

云防火墙

API 参考

文档版本	10
发布日期	2025-08-28



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.....	5
3.1 构造请求.....	5
3.2 认证鉴权.....	7
3.3 返回结果.....	9
4 API.....	11
4.1 防火墙管理.....	11
4.1.1 创建防火墙.....	11
4.1.2 查询防火墙列表.....	28
4.1.3 获取 CFW 任务执行状态.....	40
4.1.4 删除防火墙.....	46
4.1.5 查询防火墙详细信息.....	51
4.1.6 获取东西向防火墙信息.....	65
4.1.7 创建东西向防火墙.....	77
4.1.8 查询防护 VPC 数.....	84
4.1.9 标签创建接口.....	92
4.1.10 删除标签.....	98
4.1.11 更改防火墙名称.....	104
4.1.12 查询防火墙配额信息.....	108
4.1.13 查询南北向防火墙防护状态.....	114
4.2 EIP 管理.....	117
4.2.1 弹性 IP 开启关闭.....	118
4.2.2 查询 EIP 数量.....	127
4.2.3 弹性 IP 列表查询.....	132
4.2.4 一键逃生/一键恢复开关.....	140
4.2.5 添加 EIP 告警白名单.....	144
4.2.6 查看 EIP 告警白名单.....	148
4.2.7 修改 EIP 自动防护开关.....	153
4.2.8 获取 EIP 自动防护状态信息.....	159
4.3 访问控制规则管理.....	163
4.3.1 创建 ACL 规则.....	163

4.3.2 删除 ACL 规则.....	193
4.3.3 批量删除 Acl 规则.....	199
4.3.4 删除规则击中次数.....	206
4.3.5 更新 ACL 规则.....	213
4.3.6 批量更新规则动作.....	240
4.3.7 ACL 防护规则优先级设置.....	248
4.3.8 查询防护规则.....	255
4.3.9 查询规则标签.....	280
4.3.10 获取规则击中次数.....	287
4.3.11 查看 region 列表.....	294
4.3.12 查看导入状态接口.....	299
4.3.13 导入访问控制规则.....	305
4.3.14 导出访问控制规则.....	308
4.3.15 下载导入模板.....	311
4.3.16 下载导出结果.....	314
4.3.17 查询导出结果.....	316
4.3.18 下载导入规则.....	320
4.3.19 获取规则的命中次数和最后一次命中时间.....	323
4.4 黑白名单管理.....	327
4.4.1 批量添加黑白名单列表.....	327
4.4.2 批量删除黑白名单列表.....	336
4.4.3 创建黑白名单规则.....	340
4.4.4 更新黑白名单列表.....	346
4.4.5 删除黑白名单规则.....	352
4.4.6 查询黑白名单列表.....	357
4.5 地址组管理.....	363
4.5.1 添加地址组.....	363
4.5.2 添加地址组成员.....	369
4.5.3 删除地址组.....	375
4.5.4 批量删除地址组成员.....	380
4.5.5 更新地址组信息.....	385
4.5.6 查询地址组列表.....	390
4.5.7 查询地址组详细信息.....	397
4.5.8 查询地址组成员.....	402
4.5.9 删除地址组成员.....	408
4.5.10 批量删除地址组.....	413
4.5.11 更新对象配置描述.....	417
4.6 服务组管理.....	422
4.6.1 新建服务组.....	422
4.6.2 新建服务成员.....	427
4.6.3 删除服务组.....	434
4.6.4 批量删除服务组成员信息.....	439

4.6.5 修改服务组.....	444
4.6.6 获取服务组列表.....	449
4.6.7 查询服务组详情.....	455
4.6.8 查询服务成员列表.....	460
4.6.9 删除服务组成员.....	467
4.6.10 批量删除服务组.....	472
4.7 域名解析及域名组管理.....	476
4.7.1 添加域名组.....	476
4.7.2 删除域名组.....	485
4.7.3 更新域名组.....	490
4.7.4 更新 DNS 服务器列表.....	496
4.7.5 查询域名组列表.....	504
4.7.6 查询 DNS 服务器列表.....	513
4.7.7 查询域名解析 IP 地址.....	520
4.7.8 获取域名组下域名列表.....	526
4.7.9 添加域名列表.....	534
4.7.10 删除域名列表.....	543
4.7.11 查看域名组详情.....	550
4.7.12 获取域名地址解析结果.....	557
4.7.13 批量删除域名组.....	562
4.7.14 添加指定 DNS 服务器.....	570
4.7.15 获取域名解析结果.....	573
4.8 IPS 管理.....	576
4.8.1 查询 IPS 特性开关状态.....	576
4.8.2 IPS 特性开关操作.....	581
4.8.3 查询防护模式.....	586
4.8.4 切换防护模式.....	591
4.8.5 改变 IPS 规则模式.....	597
4.8.6 创建频率 IPS 规则.....	602
4.8.7 查询频率 IPS 规则信息.....	607
4.8.8 获取 IPS 规则列表.....	610
4.8.9 获取 IPS 规则细节.....	617
4.8.10 查看自定义 IPS 规则列表.....	621
4.8.11 创建自定义 IPS 规则.....	643
4.8.12 批量删除自定义 IPS 规则.....	657
4.8.13 批量更新自定义 IPS 规则的动作.....	660
4.8.14 查询自定义 IPS 规则详情.....	664
4.8.15 更新自定义 IPS 规则.....	680
4.9 日志管理.....	697
4.9.1 获取日志配置.....	698
4.9.2 创建日志配置.....	702
4.9.3 更新日志配置.....	708

4.9.4 查询防火墙日志.....	714
4.9.5 导出防火墙日志.....	730
4.9.6 查询流日志.....	738
4.9.7 查询访问控制日志.....	746
4.9.8 查询攻击日志.....	754
4.10 抓包管理.....	765
4.10.1 查询抓包任务.....	765
4.10.2 创建抓包任务.....	770
4.10.3 批量删除抓包任务.....	777
4.10.4 获取抓包任务结果.....	782
4.10.5 取消抓包任务.....	787
4.11 反病毒管理.....	791
4.11.1 查看反病毒开关.....	792
4.11.2 修改反病毒开关.....	796
4.11.3 获取防火墙反病毒规则信息.....	801
4.11.4 修改反病毒规则.....	806
4.12 告警配置管理.....	812
4.12.1 获取告警配置信息.....	812
4.12.2 修改告警配置接口.....	816
4.13 标签管理.....	822
4.13.1 查询标签信息.....	822
4.13.2 查询资源标签信息.....	827
4.13.3 保存资源标签接口.....	832
4.14 VPC 间防火墙管理.....	838
4.14.1 更新 VPC 间防火墙防护状态.....	838
4.14.2 查询 VPC 边界防火墙使用的引流 VPC 信息.....	843
4.14.3 查询 VPC 间防火墙使用的企业路由器信息.....	849
4.14.4 获取私网网段的信息.....	854
4.14.5 更新私网网段.....	859
4.14.6 创建私网网段.....	863
4.14.7 删除私网网段信息.....	867
4.15 日志分析.....	870
4.15.1 查询流量趋势.....	870
4.15.2 查询访问日志统计信息.....	877
4.15.3 查询攻击统计.....	887
4.15.4 查询攻击趋势.....	897
4.15.5 查询攻击概览.....	903
4.15.6 查询日志数量.....	910
4.15.7 查询流量日志统计详情.....	916
4.15.8 查询流量 TOP 统计.....	928
4.15.9 查询流量日志统计.....	936
4.15.10 查询会话趋势.....	946

4.15.11 查询攻击日志统计详情.....	953
4.15.12 查询攻击日志 TOP 统计.....	963
4.15.13 查询访问控制统计详情.....	972
4.16 多账号管理.....	984
4.16.1 查询账号列表.....	984
4.16.2 批量添加账号.....	991
4.16.3 批量移除账号.....	994
4.16.4 开启多账号管理.....	998
4.16.5 查询组织账号列表.....	1003
4.16.6 查询组织结构.....	1010
4.17 时间表管理.....	1017
4.17.1 查询时间表列表.....	1017
4.17.2 创建时间表.....	1024
4.17.3 更新时间表.....	1030
4.17.4 删除时间表.....	1037
4.17.5 批量删除时间表.....	1040
4.18 安全报告.....	1043
4.18.1 查询安全报告模板列表.....	1043
4.18.2 创建安全报告模板.....	1051
4.18.3 获取安全报告模板.....	1059
4.18.4 更新安全报告模板.....	1066
4.18.5 删除安全报告模板.....	1075
4.18.6 查询安全报告发送历史.....	1080
4.18.7 查询安全报告.....	1086
4.19 IPS 白名单管理.....	1102
4.19.1 更新 IPS 白名单.....	1102
4.19.2 查询 IPS 白名单列表详情.....	1107
4.19.3 批量添加 IPS 白名单.....	1112
4.19.4 批量删除 IPS 白名单.....	1117
4.20 TLS 证书管理.....	1119
4.20.1 导入 TLS 证书.....	1120
4.20.2 获取 TLS 证书.....	1123
4.21 流量过滤.....	1129
4.21.1 导入 IP 黑名单用于流量过滤.....	1129
4.21.2 获取导入的 IP 黑名单列表信息.....	1133
4.21.3 删除已经导入的 IP 黑名单.....	1138
4.21.4 导出用于流量过滤的 IP 黑名单.....	1142
4.21.5 用于流量过滤的 IP 黑名单导入失败后进行重新导入.....	1146
4.21.6 获取流量过滤功能的开关信息.....	1148
4.21.7 开启或者关闭流量过滤的 IP 黑名单功能.....	1150
A 附录.....	1154
A.1 状态码.....	1154

A.2 错误码.....	1155
A.3 获取项目 ID.....	1162
A.4 获取企业项目 ID.....	1163
A.5 获取防火墙 ID.....	1163
A.6 获取账号、IAM 用户、项目、用户组、区域、委托的名称和 ID.....	1163
A.7 地域信息表.....	1166

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](#)。

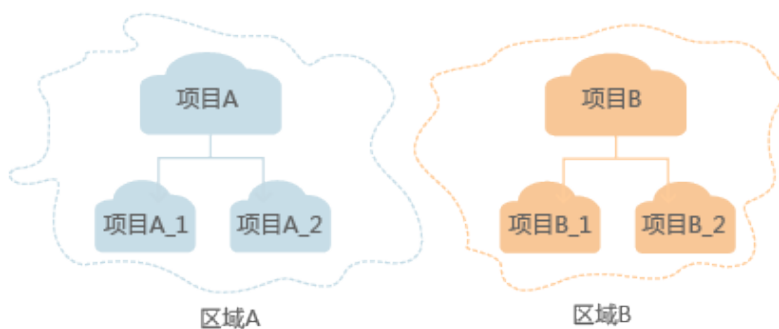
- 可用区（AZ，Availability Zone）

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。
访问控制规则管理	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	管理标签功能，包括查询资源标签信息等接口。
VPC间防火墙管理	管理VPC间防火墙接口，包括更新、查询、删除等接口。
日志分析	管理日志相关功能，包括查询攻击统计、查询日志数量、查询会话趋势等接口。
多账号管理	管理多账号，包括查询、添加、移除多账号等接口。

类型	说明
时间表管理	管理时间表，包括查询、创建、更新、删除时间表等接口。
安全报告	管理安全报告，包括查询、创建、更新、删除报告等接口。
IPS白名单管理	管理IPS白名单，包括更新、查询、批量添加、批量删除IPS白名单接口。
TLS证书管理	管理TLS证书，包括导入、获取证书接口。
流量过滤	流量封堵功能，包括开启、关闭、导入IP黑名单等接口。

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[管理员创建IAM用户](#)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-北京一”区域的Endpoint为“iam.cn-north-1.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。

例如您需要创建IAM用户，由于IAM为全局服务，则使用任一区域的Endpoint（比如“华北-北京四”区域的Endpoint：“iam.cn-north-4.myhuaweicloud.com”），并

在[管理员创建IAM用户](#)的URI部分找到resource-path (/v3.0/OS-USER/users)，拼接起来如下所示。

```
https://iam.cn-north-4.myhuaweicloud.com/v3.0/OS-USER/users
```

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[管理员创建IAM用户](#)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/v3.0/OS-USER/users
```

请求消息头

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

如下公共消息头需要添加到请求中。

- **Content-Type**：消息体的类型（格式），必选，默认取值为“application/json”，有其他取值时会在具体接口中专门说明。
- **Authorization**：签名认证信息，可选，当使用AK/SK方式认证时，使用SDK对请求进行签名的过程中会自动填充该字段。
- **X-Sdk-Date**：请求发送的时间，可选，当使用AK/SK方式认证时，使用SDK对请求进行签名的过程中会自动填充该字段。
- **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。

- **X-Project-ID**: 子项目ID, 可选, 在多项目场景中使用。如果云服务资源创建在子项目中, AK/SK认证方式下, 操作该资源的接口调用需要在请求消息头中携带X-Project-ID。
- **X-Domain-ID**: 账号ID, 可选。AK/SK认证方式下, 全局服务的接口调用时, 需在请求消息头中携带X-Domain-ID。

对于[管理员创建IAM用户](#)接口, 使用AK/SK方式认证时, 添加消息头后的请求如下所示。

```
POST https://iam.cn-north-4.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Sdk-Date: 20240416T095341Z
Authorization: SDK-HMAC-SHA256 Access=*****, SignedHeaders=content-type;host;x-sdk-date,
Signature=*****
```

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](#)。

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[管理员创建IAM用户](#)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。

- **accountid**为IAM用户所属的账号ID。
- **username**为要创建的IAM用户名。
- **email**为IAM用户的邮箱。
- *********为IAM用户的登录密码。

```
POST https://iam.cn-north-4.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Sdk-Date: 20240416T095341Z
Authorization: SDK-HMAC-SHA256 Access=*****, SignedHeaders=content-type;host;x-sdk-date,
Signature=*****
```

```
{
  "user": {
    "domain_id": "accountid",
    "name": "username",
    "password": "*****",
    "email": "email",
    "description": "IAM User Description"
  }
}
```

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。

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。
- 使用Token前请确保Token离过期有足够的时间，防止调用API的过程中Token过期导致调用API失败。

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“ABCDEFGH....”，则调用接口时将“X-Auth-Token: ABCDEFGH....”加到请求消息头即可，如下所示。

```
POST https://iam.cn-north-4.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Auth-Token: ABCDEFGH....
```

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既可以使用永久访问密钥中的AK/SK，也可以使用临时访问密钥中的AK/SK，但使用临时访问密钥的AK/SK时需要额外携带“X-Security-Token”字段，字段值为临时访问密钥的security_token。

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。

您也可以通过这个视频教程了解AK/SK认证的使用：<https://bbs.huaweicloud.com/videos/100697>。

3.3 返回结果

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

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[管理员创建IAM用户](#)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[管理员创建IAM用户](#)接口，返回如[图3-2](#)所示的消息头。

图 3-2 获取用户 Token 响应消息头

```
"X-Frame-Options": "SAMEORIGIN",
"X-IAM-ETag-id": "2562365939-d8f6f12921974cb097338ac11fceac8a",
"Transfer-Encoding": "chunked",
"Strict-Transport-Security": "max-age=31536000; includeSubdomains;",
"Server": "api-gateway",
"X-Request-Id": "af2953f2bcc67a42325a69a19e6c32a2",
"X-Content-Type-Options": "nosniff",
"Connection": "keep-alive",
"X-Download-Options": "noopen",
"X-XSS-Protection": "1; mode=block;",
"X-IAM-Trace-Id": "token_ _null_af2953f2bcc67a42325a69a19e6c32a2",
"Date": "Tue, 21 May 2024 09:03:40 GMT",
"Content-Type": "application/json; charset=utf8"
```

响应消息体

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[管理员创建IAM用户](#)接口，返回如下消息体。为篇幅起见，这里只展示部分内容。

```
{
  "user": {
    "id": "c131886aec...",
    "name": "IAMUser",
    "description": "IAM User Description",
    "areacode": "",
    "phone": "",
    "email": "****@***.com",
    "status": null,
    "enabled": true,
    "pwd_status": false,
    "access_mode": "default",
    "is_domain_owner": false,
    "xuser_id": "",
    "xuser_type": "",
    "password_expires_at": null,
    "create_time": "2024-05-21T09:03:41.000000",
    "domain_id": "d78cbac1.....",
    "xdomain_id": "30086000.....",
    "xdomain_type": "",
    "default_project_id": null
  }
}
```

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

