeBSD、4 Solaris、5 other Unix、6 网络设备、7 Mac OS、8 ios、9 android、10 others |
| attack_type | 否 | Integer | 攻击类型、包括1：访问控制、2：漏洞扫描、3：邮件攻击、4：漏洞攻击、5：Web攻击、6：密码攻击、7：劫持攻击、8：协议异常、9：特洛伊木马、10：蠕虫、11：缓冲区溢出、12：黑客工具、13：间谍软件、14：DDos泛洪、15：应用层DDos攻击、16：其他可疑行为、17：可疑DNS活动、18：网络钓鱼、19：垃圾邮件、20：其他攻击 |
| fw_instance_id | 是 | String | 防火墙ID，可通过 防火墙ID获取方式 获取 |
| ips_name | 否 | String | ips规则名称 |
| limit | 是 | Integer | 每页显示个数，范围为1-1024 |
| 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，通过返回值中的 data.records.protect_objects.object_id（.表示各对象之间层级的区分）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。此处仅取type为1的防护对象id，可通过 data.records.protect_objects.type（.表示各对象之间层级的区分）获得。 |
| 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 |
|-----------------------|------|---------|--|
| protocol | 否 | Integer | 协议类型，包括：1 FTP、2 TELNET、3 SMTP、4 DNS_TCP、5 DNS_UDP、6 DHCP、7 TFTP、8 FINGER、9 HTTP、10 POP3、11 SUNRPC_TCP、12 SUNRPC_UDP、13 NNTP、14 MSRPC_TCP、15 MSRPC_UDP、16 NETBIOS_NAME_TCP、17 NETBIOS_NAME_UDP、18 NETBIOS_SMB、19 NETBIOS_DATAGRAM、20 IMAP4、21 SNMP、22 LDAP、23 MSSQL、24 ORACAL |
| severity | 否 | Integer | 严重程度（critical: 致命，high: 高危，medium:中危，low:低危） |
| software | 否 | Integer | 影响软件，包括：0 ANY、1 ADOBE、2 APACHE、3 APPLE、4 CA、5 CISCO、6 GOOGLE_CHROME、7 HP、8 IBM、9 IE、10 IIS、11 MC_AFEE、12 MEDIA_PLAYER、13 MICROSOFT_NET、14 MICROSOFT_EDGE、15 MICROSOFT_EXCHANGE、16 MICROSOFT_OFFICE、17 MICROSOFT_OUTLOOK、18 MICROSOFT_SHARE_POINT、19 MICROSOFT_WINDOWS、20 MOZILLA、21 MSSQL、22 MYSQL、23 NOVELL、24 ORACLE、25 SAMBA、26 SAMSUNG、27 SAP、28 SCADA、29 SQUID、30 SUN、31 SYMANTEC、32 TRREND_MICRO、33 VMWARE、34 WORD_PRESS、35 Others |
| enterprise_project_id | 否 | String | 企业项目ID，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

请求参数

表 4-505 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，可通过 如何获取用户Token 获取 |

响应参数

状态码：200

表 4-506 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--|----|
| data | HttpListCustomerIpsResponseData object | |

表 4-507 HttpListCustomerIpsResponseData

| 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 | 查询获得自定义ips规则列表总数 |
| records | Array of CustomerIpsListVO objects | 自定义ips规则记录 |

表 4-508 CustomerIpsListVO

| 参数 | 参数类型 | 描述 |
|---------------|---------|---------------------------------|
| action | Integer | 动作（0：只记录日志，1：重置/拦截） |
| affected_os | Integer | 操作系统 |
| attack_type | Integer | 攻击类型 |
| config_status | Integer | 规则状态（0：初始化，1：配置中，2：配置成功，3：配置失败） |

| 参数 | 参数类型 | 描述 |
|---------------|---------|--|
| content | String | 内容json存储 |
| dst_port_type | Integer | 目的端口类型 |
| dst_ports | String | 目的端口 |
| group_id | String | 防火墙集群id |
| ips_cfw_id | String | cfw侧自定义ips规则id |
| ips_id | String | ips规则id |
| ips_name | String | ips规则名称 |
| protocol | Integer | 协议 |
| severity | Integer | 严重程度 (critical: 致命, high: 高危, medium:中危, low:低危) |
| software | Integer | 影响软件 |
| src_port_type | Integer | 源端口类型 |
| src_ports | String | 源端口 |

请求示例

获取项目id为408972e72dcd4c1a9b033e955802a36b的自定义IPS规则列表，防火墙id为e743cfaf-8164-4807-aa13-d893d83313cf，企业项目id为fb55459c-41b3-47fc-885d-540946fddda4，目标对象id为1b90f031-0c7b-4f25-95e2-b6d9940d269e。查询结果限制为1000条，偏移量为0。

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/ips/custom-rule?fw_instance_id=e743cfaf-8164-4807-aa13-d893d83313cf&enterprise_project_id=fb55459c-41b3-47fc-885d-540946fddda4&object_id=1b90f031-0c7b-4f25-95e2-b6d9940d269e&limit=1000&offset=0
```

响应示例

状态码：200

OK

```
{
  "data": {
    "limit": 1000,
    "offset": 0,
    "records": [],
    "total": 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 ListCustomerIpsSolution {

    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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListCustomerIpsRequest request = new ListCustomerIpsRequest();
        try {
            ListCustomerIpsResponse response = client.listCustomerIps(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

import os
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.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

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

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListCustomerIpsRequest{}
    response, err := client.ListCustomerIps(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 |

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 4-509 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-510 Query 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------------|------|--------|---|
| fw_instance_id | 是 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|-----------------------|------|--------|---|
| enterprise_project_id | 否 | String | 企业项目ID，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

请求参数

表 4-511 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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-512 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|-------------------------------------|--------|
| data | LogConfigDto object | 日志配置数据 |

表 4-513 LogConfigDto

| 参数 | 参数类型 | 描述 |
|--------------------------|---------|--|
| fw_instance_id | String | 防火墙id，可通过 防火墙ID获取方式 获取 |
| lts_enable | Integer | 是否开启LTS，1表示是，0表示不是 |
| lts_log_group_id | String | LTS日志分组id,可通过查询LTS（云日志服务）下查询账号下所有日志组接口获得，通过返回值中的log_groups.log_group_id（.表示各对象之间层级的区分）获得 |
| lts_attack_log_stream_id | String | 攻击日志流id,可通过查询LTS（云日志服务）下查询指定日志组下的所有日志流接口获得，通过返回值中的log_streams.log_stream_id（.表示各对象之间层级的区分）获得 |

| 参数 | 参数类型 | 描述 |
|------------------------------|---------|--|
| lts_attack_log_stream_enable | Integer | 是否开启攻击日志流, 1表示是, 0表示不是 |
| lts_access_log_stream_id | String | 访问控制日志流id,可通过查询LTS(云日志服务)下查询指定日志组下的所有日志流接口获得,通过返回值中的log_streams.log_stream_id(.表示各对象之间层级的区分)获得 |
| lts_access_log_stream_enable | Integer | 是否开启访问控制流, 1表示是, 0表示不是 |
| lts_flow_log_stream_id | String | 流量日志id,可通过查询LTS(云日志服务)下查询指定日志组下的所有日志流接口获得,通过返回值中的log_streams.log_stream_id(.表示各对象之间层级的区分)获得 |
| lts_flow_log_stream_enable | Integer | 是否开启流量日志, 1表示是, 0表示不是 |

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

查询日志配置返回值

```
{
  "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"
  }
}
```

状态码

| 状态码 | 描述 |
|-----|-----------|
| 200 | 查询日志配置返回值 |

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 4-514 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-515 Query 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|-----------------------|------|--------|---|
| fw_instance_id | 是 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |
| enterprise_project_id | 否 | String | 企业项目ID, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |

请求参数

表 4-516 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。可通过 如何获取用户Token 获取。 |

表 4-517 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, 可通过 防火墙ID获取方式 获取 |
| lts_enable | 是 | Integer | 是否开启LTS, 1表示是, 0表示不是 |
| lts_log_group_id | 是 | String | LTS日志分组id,可通过查询LTS (云日志服务)下查询账号下所有日志组接口获得, 通过返回值中的log_groups.log_group_id (.表示各对象之间层级的区分)获得 |
| lts_attack_log_stream_id | 否 | String | 攻击日志流id,可通过查询LTS (云日志服务)下查询指定日志组下的所有日志流接口获得, 通过返回值中的log_streams.log_stream_id (.表示各对象之间层级的区分)获得 |
| lts_attack_log_stream_enable | 否 | Integer | 是否开启攻击日志流, 1表示是, 0表示不是 |
| lts_access_log_stream_id | 否 | String | 访问控制日志流id,可通过查询LTS (云日志服务)下查询指定日志组下的所有日志流接口获得, 通过返回值中的log_streams.log_stream_id (.表示各对象之间层级的区分)获得 |
| lts_access_log_stream_enable | 否 | Integer | 是否开启访问控制流, 1表示是, 0表示不是 |
| lts_flow_log_stream_id | 否 | String | 流量日志id,可通过查询LTS (云日志服务)下查询指定日志组下的所有日志流接口获得, 通过返回值中的log_streams.log_stream_id (.表示各对象之间层级的区分)获得 |
| lts_flow_log_stream_enable | 否 | Integer | 是否开启流量日志, 1表示是, 0表示不是 |

响应参数

状态码: 200

表 4-518 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--------|------------------|
| data | String | 添加日志配置返回值，为防火墙id |

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");
    String projectId = "{project_id}";

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

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

import os
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")
```

```
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

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

try:
    request = AddLogConfigRequest()
    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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddLogConfigRequest{}
    ltsFlowLogStreamEnableLogConfigDto := int32(0)
```



```
ltsAccessLogStreamEnableLogConfigDto:= int32(0)
ltsAttackLogStreamEnableLogConfigDto:= int32(0)
request.Body = &model.LogConfigDto{
    LtsFlowLogStreamEnable: &ltsFlowLogStreamEnableLogConfigDto,
    LtsAccessLogStreamEnable: &ltsAccessLogStreamEnableLogConfigDto,
    LtsAttackLogStreamEnable: &ltsAttackLogStreamEnableLogConfigDto,
    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 4-519 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|--|
| project_id | 是 | String | 项目ID, 可以从调API处获取，也可以从控制台获取。 项目ID获取方式 |

表 4-520 Query 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|-----------------------|------|--------|---|
| fw_instance_id | 是 | String | 防火墙id, 可通过 防火墙ID获取方式 获取。 |
| enterprise_project_id | 否 | String | 企业项目ID, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |

请求参数

表 4-521 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。可通过 如何获取用户Token 获取。 |

表 4-522 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, 可通过 防火墙ID获取方式 获取 |
| lts_enable | 是 | Integer | 是否开启LTS, 1表示是, 0表示不是 |
| lts_log_group_id | 是 | String | LTS日志分组id, 可通过查询LTS (云日志服务) 下查询账号下所有日志组接口获得, 通过返回值中的log_groups.log_group_id (.表示各对象之间层级的区分) 获得 |
| lts_attack_log_stream_id | 否 | String | 攻击日志流id, 可通过查询LTS (云日志服务) 下查询指定日志组下的所有日志流接口获得, 通过返回值中的log_streams.log_stream_id (.表示各对象之间层级的区分) 获得 |
| lts_attack_log_stream_enable | 否 | Integer | 是否开启攻击日志流, 1表示是, 0表示不是 |

| 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,可通过查询LTS（云日志服务）下查询指定日志组下的所有日志流接口获得，通过返回值中的log_streams.log_stream_id（.表示各对象之间层级的区分）获得 |
| lts_access_log_stream_enable | 否 | Integer | 是否开启访问控制流，1表示是，0表示不是 |
| lts_flow_log_stream_id | 否 | String | 流量日志id,可通过查询LTS（云日志服务）下查询指定日志组下的所有日志流接口获得，通过返回值中的log_streams.log_stream_id（.表示各对象之间层级的区分）获得 |
| lts_flow_log_stream_enable | 否 | Integer | 是否开启流量日志，1表示是，0表示不是 |

响应参数

状态码：200

表 4-523 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--------|------------------|
| data | String | 更新日志配置返回值，为防火墙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");
        String projectId = "{project_id}";