```
{
  "error_msg": "Request body is invalid.",
  "error_code": "IAM.0011"
}
```

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	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-2 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	参数解释： 防火墙名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
enterprise_project_id	否	String	参数解释： 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 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0

参数	是否必选	参数类型	描述
tags	否	Array of tags objects	参数解释： 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 约束限制： 不涉及
flavor	是	flavor object	参数解释： 防火墙规格信息 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
charge_info	是	charge_info object	参数解释： 计费类型信息，支持包年/包月和按需，默认为按需。 约束限制： 不涉及

表 4-3 tags

参数	是否必选	参数类型	描述
key	否	String	参数解释： 资源标签键 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
value	否	String	参数解释： 资源标签值 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-4 flavor

参数	是否必选	参数类型	描述
version	是	String	参数解释： 防火墙版本 约束限制： “charge_mode” 为 “prePaid” 时，支持标准版、 专业版。 “charge_mode” 为 “postPaid” 时，仅支持专业 版。 取值范围： Standard - 标准版 Professional - 专业版 默认取值： 不涉及
extend_eip_co unt	否	Integer	参数解释： 扩展EIP数量 约束限制： 仅包周期场景下生效，当用户需 要增加EIP防护数量时使用此参 数。 取值范围： 0-2000 默认取值： 0

参数	是否必选	参数类型	描述
extend_bandwidth	否	Integer	参数解释： 扩展带宽，步长为5 约束限制： 仅包周期场景下生效，当用户需要防护带宽时使用此参数。 取值范围： 0-5000 默认取值： 不涉及
extend_vpc_count	否	Integer	参数解释： 扩展VPC数量 约束限制： 仅包周期场景下生效，当用户需要增加VPC防护数量时使用此参数。 取值范围： 0-100 默认取值： 不涉及

表 4-5 charge_info

参数	是否必选	参数类型	描述
charge_mode	是	String	参数解释： 计费模式 约束限制： 不涉及 取值范围： • prePaid：预付费，即包年/包月。 • postPaid：后付费，即按需付费 默认取值： 不涉及

参数	是否必选	参数类型	描述
period_type	否	String	参数解释： 订购周期类型 约束限制： “charge_mode”为 “prePaid”时生效，且为必选 值。 取值范围： • month：包月。 • year：包年 默认取值： 不涉及
period_num	否	Integer	参数解释： 订购时间 约束限制： “charge_mode”为 “prePaid”时， “period_num”必填。 取值范围： • 当“period_type”为 “month”时，此处取值为 1~9。 • 当“period_type”为 “year”时，此处取值为 1~3。 默认取值： 不涉及
is_auto_renew	是	Boolean	参数解释： 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。 约束限制： 创建包周期实例时可指定 取值范围： • true，为自动续订。 • false，为不自动续订 默认取值： false

参数	是否必选	参数类型	描述
is_auto_pay	是	Boolean	参数解释： 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。 约束限制： 创建包周期时可指定 取值范围： • true: 自动支付（会自动选择折扣和优惠券进行优惠，并自动从客户账户中支付），自动支付失败会生成订单、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: 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） 默认取值： false

响应参数

状态码：200

表 4-6 响应 Body 参数

参数	参数类型	描述
job_id	String	参数解释： 实例创建的任务id 取值范围： 不涉及
order_id	String	参数解释： 订单号 取值范围： 不涉及

参数	参数类型	描述
data	CreateFirewallReq object	参数解释： 创建防火墙请求体 取值范围： 不涉及

表 4-7 CreateFirewallReq

参数	参数类型	描述
name	String	参数解释： 防火墙名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
enterprise_project_id	String	参数解释： 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 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0
tags	Array of tags objects	参数解释： 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 约束限制： 不涉及

参数	参数类型	描述
flavor	flavor object	参数解释： 防火墙规格信息 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
charge_info	charge_info object	参数解释： 计费类型信息，支持包年/包月和按需，默认为按需。 约束限制： 不涉及

表 4-8 tags

参数	参数类型	描述
key	String	参数解释： 资源标签键 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
value	String	参数解释： 资源标签值 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-9 flavor

参数	参数类型	描述
version	String	参数解释： 防火墙版本 约束限制： “charge_mode”为“prePaid”时，支持标准版、专业版。 “charge_mode”为“postPaid”时，仅支持专业版。 取值范围： Standard - 标准版 Professional - 专业版 默认取值： 不涉及
extend_eip_count	Integer	参数解释： 扩展EIP数量 约束限制： 仅包周期场景下生效，当用户需要增加EIP防护数量时使用此参数。 取值范围： 0-2000 默认取值： 0
extend_bandwidth	Integer	参数解释： 扩展带宽，步长为5 约束限制： 仅包周期场景下生效，当用户需要防护带宽时使用此参数。 取值范围： 0-5000 默认取值： 不涉及

参数	参数类型	描述
extend_vpc_count	Integer	参数解释： 扩展VPC数量 约束限制： 仅包周期场景下生效，当用户需要增加VPC防护数量时使用此参数。 取值范围： 0-100 默认取值： 不涉及

表 4-10 charge_info

参数	参数类型	描述
charge_mode	String	参数解释： 计费模式 约束限制： 不涉及 取值范围： - prePaid：预付费，即包年/包月。 - postPaid：后付费，即按需付费 默认取值： 不涉及
period_type	String	参数解释： 订购周期类型 约束限制： “charge_mode”为“prePaid”时生效，且为必选值。 取值范围： - month：包月。 - year：包年 默认取值： 不涉及

参数	参数类型	描述
period_num	Integer	参数解释： 订购时间 约束限制： “charge_mode”为“prePaid”时，“period_num”必填。 取值范围： - 当“period_type”为“month”时，此处取值为1~9。 - 当“period_type”为“year”时，此处取值为1~3。 默认取值： 不涉及
is_auto_renew	Boolean	参数解释： 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。 约束限制： 创建包周期实例时可指定 取值范围： - true，为自动续订。 - false，为不自动续订 默认取值： false
is_auto_pay	Boolean	参数解释： 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。 约束限制： 创建包周期时可指定 取值范围： - true：自动支付（会自动选择折扣和优惠券进行优惠，并自动从客户账户中支付），自动支付失败会生成订单、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：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） 默认取值： false

状态码：400

表 4-11 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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

OK，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	OK，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 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 4-12 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-13 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

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	参数解释： 用户Token。可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-15 请求 Body 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
key_word	否	String	参数解释： 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 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
tags	否	Array of TagInfo objects	参数解释： 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 约束限制： 不涉及
limit	是	Integer	参数解释： 每页显示个数 约束限制： 不涉及 取值范围： 1-1024 默认取值： 不涉及

参数	是否必选	参数类型	描述
offset	是	Integer	参数解释： 偏移量：指定返回记录的开始位置 约束限制： 不涉及 取值范围： 大于或等于0 默认取值： 0

表 4-16 TagInfo

参数	是否必选	参数类型	描述
key	否	String	参数解释： 标签键 约束限制： 不涉及
values	否	Array of strings	参数解释： 标签值列表 约束限制： 不涉及

响应参数

状态码：200

表 4-17 响应 Body 参数

参数	参数类型	描述
user_support_eps	Boolean	参数解释： 是否支持企业项目 取值范围： - true：是 - false：否

参数	参数类型	描述
has_ndr	Boolean	参数解释： 是否存在NDR，NDR为原旁路版防火墙，现已停止售卖。 取值范围： • true：是 • false：不是
is_support_postpaid	Boolean	参数解释： 是否支持按需购买 取值范围： • true：是 • false：不是
is_support_basic_version	Boolean	参数解释： 是否支持基础版 取值范围： • true：是 • false：不是
is_support_buy_professional	Boolean	参数解释： 是否支持购买专业版 取值范围： • true：是 • false：不是
data	HttpFirewallInstanceListResponseData object	参数解释： 查询防火墙列表返回值data信息 取值范围： 不涉及

表 4-18 HttpFirewallInstanceListResponseData

参数	参数类型	描述
limit	Integer	参数解释： 每页显示个数 约束限制： 不涉及 取值范围： 1-1024

参数	参数类型	描述
offset	Integer	参数解释： 偏移量：指定返回记录的开始位置 约束限制： 不涉及 取值范围： 大于或等于0
project_id	String	参数解释： 租户项目ID 约束限制： 不涉及
total	Integer	参数解释： 防火墙总数量 约束限制： 不涉及
records	Array of FirewallInstanceVO objects	参数解释： 查询防火墙列表记录 约束限制： 不涉及

表 4-19 FirewallInstanceVO

参数	参数类型	描述
fw_instance_id	String	参数解释： 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。 约束限制： 不涉及
resource_id	String	参数解释： 资源id，与防火墙实例id fw_instance_id 相同。 约束限制： 不涉及
name	String	参数解释： 创建防火墙时的时间戳 约束限制： 不涉及

参数	参数类型	描述
fw_instance_name	String	参数解释： 防火墙名称 约束限制： 不涉及
enterprise_project_id	String	参数解释： 企业项目id，用户支持企业项目后，由企业项目生成的id。 约束限制： 不涉及
ha_type	Integer	参数解释： 集群类型 约束限制： 不涉及 取值范围： 0：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 1：集群模式，仅拉起两个节点作为集群
charge_mode	Integer	参数解释： 计费模式 约束限制： 不涉及 取值范围： 0：包年/包月 1：按需 默认取值： 不涉及
service_type	Integer	参数解释： 防火墙防护类型 约束限制： 不涉及 取值范围： 目前仅支持0，互联网防护

参数	参数类型	描述
engine_type	Integer	参数解释： 引擎类型 约束限制： 不涉及 取值范围： 0：自研引擎 1：山石引擎 3：天融信引擎
flavor	Flavor object	参数解释： 防火墙规格信息 约束限制： 不涉及
status	Integer	参数解释： 防火墙状态列表 约束限制： 不涉及 取值范围： -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	参数解释： 标签列表，标签键值map转化的json字符串，如{"key":"value"} 约束限制： 不涉及

表 4-20 Flavor

参数	参数类型	描述
version	Integer	参数解释： 防火墙版本 取值范围： 0：标准版 1：专业版 3：基础版，
eip_count	Integer	参数解释： EIP数量 取值范围： 不涉及
vpc_count	Integer	参数解释： VPC数量 取值范围： 不涉及
bandwidth	Integer	参数解释： 带宽，单位为mbps 取值范围： 不涉及
log_storage	Integer	参数解释： 日志存储，单位为byte 取值范围： 不涉及
default_bandwidth	Integer	参数解释： 默认防火墙带宽，单位为mbps 取值范围： 包周期标准版为10，专业版为50，按需专业版为200
default_eip_count	Integer	参数解释： 默认eip数 取值范围： 包周期标准版为20，专业版为50，按需专业版为1000
default_log_storage	Integer	参数解释： 默认日志存储，单位为byte 取值范围： 不涉及

参数	参数类型	描述
default_vpc_count	Integer	参数解释： 默认vpc数 约束限制： 包周期标准版为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

OK，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	OK，查询防火墙列表返回值

错误码

请参见[错误码](#)。

4.1.3 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 4-21 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
job_id	是	String	参数解释： 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 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

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	参数解释： 用户Token，用于携带用户身份信息，配置后可通过接口鉴权，可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及。

响应参数

状态码：200

表 4-23 响应 Body 参数

参数	参数类型	描述
data	GetCreateFirewallJobResponseData object	参数解释： 创建按需防火墙返回数据。 取值范围： 不涉及

表 4-24 GetCreateFirewallJobResponseData

参数	参数类型	描述
id	String	参数解释： 创建按需防火墙任务ID 取值范围： 不涉及

参数	参数类型	描述
status	String	参数解释： 任务执行状态，用于展示创建防火墙是否成功。 取值范围： - Running：表示任务正在执行。 - Success：表示任务执行成功。 - Failed：表示任务执行失败。
begin_time	String	参数解释： 创建时间 取值范围： 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。 例如： 2023-10-26T10:30:00Z 表示：2023年10月26日10点30分00秒，UTC时间 2023-10-26T10:30:00+0800 表示：2023年10月26日10点30分00秒，比UTC时间快8小时，也就是北京时间。
end_time	String	参数解释： 结束时间 取值范围： 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。 例如： 2023-10-26T10:30:00Z 表示：2023年10月26日10点30分00秒，UTC时间 2023-10-26T10:30:00+0800 表示：2023年10月26日10点30分00秒，比UTC时间快8小时，也就是北京时间。

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

OK，获取创建按需防火墙任务情况接口返回值

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

SDK 代码示例

SDK代码示例如下。

Java

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

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

public class ListJobSolution {

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

4.1.4 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 4-25 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
resource_id	是	String	参数解释： 防火墙ID，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-26 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token，用于携带用户身份信息，配置后可通过接口鉴权，可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及。

响应参数

状态码：200

表 4-27 响应 Body 参数

参数	参数类型	描述
data	String	参数解释： 删除防火墙时生成的任务的job_id 约束限制： 不涉及

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

OK，删除防火墙返回值

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

SDK 代码示例

SDK代码示例如下。

Java

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

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

public class DeleteFirewallSolution {

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



```
String sk = System.getenv("CLOUD_SDK_SK");
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.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 = 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	OK，删除防火墙返回值

错误码

请参见[错误码](#)。

4.1.5 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 4-28 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-29 Query 参数

参数	是否必选	参数类型	描述
offset	是	Integer	参数解释： 偏移量：指定返回记录的开始位置 约束限制： 必须为数字 取值范围： 大于或等于0 默认取值： 0

参数	是否必选	参数类型	描述
limit	是	Integer	参数解释： 每页显示个数，范围为1-1024 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
service_type	是	Integer	参数解释： 服务类型 约束限制： 目前仅支持0互联网防护 取值范围： 0 默认取值： 0
enterprise_project_id	否	String	参数解释： 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 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0：用户未开启企业项目

参数	是否必选	参数类型	描述
fw_instance_id	否	String	参数解释： 防火墙实例id，创建云防火墙后用于标志防火墙由系统自动生成的标志id，可通过调用 查询防火墙实例接口 获取 约束限制： 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 取值范围： 不涉及 默认取值： 不涉及
name	否	String	参数解释： 防火墙名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

请求参数

表 4-30 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token。可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-31 响应 Body 参数

参数	参数类型	描述
data	GetFirewallInstanceData object	参数解释： 查询防火墙实例数据 取值范围： 不涉及

表 4-32 GetFirewallInstanceData

参数	参数类型	描述
limit	Integer	参数解释： 每页显示个数 取值范围： 1-1024
offset	Integer	参数解释： 偏移量：指定返回记录的开始位 取值范围： 大于或等于0
total	Integer	参数解释： 防火墙总数 取值范围： 不涉及
records	Array of GetFirewallInstanceResponseRecord objects	参数解释： 查询防火墙实例信息列表 取值范围： 不涉及

表 4-33 GetFirewallInstanceResponseRecord

参数	参数类型	描述
fw_instance_id	String	参数解释： 防火墙实例id，创建云防火墙后用于标志 防火墙由系统自动生成的标志id。可通过 调用 查询防火墙实例接口 获取 取值范围： 不涉及
name	String	参数解释： 防火墙名称 取值范围： 不涉及
ha_type	Integer	参数解释： 集群类型 取值范围： 0：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 1：集群模式，仅拉起两个节点作为集群
charge_mode	Integer	参数解释： 计费模式 取值范围： 0：包年/包月 1：按需
service_type	Integer	参数解释： 防火墙防护类型 取值范围： 目前仅支持0，互联网防护
engine_type	Integer	参数解释： 引擎类型 取值范围： 0：自研引擎 1：山石引擎 3：天融信引擎
flavor	Flavor object	参数解释： 防火墙规格信息 取值范围： 不涉及

参数	参数类型	描述
protect_objects	Array of ProtectObjectVO objects	参数解释： 防护对象列表 取值范围： 不涉及
status	Integer	参数解释： 防火墙状态列表 取值范围： • -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	参数解释： 是否为旧引擎 取值范围： • true：是 • false：不是
is_available_obs	Boolean	参数解释： 是否支持obs 取值范围： • true：是 • false：不是
is_support_threat_tags	Boolean	参数解释： 是否支持威胁情报标签 取值范围： • true：是 • false：不是

参数	参数类型	描述
support_ipv6	Boolean	参数解释： 是否支持ipv6 取值范围： • true：是 • false：不是
feature_toggle	Map<String,Boolean>	参数解释： 特性开关 取值范围： • true：是 • false：不是
resources	Array of FirewallInstanceResource objects	参数解释： 防火墙资源列表 取值范围： 不涉及
fw_instance_name	String	参数解释： 防火墙名称 取值范围： 不涉及
enterprise_project_id	String	参数解释： 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 取值范围： 不涉及
resource_id	String	参数解释： 防火墙资源id，同fw_instance_id 取值范围： 不涉及
support_url_filtering	Boolean	参数解释： 是否支持url过滤 取值范围： • true：是 • false：不是

参数	参数类型	描述
tags	String	参数解释： 标签列表，标签键值map转化的json字符串，如{"key":"value"}" 取值范围： 不涉及

表 4-34 Flavor

参数	参数类型	描述
version	Integer	参数解释： 防火墙版本 取值范围： 0：标准版 1：专业版 3：基础版，
eip_count	Integer	参数解释： EIP数量 取值范围： 不涉及
vpc_count	Integer	参数解释： VPC数量 取值范围： 不涉及
bandwidth	Integer	参数解释： 带宽，单位为mbps 取值范围： 不涉及
log_storage	Integer	参数解释： 日志存储，单位为byte 取值范围： 不涉及
default_bandwidth	Integer	参数解释： 默认防火墙带宽，单位为mbps 取值范围： 包周期标准版为10，专业版为50，按需专业版为200

参数	参数类型	描述
default_eip_count	Integer	参数解释： 默认eip数 取值范围： 包周期标准版为20，专业版为50，按需专业版为1000
default_log_storage	Integer	参数解释： 默认日志存储，单位为byte 取值范围： 不涉及
default_vpc_count	Integer	参数解释： 默认vpc数 约束限制： 包周期标准版为0，专业版为2，按需专业版为5

表 4-35 ProtectObjectVO

参数	参数类型	描述
object_id	String	参数解释： 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。 取值范围： 不涉及
object_name	String	参数解释： 防护对象名称 取值范围： 不涉及
type	Integer	参数解释： 防护对象类型 取值范围： 0：南北向 1：东西向

表 4-36 FirewallInstanceResource

参数	参数类型	描述
resource_id	String	参数解释： 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。 取值范围： 不涉及
cloud_service_type	String	参数解释： 服务类型，用于CBC使用，特指： hws.service.type.cfw 取值范围： 不涉及
resource_type	String	参数解释： 资源类型 取值范围： 1：云防火墙:hws.resource.type.cfw 2：EIP:hws.resource.type.cfw.exp.eip 3：带宽:hws.resource.type.cfw.exp.bandwidth 4：VPC:hws.resource.type.cfw.exp
resource_spec_code	String	参数解释： 库存单位码 取值范围： 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

参数	参数类型	描述
resource_size	Integer	参数解释： 资源数量 取值范围： 不涉及
resource_size_measure_id	Integer	参数解释： 资源单位 取值范围： 不涉及

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

OK，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"
}],
"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	OK，获取存在防火墙实例响应
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
500	Internal Server Error

错误码

请参见[错误码](#)。

4.1.6 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 4-37 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-38 Query 参数

参数	是否必选	参数类型	描述
limit	是	Integer	参数解释： 每页显示个数 约束限制： 不涉及 取值范围： 1-1024 默认取值： 不涉及

参数	是否必选	参数类型	描述
offset	是	Integer	参数解释： 偏移量：指定返回记录的开始位置 约束限制： 必须为数字 取值范围： 大于或等于0 默认取值： 默认0
enterprise_project_id	否	String	参数解释： 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 约束限制： 用户未开启企业项目时为0 取值范围： 不涉及 默认取值： 不涉及
fw_instance_id	是	String	参数解释： 防火墙ID，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-39 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token。可通过 如何获取用户Token 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-40 响应 Body 参数

参数	参数类型	描述
data	GetEastWestFirewallResponseBody object	参数解释： 获取东西向防火墙列表返回data 约束限制： 不涉及

表 4-41 GetEastWestFirewallResponseBody

参数	参数类型	描述
object_id	String	参数解释： 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。 type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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。 取值范围： 不涉及
project_id	String	参数解释： 项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 取值范围： 不涉及
status	Integer	参数解释： 防护状态： 不涉及 取值范围： 0：已开启防护 1：未开启防护
firewall_associated_subnets	Array of SubnetInfo objects	参数解释： 云防火墙关联子网信息 取值范围： 不涉及
router	EulerInstance object	参数解释： 出方向关联企业路由器信息 取值范围： 不涉及
inspection_vpc	VpcDetail object	参数解释： 引流vpc信息 取值范围： 不涉及

参数	参数类型	描述
protect_infos	Array of EwProtectResourceInfo objects	参数解释： 东西向防护资源信息 取值范围： 不涉及
total	Integer	参数解释： 防护VPC总数 取值范围： 不涉及
offset	Integer	参数解释： 偏移量：指定返回记录的开始位置 取值范围： 大于或等于0
limit	Integer	参数解释： 每页显示个数 取值范围： 1-1024
mode	String	参数解释： 防护模式，值为er 取值范围： 不涉及

表 4-42 SubnetInfo

参数	参数类型	描述
availability_zone	String	参数解释： 子网所在的可用区标识，从终端节点获取，参考 终端节点 取值范围： 不涉及
cidr	String	参数解释： 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

参数	参数类型	描述
name	String	参数解释： 子网名称 取值范围： 不涉及
id	String	参数解释： 子网id 取值范围： 不涉及
gateway_ip	String	参数解释： 子网的网关 取值范围： 子网网段cidr中的IP地址
vpc_id	String	参数解释： 创建VPC产生的uuid 取值范围： 不涉及
ipv6_enable	Boolean	参数解释： 是否支持ipv6 取值范围： • true： 是 • false： 否

表 4-43 ErlInstance

参数	参数类型	描述
id	String	参数解释： ER ID，创建ER时产生的ID 取值范围： 不涉及
name	String	参数解释： ER名称 取值范围： 不涉及

参数	参数类型	描述
project_id	String	参数解释： 项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 取值范围： 不涉及
attachment_id	String	参数解释： 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。 取值范围： 不涉及

表 4-44 VpcDetail

参数	参数类型	描述
id	String	参数解释： 创建引流VPC产生的随机UUID 取值范围： 不涉及
name	String	参数解释： 引流VPC名称 取值范围： 不涉及
cidr	String	参数解释： 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

表 4-45 EwProtectResourceInfo

参数	参数类型	描述
protected_resource_type	Integer	参数解释： 防护资源类 取值范围： 0：VPC 1：VGW 2：VPN 3：PEERING
protected_resource_name	String	参数解释： 防护资源名称 取值范围： 不涉及
protected_resource_id	String	参数解释： 防护资源id 取值范围： 不涉及
protected_resource_nat_name	String	参数解释： 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。 取值范围： 不涉及
protected_resource_nat_id	String	参数解释： 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。 取值范围： 不涉及
protected_resource_project_id	String	参数解释： 防火墙支持跨账户防护，此处为防护资源租户id 取值范围： 不涉及
protected_resource_mode	String	参数解释： 防护资源模式，为er 取值范围： 不涉及

参数	参数类型	描述
status	Integer	参数解释： 防护资源的防护状态 取值范围： 0：已关联 1：未关联

状态码：500

表 4-46 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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

OK，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-aefa-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" : "inspt-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.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 = 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	OK，东西向防火墙信息响应
400	Bad Request，错误的请求

状态码	描述
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
500	Internal Server Error，内部服务错误

错误码

请参见[错误码](#)。

4.1.7 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 4-47 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-48 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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 约束限制： 用户未开启企业项目时为0 取值范围： 不涉及 默认取值： 不涉及
fw_instance_id	是	String	参数解释： 防火墙ID，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

请求参数

表 4-49 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token。可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-50 请求 Body 参数

参数	是否必选	参数类型	描述
er_id	是	String	参数解释： 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） 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
inspection_cidr	是	String	参数解释： 创建引流VPC时使用的网段 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
mode	是	String	参数解释： 东西向防火墙模式，填写er 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-51 响应 Body 参数

参数	参数类型	描述
data	CreateEWFirewallResp object	参数解释： 创建东西向防火墙返回值 取值范围： 不涉及

表 4-52 CreateEWFirewallResp

参数	参数类型	描述
id	String	参数解释： 东西向防护id，对应object_id字段 取值范围： 不涉及
er	ER object	参数解释： ER信息 取值范围： 不涉及
inspersion_vpc	CreateEWFirewallInspectVpcResp object	参数解释： 引流vpc信息 取值范围： 不涉及

表 4-53 ER

参数	参数类型	描述
er_id	String	参数解释： ER ID，创建东西向防护引用的 ID 取值范围： 不涉及
er_attach_id	String	参数解释： 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。 取值范围： 不涉及

表 4-54 CreateEWFirewallInspectVpcResp

参数	参数类型	描述
vpc_id	String	参数解释： 引流VPC的ID 取值范围： 不涉及
subnet_ids	Array of strings	参数解释： 创建的引流VPC的子网ID列表 取值范围： 不涉及

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

OK，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.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

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

```
# coding: utf-8

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",
        EriId: "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	OK，创建东西向防火墙返回值

错误码

请参见[错误码](#)。

4.1.8 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 4-55 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-56 Query 参数

参数	是否必选	参数类型	描述
object_id	是	String	参数解释： 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 约束限制： type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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。 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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 约束限制： 用户未开启企业项目时为0 取值范围： 不涉及 默认取值： 不涉及
fw_instance_id	否	String	参数解释： 防火墙ID，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

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	参数解释： 用户Token。可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-58 响应 Body 参数

参数	参数类型	描述
data	VPCProtectsVo object	参数解释： 防护VPC返回值 取值范围： 不涉及

表 4-59 VPCProtectsVo

参数	参数类型	描述
total	Integer	参数解释： 总防护VPC数 取值范围： 不涉及
self_total	Integer	参数解释： 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。 取值范围： 不涉及
other_total	Integer	参数解释： 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 取值范围： 不涉及
protect_vpcs	Array of VpcAttachmentDetail objects	参数解释： 防火墙东西向防护可进行跨账号防护VPC，protect_vpcs表示总体防护VPC列表 取值范围： 不涉及
self_protect_vpcs	Array of VpcAttachmentDetail objects	参数解释： 防火墙东西向防护可进行跨账号防护VPC，self_protect_vpcs表示本项目防护VPC列表 取值范围： 不涉及

参数	参数类型	描述
other_protect_vpcs	Array of VpcAttachmentDetail objects	参数解释： 防火墙东西向防护可进行跨账号防护VPC，other_protect_vpcs表示其他项目防护VPC列表 取值范围： 不涉及
total_assets	Integer	参数解释： 租户的所有VPC资产数量 取值范围： 不涉及

表 4-60 VpcAttachmentDetail

参数	参数类型	描述
vpc_id	String	参数解释： 东西向防护添加的防护VPC的ID 取值范围： 不涉及

状态码：500

表 4-61 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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

OK，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，错误的请求

```
{
  "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.valueOf("<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	OK，查询东西向防护返回值
400	Bad Request，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
500	Internal Server Error

错误码

请参见[错误码](#)。

4.1.9 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 4-62 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
fw_instance_id	是	String	参数解释： 防火墙ID，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

请求参数

表 4-63 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token。可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-64 请求 Body 参数

参数	是否必选	参数类型	描述
tags	否	Array of CreateTag objects	参数解释： 创建防火墙标签列表 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-65 CreateTag

参数	是否必选	参数类型	描述
key	否	String	参数解释： 标签键 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
value	否	String	参数解释： 标签值 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-66 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.getStatusCode());
            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](#)。

4.1.10 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 4-67 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
fw_instance_id	是	String	参数解释： 防火墙ID，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

请求参数

表 4-68 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token。可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-69 请求 Body 参数

参数	是否必选	参数类型	描述
tags	否	Array of ResourceTag objects	参数解释： 防火墙标签列表 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-70 ResourceTag

参数	是否必选	参数类型	描述
key	否	String	参数解释： 标签键 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
value	否	String	参数解释： 标签值 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
update_time	否	String	参数解释： 标签更新时间 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-71 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();
        }
    }
}
```

```
        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](#)。

4.1.11 更改防火墙名称

功能介绍

更改防火墙名称

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/firewall/name

表 4-72 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-73 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-74 请求 Body 参数

参数	是否必选	参数类型	描述
fw_instance_name	是	String	参数解释： 防火墙名称 约束限制： 防火墙名称为中英文，包含数字，下划线，中划线，长度为1-48 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-75 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 响应信息 取值范围： 不涉及

表 4-76 data

参数	参数类型	描述
id	String	参数解释： 防火墙ID，可通过防火墙ID获取 方式获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
name	String	参数解释： 防火墙名称 约束限制： 防火墙名称为中英文，包含数字，下划线，中划线，长度为1-48 取值范围： 不涉及 默认取值： 不涉及

状态码：400

表 4-77 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及

参数	参数类型	描述
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

修改项目ID为eb7accd78ea845078275b2ad5280a109，防火墙ID为8a996d43-75bd-48b4-b1b8-de0706b72b8f的防火墙名称

```
https://{Endpoint}/v1/5de88e3e18124fa99675fb51ceceaa31/firewall/name?fw_instance_id=d2736642-009e-47e6-b66e-cc4355a9f8c8

{
  "fw_instance_name": "CFW"
}
```

响应示例

状态码：200

OK

```
{
  "data": {
    "id": "d2736642-009e-47e6-b66e-cc4355a9f8c8",
    "name": "CFW"
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.1.12 查询防火墙配额信息

功能介绍

查询防火墙配额信息

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/cfw/{fw_instance_id}/quota

表 4-78 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-79 Query 参数

参数	是否必选	参数类型	描述
config_type	是	String	参数解释： 防火墙配额类型 约束限制： 不涉及 取值范围： ACL：ACL规则配额 DNS_DOMAIN_SET：网络域名组配额 DOMAIN：域名组域名成员配额 DOMAIN_DEVICE：域名设备配额 DNS_DOMAIN_SET_PARSE_IP：网络域名组解析IP配额 APPLICATION_DOMAIN_SET：应用域名组配额 APPLICATION_DOMAIN_SET_ITEM：应用域名组域名成员配额 APPLICATION_DOMAIN_SET_ITEM_DEVICE：应用域名组设备配额 ADDR_SET：地址组配额 ADDR_SET_ITEM：地址组IP地址成员配额 ADDR_SET_ITEM_DEVICE：地址组IP地址设备配额 SERV_SET：服务组配额 SERV_SET_ITEM：服务组服务成员配额 SERV_SET_ITEM_DEVICE：服务组服务设备配额 BLACKLIST：黑名单配额 WHITELIST：白名单配额 PRIVATE_NETWORK_SEGMENT：私网网段配额 默认取值： 不涉及

参数	是否必选	参数类型	描述
set_id	否	String	参数解释： 组Id，查询IP地址组地址成员配额，域名组域名成员配额，服务组服务成员配额时必传，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，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，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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-80 响应 Body 参数

参数	参数类型	描述
data	ShowConfigQuotaDto object	

表 4-81 ShowConfigQuotaDto

参数	参数类型	描述
item_info	ItemInfo object	参数解释： 防火墙配额信息 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
max_quota	Integer	参数解释： 防火墙成员最大配额 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
quota_type	String	参数解释： 防火墙成员配额类型 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
used_quota	Integer	参数解释： 已使用配额 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-82 ItemInfo

参数	参数类型	描述
max_quota	Integer	参数解释： 最大配额 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
used_quota	Integer	参数解释： 已使用配额 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
extras_info	Map<String,String>	参数解释： 额外参数，ACL和网络域名使用 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

状态码：400

表 4-83 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及

参数	参数类型	描述
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询项目ID为eb7accd78ea845078275b2ad5280a109防火墙ID为8a996d43-75bd-48b4-b1b8-de0706b72b8f的防火墙配额信息，类型为ACL

https://{Endpoint}/v2/5de88e3e18124fa99675fb51ceceaa31/quota?config_type=ACL

响应示例

状态码：200

OK

```
{
  "data": {
    "max_quota": 3801,
    "used_quota": 4,
    "quota_type": "ADDR_SET",
    "item_info": {
      "max_quota": 30001,
      "used_quota": 97
    }
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.1.13 查询南北向防火墙防护状态

功能介绍

查询南北向防火墙防护状态

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/eip/protection-status/{fw_instance_id}

表 4-84 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

无

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	FirewallProtectionStatusVO object	参数解释： 查询南北向防火墙防护状态响应data 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-86 FirewallProtectionStatusVO

参数	参数类型	描述
protection_status	Integer	参数解释： 防火墙防护状态，0: 正常状态, 1: bypass进行中, 2: bypass成功, 3: bypass失败, 4: 恢复中, 5: 恢复失败 取值范围： 不涉及
id	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过防火墙ID获取方式获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	参数类型	描述
object_id	String	参数解释： 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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
failed_eip_list	Array of strings	参数解释： bypass失败的eip列表。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
failed_eip_id_list	Array of strings	参数解释： bypass失败的eip id列表。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

状态码：400

表 4-87 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询项目id为eefb11b22d5944f2aa067954516df1cf，防火墙实例id为175572ed-67e8-4837-9776-e2e07080ea45的防火墙防护状态

https://{Endpoint}/v1/eefb11b22d5944f2aa067954516df1cf/eip/protection-status/175572ed-67e8-4837-9776-e2e07080ea45

响应示例

状态码：200

OK

```
{
  "data": {
    "protection_status": 0,
    "id": "175572ed-67e8-4837-9776-e2e07080ea45",
    "object_id": "af30c598-b185-4b3e-9271-3d1393acd43e"
  }
}
```

状态码

状态码	描述
200	OK
400	Bad Request

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 4-88 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-89 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 4-90 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 4-91 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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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 4-92 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 4-93 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 4-94 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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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”
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 4-95 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 4-96 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
500	Internal Server Error

错误码

请参见[错误码](#)。

4.2.2 查询 EIP 数量

功能介绍

查询EIP数量

调用方法

请参见[如何调用API](#)。

URI

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

表 4-97 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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 4-98 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 4-99 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 4-100 响应 Body 参数

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

表 4-101 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 4-102 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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, 错误的请求

```
{
  "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();
        }
    }
}
```

```
        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 = 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，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
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 4-103 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-104 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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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
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

参数	是否必选	参数类型	描述
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 4-105 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 4-106 响应 Body 参数

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

表 4-107 EipResponseData

参数	参数类型	描述
limit	Integer	每页显示个数，范围为1-1024

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
total	Integer	查询获得EIP列表总数
records	Array of EipResource objects	eip资源记录

表 4-108 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	参数解释： 关联实例类型 约束限制： 不涉及 取值范围： PORT：IPV4云服务器 NATGW：NAT网关 ELB：负载均衡器 VPN：虚拟专用网络 EVPN：虚拟专用网络 IPV6_PORT：IPV6云服务器 默认取值： 不涉及
fw_instance_name	String	防火墙名称

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 。
fw_enterprise_project_id	String	Eip绑定的防火墙企业项目id
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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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。
bypass_status	Integer	bypass状态，0: 未bypass, 1: 已bypass, 2: 失败

状态码：400

表 4-109 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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,
      "bypass_status": 0,
      "tags": "combined_order_id=CBRCS23040615138M2KW912"
    } ],
    "total": 1
  }
}
```

状态码：400

Bad Request，错误的请求

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

SDK 代码示例

SDK代码示例如下。

Java

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

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

```
public class ListEipsSolution {

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

状态码	描述
403	Forbidden，禁止访问
404	Not Found，网页未找到
500	Internal Server Error

错误码

请参见[错误码](#)。

4.2.4 一键逃生/一键恢复开关

功能介绍

一键逃生/一键恢复开关

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/eip/protect/all/{fw_instance_id}/operation

表 4-110 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-111 请求 Body 参数

参数	是否必选	参数类型	描述
bypass_operation	否	Integer	防护操作类型，1表示一键关闭防护，0表示一键恢复防护

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	data object	一键逃生/一键恢复返回数据
fail_reason	String	一键逃生失败原因

表 4-113 data

参数	参数类型	描述
protection_status	Integer	参数解释： 防火墙防护状态，0: 正常状态, 1: bypass 进行中, 2: bypass成功, 3: bypass失败, 4: 恢复中, 5: 恢复失败 取值范围： 不涉及
id	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
object_id	String	参数解释： 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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

状态码：400

表 4-114 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

一键关闭EIP防护

```
https://{Endpoint}/v1/eefb11b22d5944f2aa067954516df1cf/eip/protect/all/175572ed-67e8-4837-9776-e2e07080ea45/operation
{
  "bypass_operation" : 1
}
```

响应示例

状态码：200

OK

```
{
  "data" : {
    "protection_status" : 1,
    "id" : "175572ed-67e8-4837-9776-e2e07080ea45",
    "object_id" : "af30c598-b185-4b3e-9271-3d1393acd43e"
  }
}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.2.5 添加 EIP 告警白名单

功能介绍

添加EIP告警白名单

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/eip/alarm-whitelist

表 4-115 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-116 请求 Body 参数

参数	是否必选	参数类型	描述
eip_infos	否	Array of EipInfo objects	参数解释： EIP详情 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-117 EipInfo

参数	是否必选	参数类型	描述
eip_id	否	String	参数解释： 弹性公网ID，可通过调用弹性IP列表查询获取，通过返回值中的data.records.id（.表示各对象之间层级的区分）获得 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
fw_instance_id	否	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	是否必选	参数类型	描述
object_id	否	String	参数解释： 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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
public_ip	否	String	参数解释： IPV4地址 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
public_ipv6	否	String	参数解释： IPV6地址 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-118 响应 Body 参数

参数	参数类型	描述
data	Object	参数解释： 添加EIP告警白名单响应data 取值范围： 不涉及

表 4-119 AddEipAlarmWhitelistRespData

参数	参数类型	描述
id	String	参数解释： 防火墙实例id 约束限制： 不涉及

状态码：400

表 4-120 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

添加告警白名单，项目id为28f403ddd3f141daa6e046e85cb15519

```
https://{Endpoint}/v1/28f403ddd3f141daa6e046e85cb15519/eip/alarm-whitelist
{
  "eip_infos": [{
    "eip_id": "10d42cba-5f7d-40c5-bf69-07685ddfa17a",
    "object_id": "b6d7a4f8-796d-424b-8bc6-c7ba22a7a1b1",
    "public_ip": "124.71.185.245",
    "type": 0
  }, {
    "eip_id": "2720563d-a547-4542-8855-87f967c5be00",
```

```
{  "object_id": "ee4c6f1c-2ee1-45d2-b276-0a41c1180035",  "public_ip": "1.94.54.125",  "type": 0  },  "fw_instance_id": "0cf6f7a8-a062-455c-a5ac-1f2af9272af6"}
```

响应示例

状态码：200

OK

```
{  "data": {    "id": "0cf6f7a8-a062-455c-a5ac-1f2af9272af6"  }}
```

状态码：400

Bad Request

```
{  "error_code": "CFW.00200003",  "error_msg": "参数错误"}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.2.6 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 4-121 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，可以通过调用API获取，也可以从控制台获取。可通过 项目ID获取方式 获取

表 4-122 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 4-123 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 4-124 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 4-125 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 4-126 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.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 = 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，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到

错误码

请参见[错误码](#)。

4.2.7 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 4-127 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。可通过 项目ID获取方式 获取

表 4-128 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 4-129 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 4-130 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

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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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	是	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 4-131 响应 Body 参数

参数	参数类型	描述
data	String	

状态码：201

Created

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，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到

错误码

请参见[错误码](#)。

4.2.8 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 4-132 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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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，可以通过调用API获取，也可以从控制台获取。可通过 项目ID获取方式 获取

表 4-133 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 4-134 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 4-135 响应 Body 参数

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

表 4-136 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，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到

错误码

请参见[错误码](#)。

4.3 访问控制规则管理

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 4-137 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-138 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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，配置后可根据企业项目过滤不同企业项目下的资产，可通过 如何获取企业项目ID 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0

参数	是否必选	参数类型	描述
fw_instance_id	否	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-139 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token，用于携带用户身份信息，配置后可通过接口鉴权，可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-140 请求 Body 参数

参数	是否必选	参数类型	描述
object_id	是	String	参数解释： 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 约束限制： type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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 取值范围： 32位UUID 默认取值： 不涉及
type	是	Integer	参数解释： 规则类型，用于区分不同防护对象设置规则类型。 约束限制： 不涉及 取值范围： 0：互联网边界规则，源（source）和目的（destination）地址需要为公网IP或域名； 1：VPC间规则，源（source）和目的（destination）地址需要为私有ip； 2：NAT规则，源（source）地址需要为私网IP，目的地址为公网IP或域名。 默认取值： 不涉及

参数	是否必选	参数类型	描述
rules	是	Array of rules objects	参数解释： 添加规则请求规则列表 约束限制： 不涉及

表 4-141 rules

参数	是否必选	参数类型	描述
name	是	String	参数解释： 规则名称，由用户定义，用于标识规则 约束限制： 字符串长度为0到255 取值范围： 不涉及 默认取值： 不涉及
sequence	是	OrderRuleAcl Dto object	参数解释： 修改规则的顺序 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
address_type	是	Integer	参数解释： IP地址的互联网协议类型，用于指定IP地址的互联网协议，由客户指定 约束限制： 不涉及 取值范围： 0表示IPv4，1表示IPv6 默认取值： 不涉及

参数	是否必选	参数类型	描述
action_type	是	Integer	参数解释： 规则动作类型，用于区分规则对流量的动作 约束限制： 仅能使用数字0和1 取值范围： 0表示允许通行（permit），1表示拒绝通行（deny） 默认取值： 不涉及
status	是	Integer	参数解释： 规则启用状态，用于区分规则是否启用 约束限制： 仅能使用数字0和1 取值范围： 0表示启用，1表示禁用 默认取值： 不涉及
applications	否	Array of strings	参数解释： 规则应用协议列表 约束限制： 不涉及 取值范围： 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。 默认取值： 不涉及

参数	是否必选	参数类型	描述
long_connect_time	否	Long	参数解释： 长连接时长（s），用于表示流量产生会话保持的最大时长。 约束限制： 仅能为数字 取值范围： 1-86400000。 默认取值： 不涉及
long_connect_time_hour	否	Long	参数解释： 长连接时长对应小时数（h）。 约束限制： 仅能为数字 取值范围： 0-24000。 默认取值： 不涉及
long_connect_time_minute	否	Long	参数解释： 长连接时长对应分钟数（min）。 约束限制： 仅能为数字 取值范围： 0-60。 默认取值： 不涉及
long_connect_time_second	否	Long	参数解释： 长连接时长对应秒数（s）。 约束限制： 仅能为数字 取值范围： 0-60。 默认取值： 不涉及

参数	是否必选	参数类型	描述
long_connect_enable	是	Integer	参数解释： 用于表示是否支持长连接。 约束限制： 不涉及 取值范围： 0表示不支持，1表示支持 默认取值： 不涉及
description	否	String	参数解释： 规则描述，用于描述规则的用途。 约束限制： 不涉及 取值范围： 长度在0-255之间 默认取值： 不涉及
direction	否	Integer	参数解释： 规则方向，用于指定规则是从云上至云下，还是云下至云上 约束限制： 当规则type=0（互联网规则）或者type= 2（NAT规则）时方向值必填 取值范围： 0表示外到内（云下到云上），1表示内到外（云上到云下）， 默认取值： 不涉及
source	是	RuleAddressDtoForRequest object	参数解释： 规则源地址传输对象 约束限制： 不涉及
destination	是	RuleAddressDtoForRequest object	参数解释： 规则目的地址传输对象 约束限制： 不涉及

参数	是否必选	参数类型	描述
service	是	RuleServiceDto object	参数解释： 规则服务对象 约束限制： 不涉及
tag	否	TagsVO object	参数解释： 规则标签对象 约束限制： 不涉及

表 4-142 OrderRuleAclDto

参数	是否必选	参数类型	描述
dest_rule_id	否	String	参数解释： 目标规则ID，填写后添加的新规则置位于此规则之后，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。 约束限制： 非置顶时目标规则需要存在 取值范围： 不涉及 默认取值： 不涉及
top	否	Integer	参数解释： 规则是否置顶，用于确认规则是否置顶 约束限制： 不涉及 取值范围： 0代表非置顶，1代表置顶 默认取值： 不涉及

参数	是否必选	参数类型	描述
bottom	否	Integer	参数解释： 规则是否置底，用于确认规则是否置底 约束限制： 不涉及 取值范围： 0代表非置底，1代表置底 默认取值： 不涉及

表 4-143 RuleAddressDtoForRequest

参数	是否必选	参数类型	描述
type	是	Integer	参数解释： 源或目的地址的输入类型，用于区分不同的输入类型 约束限制： 当规则type=0（互联网规则）或者type= 2（NAT规则）时方向值（direction）必填 取值范围： 0手动输入，1关联IP地址组，2域名，3地理位置，4域名组-应用型，5多对象，6域名组-网络型，7域名-应用型。 默认取值： 不涉及
address_type	否	Integer	参数解释： IP地址互联网协议类型，用于区分不同互联网协议 约束限制： 当type为0手动输入时，此处不能为空 取值范围： 地址类型0 IPv4，1 IPv6。 默认取值： 不涉及

参数	是否必选	参数类型	描述
address	否	String	参数解释： IP地址信息，用于明确规则IP地址 约束限制： 当type为0手动输入时，此处不能为空 取值范围： 不涉及 默认取值： 不涉及
address_set_id	否	String	参数解释： 关联IP地址组ID，用于明确规则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。 约束限制： 当type为1关联IP地址组时，此处不能为空 取值范围： 不涉及 默认取值： 不涉及
address_set_name	否	String	参数解释： 关联IP地址组名称，用于明确规则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。 约束限制： 当type为1关联IP地址组时，此处不能为空 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
domain_address_name	否	String	参数解释： 域名名称或引用域名组名称，用于明确规则引用域名或域名组名称 约束限制： 当type为2（域名）或7（域名组-应用型）时，此处不能为空，长度为0-255 取值范围： 不涉及 默认取值： 不涉及
region_list_json	否	String	参数解释： 规则地域列表json值，用于明确规则引用地域名称列表 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
region_list	否	Array of IpRegionDto objects	参数解释： 规则地域列表传输值 约束限制： 不涉及
domain_set_id	否	String	参数解释： 域名组ID，用于明确规则引用域名组。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。 约束限制： type为4（应用型域名组）或6（网络域名组）时不能为空。 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
domain_set_name	否	String	参数解释： 域名组名称，用于明确规则引用域名组。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。 约束限制： type为4（应用型域名组）或6（网络域名组）时不能为空。 取值范围： 不涉及 默认取值： 不涉及
ip_address	否	Array of strings	参数解释： IP地址列表，用于明确规则引用IP地址列表。 约束限制： 当type为5（多对象）时不能为空。 取值范围： 不涉及 默认取值： 不涉及
address_set_type	否	Integer	参数解释： 地址组类型，用于明确规则引用地址组类型。 约束限制： 当address的type为1（关联IP地址组）时，此处不能为空。 取值范围： 0表示自定义地址组，1表示WAF回源IP地址组，3表示NAT64转换地址组 默认取值： 不涉及

参数	是否必选	参数类型	描述
predefined_group	否	Array of strings	参数解释： 预定义地址组ID列表，用于明确规则引用预定义地址组id列表。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。 约束限制： 当type为5（多对象）时不能为空。 取值范围： 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 默认取值： 不涉及
address_group	否	Array of strings	参数解释： 地址组id列表，用于明确规则引用地址组id列表。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。 约束限制： 当type为5（多对象）时不能为空。 取值范围： 不涉及 默认取值： 不涉及

表 4-144 IpRegionDto

参数	是否必选	参数类型	描述
region_id	否	String	参数解释： 区域ID，用于明确规则引用地域，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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
description_cn	否	String	参数解释： 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 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
description_en	否	String	参数解释： 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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
region_type	否	Integer	参数解释： 区域类型 约束限制： 不涉及 取值范围： • 0：国家 • 1：省份 • 2：大洲 默认取值： 不涉及

表 4-145 RuleServiceDto

参数	是否必选	参数类型	描述
type	是	Integer	参数解释： 服务输入类型，用于明确规则的服务输入类型。 约束限制： 不涉及 取值范围： 0为手动输入类型，1为自动输入类型 默认取值： 不涉及
protocol	否	Integer	参数解释： 服务协议类型，用于明确规则引用服务协议类型。 约束限制： type为0（手动类型）时，此处不能为空。 取值范围： 协议类型：TCP为6，UDP为17，ICMP为1，ICMPv6为58，Any为-1 默认取值： 不涉及

参数	是否必选	参数类型	描述
protocols	否	Array of integers	参数解释： 协议列表，用于明确规则引用协议列表。 约束限制： type为0（手动类型）时，此处不能为空。 取值范围： 协议类型：TCP为6，UDP为17，ICMP为1，ICMPv6为58，Any为-1 默认取值： 不涉及
source_port	否	String	参数解释： 源端口，会话发起方的端口。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
dest_port	否	String	参数解释： 目的端口，会话接收方的端口。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
service_set_id	否	String	参数解释： 服务组ID，用于明确规则引用服务组，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。 约束限制： 当address的type为1（关联IP地址组）时，此处不能为空 取值范围： 不涉及 默认取值： 不涉及
service_set_name	否	String	参数解释： 服务（协议、源端口、目的端口）组的名称，用于明确规则引用服务组，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。 约束限制： 当address的type为1（关联IP地址组）时，此处不能为空 取值范围： 不涉及 默认取值： 不涉及
custom_service	否	Array of ServiceItem objects	参数解释： 自定义服务，用于明确规则引用服务。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
predefined_group	否	Array of strings	参数解释： 预定义服务组ID列表，用于明确规则引用预定义服务组。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。 约束限制： 查询条件中query_service_set_type需要设置为1预定义服务组。 取值范围： 不涉及 默认取值： 不涉及
service_group	否	Array of strings	参数解释： 服务组ID列表，用于明确规则引用服务组。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。 约束限制： 查询条件中query_service_set_type需要设置为0自定义服务组。 取值范围： 不涉及 默认取值： 不涉及
service_group_names	否	Array of ServiceGroupVO objects	参数解释： 服务（协议、源端口、目的端口）组的名称。列表。 约束限制： 不涉及

参数	是否必选	参数类型	描述
service_set_type	否	Integer	参数解释： 服务组类型，用于明确规则引用服务组类型。 约束限制： 不涉及 取值范围： 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 默认取值： 不涉及

表 4-146 ServiceItem

参数	是否必选	参数类型	描述
protocol	否	Integer	参数解释： 协议类型，用于明确规则网络协议 约束限制： RuleServiceDto.type为0时，此处不能为空。 取值范围： • 6: TCP • 17: UDP • 1: ICMP • 58: ICMPV6 • -1: Any 默认取值： 不涉及
source_port	否	String	参数解释： 源端口，会话发起方的端口。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
dest_port	否	String	参数解释： 目的端口，会话接收方的端口。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
description	否	String	参数解释： 服务（协议、源端口、目的端口）成员 约束限制： 字符串长度0-255 取值范围： 不涉及 默认取值： 不涉及
name	否	String	参数解释： 服务（协议、源端口、目的端口）成员 约束限制： 字符串长度0-255 取值范围： 不涉及 默认取值： 不涉及

表 4-147 ServiceGroupVO

参数	是否必选	参数类型	描述
name	否	String	参数解释： 服务（协议、源端口、目的端口）组的名称。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
protocols	否	Array of integers	参数解释： 协议列表 约束限制： 不涉及 取值范围： • 6: TCP • 17: UDP • 1: ICMP • 58: ICMPV6 • -1: Any 默认取值： 不涉及
service_set_type	否	Integer	参数解释： 服务（协议、源端口、目的端口）组的类型 约束限制： 不涉及 取值范围： 0表示自定义服务组，1表示预定义服务组 默认取值： 不涉及

参数	是否必选	参数类型	描述
set_id	否	String	参数解释： 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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-148 TagsVO

参数	是否必选	参数类型	描述
tag_id	否	String	参数解释： 规则ID 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
tag_key	否	String	参数解释： 规则标签键 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
tag_value	否	String	参数解释： 规则标签值 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-149 响应 Body 参数

参数	参数类型	描述
data	RuleIdList object	参数解释： 创建规则返回值数据 约束限制： 不涉及

表 4-150 RuleIdList

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

表 4-151 RuleId

参数	参数类型	描述
id	String	参数解释： 规则ID 取值范围： 不涉及

参数	参数类型	描述
name	String	参数解释： 规则名称 取值范围： 不涉及

状态码：400

表 4-152 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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
}
}],
"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, 错误的请求