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

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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateLogConfigRequest()
        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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateLogConfigRequest{
        ltsFlowLogStreamEnableLogConfigDto:= int32(0)
        ltsAccessLogStreamEnableLogConfigDto:= int32(0)
        ltsAttackLogStreamEnableLogConfigDto:= int32(0)
        request.Body = &model.LogConfigDto{
            LtsFlowLogStreamEnable: &ltsFlowLogStreamEnableLogConfigDto,
            LtsAccessLogStreamEnable: &ltsAccessLogStreamEnableLogConfigDto,
            LtsAttackLogStreamEnable: &ltsAttackLogStreamEnableLogConfigDto,
            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 4-524 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-525 Query 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------------|------|---------|---|
| fw_instance_id | 是 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |
| direction | 否 | String | 方向, 包含in2out, out2in |
| log_type | 否 | String | 日志类型包括: internet, vpc, nat |
| start_time | 是 | Long | 开始时间, 以毫秒为单位的时间戳, 如1718936272648 |
| end_time | 是 | Long | 结束时间, 以毫秒为单位的时间戳, 如1718936272648 |
| 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 | 协议类型, 包含TCP, UDP, ICMP, ICMPV6等。 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|-----------------------|------|---------|--|
| app | 否 | String | 规则应用类型包括： “HTTP”，“HTTPS”， “TLS1”，“DNS”，“SSH”， “MYSQL”，“SMTP”， “RDP”，“RDPS”， “VNC”，“POP3”， “IMAP4”，“SMTPS”， “POP3S”，“FTPS”， “ANY”，“BGP”等。 |
| log_id | 否 | String | 文档ID,第一页为空，其他页不为空，其他页可取上一次查询最后一条数据的log_id |
| next_date | 否 | Long | 下个日期，当是第一页时空，不是第一页时不为空，其他页可取上一次查询最后一条数据的start_time |
| 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0，首页时空，非首页时不为空 |
| limit | 是 | Integer | 每页显示个数，范围为1-1024 |
| enterprise_project_id | 否 | String | 企业项目ID，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |
| 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 |
| src_province_name | 否 | String | 源省份名称 |
| dst_province_name | 否 | String | 目的省份名称 |
| src_city_name | 否 | String | 源城市名称 |
| dst_city_name | 否 | String | 目的城市名称 |

请求参数

表 4-526 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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-527 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|-----------------------------|------------|
| data | data object | 查询流日志返回值数据 |

表 4-528 data

| 参数 | 参数类型 | 描述 |
|---------|--|------------------|
| total | Integer | 查询流日志返回值记录总数 |
| limit | Integer | 每页显示个数，范围为1-1024 |
| records | Array of records objects | 记录 |

表 4-529 records

| 参数 | 参数类型 | 描述 |
|------------|---------|--------------------------------|
| bytes | Double | 字节 |
| direction | String | 方向，有内到外（in2out）和外到内（out2in）两种 |
| packets | Integer | 字节包数 |
| start_time | Long | 开始时间，以毫秒为单位的时间戳，如1718936272648 |
| end_time | Long | 结束时间，以毫秒为单位的时间戳，如1718936272648 |
| log_id | String | 文档ID |
| src_ip | String | 源IP |

| 参数 | 参数类型 | 描述 |
|-------------------|---------|--|
| src_port | Integer | 源端口 |
| dst_ip | String | 目的IP |
| app | String | 规则应用类型包括：“HTTP”，“HTTPS”，“TLS1”，“DNS”，“SSH”，“MYSQL”，“SMTP”，“RDP”，“RDPS”，“VNC”，“POP3”，“IMAP4”，“SMTPS”，“POP3S”，“FTPS”，“ANY”，“BGP”等。 |
| 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 |
| 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 |
| dst_province_id | String | 目的省份id |
| dst_province_name | String | 目的省份名称 |
| dst_city_id | String | 目的城市id |
| dst_city_name | String | 目的城市名称 |
| src_province_id | String | 源省份id |
| src_province_name | String | 源省份名称 |
| src_city_id | String | 源城市id |
| src_city_name | String | 源城市名称 |

状态码：400

表 4-530 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|-----|
| error_code | String | 错误码 |

| 参数 | 参数类型 | 描述 |
|-----------|--------|------|
| error_msg | String | 错误描述 |

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

查询流日志返回值

```
{
  "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,
      "src_province_id": "source province id",
      "src_province_name": "source province name",
      "src_city_id": "source city id",
      "src_city_name": "source city name",
      "dst_province_id": "dst province id",
      "dst_province_name": "dst province name",
      "dst_city_id": "dst city id",
      "dst_city_name": "dst city name"
    } ],
    "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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListFlowLogsRequest request = new ListFlowLogsRequest();
        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.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
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.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

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

try:
    request = ListFlowLogsRequest()
    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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListFlowLogsRequest{}
    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 | 查询流日志返回值 |
| 400 | Bad Request |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |
| 500 | Internal Server Error |

错误码

请参见[错误码](#)。

4.9.5 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 4-531 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-532 Query 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------------|------|--------|---|
| fw_instance_id | 是 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|---------|--|
| rule_id | 否 | String | 规则id, 可通过 查询防护规则接口 查询获得, 通过返回值中的 data.records.rule_id (表示各对象之间层级的区分) 获得。 |
| start_time | 是 | Long | 开始时间, 以毫秒为单位的时间戳, 如1718936272648 |
| end_time | 是 | Long | 结束时间, 以毫秒为单位的时间戳, 如1718936272648 |
| 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 | 协议类型, 包含TCP, UDP, ICMP, ICMPV6等。 |
| app | 否 | String | 规则应用类型包括: “HTTP”, “HTTPS”, “TLS1”, “DNS”, “SSH”, “MYSQL”, “SMTP”, “RDP”, “RDPS”, “VNC”, “POP3”, “IMAP4”, “SMTPS”, “POP3S”, “FTPS”, “ANY”, “BGP” 等。 |
| log_id | 否 | String | 文档ID, 第一页为空, 其他页不为空, 其他页可取上一次查询最后一条数据的log_id |
| next_date | 否 | Integer | 下个日期, 当是第一页时空, 不是第一页时不为空, 其他页可取上一次查询最后一条数据的start_time |
| 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0, 首页时空, 非首页时不为空 |
| limit | 是 | Integer | 每页显示个数, 范围为1-1024 |
| log_type | 否 | String | 日志类型包括: internet, vpc, nat |

| 参数 | 是否必选 | 参数类型 | 描述 |
|-----------------------|------|--------|---|
| enterprise_project_id | 否 | String | 企业项目ID，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |
| dst_host | 否 | String | 目标主机 |
| rule_name | 否 | String | 规则名称 |
| action | 否 | String | 动作包含permit, deny |
| src_region_name | 否 | String | 源region名称 |
| dst_region_name | 否 | String | 目的region名称 |
| src_province_name | 否 | String | 源省份名称 |
| dst_province_name | 否 | String | 目的省份名称 |
| src_city_name | 否 | String | 源城市名称 |
| dst_city_name | 否 | String | 目的城市名称 |

请求参数

表 4-533 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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-534 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 4-535 data

| 参数 | 参数类型 | 描述 |
|---------|---------------------------------|------------------|
| total | Integer | 查询访问控制日志记录总数 |
| limit | Integer | 每页显示个数，范围为1-1024 |
| records | Array of records objects | 查询访问控制日志记录 |

表 4-536 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 |
| hit_time | Long | 命中时间，以毫秒为单位的时间戳，如1718936272648 |
| 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 | 规则应用类型包括：“HTTP”，“HTTPS”，“TLS1”，“DNS”，“SSH”，“MYSQL”，“SMTP”，“RDP”，“RDPS”，“VNC”，“POP3”，“IMAP4”，“SMTPS”，“POP3S”，“FTPS”，“ANY”，“BGP”等。 |
| dst_host | String | 目标主机 |
| src_province_id | String | 源省份id |

| 参数 | 参数类型 | 描述 |
|-------------------|--------|--------|
| src_province_name | String | 源省份名称 |
| src_city_id | String | 源城市id |
| src_city_name | String | 源城市名称 |
| dst_province_id | String | 目的省份id |
| dst_province_name | String | 目的省份名称 |
| dst_city_id | String | 目的城市id |
| dst_city_name | String | 目的城市名称 |

状态码：400

表 4-537 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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

查询访问控制日志返回值

```
{
  "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,
"src_province_id" : "source province id",
"src_province_name" : "source province name",
"src_city_id" : "source city id",
"src_city_name" : "source city name",
"dst_province_id" : "dst province id",
"dst_province_name" : "dst province name",
"dst_city_id" : "dst city id",
"dst_city_name" : "dst city name"
}],
"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");
        String projectId = "{project_id}";

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

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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListAccessControlLogsRequest()
        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")
projectId := "{project_id}"

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

client := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListAccessControlLogsRequest{}
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 | 查询访问控制日志返回值 |
| 400 | Bad Request |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |
| 500 | Internal Server Error |

错误码

请参见[错误码](#)。

4.9.6 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 4-538 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-539 Query 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|---------|--|
| start_time | 是 | Long | 开始时间, 以毫秒为单位的时间戳, 如1718936272648 |
| end_time | 是 | Long | 结束时间, 以毫秒为单位的时间戳, 如1718936272648 |
| src_ip | 否 | String | 源IP |
| src_port | 否 | Integer | 源端口号 |
| dst_ip | 否 | String | 目的IP |
| dst_port | 否 | Integer | 目的端口号 |
| protocol | 否 | String | 协议类型, 包含TCP, UDP, ICMP, ICMPV6等。 |
| app | 否 | String | 规则应用类型包括: “HTTP”, “HTTPS”, “TLS1”, “DNS”, “SSH”, “MYSQL”, “SMTP”, “RDP”, “RDPS”, “VNC”, “POP3”, “IMAP4”, “SMTPS”, “POP3S”, “FTPS”, “ANY”, “BGP” 等。 |
| log_id | 否 | String | 文档ID, 第一页为空, 其他页不为空, 其他页可取上一次查询最后一条数据的log_id |
| 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, 其他页可取上一次查询最后一条数据的event_time |

| 参数 | 是否必选 | 参数类型 | 描述 |
|-----------------------|------|---------|---|
| 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0，首页时为空，非首页时不为空 |
| limit | 是 | Integer | 每页显示个数，范围为1-1024 |
| fw_instance_id | 是 | String | 防火墙id，可通过 防火墙ID获取方式 获取 |
| action | 否 | String | 动作包含permit, deny |
| direction | 否 | String | 方向，包含in2out, out2in |
| 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，包括CRITICAL、HIGH、MEDIUM、LOW |
| enterprise_project_id | 否 | String | 企业项目ID，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |
| dst_host | 否 | String | 目标主机 |
| log_type | 否 | String | 日志类型包括：internet, vpc, nat |
| 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 |
| src_province_name | 否 | String | 源省份名称 |
| dst_province_name | 否 | String | 目的省份名称 |
| src_city_name | 否 | String | 源城市名称 |
| dst_city_name | 否 | String | 目的城市名称 |

请求参数

表 4-540 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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-541 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数据 |

表 4-542 data

| 参数 | 参数类型 | 描述 |
|---------|---------------------------------|------------------|
| total | Integer | 返回攻击数据总数 |
| limit | Integer | 每页显示个数，范围为1-1024 |
| records | Array of records objects | 攻击日志记录列表 |

表 4-543 records

| 参数 | 参数类型 | 描述 |
|-------------|--------|---------------------------------|
| direction | String | 方向，包含in2out, out2in |
| action | String | 动作包含permit, deny |
| event_time | Long | 事件时间，以毫秒为单位的时间戳，如1718936272648 |
| attack_type | String | 攻击类型 |
| attack_rule | String | 攻击规则 |
| level | String | 威胁等级，包括CRITICAL、HIGH、MEDIUM、LOW |
| source | String | 来源 |

| 参数 | 参数类型 | 描述 |
|-------------------|--|--|
| packet_length | Long | 报文长度 |
| attack_rule_id | String | 攻击规则id |
| hit_time | Long | 命中时间，以毫秒为单位的时间戳，如1718936272648 |
| log_id | String | 日志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，包含TCP, UDP,ICMP,ICMPV6等。 |
| packet | String | 攻击日志报文 |
| app | String | 规则应用类型包括：“HTTP”，“HTTPS”，“TLS1”，“DNS”，“SSH”，“MYSQL”，“SMTP”，“RDP”，“RDPS”，“VNC”，“POP3”，“IMAP4”，“SMTPS”，“POP3S”，“FTPS”，“ANY”，“BGP”等。 |
| packetMessages | Array of PacketMessage objects | 攻击报文信息 |
| 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 |
| src_province_id | String | 源省份id |
| src_province_name | String | 源省份名称 |
| src_city_id | String | 源城市id |
| src_city_name | String | 源城市名称 |
| dst_province_id | String | 目的省份id |
| dst_province_name | String | 目的省份名称 |
| dst_city_id | String | 目的城市id |

| 参数 | 参数类型 | 描述 |
|---------------|--------|--------|
| dst_city_name | String | 目的城市名称 |

表 4-544 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 4-545 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------------|--------|------|
| error_code | String | 错误码 |
| error_msg | String | 错误描述 |