```
{
  "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 CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment 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.getStatusCode());
    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,
            )
        ]
        request.set_body(listRulesbody)
```

```
        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，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问

状态码	描述
404	Not Found，网页未找到
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 4-153 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	是否必选	参数类型	描述
acl_rule_id	是	String	参数解释： 规则ID，创建规则后产生的唯一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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-154 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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，配置后可根据企业项目过滤不同企业项目下的资产，可通过 如何获取企业项目ID 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0
fw_instance_id	否	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

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	参数解释： 用户Token，用于携带用户身份信息，配置后可通过接口鉴权，可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-156 响应 Body 参数

参数	参数类型	描述
data	RuleId object	参数解释： 规则ID数据

表 4-157 RuleId

参数	参数类型	描述
id	String	参数解释： 规则ID 取值范围： 不涉及
name	String	参数解释： 规则名称 取值范围： 不涉及

状态码：400

表 4-158 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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, 错误的请求

```
{
  "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.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 = 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, 错误的请求
401	Unauthorized, 错误的请求
403	Forbidden, 禁止访问
404	Not Found, 网页未找到
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 4-159 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-160 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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，配置后可根据企业项目过滤不同企业项目下的资产，可通过 如何获取企业项目ID 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0
fw_instance_id	否	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-161 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token，用于携带用户身份信息，配置后可通过接口鉴权，可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-162 请求 Body 参数

参数	是否必选	参数类型	描述
object_id	是	String	参数解释： 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 约束限制： type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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 取值范围： 32位UUID 默认取值： 不涉及

参数	是否必选	参数类型	描述
rule_ids	是	Array of strings	参数解释： 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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-163 响应 Body 参数

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

表 4-164 BatchDeleteAclRulesResponseData

参数	参数类型	描述
responseDatas	Array of BatchDeleteRuleInfo objects	参数解释： 批量删除规则返回 取值范围： 不涉及

表 4-165 BatchDeleteRuleInfo

参数	参数类型	描述
name	String	参数解释： 批量删除的ACL的名称 取值范围： 不涉及
id	String	参数解释： 批量删除的ACL的ID 取值范围： 不涉及

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 4-166 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-167 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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，配置后可根据企业项目过滤不同企业项目下的资产，可通过 如何获取企业项目ID 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0

参数	是否必选	参数类型	描述
fw_instance_id	否	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-168 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token，用于携带用户身份信息，配置后可通过接口鉴权，可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-169 请求 Body 参数

参数	是否必选	参数类型	描述
rule_ids	是	Array of strings	参数解释： 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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

响应参数

状态码：200

表 4-170 响应 Body 参数

参数	参数类型	描述
-	String	

状态码：400

表 4-171 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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

删除规则击中次数

```
{ }
```

状态码：400

Bad Request, 错误的请求

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

SDK 代码示例

SDK代码示例如下。

Java

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

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

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

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

public class DeleteAclRuleHitCountSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        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.getStatusCode());
    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.valueOf("<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	删除规则击中次数
400	Bad Request，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
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 4-172 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	是否必选	参数类型	描述
acl_rule_id	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-173 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
fw_instance_id	否	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

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	参数解释： 用户Token，用于携带用户身份信息，配置后可通过接口鉴权，可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-175 请求 Body 参数

参数	是否必选	参数类型	描述
address_type	否	Integer	参数解释： IP地址的互联网协议类型，用于指定IP地址的互联网协议，由客户指定 约束限制： 不涉及 取值范围： 0表示IPv4，1表示IPv6 默认取值： 不涉及
name	否	String	参数解释： 规则名称，由用户定义，用于标识规则 约束限制： 字符串长度为0到255 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
direction	否	Integer	参数解释： 规则方向，用于指定规则是从云上至云下，还是云下至云上 约束限制： 当规则type=0（互联网规则）或者type= 2（nat规则）时方向值必填 取值范围： 0表示外到内（云下到云上），1表示内到外（云上到云下）， 默认取值： 不涉及
action_type	否	Integer	参数解释： 规则动作类型，用于区分规则对流量的动作 约束限制： 仅能使用数字0和1 取值范围： 0表示允许通行（permit），1表示拒绝通行（deny） 默认取值： 不涉及
status	否	Integer	参数解释： 规则启用状态，用于区分规则是否启用 约束限制： 仅能使用数字0和1 取值范围： 0表示启用，1表示禁用 默认取值： 不涉及

参数	是否必选	参数类型	描述
applications	否	Array of strings	参数解释： 规则应用协议列表 约束限制： 不涉及 取值范围： 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。 默认取值： 不涉及
description	否	String	参数解释： 规则描述，用于描述规则的用途。 约束限制： 不涉及 取值范围： 长度在0-255之间 默认取值： 不涉及
long_connect_time_hour	否	Long	参数解释： 长连接时长对应小时数（h）。 约束限制： 仅能为数字 取值范围： 0-24000。 默认取值： 不涉及

参数	是否必选	参数类型	描述
long_connect_time_minute	否	Long	参数解释： 长连接时长对应分钟数（min）。 约束限制： 仅能为数字 取值范围： 0-60。 默认取值： 不涉及
long_connect_time_second	否	Long	参数解释： 长连接时长对应秒数（s）。 约束限制： 仅能为数字 取值范围： 0-60。 默认取值： 不涉及
long_connect_time	否	Long	参数解释： 长连接时长（s），用于表示流量产生会话保持的最大时长。 约束限制： 仅能为数字 取值范围： 1-86400000。 默认取值： 不涉及
long_connect_enable	否	Integer	参数解释： 用于表示是否支持长连接。 约束限制： 不涉及 取值范围： 0表示不支持，1表示支持 默认取值： 不涉及
source	否	RuleAddress Dto object	参数解释： 源地址传输对象 约束限制： 不涉及

参数	是否必选	参数类型	描述
destination	否	RuleAddressDto object	参数解释： 目的地址传输对象 约束限制： 不涉及
service	否	RuleServiceDto object	参数解释： 服务对象 约束限制： 不涉及
type	否	Integer	参数解释： 规则类型，用于区分不同防护对象设置规则类型。 约束限制： 不涉及 取值范围： 0：互联网边界规则，源（source）和目的（destination）地址需要为公网IP或域名； 1：VPC间规则，源（source）和目的（destination）地址需要为私有ip； 2：NAT规则，源（source）地址需要为私网IP，目的地址为公网IP或域名。 默认取值： 不涉及
tag	否	TagsVO object	参数解释： 规则附带标签对象 约束限制： 不涉及

表 4-176 RuleAddressDto

参数	是否必选	参数类型	描述
type	是	Integer	参数解释： 地址输入类型，用于区分不同的输入类型 约束限制： 当规则type=0（互联网规则）或者type= 2（NAT规则）时方向值（direction）必填 取值范围： 0手动输入，1关联IP地址组，2域名，3地理位置，4域名组-应用型，5多对象，6域名组-网络型，7域名-应用型。 默认取值： 不涉及
address_type	否	Integer	参数解释： IP地址互联网协议类型，用于区分不同互联网协议 约束限制： 当type为0手动输入时，此处不能为空 取值范围： 地址类型0 IPv4，1 IPv6。 默认取值： 不涉及
address	否	String	参数解释： IP地址信息，用于明确规则IP地址 约束限制： 当type为0手动输入时，此处不能为空 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
address_set_id	否	String	参数解释： 关联IP地址组ID，用于明确规则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。 约束限制： 当type为1关联IP地址组时，此处不能为空 取值范围： 不涉及 默认取值： 不涉及
address_set_name	否	String	参数解释： 关联IP地址组名称，用于明确规则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。 约束限制： 当type为1关联IP地址组时，此处不能为空 取值范围： 不涉及 默认取值： 不涉及。
domain_address_name	否	String	参数解释： 域名名称或引用域名组名称，用于明确规则引用域名或域名组名称 约束限制： 当type为2或7时不能为空，长度为0-255 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
region_list_json	否	String	参数解释： 规则地域列表json值，用于明确规则引用地域名称列表 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
region_list	否	Array of IpRegionDto objects	参数解释： 规则地域列表传输值 约束限制： 不涉及
domain_set_id	否	String	参数解释： 域名组id，用于明确规则引用域名组。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。 约束限制： type为4（应用型域名组）或6（网络域名组）时不能为空。 取值范围： 不涉及 默认取值： 不涉及
domain_set_name	否	String	参数解释： 域名组名称，用于明确规则引用域名组。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。 约束限制： type为4（应用型域名组）或6（网络域名组）时不能为空。 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
ip_address	否	Array of strings	参数解释： IP地址列表，用于明确规则引用IP地址列表。 约束限制： 当type为5（多对象）时不能为空。 取值范围： 不涉及 默认取值： 不涉及
address_group	否	Array of strings	参数解释： 地址组ID列表，用于明确规则引用地址组id列表。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。 约束限制： 当type为5（多对象）时不能为空。 取值范围： 不涉及 默认取值： 不涉及
address_group_names	否	Array of AddressGroupVO objects	参数解释： 地址组名称列表 约束限制： 不涉及

参数	是否必选	参数类型	描述
address_set_type	否	Integer	参数解释： 地址组类型，用于明确规则引用地址组类型。 约束限制： 当address的type为1（关联IP地址组）时，此处不能为空。 取值范围： 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 默认取值： 不涉及
predefined_group	否	Array of strings	参数解释： 预定义地址组id列表，用于明确规则引用预定义地址组id列表。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。 约束限制： 当type为5（多对象）时不能为空。 取值范围： 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 默认取值： 不涉及

表 4-177 IpRegionDto

参数	是否必选	参数类型	描述
region_id	否	String	参数解释： 区域ID，用于明确规则引用地域，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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
description_cn	否	String	参数解释： 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 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
description_en	否	String	参数解释： 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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
region_type	否	Integer	参数解释： 区域类型 约束限制： 不涉及 取值范围： • 0：国家 • 1：省份 • 2：大洲 默认取值： 不涉及

表 4-178 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 4-179 RuleServiceDto

参数	是否必选	参数类型	描述
type	是	Integer	参数解释： 服务输入类型，用于明确规则的服务输入类型。 约束限制： 不涉及 取值范围： 0为手动输入类型，1为自动输入类型 默认取值： 不涉及
protocol	否	Integer	参数解释： 服务协议类型，用于明确规则引用服务协议类型。 约束限制： type为0（手动类型）时，此处不能为空。 取值范围： 协议类型：TCP为6，UDP为17，ICMP为1，ICMPv6为58，Any为-1 默认取值： 不涉及
protocols	否	Array of integers	参数解释： 协议列表，用于明确规则引用协议列表。 约束限制： type为0（手动类型）时，此处不能为空。 取值范围： 协议类型：TCP为6，UDP为17，ICMP为1，ICMPv6为58，Any为-1 默认取值： 不涉及

参数	是否必选	参数类型	描述
source_port	否	String	参数解释： 源端口，会话发起方的端口。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
dest_port	否	String	参数解释： 目的端口，会话接收方的端口。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
service_set_id	否	String	参数解释： 服务组ID，用于明确规则引用服务组，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。 约束限制： 当address的type为1（关联IP地址组）时，此处不能为空 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
service_set_name	否	String	参数解释： 服务（协议、源端口、目的端口）组的名称，用于明确规则引用服务组，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。 约束限制： 当address的type为1（关联IP地址组）时，此处不能为空 取值范围： 不涉及 默认取值： 不涉及
custom_service	否	Array of ServiceItem objects	参数解释： 自定义服务，用于明确规则引用服务。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
predefined_group	否	Array of strings	参数解释： 预定义服务组ID列表，用于明确规则引用预定义服务组。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。 约束限制： 查询条件中query_service_set_type需要设置为1预定义服务组。 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
service_group	否	Array of strings	参数解释： 服务组ID列表，用于明确规则引用服务组。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。 约束限制： 查询条件中query_service_set_type需要设置为0自定义服务组。 取值范围： 不涉及 默认取值： 不涉及
service_group_names	否	Array of ServiceGroup VO objects	参数解释： 服务（协议、源端口、目的端口）组的名称。列表。 约束限制： 不涉及
service_set_type	否	Integer	参数解释： 服务组类型，用于明确规则引用服务组类型。 约束限制： 不涉及 取值范围： 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 默认取值： 不涉及

表 4-180 ServiceItem

参数	是否必选	参数类型	描述
protocol	否	Integer	参数解释： 协议类型，用于明确规则网络协议 约束限制： RuleServiceDto.type为0时，此处不能为空。 取值范围： • 6: TCP • 17: UDP • 1: ICMP • 58: ICMPV6 • -1: Any 默认取值： 不涉及
source_port	否	String	参数解释： 源端口，会话发起方的端口。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
dest_port	否	String	参数解释： 目的端口，会话接收方的端口。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
description	否	String	参数解释： 服务（协议、源端口、目的端口）成员 约束限制： 字符串长度0-255 取值范围： 不涉及 默认取值： 不涉及
name	否	String	参数解释： 服务（协议、源端口、目的端口）成员 约束限制： 字符串长度0-255 取值范围： 不涉及 默认取值： 不涉及

表 4-181 ServiceGroupVO

参数	是否必选	参数类型	描述
name	否	String	参数解释： 服务（协议、源端口、目的端口）组的名称。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
protocols	否	Array of integers	参数解释： 协议列表 约束限制： 不涉及 取值范围： • 6: TCP • 17: UDP • 1: ICMP • 58: ICMPV6 • -1: Any 默认取值： 不涉及
service_set_type	否	Integer	参数解释： 服务（协议、源端口、目的端口）组的类型 约束限制： 不涉及 取值范围： 0表示自定义服务组，1表示预定义服务组 默认取值： 不涉及
set_id	否	String	参数解释： 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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-182 TagsVO

参数	是否必选	参数类型	描述
tag_id	否	String	参数解释： 规则ID 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
tag_key	否	String	参数解释： 规则标签键 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
tag_value	否	String	参数解释： 规则标签值 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-183 响应 Body 参数

参数	参数类型	描述
data	RuleId object	参数解释： 更新规则返回值数据

表 4-184 RuleId

参数	参数类型	描述
id	String	参数解释： 规则ID 取值范围： 不涉及
name	String	参数解释： 规则名称 取值范围： 不涉及

状态码：400

表 4-185 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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, 错误的请求

```
{
  "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.getStatusCode());
    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，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
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 4-186 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-187 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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，配置后可根据企业项目过滤不同企业项目下的资产，可通过 如何获取企业项目ID 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0

参数	是否必选	参数类型	描述
fw_instance_id	否	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-188 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token，用于携带用户身份信息，配置后可通过接口鉴权，可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-189 请求 Body 参数

参数	是否必选	参数类型	描述
object_id	是	String	参数解释： 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 约束限制： type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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 取值范围： 32位UUID 默认取值： 不涉及
action	是	String	参数解释： 规则动作 约束限制： 不涉及 取值范围： enable表示允许通行（permit），disable表示拒绝通行（deny） 默认取值： 不涉及

参数	是否必选	参数类型	描述
rule_ids	是	Array of strings	参数解释： 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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-190 响应 Body 参数

参数	参数类型	描述
data	Array of strings	参数解释： 批量更新acl规则ID，为请求体中传入的规则ID 取值范围： 不涉及

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.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 = 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 4-191 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
acl_rule_id	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-192 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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，配置后可根据企业项目过滤不同企业项目下的资产，可通过 如何获取企业项目ID 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0
fw_instance_id	否	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID **默认

请求参数

表 4-193 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token，用于携带用户身份信息，配置后可通过接口鉴权，可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-194 请求 Body 参数

参数	是否必选	参数类型	描述
dest_rule_id	否	String	参数解释： 目标规则ID，填写后添加的新规则置位于此规则之后，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。 约束限制： 非置顶时目标规则需要存在 取值范围： 不涉及 默认取值： 不涉及
top	否	Integer	参数解释： 规则是否置顶，用于确认规则是否置顶 约束限制： 不涉及 取值范围： 0代表非置顶，1代表置顶 默认取值： 不涉及
bottom	否	Integer	参数解释： 规则是否置底，用于确认规则是否置底 约束限制： 不涉及 取值范围： 0代表非置底，1代表置底 默认取值： 不涉及

响应参数

状态码：200

表 4-195 响应 Body 参数

参数	参数类型	描述
data	OrderRuleId object	参数解释： 规则ID数据

表 4-196 OrderRuleId

参数	参数类型	描述
id	String	参数解释： 移动的规则ID 取值范围： 不涉及

状态码：400

表 4-197 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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, 错误的请求

```
{
  "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.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

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

```
# coding: utf-8

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

状态码	描述
400	Bad Request，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
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 4-198 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-199 Query 参数

参数	是否必选	参数类型	描述
object_id	是	String	参数解释： 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 约束限制： type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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 取值范围： 32位UUID 默认取值： 不涉及
type	否	Integer	参数解释： 规则类型，用于区分不同防护对象设置规则类型。 约束限制： 不涉及 取值范围： 0：互联网边界规则，源（source）和目的（destination）地址需要为公网IP或域名； 1：VPC间规则，源（source）和目的（destination）地址需要为私有ip； 2：NAT规则，源（source）地址需要为私网IP，目的地址为公网IP或域名。 默认取值： 不涉及

参数	是否必选	参数类型	描述
ip	否	String	参数解释： IP地址信息 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
name	否	String	参数解释： 规则名称，由用户定义，用于标识规则 约束限制： 字符串长度为0到255 取值范围： 不涉及 默认取值： 不涉及
direction	否	Integer	参数解释： 规则方向，用于指定规则是从云上至云下，还是云下至云上 约束限制： 当规则type=0（互联网规则）或者type= 2（nat规则）时方向值必填 取值范围： 0表示外到内（云下到云上），1表示内到外（云上到云下）， 默认取值： 不涉及
status	否	Integer	参数解释： 规则启用状态，用于区分规则是否启用 约束限制： 仅能使用数字0和1 取值范围： 0表示启用，1表示禁用 默认取值： 不涉及

参数	是否必选	参数类型	描述
action_type	否	Integer	参数解释： 规则动作类型，用于区分规则对流量的动作 约束限制： 仅能使用数字0和1 取值范围： 0表示允许通行（permit），1表示拒绝通行（deny） 默认取值： 不涉及
address_type	否	Integer	参数解释： IP地址的互联网协议类型，用于指定IP地址的互联网协议，由客户指定 约束限制： 不涉及 取值范围： 0表示IPv4，1表示IPv6 默认取值： 不涉及
limit	是	Integer	参数解释： 每页显示个数 约束限制： 必须为数字 取值范围： 1-1024 默认取值： 不涉及
offset	是	Integer	参数解释： 偏移量：指定返回记录的开始位置 约束限制： 必须为数字 取值范围： 大于或等于0 默认取值： 不涉及

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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，配置后可根据企业项目过滤不同企业项目下的资产，可通过 如何获取企业项目ID 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0
fw_instance_id	否	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
tags_id	否	String	参数解释： 规则标签id，创建规则时产生。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
source	否	String	参数解释： 源地址。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
destination	否	String	参数解释： 目的地址。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
service	否	String	参数解释： 服务端口 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
application	否	String	参数解释： 规则应用协议列表 约束限制： 不涉及 取值范围： 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。 默认取值： 不涉及

请求参数

表 4-200 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token，用于携带用户身份信息，配置后可通过接口鉴权，可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-201 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 查询规则列表返回值数据

表 4-202 data

参数	参数类型	描述
offset	Integer	参数解释： 偏移量：指定返回记录的开始位置 取值范围： 大于或等于0
limit	Integer	参数解释： 每页显示个数 取值范围： 1-1024

参数	参数类型	描述
total	Integer	参数解释： 查询规则列表总条数 取值范围： 大于0
object_id	String	参数解释： 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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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 取值范围： 32位UUID
up_rules_count	Integer	参数解释： 顶部规则数量 取值范围： 不涉及
records	Array of records objects	参数解释： 查询规则列表记录

表 4-203 records

参数	参数类型	描述
rule_id	String	参数解释： 规则ID 取值范围： 不涉及
order_id	Integer	参数解释： 排序id 取值范围： 不涉及

参数	参数类型	描述
applications	Array of strings	参数解释： 应用列表 取值范围： 不涉及
address_type	Integer	参数解释： IP地址的互联网协议类型，用于指定IP地址的互联网协议，由客户指定 约束限制： 不涉及 取值范围： 0：IPv4 1：IPv6 默认取值： 不涉及
name	String	参数解释： 规则名称 取值范围： 不涉及
direction	Integer	参数解释： 规则方向 取值范围： 0：外到内1：内到外
action_type	Integer	参数解释： 规则动作类型，用于区分规则对流量的动作 取值范围： 0表示允许通行（ permit ）， 1表示拒绝通行（ deny ）
status	Integer	参数解释： 规则启用状态，用于区分规则是否启用 取值范围： 0表示启用， 1表示禁用
description	String	参数解释： 规则描述，用于描述规则的用途。 取值范围： 不涉及

参数	参数类型	描述
long_connect_time	Long	参数解释： 长连接时长（s）。 取值范围： 1-86400000。
long_connect_enable	Integer	参数解释： 用于表示是否支持长连接。 取值范围： 0表示不支持，1表示支持
long_connect_time_hour	Long	参数解释： 长连接时长对应小时数（h）。 取值范围： 0-24000。
long_connect_time_minute	Long	参数解释： 长连接时长对应分钟数（min）。 取值范围： 0-60。
long_connect_time_second	Long	参数解释： 长连接时长对应秒数（s）。 取值范围： 0-60。
source	RuleAddressDtoForResponse object	参数解释： 源地址对象。
destination	RuleAddressDtoForResponse object	参数解释： 目的地址对象。
service	RuleServiceDtoForResponse object	参数解释： 目的地址对象

参数	参数类型	描述
type	Integer	参数解释： 规则类型，用于区分不同防护对象设置规则类型。 取值范围： 0：互联网边界规则，源（source）和目的（destination）地址需要为公网IP或域名； 1：VPC间规则，源（source）和目的（destination）地址需要为私有ip； 2：NAT规则，源（source）地址需要为私网IP，目的地址为公网IP或域名。
created_date	String	参数解释： 规则创建时间。 取值范围： 不涉及
modified_date	String	参数解释： 规则修改时间。 取值范围： 不涉及
last_open_time	String	参数解释： 规则最后开启时间。 取值范围： 不涉及
tag	TagsVO object	参数解释： 规则附带标签对象。

表 4-204 RuleAddressDtoForResponse

参数	参数类型	描述
type	Integer	参数解释： 地址输入类型，用于区分不同的输入类型 约束限制： 不涉及 取值范围： 0手动输入，1关联IP地址组，2域名，3地理位置，4域名组-应用型，5多对象，6域名组-网络型，7域名-应用型。 默认取值： 不涉及
address_type	Integer	参数解释： IP地址互联网协议类型，用于区分不同互联网协议 约束限制： 不涉及 取值范围： 地址类型0 IPv4，1 IPv6。 默认取值： 不涉及
address	String	参数解释： IP地址信息，用于明确规则IP地址 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	参数类型	描述
address_set_id	String	参数解释： 关联IP地址组ID，用于明确规则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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
address_set_name	String	参数解释： 关联IP地址组名称，用于明确规则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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
domain_address_name	String	参数解释： 域名名称或引用域名组名称，用于明确规则引用域名或域名组名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	参数类型	描述
region_list_json	String	参数解释： 规则地域列表json值，用于明确规则引用地域名称列表 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
region_list	Array of IpRegionDto objects	参数解释： 规则地域列表传输值 约束限制： 不涉及
domain_set_id	String	参数解释： 域名组id，用于明确规则引用域名组。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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
domain_set_name	String	参数解释： 域名组名称，用于明确规则引用域名组。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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	参数类型	描述
ip_address	Array of strings	参数解释： IP地址列表，用于明确规则引用IP地址列表。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
address_group	Array of strings	参数解释： 地址组id列表，用于明确规则引用地址组id列表。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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
address_group_names	Array of AddressGroupVO objects	参数解释： 地址组名称列表 约束限制： 不涉及
address_set_type	Integer	参数解释： 地址组类型，用于明确规则引用地址组类型。 约束限制： 不涉及 取值范围： 0表示自定义地址组，1表示WAF回源IP地址组，3表示NAT64转换地址组 默认取值： 不涉及

表 4-205 IpRegionDto

参数	参数类型	描述
region_id	String	参数解释： 区域ID，用于明确规则引用地域，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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
description_cn	String	参数解释： 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 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
description_en	String	参数解释： 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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	参数类型	描述
region_type	Integer	参数解释： 区域类型 约束限制： 不涉及 取值范围： • 0：国家 • 1：省份 • 2：大洲 默认取值： 不涉及

表 4-206 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。
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 4-207 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

参数	参数类型	描述
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 4-208 ServiceItem

参数	参数类型	描述
protocol	Integer	参数解释： 协议类型，用于明确规则网络协议 约束限制： RuleServiceDto.type为0时，此处不能为空。 取值范围： • 6: TCP • 17: UDP • 1: ICMP • 58: ICMPV6 • -1: Any 默认取值： 不涉及

参数	参数类型	描述
source_port	String	参数解释： 源端口，会话发起方的端口。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
dest_port	String	参数解释： 目的端口，会话接收方的端口。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
description	String	参数解释： 服务（协议、源端口、目的端口）成员 约束限制： 字符串长度0-255 取值范围： 不涉及 默认取值： 不涉及
name	String	参数解释： 服务（协议、源端口、目的端口）成员 约束限制： 字符串长度0-255 取值范围： 不涉及 默认取值： 不涉及

表 4-209 ServiceGroupVO

参数	参数类型	描述
name	String	参数解释： 服务（协议、源端口、目的端口）组的名称。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
protocols	Array of integers	参数解释： 协议列表 约束限制： 不涉及 取值范围： • 6: TCP • 17: UDP • 1: ICMP • 58: ICMPV6 • -1: Any 默认取值： 不涉及
service_set_type	Integer	参数解释： 服务（协议、源端口、目的端口）组的类型 约束限制： 不涉及 取值范围： 0表示自定义服务组，1表示预定义服务组 默认取值： 不涉及

参数	参数类型	描述
set_id	String	参数解释： 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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-210 TagsVO

参数	参数类型	描述
tag_id	String	参数解释： 规则ID 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
tag_key	String	参数解释： 规则标签键 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	参数类型	描述
tag_value	String	参数解释： 规则标签值 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

状态码：400

表 4-211 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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",
"modified_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",
"order_id" : 1000000,
"service" : {
  "dest_port" : "0",
  "protocol" : -1,
  "source_port" : "0",
  "type" : 0
},
"source" : {
  "address_set_id" : "48bfb09b-6f3a-4371-8ddb-05d5d7148bcc",
  "address_set_name" : "ip_group",
  "address_type" : 0,
  "type" : 1
},
"status" : 1,
"type" : "0"
}],
"total" : 1,
"up_rules_count" : 0
}
}
```

状态码：400

Bad Request, 错误的请求

```
{
  "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.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 = 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，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
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 4-212 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-213 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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，配置后可根据企业项目过滤不同企业项目下的资产，可通过 如何获取企业项目ID 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
offset	是	Integer	参数解释： 偏移量：指定返回记录的开始位置 约束限制： 必须为数字 取值范围： 大于或等于0 默认取值： 不涉及

参数	是否必选	参数类型	描述
limit	是	Integer	参数解释： 每页显示个数 约束限制： 必须为数字 取值范围： 1-1024 默认取值： 不涉及

请求参数

表 4-214 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token，用于携带用户身份信息，配置后可通过接口鉴权，可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-215 响应 Body 参数

参数	参数类型	描述
data	HttpGetAclTagResponseData object	参数解释： 获取规则标签数据

表 4-216 HttpGetAclTagResponseData

参数	参数类型	描述
offset	Integer	参数解释： 偏移量：指定返回记录的开始位置 取值范围： 大于或等于0
limit	Integer	参数解释： 每页显示个数 取值范围： 1-1024
total	Integer	参数解释： 规则标签总数 取值范围： 大于0
records	Array of TagsVO objects	参数解释： 规则标签列表

表 4-217 TagsVO

参数	参数类型	描述
tag_id	String	参数解释： 规则ID 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
tag_key	String	参数解释： 规则标签键 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	参数类型	描述
tag_value	String	参数解释： 规则标签值 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

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.cfwr.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfwr.v1.*;
import com.huaweicloud.sdk.cfwr.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.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 huaweicloudsdkcfwr.v1.region.cfwr_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfwr.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 4-218 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-219 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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，配置后可根据企业项目过滤不同企业项目下的资产，可通过 如何获取企业项目ID 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0
fw_instance_id	否	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-220 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token，用于携带用户身份信息，配置后可通过接口鉴权，可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及。

表 4-221 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 4-222 响应 Body 参数

参数	参数类型	描述
data	RuleHitCountRecords object	参数解释： 获取规则击中次数响应数据

表 4-223 RuleHitCountRecords

参数	参数类型	描述
limit	Integer	参数解释： 每页显示个数 取值范围： 1-1024
offset	Integer	参数解释： 偏移量：指定返回记录的开始位置，必须为数字 取值范围： 大于或等于0，默认0
total	Integer	参数解释： 获取规则击中次数总条数 取值范围： 不涉及
records	Array of RuleHitCountObject objects	参数解释： 规则击中次数信息列表 约束限制： 不涉及

表 4-224 RuleHitCountObject

参数	参数类型	描述
rule_id	String	参数解释： 规则ID 取值范围： 不涉及
rule_hit_count	Integer	参数解释： 规则击中次数，当acl规则被触发时，对应规则ID的击中次数会添加一次。 取值范围： 不涉及

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.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

查询项目id为0b2179bbe180d3762fb0c01a2d5725c7中acl规则id为59ff6bd9-0a76-41ec-9650-380086069965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，错误的请求

状态码	描述
403	Forbidden，禁止访问
404	Not Found，网页未找到
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 4-225 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-226 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
enterprise_project_id	否	String	参数解释： 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，配置后可根据企业项目过滤不同企业项目下的资产，可通过 如何获取企业项目ID 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0

请求参数

表 4-227 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token，用于携带用户身份信息，配置后可通过接口鉴权，可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-228 响应 Body 参数

参数	参数类型	描述
data	Object	参数解释： region列表 取值范围： 不涉及

请求示例

查询region列表，项目id为408972e72dcd4c1a9b033e955802a36b，防火墙ID为fced7179-64a4-4438-bc6d-b510a2da2f64。

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/regions?fw_instance_id=fced7179-64a4-4438-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.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 = 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，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到

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 4-229 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-230 Query 参数

参数	是否必选	参数类型	描述
object_id	是	String	参数解释： 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 约束限制： type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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 取值范围： 32位UUID 默认取值： 不涉及

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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，配置后可根据企业项目过滤不同企业项目下的资产，可通过 如何获取企业项目ID 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0

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	参数解释： 用户Token，用于携带用户身份信息，配置后可通过接口鉴权，可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

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	ShowImportStat usId object	参数解释： 查看导入状态数据

表 4-233 ShowImportStatusId

参数	参数类型	描述
id	String	参数解释： 防护对象id 取值范围： 不涉及
status	Integer	参数解释： 规则导入状态 约束限制： 不涉及 取值范围： 0: 无任务态 1: 任务等待 2: 任务执行 3: 任务成功 4: 任务失败 5: 任务部分成功 6: 任务全部失败 默认取值： 不涉及

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-dac013fdfc0c。

```
https://{Endpoint}/v1/176d761dc156471c9c9b227376594160/acl-rule/import-status?object_id=cfb245cf-28b0-4c23-8164-dac013fdfc0c
```

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-dac013fdfc0c",
    "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.cfww.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfww.v1.*;
import com.huaweicloud.sdk.cfww.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.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 huaweicloudsdkcfww.v1.region.cfww_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfww.v1 import *

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

```
ak = 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，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到

错误码

请参见[错误码](#)。

4.3.13 导入访问控制规则

功能介绍

导入访问控制规则

调用方法

请参见[如何调用API](#)。

URI

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

表 4-234 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-235 Query 参数

参数	是否必选	参数类型	描述
object_id	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-236 FormData 参数

参数	是否必选	参数类型	描述
file	是	File	参数解释： 规则文件 约束限制： 请按模板要求的文件类型和内容导入规则，否则无法成功导入 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-237 响应 Body 参数

参数	参数类型	描述
data	RuleAclResponseData object	参数解释： 导入访问控制规DATA 取值范围： 不涉及

表 4-238 RuleAclResponseData

参数	参数类型	描述
id	String	参数解释： 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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

状态码：400

表 4-239 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

向项目ID为176d761dc156471c9c9b227376594160, 防护对象ID为e21c49f9-d2fe-44cb-be9a-e96c96d1fe04导入访问控制规则

```
https://{Endpoint}/v1/5de88e3e18124fa99675fb51ceceaa31/acl-rule/import?object_id=e21c49f9-d2fe-44cb-be9a-e96c96d1fe04&enterprise_project_id=default
```

响应示例

状态码：200

OK

```
{
  "data": {
    "id": "ee4c6f1c-2ee1-45d2-b276-0a41c1180035"
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.3.14 导出访问控制规则

功能介绍

导出访问控制规则

调用方法

请参见[如何调用API](#)。

URI

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

表 4-240 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-241 请求 Body 参数

参数	是否必选	参数类型	描述
object_id	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

响应参数

状态码：200

表 4-242 响应 Body 参数

参数	参数类型	描述
data	RuleAclResponse Data object	参数解释： 导出访问控制规则响应DATA 取值范围： 不涉及

表 4-243 RuleAclResponseData

参数	参数类型	描述
id	String	参数解释： 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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

状态码：400

表 4-244 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

将项目ID为28f403ddd3f141daa6e046e85cb15519, 防护对象ID为89d537d1-2ca9-4ec6-b822-8c524628e666的访问控制规则导出到文件中

```
https://{Endpoint}/v1/28f403ddd3f141daa6e046e85cb15519/acl-rule/export
{
  "object_id" : "ee4c6f1c-2ee1-45d2-b276-0a41c1180035"
}
```

响应示例

状态码：200

OK

```
{
  "data" : {
    "id" : "ee4c6f1c-2ee1-45d2-b276-0a41c1180035"
  }
}
```

状态码：400

Bad Request

```
{
  "error_code" : "CFW.00200003",
  "error_msg" : "参数错误"
}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.3.15 下载导入模板

功能介绍

下载导入模板

调用方法

请参见[如何调用API](#)。

URI

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

表 4-245 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-246 Query 参数

参数	是否必选	参数类型	描述
object_id	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-247 响应 Body 参数

参数	参数类型	描述
-	File	参数解释： 导入规则模板文件 取值范围： 不涉及

状态码：400

表 4-248 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

下载导入模板，项目ID为28f403ddd3f141daa6e046e85cb15519，防护对象ID为bc11e773-35cc-478c-a91c-772ed077e06e

https://{Endpoint}/v1/28f403ddd3f141daa6e046e85cb15519/acl-rule/import-template?
object_id=bc11e773-35cc-478c-a91c-772ed077e06e

响应示例

状态码：400

Bad Request

```
{  
  "error_code" : "CFW.00200003",  
  "error_msg" : "参数错误"  
}
```

状态码

状态码	描述
200	OK

状态码	描述
400	Bad Request

错误码

请参见[错误码](#)。

4.3.16 下载导出结果

功能介绍

下载导出结果

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/acl-rule/export-result

表 4-249 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-250 Query 参数

参数	是否必选	参数类型	描述
object_id	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-251 响应 Body 参数

参数	参数类型	描述
-	File	参数解释： 导出的规则文件 取值范围： 不涉及

状态码：400

表 4-252 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

下载导出结果，项目ID为28f403ddd3f141daa6e046e85cb15519，防护对象ID为bc11e773-35cc-478c-a91c-772ed077e06e

https://{Endpoint}/v1/28f403ddd3f141daa6e046e85cb15519/acl-rule/export-result?object_id=bc11e773-35cc-478c-a91c-772ed077e06e

响应示例

状态码：400

Bad Request

```
{
  "error_code" : "CFW.00200003",
  "error_msg" : "参数错误"
}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.3.17 查询导出结果

功能介绍

查询导出结果

调用方法

请参见[如何调用API](#)。

URI

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

表 4-253 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-254 Query 参数

参数	是否必选	参数类型	描述
object_id	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-255 响应 Body 参数

参数	参数类型	描述
data	ExportStatusVO object	参数解释： 查询导出状态响应DATA 取值范围： 不涉及

表 4-256 ExportStatusVO

参数	参数类型	描述
id	String	参数解释： 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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	参数类型	描述
status	Integer	参数解释： 导出状态 约束限制： 不涉及 取值范围： 0：无任务态 1：任务等待 2：任务执行 3：任务成功 4：任务失败 默认取值： 不涉及

状态码：400

表 4-257 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询导出结果，项ID为28f403ddd3f141daa6e046e85cb15519，防护对象ID为bc11e773-35cc-478c-a91c-772ed077e06e

https://{Endpoint}/v1/28f403ddd3f141daa6e046e85cb15519/acl-rule/export-status?
object_id=bc11e773-35cc-478c-a91c-772ed077e06e

响应示例

状态码：200

OK

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

```
{
  "id": "bc11e773-35cc-478c-a91c-772ed077e06e",
  "status": 3
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.3.18 下载导入规则

功能介绍

下载导入规则

调用方法

请参见[如何调用API](#)。

URI

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

表 4-258 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-259 Query 参数

参数	是否必选	参数类型	描述
object_id	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-260 响应 Body 参数

参数	参数类型	描述
-	File	参数解释： 导入规则文件 取值范围： 不涉及

状态码：400

表 4-261 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

下载导入结果，项目ID为28f403ddd3f141daa6e046e85cb15519，防火墙ID为bc11e773-35cc-478c-a91c-772ed077e06e

https://{Endpoint}/v1/28f403ddd3f141daa6e046e85cb15519/acl-rule/import-result?
fw_instance_id=bc11e773-35cc-478c-a91c-772ed077e06e

响应示例

状态码：400

Bad Request

```
{
  "error_code" : "CFW.00200003",
  "error_msg" : "参数错误"
}
```

状态码

状态码	描述
200	OK

状态码	描述
400	Bad Request

错误码

请参见[错误码](#)。

4.3.19 获取规则的命中次数和最后一次命中时间

功能介绍

获取规则的命中次数和最后一次命中时间

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/cfw/{fw_instance_id}/acl-rule/hit-info/batch-query

表 4-262 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-263 请求 Body 参数

参数	是否必选	参数类型	描述
rule_ids	是	Array of strings	参数解释： 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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-264 响应 Body 参数

参数	参数类型	描述
data	HitStatusResponseData object	参数解释： 获取规则的命中次数和最后一次命中时间响应DATA 取值范围： 不涉及

表 4-265 HitStatusResponseData

参数	参数类型	描述
records	Array of RuleHitStatusListVO objects	RuleHitStatusListVO

表 4-266 RuleHitStatusListVO

参数	参数类型	描述
rule_id	String	参数解释： 规则ID 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
rule_hit_count	Long	参数解释： 规则命中次数 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	参数类型	描述
rule_last_hit_time	Long	参数解释： 命中时间 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

状态码：400

表 4-267 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询项目ID为28f403ddd3f141daa6e046e85cb15519，防火墙ID为bc11e773-35cc-478c-a91c-772ed077e06e下ACL规则的命中次数和最后一次命中时间

```
https://{Endpoint}/v1/28f403ddd3f141daa6e046e85cb15519/cfw/bc11e773-35cc-478c-a91c-772ed077e06e/acl-rule/hit-info/batch-query
{
  "rule_ids" : [ "323289bf-252d-4f1b-b579-7f20d850e6cc" ]
}
```

响应示例

状态码：200

OK

```
{
  "data" : {
    "records" : [ {
      "rule_id" : "93e4e0f8-45b0-42a3-a3f6-09a10943e823",
```