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

查询攻击日志返回值

```
{
  "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" : "+hZUJZMhV+hY/AaHMCABFKABpXPNAADAGof1kVe6QZF
+UMcTQH5B0wdaz888+uoAYA0VyNQAAAQEICjrmikVb9JLCR0VUIC9uaWNUJlwcG9ydHMIMkMvVHJpJTZFaX
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,
"src_province_id" : "source province id",
"src_province_name" : "source province name",
"src_city_id" : "source city id",
"src_city_name" : "source city name",
"dst_province_id" : "dst province id",
"dst_province_name" : "dst province name",
"dst_city_id" : "dst city id",
"dst_city_name" : "dst city name"
} ],
"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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListAttackLogsRequest request = new ListAttackLogsRequest();
        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.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
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.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

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

try:
    request = ListAttackLogsRequest()
    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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAttackLogsRequest{}
    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 | 查询攻击日志返回值 |
| 400 | Bad Request |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |
| 500 | Internal Server Error |

错误码

请参见[错误码](#)。

4.10 抓包管理

4.10.1 查询抓包任务

功能介绍

查询抓包任务

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/capture-task

表 4-546 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-547 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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |
| fw_instance_id | 是 | String | 防火墙id，可通过 防火墙ID获取方式 获取 |
| 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 |

请求参数

表 4-548 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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-549 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|---|-------------|
| data | HttpQueryCaptureTaskResponseData object | 查询抓包任务返回值数据 |

表 4-550 HttpQueryCaptureTaskResponseData

| 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 |

| 参数 | 参数类型 | 描述 |
|---------|--|--------|
| total | Long | 抓包任务总数 |
| records | Array of CaptureTaskVO objects | 抓包任务列表 |

表 4-551 CaptureTaskVO

| 参数 | 参数类型 | 描述 |
|---------------------|---------|--|
| capture_size | String | 抓包大小,如500kb,500mb |
| created_date | String | 抓包创建时间,如2024/08/31 10:17:30 |
| dest_address | String | 目的地址 |
| dest_address_type | Integer | 目的地址类型0 ipv4, 1 ipv6 |
| dest_port | String | 目的端口 |
| duration | Integer | 抓包时长,以分钟为单位 |
| is_deleted | Integer | 是否被删除, 0否 1是 |
| max_packets | Integer | 最大抓包数,以个为单位 |
| modified_date | String | 修改日期,如2024/08/31 10:17:30 |
| name | String | 抓包任务名称 |
| 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 |
| remaining_days | Integer | 剩余保留天数 |
| source_address | String | 源地址 |
| source_address_type | Integer | 源地址类型0 ipv4, 1 ipv6 |
| source_port | String | 源端口 |
| status | Integer | 抓包任务状态, 如成功(1), 运行中(2), 已截止(4), 截止中(5) |
| task_id | String | 抓包任务id |

请求示例

向项目id为09bb24e6fe80d23d2fa2c010b53b418c, 防火墙id为ebf891cd-2163-48a0-9963-6309f99dd3c4查询50条抓包任务。


```
https://{Endpoint}/v1/09bb24e6fe80d23d2fa2c010b53b418c/capture-task?  
fw_instance_id=ebf891cd-2163-48a0-9963-6309f99dd3c4&enterprise_project_id=default&limit=50&offset=0
```

响应示例

状态码：200

查询抓包任务列表返回值

```
{  
  "data": {  
    "limit": 50,  
    "offset": 0,  
    "records": [ {  
      "capture_size": "--",  
      "created_date": "2024/06/04 11:23:50",  
      "dest_address": "2.2.2.2",  
      "dest_address_type": 0,  
      "dest_port": "",  
      "duration": 3,  
      "is_deleted": 0,  
      "max_packets": 100000,  
      "modified_date": "2024/06/04 11:23:50",  
      "name": "zhuabaotest",  
      "protocol": -1,  
      "remaining_days": 7,  
      "source_address": "1.1.1.1",  
      "source_address_type": 0,  
      "source_port": "",  
      "status": 2,  
      "task_id": "24e6bb6d-d335-48fd-b9c7-bbaaa851d209"  
    } ],  
    "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 ListCaptureTaskSolution {  
    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");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()
```

```
.withProjectId(projectId)
.withAk(ak)
.withSk(sk);

CfwClient client = CfwClient.newBuilder()
    .withCredential(auth)
    .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
    .build();
ListCaptureTaskRequest request = new ListCaptureTaskRequest();
try {
    ListCaptureTaskResponse response = client.listCaptureTask(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

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListCaptureTaskRequest()
        response = client.list_capture_task(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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListCaptureTaskRequest{}
    response, err := client.ListCaptureTask(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 | 查询抓包任务列表返回值 |

错误码

请参见[错误码](#)。

4.10.2 创建抓包任务

功能介绍

创建抓包任务，每个任务只能执行一次。

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/capture-task

表 4-552 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-553 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, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |
| fw_instance_id | 是 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

请求参数

表 4-554 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。可通过 如何获取用户Token 获取。 |

表 4-555 请求 Body 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|-------------|------|--|--------------|
| destination | 是 | CaptureRuleAddressDto object | 抓包规则目的地址请求体 |
| duration | 是 | Integer | 抓包时长, 以分钟为单位 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|-------------|------|--|-------------|
| max_packets | 是 | Integer | 最大抓包数，以个为单位 |
| name | 是 | String | 抓包任务名称 |
| service | 是 | CaptureServiceDto object | 抓包任务服务请求体 |
| source | 是 | CaptureRuleAddressDto object | 抓包规则源地址请求体 |

表 4-556 CaptureServiceDto

| 参数 | 是否必选 | 参数类型 | 描述 |
|-------------|------|---------|--|
| dest_port | 否 | String | 目的端口 |
| 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 | 否 | String | 源端口 |

表 4-557 CaptureRuleAddressDto

| 参数 | 是否必选 | 参数类型 | 描述 |
|--------------|------|---------|------------------------|
| address | 是 | String | 地址 |
| address_type | 是 | Integer | 目的地址类型0 ipv4, 1 ipv6 |
| type | 是 | Integer | 输入地址类型, 目前只支持0, 手工输入类型 |

响应参数

状态码：200

表 4-558 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--------------------------------------|------------|
| data | CaptureTaskId object | 创建抓包任务id信息 |

表 4-559 CaptureTaskId

| 参数 | 参数类型 | 描述 |
|------|--------|-------|
| id | String | 防火墙id |
| name | String | 防火墙名称 |

请求示例

向项目id为09bb24e6fe80d23d2fa2c010b53b418c，防火墙id为ebf891cd-2163-48a0-9963-6309f99dd3c4添加抓包任务，任务名称为zhuabaotest，最大抓包数为100000，抓包时长为3分钟，剩余保留天数为7天，源地址协议为tcp协议，地址类型为ipv4，地址为1.1.1.1，端口号为1-65535，目的地址协议为tcp协议，地址类型为ipv4，地址为2.2.2.2，端口号为1-65535。

```
https://{Endpoint}/v1/09bb24e6fe80d23d2fa2c010b53b418c/capture-task?  
fw_instance_id=ebf891cd-2163-48a0-9963-6309f99dd3c4&enterprise_project_id=default
```

```
{  
  "name": "zhuabaotest",  
  "max_packets": 100000,  
  "duration": 3,  
  "source": {  
    "type": 0,  
    "address_type": 0,  
    "address": "1.1.1.1"  
  },  
  "destination": {  
    "type": 0,  
    "address_type": 0,  
    "address": "2.2.2.2"  
  },  
  "service": {  
    "protocol": -1,  
    "source_port": "",  
    "dest_port": ""  
  }  
}
```

响应示例

状态码：200

创建抓包任务返回值

```
{  
  "data": {  
    "id": "ebf891cd-2163-48a0-9963-6309f99dd3c4",  
    "name": "test"  
  }  
}
```

SDK 代码示例

SDK代码示例如下。

Java

向项目id为09bb24e6fe80d23d2fa2c010b53b418c，防火墙id为ebf891cd-2163-48a0-9963-6309f99dd3c4添加抓包任务，任务名称为zhuabaotest，

最大抓包数为100000，抓包时长为3分钟，剩余保留天数为7天，源地址协议为tcp协议，地址类型为ipv4，地址为1.1.1.1，端口号为1-65535，目的地址协议为tcp协议，地址类型为ipv4，地址为2.2.2.2，端口号为1-65535。

```
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 CreateCaptureTaskSolution {

    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");
        String projectId = "{project_id}";

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

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

        CreateCaptureTaskRequest request = new CreateCaptureTaskRequest();
        CaptureTaskDto body = new CaptureTaskDto();
        CaptureRuleAddressDto sourcebody = new CaptureRuleAddressDto();
        sourcebody.withAddress("1.1.1.1")
            .withAddressType(0)
            .withType(0);
        CaptureServiceDto servicebody = new CaptureServiceDto();
        servicebody.withDestPort("")
            .withProtocol(-1)
            .withSourcePort("");
        CaptureRuleAddressDto destinationbody = new CaptureRuleAddressDto();
        destinationbody.withAddress("2.2.2.2")
            .withAddressType(0)
            .withType(0);
        body.withSource(sourcebody);
        body.withService(servicebody);
        body.withName("zhuabaotest");
        body.withMaxPackets(100000);
        body.withDuration(3);
        body.withDestination(destinationbody);
        request.withBody(body);
        try {
            CreateCaptureTaskResponse response = client.createCaptureTask(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09bb24e6fe80d23d2fa2c010b53b418c，防火墙id为ebf891cd-2163-48a0-9963-6309f99dd3c4添加抓包任务，任务名称为zhuabaotest，最大抓包数为100000，抓包时长为3分钟，剩余保留天数为7天，源地址协议为tcp协议，地址类型为ipv4，地址为1.1.1.1，端口号为1-65535，目的地址协议为tcp协议，地址类型为ipv4，地址为2.2.2.2，端口号为1-65535。

```
# coding: utf-8

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateCaptureTaskRequest()
        sourcebody = CaptureRuleAddressDto(
            address="1.1.1.1",
            address_type=0,
            type=0
        )
        servicebody = CaptureServiceDto(
            dest_port="",
            protocol=-1,
            source_port=""
        )
        destinationbody = CaptureRuleAddressDto(
            address="2.2.2.2",
            address_type=0,
            type=0
        )
        request.body = CaptureTaskDto(
            source=sourcebody,
            service=servicebody,
            name="zhuabaotest",
            max_packets=100000,
            duration=3,
            destination=destinationbody
        )
        response = client.create_capture_task(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09bb24e6fe80d23d2fa2c010b53b418c，防火墙id为ebf891cd-2163-48a0-9963-6309f99dd3c4添加抓包任务，任务名称为zhuabaotest，最大抓包数为100000，抓包时长为3分钟，剩余保留天数为7天，源地址协议为tcp协议，地址类型为ipv4，地址为1.1.1.1，端口号为1-65535，目的地址协议为tcp协议，地址类型为ipv4，地址为2.2.2.2，端口号为1-65535。

```
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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateCaptureTaskRequest{}
    sourcebody := &model.CaptureRuleAddressDto{
        Address: "1.1.1.1",
        AddressType: int32(0),
        Type: int32(0),
    }
    destPortService := ""
    sourcePortService := ""
    servicebody := &model.CaptureServiceDto{
        DestPort: &destPortService,
        Protocol: int32(-1),
        SourcePort: &sourcePortService,
    }
    destinationbody := &model.CaptureRuleAddressDto{
        Address: "2.2.2.2",
        AddressType: int32(0),
        Type: int32(0),
    }
    request.Body = &model.CaptureTaskDto{
        Source: sourcebody,
        Service: servicebody,
        Name: "zhuabaotest",
        MaxPackets: int32(100000),
        Duration: int32(3),
    }
}
```

```
        Destination: destinationbody,
    }
    response, err := client.CreateCaptureTask(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 | 创建抓包任务返回值 |

错误码

请参见[错误码](#)。

4.10.3 批量删除抓包任务

功能介绍

批量删除抓包任务

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/capture-task/batch-delete

表 4-560 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|--|
| project_id | 是 | String | 项目ID, 可以从调API处获取，也可以从控制台获取。 项目ID获取方式 |

表 4-561 Query 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|-----------------------|------|--------|---|
| fw_instance_id | 是 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |
| enterprise_project_id | 否 | String | 企业项目ID, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |

请求参数

表 4-562 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。可通过 如何获取用户Token 获取。 |

表 4-563 请求 Body 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------|------|------------------|--|
| task_ids | 是 | Array of strings | 抓包任务id列表, 抓包任务id可通过 查询抓包任务接口 查询获得, 通过返回值中的 data.records.task_id (表示各对象之间层级的区分) 获得。 |

响应参数

状态码: 200

表 4-564 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--------------------------------------|------------|
| data | CaptureTaskId object | 删除抓包任务返回数据 |

表 4-565 CaptureTaskId

| 参数 | 参数类型 | 描述 |
|------|--------|-------|
| id | String | 防火墙id |
| name | String | 防火墙名称 |

请求示例

向项目id为09bb24e6fe80d23d2fa2c010b53b418c，防火墙id为ebf891cd-2163-48a0-9963-6309f99dd3c4删除抓包任务，抓包任务id为24e6bb6d-d335-48fd-b9c7-bbaaa851d209。