```
"rule_hit_count" : 1,  
"rule_last_hit_time" : 13516541651  
  }]  
}  
}
```

状态码：400

Bad Request

```
{  
  "error_code" : "CFW.00200003",  
  "error_msg" : "参数错误"  
}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.4 黑白名单管理

4.4.1 批量添加黑白名单列表

功能介绍

批量添加黑白名单列表

调用方法

请参见[如何调用API](#)。

URI

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

表 4-268 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-269 请求 Body 参数

参数	是否必选	参数类型	描述
list_items	是	Array of BlackWhiteInfo objects	参数解释： 黑白名单信息列表 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
list_type	是	Integer	参数解释： 黑白名单列表类型 约束限制： 不涉及 取值范围： 4：黑名单 5：白名单 默认取值： 不涉及

参数	是否必选	参数类型	描述
object_id	是	String	参数解释： 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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-270 BlackWhiteInfo

参数	是否必选	参数类型	描述
address	是	String	参数解释： IP地址 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
address_type	否	Integer	参数解释： IP地址类型 约束限制： 不涉及 取值范围： 0：IPV4 1：IPV6 默认取值： 不涉及
description	否	String	参数解释： 描述 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
direction	是	Integer	参数解释： 地址方向 约束限制： 不涉及 取值范围： 0：源地址 1：目的地址 默认取值： 不涉及
port	否	String	参数解释： 端口 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
protocol	是	Integer	参数解释： 协议类型 约束限制： 不涉及 取值范围： -1： ANY 1： ICMP 6： TCP 17： UDP 58： ICMPV6 132： SCTP 手动类型不为空，自动类型为空 默认取值： 不涉及

响应参数

状态码：200

表 4-271 响应 Body 参数

参数	参数类型	描述
data	AddBlackWhiteListBatchVO object	

表 4-272 AddBlackWhiteListBatchVO

参数	参数类型	描述
duplicated_list	Array of BlackWhiteInfo objects	参数解释： 黑白名单重复添加列表 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	参数类型	描述
failed_list	Array of BlackWhiteInfo objects	参数解释： 黑白名单添加失败列表 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
success_list	Array of BlackWhiteInfo objects	参数解释： 黑白名单添加成功列表 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-273 BlackWhiteInfo

参数	参数类型	描述
address	String	参数解释： IP地址 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
address_type	Integer	参数解释： IP地址类型 约束限制： 不涉及 取值范围： 0: IPV4 1: IPV6 默认取值： 不涉及

参数	参数类型	描述
description	String	参数解释： 描述 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
direction	Integer	参数解释： 地址方向 约束限制： 不涉及 取值范围： 0：源地址 1：目的地址 默认取值： 不涉及
port	String	参数解释： 端口 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	参数类型	描述
protocol	Integer	参数解释： 协议类型 约束限制： 不涉及 取值范围： -1：ANY 1：ICMP 6：TCP 17：UDP 58：ICMPV6 132：SCTP 手动类型不为空，自动类型为空 默认取值： 不涉及

状态码：400

表 4-274 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

批量添加黑名单列表，项目id为5de88e3e18124fa99675fb51ceceaa31，将ip地址1.2.3.4、2.3.4.5、3.4.5.6添加进黑名单

```
https://{Endpoint}/v1/5de88e3e18124fa99675fb51ceceaa31/black-white-lists
{
  "object_id" : "c5bee3c1-63aa-405f-8a38-8bac2b5674e1",
  "list_type" : 4,
  "list_items" : [ {
    "address" : "1.2.3.4",
    "direction" : 0,
```

```
{
  "protocol": -1
}, {
  "address": "2.3.4.5",
  "direction": 0,
  "protocol": -1
}, {
  "address": "3.4.5.6",
  "direction": 0,
  "protocol": -1
}
}]
}
```

响应示例

状态码：200

OK

```
{
  "data": {
    "duplicated_list": [ ],
    "failed_list": [ ],
    "success_list": [ {
      "address": "1.2.3.4",
      "address_type": 0,
      "direction": 0,
      "port": "1-65535",
      "protocol": -1
    }, {
      "address": "2.3.4.5",
      "address_type": 0,
      "direction": 0,
      "port": "1-65535",
      "protocol": -1
    }, {
      "address": "3.4.5.6",
      "address_type": 0,
      "direction": 0,
      "port": "1-65535",
      "protocol": -1
    }
  ]
}
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.4.2 批量删除黑白名单列表

功能介绍

批量删除黑白名单列表

调用方法

请参见[如何调用API](#)。

URI

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

表 4-275 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-276 请求 Body 参数

参数	是否必选	参数类型	描述
list_ids	是	Array of strings	参数解释： 黑白名单列表ID 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
list_type	是	Integer	参数解释： 黑白名单列表类型 约束限制： 不涉及 取值范围： 4：黑名单 5：白名单 默认取值： 不涉及

参数	是否必选	参数类型	描述
object_id	是	String	参数解释： 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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-277 响应 Body 参数

参数	参数类型	描述
data	Array of strings	参数解释： 批量删除黑白名单列表id 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

状态码：400

表 4-278 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

批量删除黑名单，项目id为5de88e3e18124fa99675fb51ceceaa31，删除list_id为1662d45b-8a39-4119-b905-c429ad038cf0、6bb15c8e-dbe1-4314-bbde-90c806cb0663、4d574d5d-3867-4a7f-9645-95ccfdf3f3a0的黑名单

```
https://{Endpoint}/v1/5de88e3e18124fa99675fb51ceceaa31/black-white-list
{
  "list_ids": [ "1662d45b-8a39-4119-b905-c429ad038cf0", "6bb15c8e-dbe1-4314-bbde-90c806cb0663",
    "4d574d5d-3867-4a7f-9645-95ccfdf3f3a0" ],
  "list_type": 4,
  "object_id": "c5bee3c1-63aa-405f-8a38-8bac2b5674e1"
}
```

响应示例

状态码：200

OK

```
{
  "data": [ "1662d45b-8a39-4119-b905-c429ad038cf0" ]
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.4.3 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 4-279 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-280 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 4-281 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 4-282 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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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 4-283 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 4-284 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 4-285 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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, 错误的请求

```
{
  "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.getHttpStatusCode());  
        System.out.println(e.getRequestId());  
        System.out.println(e.getErrorCode());  
        System.out.println(e.getErrorMsg());  
    }  
}  
}
```

Python

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

```
# 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，错误的请求
401	Unauthorized，错误的请求

状态码	描述
403	Forbidden，禁止访问
404	Not Found，网页未找到
500	Internal Server Error

错误码

请参见[错误码](#)。

4.4.4 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 4-286 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API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 4-287 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 4-288 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 4-289 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 4-290 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 4-291 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 4-292 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

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

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/black-white-list/9d80d070b6d44942af73c9c3d38e042b
{
  "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, 错误的请求

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

SDK 代码示例

SDK代码示例如下。

Java

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

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

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

public class UpdateBlackWhiteListSolution {

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

```
        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，错误的请求

状态码	描述
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
500	Internal Server Error

错误码

请参见[错误码](#)。

4.4.5 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 4-293 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API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 4-294 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 4-295 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 4-296 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 4-297 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 4-298 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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, 错误的请求

```
{
  "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.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 = 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，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
500	Internal Server Error

错误码

请参见[错误码](#)。

4.4.6 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 4-299 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-300 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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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 4-301 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 4-302 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 4-303 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 4-304 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 4-305 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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，错误的请求

```
{
  "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.cfww.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfww.v1.*;
import com.huaweicloud.sdk.cfww.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.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 huaweicloudsdkcfww.v1.region.cfww_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfww.v1 import *

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

```
ak = 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 = 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, 错误的请求
401	Unauthorized, 错误的请求
403	Forbidden, 禁止访问
404	Not Found, 网页未找到
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 4-306 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-307 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 4-308 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 4-309 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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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，此处只能使用type为0的防护对象id
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 4-310 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 4-311 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 4-312 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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": "9dffc62-23bf-4456-83fa-80fa0fee47db",
    "name": "name"
  }
}
```

状态码：400

Bad Request, 错误的请求

```
{
  "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

状态码	描述
400	Bad Request, 错误的请求
401	Unauthorized, 错误的请求
403	Forbidden, 禁止访问
404	Not Found, 网页未找到
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 4-313 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-314 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 4-315 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 4-316 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 4-317 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 4-318 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 4-319 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 4-320 AddressItemIdWithoutName

参数	参数类型	描述
id	String	地址组成员id

表 4-321 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 4-322 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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，错误的请求

```
{
  "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 = [
        AddAddressItemsInfoDtoAddressItems(
            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，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
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 4-323 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API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 4-324 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	AddressSetId object	删除地址组返回数据

表 4-327 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 4-328 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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, 错误的请求

```
{  
  "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.cfsw.v1.region.CfwRegion;  
import com.huaweicloud.sdk.cfsw.v1.*;  
import com.huaweicloud.sdk.cfsw.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.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 = 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，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
500	Internal Server Error

错误码

请参见[错误码](#)。

4.5.4 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 4-329 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

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。

表 4-332 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。
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 4-333 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.getStatusCode());
    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.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.valueOf("<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](#)。

4.5.5 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 4-334 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API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 4-335 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目ID，用户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 4-336 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。可通过 如何获取用户Token 获取。

表 4-337 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 4-338 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 4-339 UpdateAddressSetResponseData

参数	参数类型	描述
id	String	地址组id

参数	参数类型	描述
name	String	地址组名称

状态码：400

表 4-340 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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, 错误的请求

```
{  
  "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.getHttpStatusCode());
            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，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
500	Internal Server Error

错误码

请参见[错误码](#)。

4.5.6 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 4-341 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-342 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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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
address_type	否	Integer	地址类型0 ipv4， 1 ipv6

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_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 4-343 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。

响应参数

状态码：200

表 4-344 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 4-345 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

参数	参数类型	描述
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 4-346 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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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 4-347 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及

参数	参数类型	描述
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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，错误的请求

```
{
  "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.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 = 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, 错误的请求
401	Unauthorized, 错误的请求
403	Forbidden, 禁止访问
404	Not Found, 网页未找到
500	Internal Server Error

错误码

请参见[错误码](#)。

4.5.7 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 4-348 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 可以通过调用API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 4-349 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_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 4-350 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	data object	查询地址组详情数据

表 4-352 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

参数	参数类型	描述
address_type	Integer	地址类型0 ipv4, 1 ipv6

状态码：400

表 4-353 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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, 错误的请求

```
{
  "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.setSetId("{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.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 = 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，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
500	Internal Server Error

错误码

请参见[错误码](#)。

4.5.8 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 4-354 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-355 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
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 4-356 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 4-357 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 4-358 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 4-359 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 4-360 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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, 错误的请求

```
{
  "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.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 = 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，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
500	Internal Server Error

错误码

请参见[错误码](#)。

4.5.9 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 4-361 路径参数

参数	是否必选	参数类型	描述
item_id	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-362 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
fw_instance_id	否	String	参数解释： 防火墙ID，字段已废弃 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-363 响应 Body 参数

参数	参数类型	描述
data	AddressItemId object	

表 4-364 AddressItemId

参数	参数类型	描述
id	String	参数解释： 地址组成员ID 取值范围： 不涉及
name	String	参数解释： IP地址 取值范围： 不涉及

状态码：400

表 4-365 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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

OK

```
{
  "data": {
    "id": "0396037e-a0fe-404b-b1ea-37677a1816c0",
    "name": "1.1.1.1"
  }
}
```

状态码：400**Bad Request**

```
{
  "error_code" : "CFW.00200003",
  "error_msg" : "参数错误"
}
```

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 DeleteAddressItemSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        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();
        DeleteAddressItemRequest request = new DeleteAddressItemRequest();
        request.withItemId("{item_id}");
        try {
            DeleteAddressItemResponse response = client.deleteAddressItem(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.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 = DeleteAddressItemRequest()
        request.item_id = "{item_id}"
        response = client.delete_address_item(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    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.DeleteAddressItemRequest{  
    request.ItemId = "{item_id}"  
    response, err := client.DeleteAddressItem(request)  
    if err == nil {  
        fmt.Printf("%v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

更多

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

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.5.10 批量删除地址组

功能介绍

批量删除地址组

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/address-sets/batch-delete

表 4-366 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-367 请求 Body 参数

参数	是否必选	参数类型	描述
object_id	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	是否必选	参数类型	描述
set_ids	是	Array of strings	参数解释： 地址组ID，可以通过调用[查询地址组列表接口]获得，通过返回值中的data.records.set_id获得 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

响应参数

状态码：200

表 4-368 响应 Body 参数

参数	参数类型	描述
data	Array of data objects	参数解释： 地址组信息 取值范围： 不涉及

表 4-369 data

参数	参数类型	描述
name	String	参数解释： 地址组名称 取值范围： 不涉及
id	String	参数解释： 地址组ID 取值范围： 不涉及

状态码：400

表 4-370 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

批量删除地址组，项目ID为eefb11b22d5944f2aa067954516df1cf，object ID为be6a36cf-3f3a-4e49-b437-0ec2ae6b3135，地址组ID为2d986d68-1c12-4d68-96aa-7c7689b6c393，c2fdf443-1325-4403-97cd-7dfed76a3da5

```
https://{Endpoint}/v1/eefb11b22d5944f2aa067954516df1cf/address-sets/batch-delete
{
  "object_id": "be6a36cf-3f3a-4e49-b437-0ec2ae6b3135",
  "set_ids": [ "2d986d68-1c12-4d68-96aa-7c7689b6c393", "c2fdf443-1325-4403-97cd-7dfed76a3da5" ]
}
```

响应示例

状态码：200

OK

```
{
  "data": [ {
    "name": "test",
    "id": "6d09fd06-cfb6-4c08-94b1-a7cfe3ef9413"
  } ]
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

状态码

状态码	描述
200	OK

状态码	描述
400	Bad Request

错误码

请参见[错误码](#)。

4.5.11 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}/config/object/description

表 4-371 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

请求参数

表 4-372 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 4-373 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，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
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。
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 4-374 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/config/object/description

```
{
  "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"
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

```
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 UpdateObjectConfigDescSolution {

    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();
UpdateObjectConfigDescRequest request = new UpdateObjectConfigDescRequest();
ObjectInfoDto body = new ObjectInfoDto();
body.withType("ADDR_SET");
body.withSetId("4fc0f061-3f3f-4a16-821c-30d7f6fb895d");
body.withItemId("0eac72c9-da3e-4386-b2f2-eb7dbc37d25f");
body.withFwInstanceId("b273e6c6-59f3-44cc-9a0d-89824814b474");
body.withDescription("test");
request.withBody(body);
try {
    UpdateObjectConfigDescResponse response = client.updateObjectConfigDesc(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrMsg());
}
}
```

Python

更新项目id为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

```
# 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.valueOf("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateObjectConfigDescRequest()
        request.body = ObjectInfoDto(
            type="ADDR_SET",
            set_id="4fc0f061-3f3f-4a16-821c-30d7f6fb895d",
            item_id="0eac72c9-da3e-4386-b2f2-eb7dbc37d25f",
            fw_instance_id="b273e6c6-59f3-44cc-9a0d-89824814b474",
            description="test"
        )
```

```
response = client.update_object_config_desc(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

```
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.UpdateObjectConfigDescRequest{}
    request.Body = &model.ObjectInfoDto{
        Type: "ADDR_SET",
        SetId: "4fc0f061-3f3f-4a16-821c-30d7f6fb895d",
        ItemId: "0eac72c9-da3e-4386-b2f2-eb7dbc37d25f",
        FwInstanceId: "b273e6c6-59f3-44cc-9a0d-89824814b474",
        Description: "test",
    }
    response, err := client.UpdateObjectConfigDesc(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 4-375 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-376 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目ID，用户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 4-377 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。可通过 如何获取用户Token 获取。

表 4-378 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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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，此处只能使用type为0的防护对象id
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 4-379 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 4-380 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 4-381 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

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

```
https://{Endpoint}/v1/9d80d070b6d44942af73c9c3d38e0429/service-set
{
  "object_id" : "cfebd347-b655-4b84-b938-3c54317599b2",
  "name" : "ceshi",
  "description" : ""
}
```

响应示例

状态码：200

创建服务组返回值

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

状态码：400

Bad Request，错误的请求

```
{
  "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.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

Python

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

```
# coding: utf-8

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，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
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 4-382 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-383 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目ID，用户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 4-384 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。可通过 如何获取用户Token 获取。

表 4-385 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。

参数	是否必选	参数类型	描述
service_items	是	Array of service_items objects	服务组成员列表

表 4-386 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, 自动类型为为空
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 4-387 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 4-388 ServiceItemIds

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

表 4-389 items

参数	参数类型	描述
id	String	服务组成员id

状态码：400

表 4-390 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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，错误的请求

```
{
  "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，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
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 4-391 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API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 4-392 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 4-393 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 4-394 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 4-395 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 4-396 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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，错误的请求

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

SDK 代码示例

SDK代码示例如下。

Java

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

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

public class DeleteServiceSetSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        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();
DeleteServiceSetRequest request = new DeleteServiceSetRequest();
request.setSetId("{set_id}");
try {
    DeleteServiceSetResponse response = client.deleteServiceSet(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.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 = DeleteServiceSetRequest()
        request.set_set_id = "{set_id}"
        response = client.delete_service_set(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

```
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cfw/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    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.DeleteServiceSetRequest{}
    request.SetId = "{set_id}"
    response, err := client.DeleteServiceSet(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	删除服务组返回值
400	Bad Request，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
500	Internal Server Error

错误码

请参见[错误码](#)。

4.6.4 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 4-397 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API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	否	String	防火墙ID，可通过 防火墙ID获取方式 获取

请求参数

表 4-399 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。

表 4-400 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 4-401 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 4-402 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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, 错误的请求

```
{
  "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.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 = 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，错误的请求

错误码

请参见[错误码](#)。

4.6.5 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 4-403 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API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 4-404 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 4-405 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 4-406 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 4-407 响应 Body 参数

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

表 4-408 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 4-409 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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，错误的请求

```
{
  "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.setSetId("{set_id}");
        UpdateServiceSetUsingPUTRequestBody body = new UpdateServiceSetUsingPUTRequestBody();
        body.setDescription("描述");
        body.setName("ceshi2");
        request.setBody(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，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
500	Internal Server Error

错误码

请参见[错误码](#)。

4.6.6 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 4-410 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-411 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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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 4-412 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 4-413 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 4-414 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 4-415 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 4-416 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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, 错误的请求

```
{
  "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，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
500	Internal Server Error

错误码

请参见[错误码](#)。

4.6.7 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 4-417 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API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 4-418 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 4-419 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。可通过 如何获取用户Token 获取。

响应参数

状态码：200

表 4-420 响应 Body 参数

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

表 4-421 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
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 4-422 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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, 错误的请求

```
{  
  "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.cfsw.v1.region.CfwRegion;  
import com.huaweicloud.sdk.cfsw.v1.*;  
import com.huaweicloud.sdk.cfsw.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.setSetId("{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.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 = 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，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
500	Internal Server Error

错误码

请参见[错误码](#)。

4.6.8 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 4-423 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-424 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。
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 4-425 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 4-426 响应 Body 参数

参数	参数类型	描述
data	data object	服务组成员列表

表 4-427 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 4-428 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 4-429 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

状态码：401

表 4-430 响应 Body 参数

参数	参数类型	描述
-	String	

状态码：403

表 4-431 响应 Body 参数

参数	参数类型	描述
-	String	

状态码：404

表 4-432 响应 Body 参数

参数	参数类型	描述
-	String	

状态码：500

表 4-433 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，错误的请求

```
{
  "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.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 = 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，错误的请求
401	Unauthorized，错误的请求

状态码	描述
403	Forbidden，禁止访问
404	Not Found，网页未找到
500	Internal Server Error

错误码

请参见[错误码](#)。

4.6.9 删除服务组成员

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 4-434 路径参数

参数	是否必选	参数类型	描述
item_id	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-435 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	否	String	参数解释： 防火墙ID，字段已废弃 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
enterprise_project_id	否	String	参数解释： 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 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

无

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	DeleteServiceItemResponseBodyData object	

表 4-437 DeleteServiceItemResponseBodyData

参数	参数类型	描述
id	String	参数解释： 服务组成员ID 取值范围： 不涉及
name	String	参数解释： 源、目的端口 取值范围： 不涉及

状态码：400

表 4-438 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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：400

Bad Request

```
{
  "error_code" : "CFW.00200003",
  "error_msg" : "参数错误"
}
```

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.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 = 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	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.6.10 批量删除服务组

功能介绍

批量删除服务组

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/service-sets/batch-delete

表 4-439 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-440 请求 Body 参数

参数	是否必选	参数类型	描述
object_id	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	是否必选	参数类型	描述
set_ids	是	Array of strings	参数解释： 服务组ID，可以通过调用[获取服务组列表接口]获得，通过返回值中的data.records.set_id获得 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

响应参数

状态码：200

表 4-441 响应 Body 参数

参数	参数类型	描述
data	Array of data objects	参数解释： 服务组信息 取值范围： 不涉及

表 4-442 data

参数	参数类型	描述
name	String	参数解释： 服务组名称 取值范围： 不涉及
id	String	参数解释： 服务组ID 取值范围： 不涉及

状态码：400

表 4-443 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

批量删除服务组，项目ID为eefb11b22d5944f2aa067954516df1cf，object ID为be6a36cf-3f3a-4e49-b437-0ec2ae6b3135，服务组ID为6d09fd06-cfb6-4c08-94b1-a7cfe3ef9413，9fb75958-aab1-48b6-a281-d029345cc9a4

```
https://{Endpoint}/v1/eefb11b22d5944f2aa067954516df1cf/service-sets/batch-delete
{
  "object_id": "be6a36cf-3f3a-4e49-b437-0ec2ae6b3135",
  "set_ids": [ "6d09fd06-cfb6-4c08-94b1-a7cfe3ef9413", "9fb75958-aab1-48b6-a281-d029345cc9a4" ]
}
```