```
https://{Endpoint}/v1/09bb24e6fe80d23d2fa2c010b53b418c/capture-task/batch-delete?fw_instance_id=ebf891cd-2163-48a0-9963-6309f99dd3c4&enterprise_project_id=default
```

```
{
  "task_ids" : [ "24e6bb6d-d335-48fd-b9c7-bbaaa851d209" ]
}
```

响应示例

状态码：200

删除抓包任务返回值

```
{
  "data" : {
    "id" : "ebf891cd-2163-48a0-9963-6309f99dd3c4",
    "name" : "test"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

向项目id为09bb24e6fe80d23d2fa2c010b53b418c，防火墙id为ebf891cd-2163-48a0-9963-6309f99dd3c4删除抓包任务，抓包任务id为24e6bb6d-d335-48fd-b9c7-bbaaa851d209。

```
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 DeleteCaptureTaskSolution {
```

```
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");
    String projectId = "{project_id}";

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

    CfwClient client = CfwClient.newBuilder()
        .withCredential(auth)
        .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
        .build();
    DeleteCaptureTaskRequest request = new DeleteCaptureTaskRequest();
    DeleteCaptureTaskDto body = new DeleteCaptureTaskDto();
    List<String> listbodyTaskIds = new ArrayList<>();
    listbodyTaskIds.add("24e6bb6d-d335-48fd-b9c7-bbaaa851d209");
    body.withTaskIds(listbodyTaskIds);
    request.withBody(body);
    try {
        DeleteCaptureTaskResponse response = client.deleteCaptureTask(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09bb24e6fe80d23d2fa2c010b53b418c，防火墙id为ebf891cd-2163-48a0-9963-6309f99dd3c4删除抓包任务，抓包任务id为24e6bb6d-d335-48fd-b9c7-bbaaa851d209。

```
# coding: utf-8

import os
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")
    projectId = "{project_id}"

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

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

try:
    request = DeleteCaptureTaskRequest()
    listTaskIdsbody = [
        "24e6bb6d-d335-48fd-b9c7-bbaaa851d209"
    ]
    request.body = DeleteCaptureTaskDto(
        task_ids=listTaskIdsbody
    )
    response = client.delete_capture_task(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09bb24e6fe80d23d2fa2c010b53b418c，防火墙id为ebf891cd-2163-48a0-9963-6309f99dd3c4删除抓包任务，抓包任务id为24e6bb6d-d335-48fd-b9c7-bbaaa851d209。

```
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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteCaptureTaskRequest{}
    var listTaskIdsbody = []string{
        "24e6bb6d-d335-48fd-b9c7-bbaaa851d209",
    }
    request.Body = &model.DeleteCaptureTaskDto{
        TaskIds: listTaskIdsbody,
    }
    response, err := client.DeleteCaptureTask(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 | 删除抓包任务返回值 |

错误码

请参见[错误码](#)。

4.10.4 获取抓包任务结果

功能介绍

获取抓包任务结果

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/capture-task/capture-result

表 4-566 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|--|
| project_id | 是 | String | 项目ID, 可以从调API处获取，也可以从控制台获取。 项目ID获取方式 |

表 4-567 Query 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------------|------|--------|---|
| fw_instance_id | 是 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |

| 参数 | 是否必选 | 参数类型 | 描述 |
|-----------------------|------|------------------|---|
| task_id | 是 | String | 抓包任务id, 可通过 查询抓包任务接口 查询获得, 通过返回值中的data.records.task_id (表示各对象之间层级的区分) 获得。 |
| type | 否 | Integer | 是否指定公网ip下载, 0: 无限制, 1: 指定公网ip下载 |
| ip | 否 | Array of strings | 指定公网ip |
| enterprise_project_id | 否 | String | 企业项目ID, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |

请求参数

表 4-568 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。可通过 如何获取用户Token 获取。 |

响应参数

状态码：200

表 4-569 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|---|---------|
| data | CaptureResultUrlVO object | 抓包结果url |

表 4-570 CaptureResultUrlVO

| 参数 | 参数类型 | 描述 |
|---------|--------|-----------------------|
| captcha | String | 下载链接提取码, 用于打开下载链接时使用。 |
| expires | Long | 下载链接过期时间 |

| 参数 | 参数类型 | 描述 |
|----------------|-------------------------------------|---|
| file_list | Array of CaptureFile objects | 抓包文件列表，当环境不支持obs文件夹分享时使用。当此字段存在时，无 captch, expires, url返回值。 |
| request_header | HostHeaderInfo object | 主机请求头，当环境不支持obs文件夹分享时使用 |
| url | String | 下载链接 |

表 4-571 CaptureFile

| 参数 | 参数类型 | 描述 |
|-----------|--------|------|
| file_name | String | 文件名称 |
| url | String | 下载链接 |
| file_path | String | 文件路径 |

表 4-572 HostHeaderInfo

| 参数 | 参数类型 | 描述 |
|------|--------|----|
| Host | String | 主机 |

请求示例

查询项目id为09bb24e6fe80d23d2fa2c010b53b418c，防火墙id为ebf891cd-2163-48a0-9963-6309f99dd3c4下的抓包任务结果。

```
https://{Endpoint}/v1/09bb24e6fe80d23d2fa2c010b53b418c/capture-task/capture-result?fw_instance_id=14f99b45-47df-4e40-aa5d-cf3700ce9c8b&enterprise_project_id=default&task_id=ea47d7f9-8b93-45de-978d-b4bf0613cfed&type=0
```

响应示例

状态码：200

获取抓包结果

```
{
  "data": {
    "captcha": "696821",
    "expires": 30,
    "url": "testurl"
  }
}
```

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 ListCaptureResultSolution {

    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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListCaptureResultRequest request = new ListCaptureResultRequest();
        try {
            ListCaptureResultResponse response = client.listCaptureResult(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

import os
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.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

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

try:
    request = ListCaptureResultRequest()
    response = client.list_capture_result(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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListCaptureResultRequest{}
    response, err := client.ListCaptureResult(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 | 获取抓包结果 |

错误码

请参见[错误码](#)。

4.10.5 取消抓包任务

功能介绍

取消抓包任务

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/capture-task/stop

表 4-573 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID, 可以从调API处获取, 也可以从控制台获取。 项目ID获取方式 |

表 4-574 Query 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|-----------------------|------|--------|---|
| fw_instance_id | 是 | String | 防火墙id, 可通过 防火墙ID获取方式 获取 |
| enterprise_project_id | 否 | String | 企业项目ID, 用户根据组织规划企业项目, 对应的ID为企业项目ID, 可通过 如何获取企业项目ID 获取, 用户未开启企业项目时为0 |

请求参数

表 4-575 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。可通过 如何获取用户Token 获取。 |

表 4-576 请求 Body 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|---------|------|--------|--|
| task_id | 是 | String | 抓包任务id，可通过 查询抓包任务接口 查询获得，通过返回值中的data.records.task_id（.表示各对象之间层级的区分）获得。 |

响应参数

状态码：200

表 4-577 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--------------------------------------|------------|
| data | CaptureTaskId object | 停止抓包任务id信息 |

表 4-578 CaptureTaskId

| 参数 | 参数类型 | 描述 |
|------|--------|-------|
| id | String | 防火墙id |
| name | String | 防火墙名称 |

请求示例

向项目id为09bb24e6fe80d23d2fa2c010b53b418c，防火墙id为ebf891cd-2163-48a0-9963-6309f99dd3c4停止任务id为1dcddab3-7d79-4a56-9e67-4aa0962b98ad的抓包任务。

```
https://{Endpoint}/v1/09bb24e6fe80d23d2fa2c010b53b418c/capture-task/stop?fw_instance_id=ebf891cd-2163-48a0-9963-6309f99dd3c4&enterprise_project_id=default  
  
{  
  "task_id": "1dcddab3-7d79-4a56-9e67-4aa0962b98ad"  
}
```

响应示例

状态码：200

截止抓包任务返回值

```
{
  "data" : {
    "id" : "ebf891cd-2163-48a0-9963-6309f99dd3c4",
    "name" : "test"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

向项目id为09bb24e6fe80d23d2fa2c010b53b418c，防火墙id为ebf891cd-2163-48a0-9963-6309f99dd3c4停止任务id为1dcddab3-7d79-4a56-9e67-4aa0962b98ad的抓包任务。

```
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 CancelCaptureTaskSolution {

    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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        CancelCaptureTaskRequest request = new CancelCaptureTaskRequest();
        CancelCaptureTaskDto body = new CancelCaptureTaskDto();
        body.withTaskId("1dcddab3-7d79-4a56-9e67-4aa0962b98ad");
        request.withBody(body);
        try {
            CancelCaptureTaskResponse response = client.cancelCaptureTask(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09bb24e6fe80d23d2fa2c010b53b418c，防火墙id为ebf891cd-2163-48a0-9963-6309f99dd3c4停止任务id为1dcddab3-7d79-4a56-9e67-4aa0962b98ad的抓包任务。

```
# coding: utf-8

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CancelCaptureTaskRequest()
        request.body = CancelCaptureTaskDto(
            task_id="1dcddab3-7d79-4a56-9e67-4aa0962b98ad"
        )
        response = client.cancel_capture_task(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09bb24e6fe80d23d2fa2c010b53b418c，防火墙id为ebf891cd-2163-48a0-9963-6309f99dd3c4停止任务id为1dcddab3-7d79-4a56-9e67-4aa0962b98ad的抓包任务。

```
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")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := cfw.NewCfwClient(  
        cfw.CfwClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.CancelCaptureTaskRequest{}  
    request.Body = &model.CancelCaptureTaskDto{  
        TaskId: "1dcddab3-7d79-4a56-9e67-4aa0962b98ad",  
    }  
    response, err := client.CancelCaptureTask(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 | 截止抓包任务返回值 |

错误码

请参见[错误码](#)。

4.11 反病毒管理

4.11.1 查看反病毒开关

功能介绍

查看反病毒开关

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/anti-virus/switch

表 4-579 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID，可以从调API处获取，也可以从控制台获取。可通过 项目ID获取方式 获取 |

表 4-580 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，通过返回值中的 data.records.protect_objects.object_id（.表示各对象之间层级的区分）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。此处仅取type为1的防护对象id，可通过 data.records.protect_objects.type（.表示各对象之间层级的区分）获得。 |
| enterprise_project_id | 否 | String | 企业项目ID，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

请求参数

表 4-581 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，可通过 如何获取用户Token 获取 |

响应参数

状态码：200

表 4-582 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--------------------|---------|
| data | AntiVirusVO object | 反病毒开关数据 |

表 4-583 AntiVirusVO

| 参数 | 参数类型 | 描述 |
|-------------------|---------|---------------------|
| anti_virus_status | Integer | 反病毒开关状态，0表示关闭，1表示开启 |
| id | String | 防护对象id |

请求示例

查看项目id为408972e72dcd4c1a9b033e955802a36b的反病毒开关，防火墙id为e743cfaf-8164-4807-aa13-d893d83313cf，企业项目id为fb55459c-41b3-47fc-885d-540946fddda4，目标防护对象id为1b90f031-0c7b-4f25-95e2-b6d9940d269e，引擎类型为1。

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/anti-virus/switch?fw_instance_id=e743cfaf-8164-4807-aa13-d893d83313cf&enterprise_project_id=fb55459c-41b3-47fc-885d-540946fddda4&object_id=1b90f031-0c7b-4f25-95e2-b6d9940d269e&engine_type=1
```

响应示例

状态码：200

查看反病毒开关返回值

```
{  "data": {    "anti_virus_status": 1,    "id": "1b90f031-0c7b-4f25-95e2-b6d9940d269e"  }
```

```
}  
}
```

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 ShowAntiVirusSwitchSolution {  
  
    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");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        CfwClient client = CfwClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowAntiVirusSwitchRequest request = new ShowAntiVirusSwitchRequest();  
        try {  
            ShowAntiVirusSwitchResponse response = client.showAntiVirusSwitch(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  
  
import os  
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowAntiVirusSwitchRequest{}
    response, err := client.ShowAntiVirusSwitch(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 | 查看反病毒开关返回值 |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |

错误码

请参见[错误码](#)。

4.11.2 修改反病毒开关

功能介绍

修改反病毒开关

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/anti-virus/switch

表 4-584 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID，可以从调API处获取，也可以从控制台获取。可通过 项目ID获取方式 获取 |

表 4-585 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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

请求参数

表 4-586 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，可通过 如何获取用户Token 获取 |

表 4-587 请求 Body 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|-------------------|------|---------|---------|
| anti_virus_status | 否 | Integer | 反病毒开关状态 |
| object_id | 否 | String | 防护对象ID |

响应参数

状态码：200

表 4-588 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|-------------------------------------|--------------|
| data | ResponseData object | 更新反病毒开关返回值数据 |

表 4-589 ResponseData

| 参数 | 参数类型 | 描述 |
|----|--------|------|
| id | String | 数据id |

请求示例

打开反病毒开关

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/anti-virus/switch?  
fw_instance_id=e743cfaf-8164-4807-aa13-  
d893d83313cf&enterprise_project_id=fb55459c-41b3-47fc-885d-540946fddda4
```

```
{  
  "object_id" : "1b90f031-0c7b-4f25-95e2-b6d9940d269e",  
  "anti_virus_status" : 1  
}
```

响应示例

状态码：200

更新反病毒开关返回值

```
{  
  "data" : {  
    "id" : "1b90f031-0c7b-4f25-95e2-b6d9940d269e"  
  }  
}
```

SDK 代码示例

SDK代码示例如下。

Java

打开反病毒开关

```
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 UpdateAntiVirusSwitchSolution {  
  
    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");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        CfwClient client = CfwClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))  
            .build();
```

```
UpdateAntiVirusSwitchRequest request = new UpdateAntiVirusSwitchRequest();
AntiVirusSwitchDto body = new AntiVirusSwitchDto();
body withObjectId("1b90f031-0c7b-4f25-95e2-b6d9940d269e");
body withAntiVirusStatus(1);
request withBody(body);
try {
    UpdateAntiVirusSwitchResponse response = client.updateAntiVirusSwitch(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

打开反病毒开关

```
# coding: utf-8

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateAntiVirusSwitchRequest()
        request.body = AntiVirusSwitchDto(
            object_id="1b90f031-0c7b-4f25-95e2-b6d9940d269e",
            anti_virus_status=1
        )
        response = client.update_anti_virus_switch(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

打开反病毒开关


```
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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateAntiVirusSwitchRequest{}
    objectIdAntiVirusSwitchDto := "1b90f031-0c7b-4f25-95e2-b6d9940d269e"
    antiVirusStatusAntiVirusSwitchDto := int32(1)
    request.Body = &model.AntiVirusSwitchDto{
        ObjectId: &objectIdAntiVirusSwitchDto,
        AntiVirusStatus: &antiVirusStatusAntiVirusSwitchDto,
    }
    response, err := client.UpdateAntiVirusSwitch(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 | 更新反病毒开关返回值 |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |

错误码

请参见[错误码](#)。

4.11.3 获取防火墙反病毒规则信息

功能介绍

获取防火墙反病毒规则信息

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/anti-virus/rule

表 4-590 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID，可以从调API处获取，也可以从控制台获取。可通过 项目ID获取方式 获取 |

表 4-591 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，通过返回值中的 data.records.protect_objects.object_id（.表示各对象之间层级的区分）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。此处仅取type为1的防护对象id，可通过 data.records.protect_objects.type（.表示各对象之间层级的区分）获得。 |
| limit | 是 | Integer | 每页显示个数，范围为1-1024 |

| 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0，首页时为空，非首页时不为空 |
| enterprise_project_id | 否 | String | 企业项目ID，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

请求参数

表 4-592 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，可通过 如何获取用户Token 获取 |

响应参数

状态码：200

表 4-593 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--|--------------|
| data | AntiVirusRuleVO object | 查询反病毒规则返回值数据 |

表 4-594 AntiVirusRuleVO

| 参数 | 参数类型 | 描述 |
|-----------------------|---|-------------|
| id | String | 反病毒规则id |
| scan_protocol_configs | Array of ScanProtocolConfig objects | 反病毒扫描协议列表 |
| total | Integer | 反病毒扫描协议列表总数 |

表 4-595 ScanProtocolConfig

| 参数 | 参数类型 | 描述 |
|---------------|---------|--|
| action | Integer | 反病毒动作, 0: 观察 1: 拦截 2: 禁用 |
| protocol_type | Integer | 协议类型, 包括0: HTTP、1: SMTP、2: POP3、3: IMAP4、4: FTP、5: SMB、6: 恶意访问、7: IM |

请求示例

查询项目id为408972e72dcd4c1a9b033e955802a36b的防病毒规则, 防火墙id为e743cfaf-8164-4807-aa13-d893d83313cf, 企业项目id为fb55459c-41b3-47fc-885d-540946fddda4, 目标对象id为1b90f031-0c7b-4f25-95e2-b6d9940d269e, 引擎类型为1。查询结果限制为50条, 偏移量为0。

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/anti-virus/rule?fw_instance_id=e743cfaf-8164-4807-aa13-d893d83313cf&enterprise_project_id=fb55459c-41b3-47fc-885d-540946fddda4&limit=50&offset=0&object_id=1b90f031-0c7b-4f25-95e2-b6d9940d269e&engine_type=1
```

响应示例

状态码: 200

查询反病毒规则返回值

```
{
  "data": {
    "scan_protocol_configs": [ {
      "action": 0,
      "protocol_type": 0
    }, {
      "action": 1,
      "protocol_type": 1
    }, {
      "action": 2,
      "protocol_type": 2
    }, {
      "action": 2,
      "protocol_type": 4
    }, {
      "action": 2,
      "protocol_type": 3
    }, {
      "action": 2,
      "protocol_type": 6
    }, {
      "action": 2,
      "protocol_type": 5
    }
  ],
  "total": 7
}
```

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 ShowAntiVirusRuleSolution {

    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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowAntiVirusRuleRequest request = new ShowAntiVirusRuleRequest();
        try {
            ShowAntiVirusRuleResponse response = client.showAntiVirusRule(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

import os
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.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

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

try:
    request = ShowAntiVirusRuleRequest()
    response = client.show_anti_virus_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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowAntiVirusRuleRequest{}
    response, err := client.ShowAntiVirusRule(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 | 查询反病毒规则返回值 |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |

错误码

请参见[错误码](#)。

4.11.4 修改反病毒规则

功能介绍

修改反病毒规则

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/anti-virus/rule

表 4-596 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID，可以从调API处获取，也可以从控制台获取。可通过 项目ID获取方式 获取 |

表 4-597 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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

请求参数

表 4-598 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，可通过 如何获取用户Token 获取 |

表 4-599 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 |
| scan_protocol_configs | 否 | Array of ScanProtocol Config objects | 扫描协议配置列表 |

表 4-600 ScanProtocolConfig

| 参数 | 是否必选 | 参数类型 | 描述 |
|---------------|------|---------|---|
| action | 否 | Integer | 反病毒动作，0：观察 1：拦截 2：禁用 |
| protocol_type | 否 | Integer | 协议类型，包括0：HTTP、1：SMTP、2：POP3、3：IMAP4、4：FTP、5：SMB、6：恶意访问、7：IM |

响应参数

状态码：200

表 4-601 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|-------------------------------------|----------------|
| data | ResponseData object | 更新反病毒规则列表返回值数据 |

表 4-602 ResponseData

| 参数 | 参数类型 | 描述 |
|----|--------|------|
| id | String | 数据id |

请求示例

修改项目id为408972e72dcd4c1a9b033e955802a36b的反病毒规则为”观察“，防火墙id为e743cfaf-8164-4807-aa13-d893d83313cf，企业项目id为fb55459c-41b3-47fc-885d-540946fddda4。

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/anti-virus/rule?
fw_instance_id=e743cfaf-8164-4807-aa13-
d893d83313cf&enterprise_project_id=fb55459c-41b3-47fc-885d-540946fddda4

{
  "object_id" : "1b90f031-0c7b-4f25-95e2-b6d9940d269e",
  "scan_protocol_configs" : [ {
    "protocol_type" : 2,
    "action" : 0
  } ]
}
```

响应示例

状态码：200

更新反病毒规则返回值

```
{
  "data" : {
    "id" : "1b90f031-0c7b-4f25-95e2-b6d9940d269e"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

修改项目id为408972e72dcd4c1a9b033e955802a36b的反病毒规则为”观察“，防火墙id为e743cfaf-8164-4807-aa13-d893d83313cf，企业项目id为fb55459c-41b3-47fc-885d-540946fddda4。

```
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 UpdateAntiVirusRuleSolution {
    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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateAntiVirusRuleRequest request = new UpdateAntiVirusRuleRequest();
        AntiVirusRuleDto body = new AntiVirusRuleDto();
        List<ScanProtocolConfig> listbodyScanProtocolConfigs = new ArrayList<>();
        listbodyScanProtocolConfigs.add(
            new ScanProtocolConfig()
                .withAction(0)
                .withProtocolType(2)
        );
        body.withScanProtocolConfigs(listbodyScanProtocolConfigs);
        body.withObjectId("1b90f031-0c7b-4f25-95e2-b6d9940d269e");
        request.withBody(body);
        try {
            UpdateAntiVirusRuleResponse response = client.updateAntiVirusRule(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

修改项目id为408972e72dcd4c1a9b033e955802a36b的反病毒规则为”观察“，防火墙id为e743cfaf-8164-4807-aa13-d893d83313cf，企业项目id为fb55459c-41b3-47fc-885d-540946fddda4。

```
# coding: utf-8

import os
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.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

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

try:
    request = UpdateAntiVirusRuleRequest()
    listScanProtocolConfigsbody = [
        ScanProtocolConfig(
            action=0,
            protocol_type=2
        )
    ]
    request.body = AntiVirusRuleDto(
        scan_protocol_configs=listScanProtocolConfigsbody,
        object_id="1b90f031-0c7b-4f25-95e2-b6d9940d269e"
    )
    response = client.update_anti_virus_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

修改项目id为408972e72dcd4c1a9b033e955802a36b的反病毒规则为”观察“，防火墙id为e743cfaf-8164-4807-aa13-d893d83313cf，企业项目id为fb55459c-41b3-47fc-885d-540946fddda4。

```
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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```
Build()  
  
request := &model.UpdateAntiVirusRuleRequest{  
    actionScanProtocolConfigs:= int32(0)  
    protocolTypeScanProtocolConfigs:= int32(2)  
    var listScanProtocolConfigbody = []model.ScanProtocolConfig{  
        {  
            Action: &actionScanProtocolConfigs,  
            ProtocolType: &protocolTypeScanProtocolConfigs,  
        },  
    }  
    objectIdAntiVirusRuleDto:= "1b90f031-0c7b-4f25-95e2-b6d9940d269e"  
    request.Body = &model.AntiVirusRuleDto{  
        ScanProtocolConfigs: &listScanProtocolConfigbody,  
        ObjectId: &objectIdAntiVirusRuleDto,  
    }  
    response, err := client.UpdateAntiVirusRule(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 | 更新反病毒规则返回值 |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |

错误码

请参见[错误码](#)。

4.12 告警配置管理

4.12.1 获取告警配置信息

功能介绍

获取告警配置信息

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/cfw/alarm/config

表 4-603 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID，可以从调API处获取，也可以从控制台获取。可通过 项目ID获取方式 获取 |

表 4-604 Query 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|-----------------------|------|--------|---|
| fw_instance_id | 是 | String | 防火墙ID，可通过 防火墙ID获取方式 获取 |
| enterprise_project_id | 否 | String | 企业项目ID，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

请求参数

表 4-605 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，可通过 如何获取用户Token 获取 |

响应参数

状态码：200

表 4-606 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|---------------|--|--------|
| alarm_configs | Array of AlarmConfig objects | 告警配置列表 |

表 4-607 AlarmConfig

| 参数 | 参数类型 | 描述 |
|-------------------|---------|---|
| alarm_time_period | Integer | 告警周期, 0: 全天, 1: 8时到22时 |
| alarm_type | Integer | 告警类型 0:攻击告警; 1:流量超额预警; 2:EIP未防护告警; 3:威胁情报告警 |
| enable_status | Integer | 告警状态 0:失效; 1:生效 |
| frequency_count | Integer | 告警触发频次 |
| frequency_time | Integer | 告警频次时间范围, 以分钟为单位 |
| language | String | 告警语言, zh-cn为中文, en-us为英文 |
| name | String | 通知群组名称 |
| severity | String | 告警等级, 当type为0和4时, severity为CRITICAL,HIGH,MEDIUM,LOW四种等级的组合字符串, 当type为2时, severity固定为3 |
| topic_urn | String | 告警topic的urn |
| username | String | 用户名称, 为cfw |

请求示例

查询项目id为408972e72dcd4c1a9b033e955802a36b的防火墙告警配置, 防火墙id为8de37729-026d-412f-ba82-80c46f388d14。

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/cfw/alarm/config?fw_instance_id=8de37729-026d-412f-ba82-80c46f388d14
```

响应示例

状态码: 200

查询告警配置列表返回值

```
{
  "alarm_configs": [ {
    "alarm_time_period": 0,
    "alarm_type": 1,
    "enable_status": 0,
    "severity": "1"
  }, {
    "alarm_time_period": 0,
    "alarm_type": 2,
    "enable_status": 0,
    "severity": "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 ShowAlarmConfigSolution {