响应示例

状态码：200

OK

```
{
  "data": [ {
    "name": "test",
    "id": "6d09fd06-cfb6-4c08-94b1-a7cfe3ef9413"
  }, {
    "name": "test2",
    "id": "9fb75958-aab1-48b6-a281-d029345cc9a4"
  } ]
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

状态码

状态码	描述
200	OK
400	Bad Request

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 4-444 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-445 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0
fw_instance_id	否	String	参数解释： 防火墙ID，字段废弃 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

请求参数

表 4-446 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token。可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-447 请求 Body 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
object_id	是	String	参数解释： 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 约束限制： type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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。 取值范围： 不涉及 默认取值： 不涉及
name	是	String	参数解释： 域名组名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
description	否	String	参数解释： 域名组描述 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
domain_names	是	Array of DomainSetInfoDto objects	参数解释： 域名信息列表 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
domain_set_type	否	Integer	参数解释： 域名组类型 约束限制： 不涉及 取值范围： 0：应用域名组 1：网络域名组 默认取值： 不涉及

表 4-448 DomainSetInfoDto

参数	是否必选	参数类型	描述
domain_name	是	String	参数解释： 域名 约束限制： 不涉及 取值范围： 域名格式，如 www.example.com 默认取值： 不涉及
description	否	String	参数解释： 域名描述 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-449 响应 Body 参数

参数	参数类型	描述
data	DomainSetResponseData object	参数解释： 添加域名组返回数据 取值范围： 不涉及

表 4-450 DomainSetResponseData

参数	参数类型	描述
id	String	参数解释： 域名组id 取值范围： 不涉及

参数	参数类型	描述
name	String	参数解释： 域名组名称 取值范围： 不涉及

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

OK，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	OK，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 4-451 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
set_id	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-452 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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，配置后可根据企业项目过滤不同企业项目下的资产，可通过 如何获取企业项目ID 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0
fw_instance_id	是	String	参数解释： 防火墙ID，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

请求参数

表 4-453 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token。可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-454 响应 Body 参数

参数	参数类型	描述
data	DomainSetResponseData object	参数解释： 删除域名组返回值data 取值范围： 不涉及

表 4-455 DomainSetResponseData

参数	参数类型	描述
id	String	参数解释： 域名组id 取值范围： 不涉及
name	String	参数解释： 域名组名称 取值范围： 不涉及

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

OK，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.setSetId("{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.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 = 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	OK，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 4-456 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
set_id	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-457 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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，配置后可根据企业项目过滤不同企业项目下的资产，可通过 如何获取企业项目ID 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0
fw_instance_id	是	String	参数解释： 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 。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

请求参数

表 4-458 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token。可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-459 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	参数解释： 域名组名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
description	否	String	参数解释： 域名组描述 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-460 响应 Body 参数

参数	参数类型	描述
data	DomainSetResponseData object	参数解释： 更新的域名组返回值data 取值范围： 不涉及

表 4-461 DomainSetResponseData

参数	参数类型	描述
id	String	参数解释： 域名组id 取值范围： 不涉及
name	String	参数解释： 域名组名称 取值范围： 不涉及

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

OK，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.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

更新项目id为9d80d070b6d44942af73c9c3d38e0429下的防火墙ID为7a004e79-0b8b-4679-ab20-267f3946e8ba的域名组id为94da194d-24b2-4f60-919e-cf0bc76c75b3，修改名称为test。

```
# coding: utf-8

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	OK，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 4-462 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-463 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
enterprise_project_id	否	String	参数解释： 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，配置后可根据企业项目过滤不同企业项目下的资产，可通过 如何获取企业项目ID 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0

请求参数

表 4-464 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token。可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-465 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	参数解释： DNS服务器列表 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
health_check_domain_name	否	String	参数解释： 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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-466 dns_server

参数	是否必选	参数类型	描述
server_ip	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
is_customized	是	Integer	参数解释： 是否是用户自定义的dns服务器 约束限制： 不涉及 取值范围： • 0：否 • 1：是 默认取值： 不涉及
is_applied	是	Integer	参数解释： 是否应用 约束限制： 不涉及 取值范围： • 0：否 • 1：是 默认取值： 不涉及

响应参数

状态码：200

表 4-467 响应 Body 参数

参数	参数类型	描述
data	Array of strings	参数解释： 域名服务器列表 取值范围： 不涉及

状态码：400

表 4-468 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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

OK，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, 错误的请求

```
{
  "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,  
            )  
        ]  
        request.set_body(listDnsServerbody)  
        response = client.updateDnsServers(request)  
        print(response.to_dict())  
    except exceptions.ApiException as e:  
        print("The api has error: %s" % e.to_dict())  
    except exceptions.ApiException as e:  
        print("The api has error: %s" % e.to_dict())
```

```
        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	OK，更新Dns服务器响应
400	Bad Request，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
500	Internal Server Error，系统内部错误

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 4-469 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-470 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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，配置后可根据企业项目过滤不同企业项目下的资产，可通过 如何获取企业项目ID 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0
fw_instance_id	是	String	参数解释： 防火墙ID，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
limit	是	Integer	参数解释： 每页显示个数 约束限制： 不涉及 取值范围： 1-1024 默认取值： 不涉及
offset	是	Integer	参数解释： 偏移量：指定返回记录的开始位置 约束限制： 不涉及 取值范围： 大于等于0 默认取值： 0
object_id	是	String	参数解释： 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 约束限制： type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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。 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
key_word	否	String	参数解释： 关键字，可使用域名组名称或描述 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
domain_set_type	否	Integer	参数解释： 域名组类型 约束限制： 不涉及 取值范围： 0：应用域名组 1：网络域名组 默认取值： 不涉及
config_status	否	Integer	参数解释： 配置状态 约束限制： 不涉及 取值范围： -1：未配置态 0：配置失败 1：配置成功 2：配置中 3：正常 4：配置异常 默认取值： 不涉及

请求参数

表 4-471 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token。可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-472 响应 Body 参数

参数	参数类型	描述
data	ListDomainsetsResponseData object	参数解释： 查询域名组列表返回值数据 取值范围： 不涉及

表 4-473 ListDomainsetsResponseData

参数	参数类型	描述
limit	Integer	参数解释： 每页显示个数 取值范围： 1-1024
offset	Integer	参数解释： 偏移量：指定返回记录的开始位置 取值范围： 大于等于0

参数	参数类型	描述
total	Integer	参数解释： 域名组总数 取值范围： 大于等于0
records	Array of DomainSetVo objects	参数解释： 域名组列表 取值范围： 不涉及

表 4-474 DomainSetVo

参数	参数类型	描述
set_id	String	参数解释： 域名组id 取值范围： 不涉及
name	String	参数解释： 域名组名称 取值范围： 不涉及
description	String	参数解释： 域名组描述 取值范围： 不涉及
ref_count	Integer	参数解释： 域名组被规则引用次数 取值范围： 不涉及
domain_set_type	Integer	参数解释： 域名组类型 取值范围： 0：应用域名组 1：网络域名组

参数	参数类型	描述
config_status	Integer	参数解释： 配置状态 取值范围： • -1：未配置态 • 0：配置失败 • 1：配置成功 • 2：配置中 • 3：正常 • 4：配置异常
rules	Array of UseRuleVO objects	参数解释： 使用规则id列表 取值范围： 不涉及

表 4-475 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

OK，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.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 = 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	OK，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 4-476 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-477 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	参数解释： 每页显示个数 约束限制： 不涉及 取值范围： 1-1024 默认取值： 不涉及
offset	否	Integer	参数解释： 偏移量：指定返回记录的开始位置 约束限制： 不涉及 取值范围： 大于或等于0 默认取值： 0

参数	是否必选	参数类型	描述
fw_instance_id	否	String	参数解释： 防火墙ID，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
enterprise_project_id	否	String	参数解释： 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，配置后可根据企业项目过滤不同企业项目下的资产，可通过 如何获取企业项目ID 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0

请求参数

表 4-478 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token。可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-479 响应 Body 参数

参数	参数类型	描述
data	Array of DnsServersResponseDTO objects	参数解释： dns服务器列表 取值范围： 不涉及
total	Integer	参数解释： dns服务器总数 取值范围： 不涉及

表 4-480 DnsServersResponseDTO

参数	参数类型	描述
id	Integer	参数解释： 域名服务器id 取值范围： 不涉及
is_applied	Integer	参数解释： 域名服务器是否应用 取值范围： 0：否 1：是
is_customized	Integer	参数解释： 域名服务器是否是用户自定义的dns服务器 取值范围： 0：否 1：是
server_ip	String	参数解释： DNS服务器IP 取值范围： 不涉及

参数	参数类型	描述
health_check_domain_name	String	参数解释： 健康检查域名 取值范围： 不涉及

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

OK，获取dns服务器响应

```
{  
  "data": [  
    {  
      "health_check_domain_name": "www.test1.com",  
      "id": 7106,  
      "is_applied": 1,  
      "is_customized": 0,  
      "server_ip": "114.114.114.114",  
      "status": 2  
    }, {  
      "health_check_domain_name": "www.test1.com",  
      "id": 7111,  
      "is_applied": 0,  
      "is_customized": 0,  
      "server_ip": "119.29.29.29"  
    }, {  
      "health_check_domain_name": "www.test1.com",  
      "id": 7107,  
      "is_applied": 1,  
      "is_customized": 0,  
      "server_ip": "223.5.5.5",  
      "status": 2  
    }  
  ],  
  "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 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.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 = 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	OK，获取dns服务器响应
401	Unauthorized，错误的请求

状态码	描述
403	Forbidden，禁止访问
404	Not Found，网页未找到
500	Internal Server Error，系统内部错误

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 4-481 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
domain_name	是	String	参数解释： 域名 约束限制： 域名格式，如 www.example.com 取值范围： 不涉及 默认取值： 不涉及

表 4-482 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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，配置后可根据企业项目过滤不同企业项目下的资产，可通过 如何获取企业项目ID 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0
fw_instance_id	否	Object	参数解释： 防火墙ID，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

请求参数

表 4-483 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token。可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

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	Array of strings	参数解释： 域名解析IP列表 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

状态码：400

表 4-485 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及

参数	参数类型	描述
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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

OK，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，错误的请求

```
{
  "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	OK，查询域名有效性返回值
400	Bad Request，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问

状态码	描述
404	Not Found，网页未找到
500	Internal Server Error，系统内部错误

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 4-486 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
domain_set_id	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-487 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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，配置后可根据企业项目过滤不同企业项目下的资产，可通过 如何获取企业项目ID 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0
fw_instance_id	是	String	参数解释： 防火墙ID，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
limit	是	Integer	参数解释： 每页显示个数 约束限制： 不涉及 取值范围： 1-1024 默认取值： 不涉及
offset	是	Integer	参数解释： 偏移量：指定返回记录的开始位置 约束限制： 不涉及 取值范围： 大于或等于0 默认取值： 0
domain_name	否	String	参数解释： 域名名称 约束限制： 域名格式，如 www.example.com 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
object_id	否	String	参数解释： 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 约束限制： type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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。 取值范围： 不涉及 默认取值： 不涉及

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	参数解释： 用户Token。可通过如何获取用户Token获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

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	ListDomainResponseData object	参数解释： 查询域名列表返回值数据 取值范围： 不涉及

表 4-490 ListDomainResponseData

参数	参数类型	描述
limit	Integer	参数解释： 每页显示个数 取值范围： 1-1024
offset	Integer	参数解释： 偏移量：指定返回记录的开始位置 取值范围： 大于或等于0
project_id	String	参数解释： 项目ID 取值范围： 不涉及
records	Array of DomainInfo objects	参数解释： 域名信息列表 取值范围： 不涉及
set_id	String	参数解释： 域名组id 取值范围： 不涉及
total	Integer	参数解释： 域名总数 取值范围： 不涉及

表 4-491 DomainInfo

参数	参数类型	描述
domain_address_id	String	参数解释： 域名地址id 取值范围： 不涉及
domain_name	String	参数解释： 域名 取值范围： 不涉及
description	String	参数解释： 域名描述 取值范围： 不涉及

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

OK, 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.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 = 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	OK, 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 4-492 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
set_id	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-493 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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，配置后可根据企业项目过滤不同企业项目下的资产，可通过 如何获取企业项目ID 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0
fw_instance_id	否	String	参数解释： 防火墙ID，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

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	参数解释： 用户Token。可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-495 请求 Body 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 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 。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
object_id	是	String	参数解释： 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 约束限制： type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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。 取值范围： 不涉及 默认取值： 不涉及
domain_names	是	Array of DomainSetInfoDto objects	参数解释： 域名列表 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-496 DomainSetInfoDto

参数	是否必选	参数类型	描述
domain_name	是	String	参数解释： 域名 约束限制： 不涉及 取值范围： 域名格式，如 www.example.com 默认取值： 不涉及
description	否	String	参数解释： 域名描述 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-497 响应 Body 参数

参数	参数类型	描述
data	DomainSetResponseData object	参数解释： 域名组返回值数据 取值范围： 不涉及

表 4-498 DomainSetResponseData

参数	参数类型	描述
id	String	参数解释： 域名组id 取值范围： 不涉及

参数	参数类型	描述
name	String	参数解释： 域名组名称 取值范围： 不涉及

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

OK，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.setSetId("{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.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

给项目id为14181c1245cf4fd786824efe1e2b9388下的域名组添加域名，防火墙ID为546af3f8-88e9-47f2-a205-2346d7090925，防护对象id为ae42418e-f077-41a0-9d3b-5b2f5ad9102b，域名组id为78719348-6d79-477e-acec-676a29842ab2，域名为www.bnm.com和www.vbc.com

```
# coding: utf-8

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	OK，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 4-499 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
set_id	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-500 Query 参数

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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，配置后可根据企业项目过滤不同企业项目下的资产，可通过 如何获取企业项目ID 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0
fw_instance_id	否	String	参数解释： 防火墙ID，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

请求参数

表 4-501 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token。可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-502 请求 Body 参数

参数	是否必选	参数类型	描述
object_id	是	String	参数解释： 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 约束限制： type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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。 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
domain_address_ids	是	Array of strings	参数解释： 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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-503 响应 Body 参数

参数	参数类型	描述
data	DomainSetResponseData object	参数解释： 删除域名列表返回值数据 取值范围： 不涉及

表 4-504 DomainSetResponseData

参数	参数类型	描述
id	String	参数解释： 域名组id 取值范围： 不涉及
name	String	参数解释： 域名组名称 取值范围： 不涉及

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

OK，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.setSetId("{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.setDomainAddressIds(listbodyDomainAddressIds);
body.setObjectId("ae42418e-f077-41a0-9d3b-5b2f5ad9102b");
request.setBody(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.getStatusCode());
    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.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 = 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"
    }
```

```
        "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	OK，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 4-505 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
domain_set_id	是	String	参数解释： 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 约束限制： URL域名组无法解析 取值范围： 不涉及 默认取值： 不涉及

表 4-506 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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，配置后可根据企业项目过滤不同企业项目下的资产，可通过 如何获取企业项目ID 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0

请求参数

表 4-507 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token。可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-508 响应 Body 参数

参数	参数类型	描述
data	DomainSetVo object	参数解释： 域名组详细信息 取值范围： 不涉及

表 4-509 DomainSetVo

参数	参数类型	描述
set_id	String	参数解释： 域名组id 取值范围： 不涉及
name	String	参数解释： 域名组名称 取值范围： 不涉及
description	String	参数解释： 域名组描述 取值范围： 不涉及
ref_count	Integer	参数解释： 域名组被规则引用次数 取值范围： 不涉及
domain_set_type	Integer	参数解释： 域名组类型 取值范围： 0：应用域名组 1：网络域名组
config_status	Integer	参数解释： 配置状态 取值范围： -1：未配置态 0：配置失败 1：配置成功 2：配置中 3：正常 4：配置异常
rules	Array of UseRuleVO objects	参数解释： 使用规则id列表 取值范围： 不涉及

表 4-510 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

OK，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.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 = 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	OK，域名组详细信息响应值
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到

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 4-511 路径参数

参数	是否必选	参数类型	描述
domain_address_id	是	String	参数解释： 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 约束限制： 仅网络型域名组需填写此参数 取值范围： 不涉及 默认取值： 不涉及
project_id	是	String	参数解释： 项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-512 Query 参数

参数	是否必选	参数类型	描述
address_type	否	Integer	参数解释： 地址类型 约束限制： 不涉及 取值范围： 0: ipv4 1: ipv6 默认取值： 不涉及
domain_set_id	是	String	参数解释： 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 约束限制： 仅网络型域名组需填写此参数 取值范围： 不涉及 默认取值： 不涉及
fw_instance_id	是	String	参数解释： 防火墙ID，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	参数解释： 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，配置后可根据企业项目过滤不同企业项目下的资产，可通过 如何获取企业项目ID 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0

请求参数

表 4-513 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token。可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-514 响应 Body 参数

参数	参数类型	描述
data	Array of strings	参数解释： 域名解析数据，包括域名IP列表 取值范围： 不涉及

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

OK，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();
        }
    }
}
```



```
        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	OK，域名的ip解析数据
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到

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 4-515 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-516 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	否	String	参数解释： 防火墙ID，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
enterprise_project_id	否	String	参数解释： 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，配置后可根据企业项目过滤不同企业项目下的资产，可通过 如何获取企业项目ID 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 0

请求参数

表 4-517 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释： 用户Token。可通过 如何获取用户Token 获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-518 请求 Body 参数

参数	是否必选	参数类型	描述
object_id	否	String	参数解释： 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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	是否必选	参数类型	描述
set_ids	否	Array of strings	参数解释： 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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-519 响应 Body 参数

参数	参数类型	描述
data	HttpBatchDeleteDomainSetResponseDatas object	参数解释： 批量删除域名组响应信息 取值范围： 不涉及

表 4-520 HttpBatchDeleteDomainSetResponseDatas

参数	参数类型	描述
responseDatas	Array of DomainSetId objects	参数解释： 批量删除域名组返回数据列表 取值范围： 不涉及

表 4-521 DomainSetId

参数	参数类型	描述
name	String	参数解释： 域名组名称 取值范围： 不涉及
id	String	参数解释： 域名组ID 取值范围： 不涉及

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

OK，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.getErrMsg());
        }
    }
}
```

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 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	OK，批量删除域名组请求的响应数据
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到

错误码

请参见[错误码](#)。

4.7.14 添加指定 DNS 服务器

功能介绍

添加指定DNS服务器

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/dns/server/{server_ip}/status

表 4-522 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
server_ip	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-523 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-524 响应 Body 参数

参数	参数类型	描述
data	String	参数解释： 连通性测试是否成功 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

状态码：400

表 4-525 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

项目id为28f403ddd3f141daa6e046e85cb15519，防火墙id为d06dad70-b3be-4480-86bc-b3d139e9938b的防火墙添加ip为78.45.12.56的DNS服务器

https://{Endpoint}/cfw/v1/28f403ddd3f141daa6e046e85cb15519/dns/server/78.45.12.56/status?fw_instance_id=d06dad70-b3be-4480-86bc-b3d139e9938b&enterprise_project_id=0

响应示例

状态码：200

OK

```
{
  "data": "Connection test success"
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.7.15 获取域名解析结果

功能介绍

获取域名解析结果

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/rule/domain/resolve-ip-list/{domain_address_id}

表 4-526 路径参数

参数	是否必选	参数类型	描述
domain_address_id	是	String	参数解释： 域名id 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-527 Query 参数

参数	是否必选	参数类型	描述
address_type	否	String	参数解释： 解析ip类型IPV4或IPV6 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-528 响应 Body 参数

参数	参数类型	描述
data	ParseIpListResponse object	参数解释： 查询域名解析的ip列表 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-529 ParseIpListResponse

参数	参数类型	描述
excess_ip	Array of strings	参数解释： 超过限制的ip列表 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
parsed_success_ip	Array of strings	参数解释： 成功解析的ip列表 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

状态码：400

表 4-530 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

获取项目id为5de88e3e18124fa99675fb51ceceaa31，防火墙id为e21c49f9-d2fe-44cb-be9a-e96c96d1fe的防火墙下，domain地址id为09d7f93f-b23b-40ff-a590-cf4d9604fb40的domian解析结果

```
https://{Endpoint}/cfw/v2/5de88e3e18124fa99675fb51ceceaa31/rule/domain/resolve-ip-list/09d7f93f-b23b-40ff-a590-cf4d9604fb40?fw_instance_id=e21c49f9-d2fe-44cb-be9a-e96c96d1fe04&domain_address_id=09d7f93f-b23b-40ff-a590-cf4d9604fb40
```

响应示例

状态码：200

OK

```
{
  "data": {
    "excess_ip": [ ],
    "parsed_success_ip": [ "100.93.12.160" ]
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

状态码

状态码	描述
200	OK
400	Bad Request

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 4-531 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API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
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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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 4-533 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	IpsSwitchResponseDTO object	查询ips开关返回值数据

表 4-535 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.cfww.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfww.v1.*;
import com.huaweicloud.sdk.cfww.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.getErrMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfww.v1.region.cfww_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfww.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = 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 = 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，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
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 4-536 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-537 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 4-538 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 4-539 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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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 4-540 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 4-541 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" : 2,
  "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_2);
        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.getStatusCode());
            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 = ChangepsSwitchStatusRequest()
    request.body = IpsSwitchDTO(
        status=1,
        ips_type=2,
        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_2,
        }
    }
```

```
    ObjectID: "1530de8a-522d-4771-9067-9fa4e2f53b48",
  }
  response, err := client.ChangelpsSwitchStatus(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，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
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 4-542 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-543 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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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 4-544 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 4-545 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 4-546 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，观察模式为默认模式

状态码：400

表 4-547 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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, 错误的请求