    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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowAlarmConfigRequest request = new ShowAlarmConfigRequest();
        try {
            ShowAlarmConfigResponse response = client.showAlarmConfig(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

import os
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.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

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

try:
    request = ShowAlarmConfigRequest()
    response = client.show_alarm_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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowAlarmConfigRequest{}
    response, err := client.ShowAlarmConfig(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 | 查询告警配置列表返回值 |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |

错误码

请参见[错误码](#)。

4.12.2 修改告警配置接口

功能介绍

修改告警配置接口

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/cfw/alarm/config

表 4-608 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID，可以从调API处获取，也可以从控制台获取。可通过 项目ID获取方式 获取 |

表 4-609 Query 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|-----------------------|------|--------|---|
| fw_instance_id | 是 | String | 防火墙ID，可通过 防火墙ID获取方式 获取 |
| enterprise_project_id | 否 | String | 企业项目ID，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

请求参数

表 4-610 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，可通过 如何获取用户Token 获取 |

表 4-611 请求 Body 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|-------------------|------|---------|---|
| alarm_time_period | 否 | Integer | 告警周期，0：全天，1：8时到22时 |
| alarm_type | 否 | Integer | 告警类型 0：攻击告警；1：流量超额预警；2：EIP未防护告警；3：威胁情报告警 |
| enable_status | 否 | Integer | 告警状态 0：失效；1：生效 |
| frequency_count | 否 | Integer | 告警触发频次 |
| frequency_time | 否 | Integer | 告警频次时间范围，以分钟为单位 |
| language | 否 | String | 告警语言，zh-cn为中文，en-us为英文 |
| severity | 否 | String | 告警等级，当type为0和4时，severity为CRITICAL,HIGH,MEDIUM,LOW四种等级的组合字符串，当type为2时，severity固定为3 |
| topic_urn | 否 | String | 告警urn |
| username | 否 | String | 用户名称，为cfw |

响应参数

状态码：200

表 4-612 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|-------------------------------------|-------|
| data | ResponseData object | 通用返回体 |

表 4-613 ResponseData

| 参数 | 参数类型 | 描述 |
|----|--------|------|
| id | String | 数据id |

请求示例

配置项目id为408972e72dcd4c1a9b033e955802a36b的防火墙告警，防火墙id为8de37729-026d-412f-ba82-80c46f388d14。告警类型为1，告警时间周期为0，启用状态为0，严重程度为0。告警策略为防护带宽使用率达到80%时，每天告警一次。告警名称为test-wyl，主题URN为urn:smn:cn-north-7:408972e72dcd4c1a9b033e955802a36b:test-wyl。

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/cfw/alarm/config?fw_instance_id=8de37729-026d-412f-ba82-80c46f388d14

{
  "alarm_time_period" : 0,
  "alarm_type" : 1,
  "enable_status" : 0,
  "severity" : "0",
  "frequency_count" : 1,
  "frequency_time" : 1,
  "topic_urn" : "urn:smn:cn-north-7:408972e72dcd4c1a9b033e955802a36b:test-wyl"
}
```

响应示例

状态码：200

OK

```
{
  "data" : {
    "id" : "8de37729-026d-412f-ba82-80c46f388d14"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

配置项目id为408972e72dcd4c1a9b033e955802a36b的防火墙告警，防火墙id为8de37729-026d-412f-ba82-80c46f388d14。告警类型为1，告警时间周期为0，启用状态为0，严重程度为0。告警策略为防护带宽使用率达到80%时，每天告警一次。告警名称为test-wyl，主题URN为urn:smn:cn-north-7:408972e72dcd4c1a9b033e955802a36b:test-wyl。

```
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 UpdateAlarmConfigSolution {

    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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateAlarmConfigRequest request = new UpdateAlarmConfigRequest();
        UpdateAttackLogAlarmConfigDto body = new UpdateAttackLogAlarmConfigDto();
        body.withTopicUrn("urn:smn:cn-north-7:408972e72dcd4c1a9b033e955802a36b:test-wyl");
        body.withSeverity("0");
        body.withFrequencyTime(1);
        body.withFrequencyCount(1);
        body.withEnableStatus(0);
        body.withAlarmType(1);
        body.withAlarmTimePeriod(0);
        request.withBody(body);
        try {
            UpdateAlarmConfigResponse response = client.updateAlarmConfig(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

配置项目id为408972e72dcd4c1a9b033e955802a36b的防火墙告警，防火墙id为8de37729-026d-412f-ba82-80c46f388d14。告警类型为1，告警时间周期为0，启用状态为0，严重程度为0。告警策略为防护带宽使用率达到80%时，每天告警一次。告警名称为test-wyl，主题URN为urn:smn:cn-north-7:408972e72dcd4c1a9b033e955802a36b:test-wyl。

```
# coding: utf-8

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateAlarmConfigRequest()
        request.body = UpdateAttackLogAlarmConfigDto(
            topic_urn="urn:smn:cn-north-7:408972e72dcd4c1a9b033e955802a36b:test-wyl",
            severity="0",
            frequency_time=1,
            frequency_count=1,
            enable_status=0,
            alarm_type=1,
            alarm_time_period=0
        )
        response = client.update_alarm_config(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

配置项目id为408972e72dcd4c1a9b033e955802a36b的防火墙告警，防火墙id为8de37729-026d-412f-ba82-80c46f388d14。告警类型为1，告警时间周期为0，启用状态为0，严重程度为0。告警策略为防护带宽使用率达到80%时，每天告警一次。告警名称为test-wyl，主题URN为urn:smn:cn-north-7:408972e72dcd4c1a9b033e955802a36b:test-wyl。

```
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")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
```

```
Build()

client := cfw.NewCfwClient(
    cfw.CfwClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateAlarmConfigRequest{
    topicUrnUpdateAttackLogAlarmConfigDto:= "urn:smn:cn-
north-7:408972e72dcd4c1a9b033e955802a36b:test-wyl"
    severityUpdateAttackLogAlarmConfigDto:= "0"
    frequencyTimeUpdateAttackLogAlarmConfigDto:= int32(1)
    frequencyCountUpdateAttackLogAlarmConfigDto:= int32(1)
    enableStatusUpdateAttackLogAlarmConfigDto:= int32(0)
    alarmTypeUpdateAttackLogAlarmConfigDto:= int32(1)
    alarmTimePeriodUpdateAttackLogAlarmConfigDto:= int32(0)
    request.Body = &model.UpdateAttackLogAlarmConfigDto{
        TopicUrn: &topicUrnUpdateAttackLogAlarmConfigDto,
        Severity: &severityUpdateAttackLogAlarmConfigDto,
        FrequencyTime: &frequencyTimeUpdateAttackLogAlarmConfigDto,
        FrequencyCount: &frequencyCountUpdateAttackLogAlarmConfigDto,
        EnableStatus: &enableStatusUpdateAttackLogAlarmConfigDto,
        AlarmType: &alarmTypeUpdateAttackLogAlarmConfigDto,
        AlarmTimePeriod: &alarmTimePeriodUpdateAttackLogAlarmConfigDto,
    }
}
response, err := client.UpdateAlarmConfig(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 |

错误码

请参见[错误码](#)。

4.13 标签管理

4.13.1 查询标签信息

功能介绍

查询标签信息

调用方法

请参见[如何调用API](#)。

URI

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

表 4-614 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------------|------|--------|---|
| project_id | 是 | String | 项目ID，可以从调API处获取，也可以从控制台获取。可通过 项目ID获取方式 获取 |

表 4-615 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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

请求参数

表 4-616 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，可通过 如何获取用户Token 获取 |

响应参数

状态码：200

表 4-617 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|----------------------------------|-------|
| tags | Array of TagValue objects | tag列表 |

表 4-618 TagValue

| 参数 | 参数类型 | 描述 |
|-------|------------------|------|
| key | String | tag键 |
| value | Array of strings | tag值 |

请求示例

查询防火墙id为e743cfaf-8164-4807-aa13-d893d83313cf的标签信息

```
https://{Endpoint}/v2/408972e72dcd4c1a9b033e955802a36b/cfw-cfw/tags?  
fw_instance_id=e743cfaf-8164-4807-aa13-  
d893d83313cf&enterprise_project_id=fb55459c-41b3-47fc-885d-540946fddda4
```

响应示例

状态码：200

查询标签返回值

```
{  
  "tags": [ {  
    "key": "test",  
    "value": [ "1" ]  
  }, {  
    "key": "tag_zjw-1",  
    "value": [ "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 ListProjectTagsSolution {

    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");
        String projectId = "{project_id}";

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

        CfwClient client = CfwClient.newBuilder()
            .withCredential(auth)
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))
            .build();
        ListProjectTagsRequest request = new ListProjectTagsRequest();
        try {
            ListProjectTagsResponse response = client.listProjectTags(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

import os
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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

```
.build()

try:
    request = ListProjectTagsRequest()
    response = client.list_project_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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListProjectTagsRequest{}
    response, err := client.ListProjectTags(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 | 查询标签返回值 |

| 状态码 | 描述 |
|-----|--------------|
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |

错误码

请参见[错误码](#)。

4.13.2 查询资源标签信息

功能介绍

查询资源标签信息

调用方法

请参见[如何调用API](#)。

URI

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

表 4-619 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------------|------|--------|---|
| fw_instance_id | 是 | String | 防火墙ID，可通过 防火墙ID获取方式 获取 |
| project_id | 是 | String | 项目ID，可以从调API处获取，也可以从控制台获取。可通过 项目ID获取方式 获取 |

表 4-620 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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

请求参数

表 4-621 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，可通过 如何获取用户Token 获取 |

响应参数

状态码：200

表 4-622 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--|--------|
| tags | Array of ResourceTag objects | 资源标签列表 |

表 4-623 ResourceTag

| 参数 | 参数类型 | 描述 |
|-------------|--------|--------|
| key | String | 标签键 |
| value | String | 标签值 |
| update_time | String | 标签更新时间 |

请求示例

查询项目id为408972e72dcd4c1a9b033e955802a36b的防火墙资源标签

```
https://{Endpoint}/v2/408972e72dcd4c1a9b033e955802a36b/cfw-cfw/e743cfaf-8164-4807-aa13-d893d83313cf/tags?fw_instance_id=e743cfaf-8164-4807-aa13-d893d83313cf&enterprise_project_id=fb55459c-41b3-47fc-885d-540946fddda4
```

响应示例

状态码：200

查询资源标签返回值

```
{
  "tags": [ {
    "key": "test",
    "value": "1"
  }, {
    "key": "tag_zjw-1",
    "value": "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 ListResourceTagsSolution {  
    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");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        CfwClient client = CfwClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CfwRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListResourceTagsRequest request = new ListResourceTagsRequest();  
        request.withFwInstanceId("{fw_instance_id}");  
        try {  
            ListResourceTagsResponse response = client.listResourceTags(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  
  
import os  
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListResourceTagsRequest()
        request.fw_instance_id = "{fw_instance_id}"
        response = client.list_resource_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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListResourceTagsRequest{}
    request.FwInstanceId = "{fw_instance_id}"
    response, err := client.ListResourceTags(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 | 查询资源标签返回值 |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |

错误码

请参见[错误码](#)。

4.13.3 保存资源标签接口

功能介绍

保存资源标签接口

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/cfw-cfw/{fw_instance_id}/tags/save

表 4-624 路径参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|----------------|------|--------|---|
| fw_instance_id | 是 | String | 防火墙ID，可通过 防火墙ID获取方式 获取 |
| project_id | 是 | String | 项目ID，可以从调API处获取，也可以从控制台获取。可通过 项目ID获取方式 获取 |

表 4-625 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，用户根据组织规划企业项目，对应的ID为企业项目ID，可通过 如何获取企业项目ID 获取，用户未开启企业项目时为0 |

请求参数

表 4-626 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，可通过 如何获取用户Token 获取 |

表 4-627 请求 Body 参数

| 参数 | 是否必选 | 参数类型 | 描述 |
|------|------|--|-----------|
| tags | 否 | Array of CreateTag objects | 创建防火墙标签列表 |

表 4-628 CreateTag

| 参数 | 是否必选 | 参数类型 | 描述 |
|-------|------|--------|-----|
| key | 否 | String | 标签键 |
| value | 否 | String | 标签值 |

响应参数

状态码：200

表 4-629 响应 Body 参数

| 参数 | 参数类型 | 描述 |
|------|--------|----|
| data | String | |

请求示例

保存项目id为408972e72dcd4c1a9b033e955802a36b的防火墙资源标签，防火墙id为e743cfaf-8164-4807-aa13-d893d83313cf，企业项目id为fb55459c-41b3-47fc-885d-540946fddda4。请求体中包含两个标签，分别为key:test、value:1和key:tag_zjw-1、value:1。

```
https://{Endpoint}/v2/408972e72dcd4c1a9b033e955802a36b/cfw-cfw/e743cfaf-8164-4807-aa13-d893d83313cf/tags/save?fw_instance_id=e743cfaf-8164-4807-aa13-d893d83313cf&enterprise_project_id=fb55459c-41b3-47fc-885d-540946fddda4
```

```
{
  "tags": [ {
    "key": "test",
    "value": "1"
  }, {
    "key": "tag_zjw-1",
    "value": "1"
  } ]
}
```

响应示例

状态码：200

保存标签返回值

```
{}
```

SDK 代码示例

SDK代码示例如下。

Java

保存项目id为408972e72dcd4c1a9b033e955802a36b的防火墙资源标签，防火墙id为e743cfaf-8164-4807-aa13-d893d83313cf，企业项目id为fb55459c-41b3-47fc-885d-540946fddda4。请求体中包含两个标签，分别为key:test、value:1和key:tag_zjw-1、value:1。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.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 SaveTagsSolution {

    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");
```

```
String projectId = "{project_id}";

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

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

SaveTagsRequest request = new SaveTagsRequest();
request.withFwInstanceId("{fw_instance_id}");
CreateTagsDto body = new CreateTagsDto();
List<CreateTag> listbodyTags = new ArrayList<>();
listbodyTags.add(
    new CreateTag()
        .withKey("test")
        .withValue("1")
);
listbodyTags.add(
    new CreateTag()
        .withKey("tag_zjw-1")
        .withValue("1")
);
body.withTags(listbodyTags);
request.withBody(body);
try {
    SaveTagsResponse response = client.saveTags(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

保存项目id为408972e72dcd4c1a9b033e955802a36b的防火墙资源标签，防火墙id为e743cfaf-8164-4807-aa13-d893d83313cf，企业项目id为fb55459c-41b3-47fc-885d-540946fddda4。请求体中包含两个标签，分别为key:test、value:1和key:tag_zjw-1、value:1。

```
# coding: utf-8

import os
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.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"
```

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

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

try:
    request = SaveTagsRequest()
    request.fw_instance_id = "{fw_instance_id}"
    listTagsbody = [
        CreateTag(
            key="test",
            value="1"
        ),
        CreateTag(
            key="tag_zjw-1",
            value="1"
        )
    ]
    request.body = CreateTagsDto(
        tags=listTagsbody
    )
    response = client.save_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

保存项目id为408972e72dcd4c1a9b033e955802a36b的防火墙资源标签，防火墙id为e743cfaf-8164-4807-aa13-d893d83313cf，企业项目id为fb55459c-41b3-47fc-885d-540946fddda4。请求体中包含两个标签，分别为key:test、value:1和key:tag_zjw-1、value:1。

```
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")
    projectId := "{project_id}"

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

    client := cfw.NewCfwClient(
        cfw.CfwClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

```
WithCredential(auth).
Build()

request := &model.SaveTagsRequest{}
request.FwInstanceId = "{fw_instance_id}"
keyTags:= "test"
valueTags:= "1"
keyTags1:= "tag_zjw-1"
valueTags1:= "1"
var listTagsbody = []model.CreateTag{
    {
        Key: &keyTags,
        Value: &valueTags,
    },
    {
        Key: &keyTags1,
        Value: &valueTags1,
    },
}
request.Body = &model.CreateTagsDto{
    Tags: &listTagsbody,
}
response, err := client.SaveTags(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 | 保存标签返回值 |
| 201 | Created |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |

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。

| 模块 | 状态码 | 错误码 | 错误信息 | 描述 | 处理措施 |
|------|-----|------------------|----------|----------|--------------|
| 公共模块 | 400 | CFW.0010 9004 | HTTP请求错误 | HTTP请求错误 | 请稍后重试或联系技术支持 |

| 模块 | 状态码 | 错误码 | 错误信息 | 描述 | 处理措施 |
|-----------|-----|--------------|--------------|--------------|----------------------|
| | 400 | CFW.00300001 | 查询数据库错误 | 查询数据库错误 | 请联系技术支持 |
| | 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.00800001 | 查询ETCD异常 | 查询ETCD异常 | 请联系技术支持 |
| | 400 | CFW.00800002 | 查询ETCD异常 | 查询ETCD异常 | 请联系技术支持 |
| | 400 | CFW.00800003 | 查询ETCD异常 | 查询ETCD异常 | 请联系技术支持 |
| | 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 |
| DNS解析 | 400 | CFW.00200005 | 请求中携带的域名组不存在 | 请求中携带的域名组不存在 | 请检查请求中携带的域名组是否存在 |
| 删除域名 | 400 | CFW.00200005 | 请求中携带的域名组不存在 | 请求中携带的域名组不存在 | 请检查请求中携带的域名组是否存在 |
| 删除域名组 | 400 | CFW.00200004 | 删除的域名组被引用 | 删除的域名组被引用 | 请删除引用该域名组的规则后再删除该域名组 |
| | 400 | CFW.00200005 | 请求中携带的域名组不存在 | 请求中携带的域名组不存在 | 请检查请求中携带的域名组是否存在 |
| 更新域名组 | 400 | CFW.00200005 | 请求中携带的域名组不存在 | 请求中携带的域名组不存在 | 请检查请求中携带的域名组是否存在 |
| 查询访问控制日志 | 400 | CFW.00500002 | 时间间距错误 | 时间间距错误 | 请联系技术支持 |
| 查询攻击日志 | 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 |
| 查询流量日志 | 400 | CFW.00500002 | 时间间距错误 | 时间间距错误 | 请联系技术支持 |
| 查询防护EIP列表 | 400 | CFW.00200030 | 地址类型错误 | 地址类型错误 | 请联系技术支持 |
| | 400 | CFW.00200016 | 实例状态错误 | 实例状态错误 | 请联系技术支持 |

| 模块 | 状态码 | 错误码 | 错误信息 | 描述 | 处理措施 |
|---------|-----|--------------|-----------------------|-----------------------|--------------------|
| 查询规则列表 | 400 | CFW.00200030 | 地址类型错误 | 地址类型错误 | 请联系技术支持 |
| 添加ACL规则 | 400 | CFW.00200001 | 规则方向不能为空 | 规则方向不能为空 | 请检查请求参数中规则的方向是否为空 |
| | 400 | CFW.00200005 | 请求中携带的地址组id不存在 | 请求中携带的地址组id不存在 | 请检查请求中携带的地址组id是否存在 |
| | 400 | CFW.00200005 | 请求中携带的服务组id不存在 | 请求中携带的服务组id不存在 | 请检查请求中携带的服务组id是否存在 |
| | 400 | CFW.00200005 | 请求中携带的域名组不存在 | 请求中携带的域名组不存在 | 请检查请求中携带的域名组是否存在 |
| | 400 | CFW.00200007 | 请求中携带的规则名称与数据库中的名称有重复 | 请求中携带的规则名称与数据库中的名称有重复 | 请删除请求中重复的规则 |
| | 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.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.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.00200032 | 引擎不支持IPv6 | 引擎不支持IPv6 | 请联系技术支持 |
| | 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.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 |
| 更新ACL规则 | 400 | CFW.00200005 | 请求中携带的地址组id不存在 | 请求中携带的地址组id不存在 | 请检查请求中携带的地址组id是否正确 |
| | 400 | CFW.00200005 | 请求中携带的服务组id不存在 | 请求中携带的服务组id不存在 | 请检查请求中携带的服务组id是否正确 |

| 模块 | 状态码 | 错误码 | 错误信息 | 描述 | 处理措施 |
|------------|-----|------------------|-----------------------|-----------------------|-----------------------|
| | 400 | CFW.0020 0005 | 请求中携带的域名组不存在 | 请求中携带的域名组不存在 | 请检查请求中携带的域名组是否正确 |
| | 400 | CFW.0020 0007 | 请求中携带的规则名称与数据库中的名称有重复 | 请求中携带的规则名称与数据库中的名称有重复 | 请删除请求中重复的规则 |
| | 400 | CFW.0020 0025 | 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.0020 0026 | 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.0020 0028 | 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.0040 0010 | 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 |
| 修改ACL规则优先级 | 400 | CFW.0040 0002 | 无需操作 | 无需操作 | 请联系技术支持 |
| 删除规则击中次数 | 400 | CFW.0040 0006 | 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 |
| 创建抓包任务 | 400 | CFW.0020 0028 | 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.0020 0032 | 引擎不支持IPv6 | 引擎不支持IPv6 | 请联系技术支持 |
| 查询抓包任务列表 | 400 | CFW.0020 0030 | 地址类型错误 | 地址类型错误 | 请联系技术支持 |
| 创建东西向防火墙 | 400 | CFW.0070 0001 | 关联ER不存在 | 关联ER不存在 | 关联ER不存在 |
| | 400 | CFW.0070 0002 | 关联VPC不存在 | 关联VPC不存在 | 请检查VPC是否存在 |
| | 400 | CFW.0070 0003 | 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.0070 0004 | 子网创建失败 | 子网创建失败 | 请联系技术支持 |

| 模块 | 状态码 | 错误码 | 错误信息 | 描述 | 处理措施 |
|-----------|-----|--------------|------------------------|------------------------|------------------|
| | 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 |
| 修改东西向防护状态 | 400 | CFW.00200016 | 实例状态错误 | 实例状态错误 | 请联系技术支持 |
| 创建防火墙 | 400 | CFW.00600003 | 查询可用规格为空 | 查询可用规格为空 | 请联系技术支持 |
| 删除防火墙 | 400 | CFW.00200016 | 实例状态错误 | 实例状态错误 | 请联系技术支持 |
| 添加地址组 | 400 | CFW.00200001 | 地址组名称为空 | 地址组名称为空 | 请检查请求中地址组名称是否为空 |
| | 400 | CFW.00200007 | 请求中携带的地址组名称与数据库中的名称有重复 | 请求中携带的地址组名称与数据库中的名称有重复 | 请删除请求中重复的地址组名称 |
| | 400 | CFW.00200032 | 引擎不支持IPv6 | 引擎不支持IPv6 | 请联系技术支持 |
| | 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 |
| 添加地址组成员列表 | 400 | CFW.00200001 | 地址组成员列表为空 | 地址组成员列表为空 | 请检查请求中地址组成员是否为空 |
| 获取地址组列表 | 400 | CFW.00200030 | 地址类型错误 | 地址类型错误 | 请联系技术支持 |
| 更新地址组 | 400 | CFW.00200005 | 请求中携带的地址组不存在 | 请求中携带的地址组不存在 | 请检查请求中携带的地址组是否存在 |
| | 400 | CFW.00200007 | 请求中携带的地址组名称与数据库中的名称有重复 | 请求中携带的地址组名称与数据库中的名称有重复 | 请删除请求中重复的地址组名称 |

| 模块 | 状态码 | 错误码 | 错误信息 | 描述 | 处理措施 |
|---------|-----|--------------|------------------------|------------------------|----------------------|
| | 400 | CFW.00200016 | 实例状态错误 | 实例状态错误 | 请联系技术支持 |
| 删除地址组 | 400 | CFW.00200004 | 删除的地址组被引用 | 删除的地址组被引用 | 请删除引用该地址组的规则后再删除该地址组 |
| | 400 | CFW.00200005 | 请求中携带的地址组不存在 | 请求中携带的地址组不存在 | 请检查请求中携带的地址组是否存在 |
| 更新地址组成员 | 400 | CFW.00400004 | 成员已存在 | 成员已存在 | 请删除一些地址组成员 |
| 添加服务组 | 400 | CFW.00200007 | 请求中携带的服务组名称与数据库中的名称有重复 | 请求中携带的服务组名称与数据库中的名称有重复 | 请删除请求中重复的服务组名称 |
| | 400 | CFW.00200024 | 添加服务组数量超过最大限制 | 添加服务组数量超过最大限制 | 删除已有服务组后再重新添加 |
| 添加服务组成员 | 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.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 |
| 更新服务组 | 400 | CFW.00200005 | 请求中携带的服务组不存在 | 请求中携带的服务组不存在 | 请检查请求中携带的服务组是否存在 |
| | 400 | CFW.00200007 | 请求中携带的地址组名称与数据库中的名称有重复 | 请求中携带的地址组名称与数据库中的名称有重复 | 请删除请求中重复的服务组名称 |
| 删除服务组 | 400 | CFW.00200004 | 删除的服务组被引用 | 删除的服务组被引用 | 请删除引用该服务组的规则后再删除该地址组 |
| | 400 | CFW.00200005 | 请求中携带的服务组不存在 | 请求中携带的服务组不存在 | 请检查请求中携带的服务组是否存在 |
| 更新服务组成员 | 400 | CFW.00400004 | 成员已存在 | 成员已存在 | 请删除一些服务组成员 |
| 添加黑白名单 | 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.00200016 | 实例状态错误 | 实例状态错误 | 请联系技术支持 |

| 模块 | 状态码 | 错误码 | 错误信息 | 描述 | 处理措施 |
|-----------|-----|--------------|-----------------------|-----------------------|-------------------|
| | 400 | CFW.00200032 | 引擎不支持IPv6 | 引擎不支持IPv6 | 请联系技术支持 |
| | 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 |
| 更新黑白名单 | 400 | CFW.00200005 | 请求中携带的黑白名单不存在 | 请求中携带的黑白名单不存在 | 请检查请求中携带的黑白名单是否存在 |
| | 400 | CFW.00200005 | 请求中携带的黑白名单不存在 | 请求中携带的黑白名单不存在 | 请检查请求中携带的黑白名单不存在 |
| | 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.00200032 | 引擎不支持IPv6 | 引擎不支持IPv6 | 请联系技术支持 |
| | 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.00200016 | 实例状态错误 | 实例状态错误 | 请联系技术支持 |
| | 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.00200028 | 地址类型不一致 | 地址类型不一致 | 请确保地址类型一致 |
| 删除黑白名单 | 400 | CFW.00200005 | 请求中携带的黑白名单不存在 | 请求中携带的黑白名单不存在 | 请检查请求中携带的黑白名单不存在 |
| 自定义IPS规则 | 400 | CFW.00200016 | 实例状态错误 | 实例状态错误 | 请联系技术支持 |
| 修改IPS开关状态 | 400 | CFW.00200023 | 调用后台修改IPS开关状态接口失败 | 调用后台修改IPS开关状态接口失败 | 请稍后重试或联系技术支持 |

| 模块 | 状态码 | 错误码 | 错误信息 | 描述 | 处理措施 |
|-----------|-----|--------------|----------|----------|---------|
| | 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 |
| 修改EIP防护状态 | 400 | CFW.00200016 | 实例状态错误 | 实例状态错误 | 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



A.4 获取企业项目 ID

本章节介绍如何通过控制台获取企业项目ID。

从控制台获取企业项目 ID

- 步骤1** [登录管理控制台](#)。
 - 步骤2** 单击页面右上方的“企业 > 项目管理”。
 - 步骤3** 在“名称/ID”列复制企业项目ID。
- 结束

A.5 获取防火墙 ID

防火墙id: fw_instance_Id, 是创建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“查询防火墙列表”获取。

- 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。

A.6 获取账号、IAM 用户、项目、用户组、区域、委托的名称和 ID

获取账号、IAM 用户、项目的名称和 ID

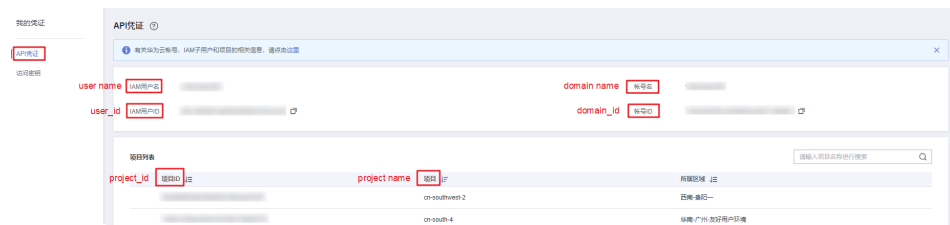
- 从控制台获取账号名、账号ID、用户名、用户ID、项目名称、项目ID
 - 在华为云首页右上角, 单击“控制台”。
 - 在右上角的用户名中选择“我的凭证”。

图 A-2 进入我的凭证



- c. 在“我的凭证”界面，API凭证页签中，查看账号名、账号ID、用户名、用户ID、项目名称、项目ID。
每个区域的项目ID有所不同，需要根据业务所在的区域获取对应的项目ID。

图 A-3 查看账号名、账号 ID、用户名、用户 ID、项目名称、项目 ID



- 调用API获取用户ID、项目ID
 - 获取用户ID请参考：[管理员查询IAM用户列表](#)。
 - 获取项目ID请参考：[查询指定条件下的项目列表](#)。

获取用户组名称和 ID

步骤1 登录华为云云服务平台，进入IAM控制台，选择“用户组”页签。

步骤2 单击需要查询的用户组前的下拉框，即可查询用户组名称、用户组ID。

图 A-4 查询用户组名称、用户组 ID



----结束

获取区域 ID

步骤1 登录华为云云服务平台，进入IAM控制台，选择“项目”页签。

步骤2 “项目”列的内容即为所属区域对应的ID。

图 A-5 查看区域 ID



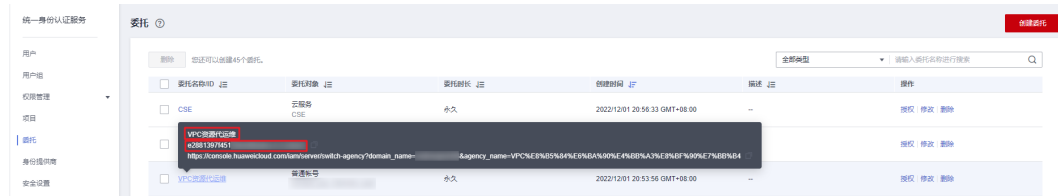
----结束

获取委托名称和 ID

步骤1 登录华为云云服务平台，进入IAM控制台，选择“委托”页签。

步骤2 鼠标移动到需要查询名称和ID的委托上，黑色框中出现的第一行为委托名称，第二行为委托ID。

图 A-6 查看委托 ID



----结束

A.7 地域信息表

大洲信息表

| 大洲 | 编码 |
|-----|----|
| 北美洲 | NA |
| 大洋洲 | OA |
| 非洲 | AF |
| 南极洲 | AN |
| 南美洲 | SA |

| 大洲 | 编码 |
|----|----|
| 欧洲 | EU |
| 亚洲 | AS |

国家/省份信息表

| 国家 | 编码 |
|-----------|----|
| 阿尔巴尼亚 | AL |
| 阿尔及利亚 | DZ |
| 阿富汗 | AF |
| 阿拉伯利比亚民众国 | LY |
| 阿拉伯联合酋长国 | AE |
| 阿鲁巴 | AW |
| 阿曼 | OM |
| 阿塞拜疆 | AZ |
| 埃及 | EG |
| 埃塞俄比亚 | ET |
| 爱尔兰 | IE |
| 爱沙尼亚 | EE |
| 安道尔 | AD |
| 安哥拉 | AO |
| 安圭拉 | AI |
| 安提瓜和巴布达 | AG |
| 奥地利 | AT |
| 奥兰群岛 | AX |
| 澳大利亚 | AU |
| 巴巴多斯 | BB |
| 巴布亚新几内亚 | PG |
| 巴哈马 | BS |
| 巴基斯坦 | PK |
| 巴拉圭 | PY |

| 国家 | 编码 |
|------------|----|
| 巴林 | BH |
| 巴西 | BR |
| 白俄罗斯 | BY |
| 百慕大 | BM |
| 保加利亚 | BG |
| 贝宁 | BJ |
| 比利时 | BE |
| 冰岛 | IS |
| 波兰 | PL |
| 波斯尼亚和黑山共和国 | BA |
| 博茨瓦纳 | BW |
| 不丹 | BT |
| 布基纳法索 | BF |
| 布隆迪 | BI |
| 朝鲜 | KP |
| 赤道几内亚 | GQ |
| 丹麦 | DK |
| 德国 | DE |
| 东帝汶 | TL |
| 多哥 | TG |
| 多米尼加 | DM |
| 多米尼加共和国 | DO |
| 俄罗斯 | RU |
| 厄立特里亚 | ER |
| 法国 | FR |
| 法罗群岛 | FO |
| 法属圭亚那 | GF |
| 法属南部领土 | TF |
| 菲律宾 | PH |
| 斐济 | FJ |

| 国家 | 编码 |
|--------|----|
| 芬兰 | FI |
| 佛得角 | CV |
| 福克兰群岛 | FK |
| 冈比亚 | GM |
| 刚果 | CG |
| 格恩西岛 | GG |
| 格陵兰 | GL |
| 格鲁吉亚 | GE |
| 圭亚那 | GY |
| 哈萨克斯坦 | KZ |
| 韩国 | KR |
| 荷兰 | NL |
| 黑山共和国 | ME |
| 吉布提 | DJ |
| 吉尔吉克斯坦 | KG |
| 几内亚 | GN |
| 几内亚比绍 | GW |
| 加纳 | GH |
| 加蓬 | GA |
| 柬埔寨 | KH |
| 捷克共和国 | CZ |
| 津巴布韦 | ZW |
| 喀麦隆 | CM |
| 卡塔尔 | QA |
| 科科斯群岛 | CC |
| 科摩罗 | KM |
| 科威特 | KW |
| 克罗地亚 | HR |
| 肯尼亚 | KE |
| 库克群岛 | CK |

| 国家 | 编码 |
|---------|----|
| 拉脱维亚 | LV |
| 莱索托 | LS |
| 老挝 | LA |
| 黎巴嫩 | LB |
| 立陶宛 | LT |
| 利比里亚 | LR |
| 列支敦士登 | LI |
| 留尼汪 | RE |
| 卢森堡 | LU |
| 卢旺达 | RW |
| 罗马尼亚 | RO |
| 马达加斯加 | MG |
| 马尔代夫 | MV |
| 马耳他 | MT |
| 马拉维 | MW |
| 马来西亚 | MY |
| 马里 | ML |
| 马绍尔群岛 | MH |
| 马提尼克群岛 | MQ |
| 马约特 | YT |
| 曼岛 | IM |
| 毛里求斯 | MU |
| 毛里塔尼亚 | MR |
| 蒙古 | MN |
| 孟加拉 | BD |
| 密克罗尼西亚 | FM |
| 缅甸 | MM |
| 摩尔多瓦共和国 | MD |
| 摩洛哥 | MA |
| 摩纳哥 | MC |

| 国家 | 编码 |
|--------------|----|
| 莫桑比克 | MZ |
| 纳米比亚 | NA |
| 南非 | ZA |
| 南乔治亚岛和南桑威齐群岛 | GS |
| 瑙鲁 | NR |
| 尼泊尔 | NP |
| 尼日尔 | NE |
| 尼日利亚 | NG |
| 挪威 | NO |
| 诺福克岛 | NF |
| 帕劳 | PW |
| 葡萄牙 | PT |
| 前南斯拉夫马其顿共和国 | MK |
| 日本 | JP |
| 瑞典 | SE |
| 瑞士 | CH |
| 塞拉利昂 | SL |
| 塞内加尔 | SN |
| 塞浦路斯 | CY |
| 塞舌尔群岛 | SC |
| 沙特阿拉伯 | SA |
| 圣诞岛 | CX |
| 圣多美和普林西比 | ST |
| 圣赫勒拿 | SH |
| 圣基茨和尼维斯 | KN |
| 圣卢西亚 | LC |
| 圣马力诺 | SM |
| 圣皮埃尔和密克隆 | PM |
| 圣座（梵蒂冈） | VA |
| 斯里兰卡 | LK |

| 国家 | 编码 |
|-----------|----|
| 斯洛伐克共和国 | SK |
| 斯洛文尼亚 | SI |
| 斯瓦尔巴特和扬马延 | SJ |
| 斯威士兰 | SZ |
| 苏里南 | SR |
| 所罗门群岛 | SB |
| 索马里 | SO |
| 塔吉克斯坦 | TJ |
| 泰国 | TH |
| 坦桑尼亚 | TZ |
| 特克斯和凯科斯群岛 | TC |
| 特立尼达和多巴哥 | TT |
| 突尼斯 | TN |
| 图瓦卢 | TV |
| 土耳其 | TR |
| 土库曼斯坦 | TM |
| 瓦努阿图 | VU |
| 文莱 | BN |
| 乌干达 | UG |
| 乌克兰 | UA |
| 乌拉圭 | UY |
| 乌兹别克斯坦 | UZ |
| 西班牙 | ES |
| 希腊 | GR |
| 象牙海岸 | CI |
| 新加坡 | SG |
| 新喀里多尼亚 | NC |
| 新西兰 | NZ |
| 匈牙利 | HU |
| 牙买加 | JM |

| 国家 | 编码 |
|---------|----|
| 亚美尼亚 | AM |
| 也门 | YE |
| 伊拉克 | IQ |
| 以色列 | IL |
| 意大利 | IT |
| 印度 | IN |
| 印度尼西亚 | ID |
| 英国 | GB |
| 英属维京群岛 | VG |
| 英属印度洋领地 | IO |
| 约旦 | JO |
| 越南 | VN |
| 赞比亚 | ZM |
| 泽西岛 | JE |
| 扎伊尔 | CD |
| 乍得 | TD |
| 直布罗陀 | GI |
| 中非共和国 | CF |
| 中国澳门 | MO |
| 中国大陆 | CN |
| 中国台湾 | TW |
| 中国香港 | HK |