```
{
  "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.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 = 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，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
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 4-548 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-549 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 4-550 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 4-551 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 ，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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 4-552 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 4-553 data

参数	参数类型	描述
id	String	防护对象id

状态码：400

表 4-554 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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, 错误的请求

```
{
  "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 = ChangepsProtectModeRequest()  
        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.ChangelpsProtectModeRequest{}
    request.Body = &model.IpsProtectDto{
        Mode: int32(1),
        ObjectId: "cfebd347-b655-4b84-b938-3c54317599b2",
    }
    response, err := client.ChangelpsProtectMode(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	修改ips防护模式请求体
400	Bad Request，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
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 4-555 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。可通过 项目ID获取方式 获取

表 4-556 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 4-557 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 4-558 请求 Body 参数

参数	是否必选	参数类型	描述
ips_ids	是	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	是	String	防护对象id
status	是	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

响应参数

状态码：200

表 4-559 响应 Body 参数

参数	参数类型	描述
data	IpsRuleChangeRespBody object	

表 4-560 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

状态码：201

Created

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 ChangelpsRuleModeSolution {

    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();
ChangelpsRuleModeRequest request = new ChangelpsRuleModeRequest();
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 {
    ChangelpsRuleModeResponse response = client.changelpsRuleMode(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 = ChangelpsRuleModeRequest()
        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.ChangelpsRuleModeRequest{}
    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.ChangelpsRuleMode(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, 错误的请求
403	Forbidden, 禁止访问
404	Not Found, 网页未找到

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 4-561 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。可通过 项目ID获取方式 获取

表 4-562 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

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 4-563 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 4-564 请求 Body 参数

参数	是否必选	参数类型	描述
action	是	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	是	String	高级ips规则id
ips_rule_type	是	Integer	ips规则类型：0表示敏感目录扫描、1表示反弹xshell
object_id	是	String	防护对象id
param	是	String	包含特殊参数的JSON字符串
status	是	Integer	频率ips规则状态，0表示关闭，1表示打开

响应参数

状态码：200

表 4-565 响应 Body 参数

参数	参数类型	描述
data	ResponseData object	通用返回体

表 4-566 ResponseData

参数	参数类型	描述
id	String	数据id

状态码：201

Created

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.getErrMsg());
    }
}
```

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.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 = 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，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到

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 4-567 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。可通过 项目ID获取方式 获取

表 4-568 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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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 4-569 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 4-570 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 4-571 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 4-572 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，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到

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 4-573 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。可通过 项目ID获取方式 获取

表 4-574 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
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

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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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。
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 4-575 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

响应参数

状态码：200

表 4-576 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 4-577 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 4-578 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
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，错误的请求
403	Forbidden，禁止访问

状态码	描述
404	Not Found，网页未找到

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 4-579 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API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
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 4-581 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	Array of IpsRuleUpdateTimeVO objects	查询ips规则细节时间数据

表 4-583 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，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到

错误码

请参见[错误码](#)。

4.8.10 查看自定义 IPS 规则列表

功能介绍

查看自定义IPS规则列表

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/ips/custom-rule

表 4-584 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-585 Query 参数

参数	是否必选	参数类型	描述
action_type	否	Integer	参数解释： 自定义IPS规则执行动作,仅更新自定义IPS规则场景下需要设置，其他场景无需设置 约束限制： 不涉及 取值范围： 0：只记录日志 1：重置/拦截 默认取值： 不涉及

参数	是否必选	参数类型	描述
affected_os	否	Integer	参数解释： 影响操作系统 约束限制： 不涉及 取值范围： 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 默认取值： 不涉及

参数	是否必选	参数类型	描述
attack_type	否	Integer	参数解释： 攻击类型 约束限制： 不涉及 取值范围： 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 默认取值： 不涉及
ips_name	否	String	参数解释： ips规则名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
ips_id	否	String	参数解释： ips规则ID， 可通过调用获取ips规则列表获取，通过data.records.ips_id获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
protocol	否	Integer	参数解释： 协议类型 约束限制： 不涉及 取值范围： 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: ORACLE 默认取值： 不涉及

参数	是否必选	参数类型	描述
severity	否	Integer	参数解释： 严重程度 约束限制： 不涉及 取值范围： 0：致命 1：高危 2：中危 3：低危 默认取值： 不涉及

参数	是否必选	参数类型	描述
software	否	Integer	参数解释： 影响软件 约束限制： 不涉及 取值范围： 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: TREND_MICRO

参数	是否必选	参数类型	描述
			33: VMWARE 34: WORD_PRESS 35: Others 默认取值: 不涉及
object_id	否	String	参数解释: 防护对象ID, 该字段已废弃 约束限制: 不涉及 取值范围: 32位UUID 默认取值: 不涉及
enterprise_project_id	否	String	参数解释: 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 约束限制: 不涉及 取值范围: 32位UUID 默认取值: 不涉及
fw_instance_id	是	String	参数解释: 防火墙ID, 用户创建防火墙实例后产生的唯一ID, 配置后可区分不同防火墙, 可通过 防火墙ID获取方式 获取 约束限制: 不涉及 取值范围: 32位UUID 默认取值: 不涉及

参数	是否必选	参数类型	描述
limit	是	Integer	参数解释： 查询返回记录的数量限制 约束限制： 不涉及 取值范围： 1-1024 默认取值： 不涉及
offset	是	Integer	参数解释： 偏移量，表示查询该偏移量后面的记录 约束限制： 不涉及 取值范围： 0 - 1024 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-586 响应 Body 参数

参数	参数类型	描述
data	CustomerIpsPageInfo object	

表 4-587 CustomerIpsPageInfo

参数	参数类型	描述
limit	Integer	参数解释： 查询返回记录的数量限制 约束限制： 不涉及 取值范围： 1-1024 默认取值： 不涉及
offset	Integer	参数解释： 偏移量，表示查询该偏移量后面的记录 约束限制： 不涉及 取值范围： 0 - 1024 默认取值： 不涉及
records	Array of CustomerIpsListVO objects	参数解释： 自定义IPS规则列表 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
total	Integer	参数解释： 自定义IPS规则数量 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-588 CustomerIpsListVO

参数	参数类型	描述
action	Integer	参数解释： 自定义IPS规则执行动作,仅更新自定义IPS规则场景下需要设置，其他场景无需设置 约束限制： 不涉及 取值范围： 0：只记录日志 1：重置/拦截 默认取值： 不涉及
affected_os	Integer	参数解释： 影响操作系统 约束限制： 不涉及 取值范围： 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 默认取值： 不涉及

参数	参数类型	描述
attack_type	Integer	参数解释： 攻击类型 约束限制： 不涉及 取值范围： 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 默认取值： 不涉及
config_status	Integer	参数解释： 规则配置状态 约束限制： 不涉及 取值范围： 0：初始化 1：配置中 2：配置成功 3：配置失败 默认取值： 不涉及

参数	参数类型	描述
content	String	参数解释： 匹配IPS攻击的内容 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
dst_port_type	Integer	参数解释： 端口操作类型 约束限制： 不涉及 取值范围： -1：Any 0：包含 1：排除 默认取值： 不涉及
dst_ports	String	参数解释： 端口 约束限制： 不涉及 取值范围： 1 - 65535 默认取值： 不涉及
ips_cfw_id	String	参数解释： cfw侧自定义IPS规则id 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	参数类型	描述
ips_id	String	参数解释： ips规则ID， 可通过调用获取ips规则列表获取，通过data.records.ips_id获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
ips_name	String	参数解释： ips规则名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	参数类型	描述
protocol	Integer	参数解释： 协议类型 约束限制： 不涉及 取值范围： 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: ORACLE 默认取值： 不涉及

参数	参数类型	描述
severity	Integer	参数解释： 严重程度 约束限制： 不涉及 取值范围： 0：致命 1：高危 2：中危 3：低危 默认取值： 不涉及

参数	参数类型	描述
software	Integer	参数解释： 影响软件 约束限制： 不涉及 取值范围： 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: TREND_MICRO

参数	参数类型	描述
		33: VMWARE 34: WORD_PRESS 35: Others 默认取值: 不涉及
src_port_type	Integer	参数解释: 端口操作类型 约束限制: 不涉及 取值范围: -1: Any 0: 包含 1: 排除 默认取值: 不涉及
src_ports	String	参数解释: 端口 约束限制: 不涉及 取值范围: 不涉及 默认取值: 不涉及
group_id	String	参数解释: 集群ID 约束限制: 不涉及 取值范围: 不涉及 默认取值: 不涉及

状态码：400

表 4-589 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询项目ID为eb7accd78ea845078275b2ad5280a109防火墙ID为8a996d43-75bd-48b4-b1b8-de0706b72b8f的自定义IPS规则列表，偏移量为0，返回数量限制为10

```
https://{Endpoint}/v1/eb7accd78ea845078275b2ad5280a109/ips/custom-rule?fw_instance_id=8a996d43-75bd-48b4-b1b8-de0706b72b8f&offset=0&limit=10
```

响应示例

状态码：200

OK

```
{
  "data": {
    "limit": 1000,
    "offset": 0,
    "records": [ {
      "action": 0,
      "affected_os": 1,
      "attack_type": 1,
      "config_status": 2,
      "content": "\"test\";distance:0;within:65535;nocase;",
      "dst_port_type": 1,
      "dst_ports": "2",
      "ips_cfw_id": "3d42ef53-9b44-4d8c-b83e-b1cce27bd543",
      "ips_id": "350000",
      "ips_name": "自定义IPS",
      "protocol": 9,
      "severity": 0,
      "software": 1,
      "src_port_type": 0,
      "src_ports": "1"
    } ],
    "total": 1
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
}
```



```
"error_msg": "参数错误"
}
```

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 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
400	Bad Request

错误码

请参见[错误码](#)。

4.8.11 创建自定义 IPS 规则

功能介绍

创建自定义IPS规则

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/ips/custom-rule

表 4-590 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-591 请求 Body 参数

参数	是否必选	参数类型	描述
action_type	是	Integer	参数解释： 自定义IPS规则执行动作,仅更新自定义IPS规则场景下需要设置，其他场景无需设置 约束限制： 不涉及 取值范围： 0：只记录日志 1：重置/拦截 默认取值： 不涉及

参数	是否必选	参数类型	描述
affected_os	是	Integer	参数解释： 影响操作系统 约束限制： 不涉及 取值范围： 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 默认取值： 不涉及

参数	是否必选	参数类型	描述
attack_type	是	Integer	参数解释： 攻击类型 约束限制： 不涉及 取值范围： 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 默认取值： 不涉及
contents	是	Array of lpsContent objects	参数解释： 匹配IPS攻击的报文内容 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
direction	是	Integer	参数解释： 方向 约束限制： 不涉及 取值范围： -1：Any 0：客户端到服务端 1：服务端到客户端 默认取值： 不涉及
dst_port	是	Port object	参数解释： 端口信息 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
object_id	否	String	参数解释： 防护对象ID，字段废弃 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	是否必选	参数类型	描述
ips_name	是	String	参数解释： ips规则名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
protocol	是	Integer	参数解释： 协议类型 约束限制： 不涉及 取值范围： 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: ORACLE 默认取值： 不涉及

参数	是否必选	参数类型	描述
severity	是	Integer	参数解释： 严重程度 约束限制： 不涉及 取值范围： 0：致命 1：高危 2：中危 3：低危 默认取值： 不涉及

参数	是否必选	参数类型	描述
software	是	Integer	参数解释： 影响软件 约束限制： 不涉及 取值范围： 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: TREND_MICRO

参数	是否必选	参数类型	描述
			33: VMWARE 34: WORD_PRESS 35: Others 默认取值: 不涉及
src_port	是	Port object	参数解释: 端口信息 约束限制: 不涉及 取值范围: 不涉及 默认取值: 不涉及

表 4-592 IpsContent

参数	是否必选	参数类型	描述
content	否	String	参数解释: 内容 约束限制: 不涉及 取值范围: 不涉及 默认取值: 不涉及
depth	否	Integer	参数解释: 匹配特征时，截止匹配的位置 约束限制: 不涉及 取值范围: 不涉及 默认取值: 不涉及

参数	是否必选	参数类型	描述
is_hex	否	Boolean	参数解释： 报文内容是否为十六进制 约束限制： 不涉及 取值范围： true： 是十六进制 false： 不是十六进制 默认取值： 不涉及
is_ignore	否	Boolean	参数解释： 是否忽略大小写 约束限制： 不涉及 取值范围： true： 忽略 false： 不忽略 默认取值： 不涉及
is_uri	否	Boolean	参数解释： 是否匹配URL中跟“内容”一致的字段 约束限制： 不涉及 取值范围： true： 匹配 false： 不匹配 默认取值： 不涉及
offset	否	Integer	参数解释： 匹配特征时开始的位置 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
relative_position	否	Integer	参数解释： 匹配特征时，指定开始的位置 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-593 Port

参数	是否必选	参数类型	描述
port_type	否	Integer	参数解释： 端口操作类型 约束限制： 不涉及 取值范围： -1：Any 0：包含 1：排除 默认取值： 不涉及
ports	否	String	参数解释： 端口 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-594 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 新增自定义IPS规则响应数据 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-595 data

参数	参数类型	描述
id	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

状态码：400

表 4-596 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

创建自定义IPS规则，项目ID为28f403ddd3f141daa6e046e85cb15519，防火墙ID为d06dad70-b3be-4480-86bc-b3d139e9938b，规则名称为IPS_test，攻击类型为访问控制，影响软件为ADOBE，操作系统为WINDOWS，方向为服务器到客户端，协议类型为HTTP，源类型为包括端口1，目的类型排除目的端口2，动作为观察，内容为content:"test;"distance:0;within:65535;nocase;

https://{Endpoint}/v1/28f403ddd3f141daa6e046e85cb15519/ips/custom-rule

```
{
  "fw_instance_id": "d06dad70-b3be-4480-86bc-b3d139e9938b",
  "ips_name": "IPS_test",
  "direction": 1,
  "src_port": {
    "port_type": 1,
    "ports": "1"
  },
  "dst_port": {
    "port_type": 0,
    "ports": "2"
  },
  "contents": [ {
    "content": "12",
    "is_hex": true,
    "is_ignore": false,
    "is_uri": false,
    "relative_position": 0,
    "offset": 0,
    "depth": 65535
  } ],
  "action_type": 0,
  "severity": 1,
  "attack_type": 1,
  "software": 4,
  "affected_os": 5,
  "protocol": 9
}
```

响应示例

状态码：200

OK

```
{
  "data": {
    "id": "bc11e773-35cc-478c-a91c-772ed077e06e"
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```


状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.8.12 批量删除自定义 IPS 规则

功能介绍

批量删除自定义IPS规则

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/ips/custom-rule/batch-delete

表 4-597 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-598 请求 Body 参数

参数	是否必选	参数类型	描述
action_type	否	Integer	参数解释： 自定义IPS规则执行动作,仅更新自定义IPS规则场景下需要设置，其他场景无需设置 约束限制： 不涉及 取值范围： 0：只记录日志 1：重置/拦截 默认取值： 不涉及
ips_ids	是	Array of strings	参数解释： ips规则ID， 可通过调用获取ips规则列表获取，通过data.records.ips_id获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

响应参数

状态码：200

表 4-599 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 批量操作自定义IPS规则响应数据 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-600 data

参数	参数类型	描述
id	String	参数解释： 防火墙id 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

状态码：400

表 4-601 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

批量删除项目id为408972e72dcd4c1a9b033e955802a36b，防火墙id为d06dad70-b3be-4480-86bc-b3d139e9938b，ips id为cd6b6abe-9d28-4063-a61a-3d27ccaa2f4c和7fb1b9b8-7462-4231-96c9-971c75557288的自定义IPS规则列表

```
https://{Endpoint}/v1/28f403ddd3f141daa6e046e85cb15519/ips/custom-rule/batch-delete
```

```
{
  "fw_instance_id": "d06dad70-b3be-4480-86bc-b3d139e9938b",
  "ips_ids": [ "cd6b6abe-9d28-4063-a61a-3d27ccaa2f4c", "7fb1b9b8-7462-4231-96c9-971c75557288" ]
}
```

响应示例

状态码：200

OK

```
{
  "data": {
    "id": "bc11e773-35cc-478c-a91c-772ed077e06e"
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.8.13 批量更新自定义IPS规则的动作

功能介绍

批量更新自定义IPS规则的动作

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/ips/custom-rule/action

表 4-602 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-603 请求 Body 参数

参数	是否必选	参数类型	描述
action_type	否	Integer	参数解释： 自定义IPS规则执行动作,仅更新自定义IPS规则场景下需要设置，其他场景无需设置 约束限制： 不涉及 取值范围： 0：只记录日志 1：重置/拦截 默认取值： 不涉及

参数	是否必选	参数类型	描述
ips_ids	是	Array of strings	参数解释： ips规则ID， 可通过调用获取ips规则列表获取，通过data.records.ips_id获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
fw_instance_id	是	String	参数解释： 防火墙ID， 用户创建防火墙实例后产生的唯一ID， 配置后可区分不同防火墙， 可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

响应参数

状态码：200

表 4-604 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 批量操作自定义IPS规则响应数据 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-605 data

参数	参数类型	描述
id	String	参数解释： 防火墙id 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

状态码：400

表 4-606 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

批量更新项目id为408972e72dcd4c1a9b033e955802a36b，防火墙id为d06dad70-b3be-4480-86bc-b3d139e9938b，ips id为797d82a4-9628-4e2b-aa77-55c17ed3aeb4和ceea5743-5e1a-4221-a543-e586c03eeeca的自定义IPS规则列表的action为拦截

```
https://{Endpoint}/v1/408972e72dcd4c1a9b033e955802a36b/ips/custom-rule/action
{
  "fw_instance_id": "d06dad70-b3be-4480-86bc-b3d139e9938b",
  "action_type": 1,
  "ips_ids": [ "797d82a4-9628-4e2b-aa77-55c17ed3aeb4", "ceea5743-5e1a-4221-a543-e586c03eeeca" ]
}
```

响应示例

状态码：200

OK

```
{
  "data": {
    "id": "bc11e773-35cc-478c-a91c-772ed077e06e"
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.8.14 查询自定义 IPS 规则详情

功能介绍

功能说明：自定义IPS规则详情，特性:根据路径输入的IPS ID查看对应的自定义IPS详情

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/ips/custom-rule/{ips_cfw_id}

表 4-607 路径参数

参数	是否必选	参数类型	描述
ips_cfw_id	是	String	参数解释： cfw侧自定义IPS规则id 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-608 Query 参数

参数	是否必选	参数类型	描述
object_id	否	String	参数解释： 防护对象ID，该字段已废弃 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-609 响应 Body 参数

参数	参数类型	描述
data	CustomerIpsVO object	参数解释： 自定义IPS规则返回数据 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-610 CustomerIpsVO

参数	参数类型	描述
action	Integer	参数解释： 自定义IPS规则执行动作,仅更新自定义IPS规则场景下需要设置，其他场景无需设置 约束限制： 不涉及 取值范围： 0：只记录日志 1：重置/拦截 默认取值： 不涉及
group_id	String	参数解释： 防火墙集群id，参数已废弃 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
affected_os	Integer	参数解释： 影响操作系统 约束限制： 不涉及 取值范围： 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 默认取值： 不涉及

参数	参数类型	描述
attack_type	Integer	参数解释： 攻击类型 约束限制： 不涉及 取值范围： 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 默认取值： 不涉及
config_status	Integer	参数解释： 规则状态 约束限制： 不涉及 取值范围： 0：初始化 1：配置中 2：配置成功 3：配置失败 默认取值： 不涉及

参数	参数类型	描述
contents	Array of IpsContent objects	参数解释： 匹配IPS攻击的报文内容 约束限制： 不涉及 取值范围： 默认取值： 不涉及
direction	Integer	参数解释： 方向 约束限制： 不涉及 取值范围： 0：客户端到服务端 1：服务端到客户端 默认取值： 不涉及
dst_port	Port object	参数解释： 端口信息 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
ips_cfw_id	String	参数解释： cfw侧自定义IPS规则id 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	参数类型	描述
ips_id	String	参数解释： ips规则ID， 可通过调用获取ips规则列表获取，通过data.records.ips_id获取。 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
ips_name	String	参数解释： ips规则名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	参数类型	描述
protocol	Integer	参数解释： 协议类型 约束限制： 不涉及 取值范围： 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: ORACLE 默认取值： 不涉及

参数	参数类型	描述
severity	Integer	参数解释： 严重程度 约束限制： 不涉及 取值范围： 0：致命 1：高危 2：中危 3：低危 默认取值： 不涉及

参数	参数类型	描述
software	Integer	参数解释： 影响软件 约束限制： 不涉及 取值范围： 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: TREND_MICRO

参数	参数类型	描述
		33: VMWARE 34: WORD_PRESS 35: Others 默认取值: 不涉及
src_port	Port object	参数解释: 端口信息 约束限制: 不涉及 取值范围: 不涉及 默认取值: 不涉及

表 4-611 IpsContent

参数	参数类型	描述
content	String	参数解释: 内容 约束限制: 不涉及 取值范围: 不涉及 默认取值: 不涉及
depth	Integer	参数解释: 匹配特征时, 截止匹配的位置 约束限制: 不涉及 取值范围: 不涉及 默认取值: 不涉及

参数	参数类型	描述
is_hex	Boolean	参数解释： 报文内容是否为十六进制 约束限制： 不涉及 取值范围： true： 是十六进制 false： 不是十六进制 默认取值： 不涉及
is_ignore	Boolean	参数解释： 是否忽略大小写 约束限制： 不涉及 取值范围： true： 忽略 false： 不忽略 默认取值： 不涉及
is_uri	Boolean	参数解释： 是否匹配URL中跟“内容”一致的字段 约束限制： 不涉及 取值范围： true： 匹配 false： 不匹配 默认取值： 不涉及
offset	Integer	参数解释： 匹配特征时开始的位置 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	参数类型	描述
relative_position	Integer	参数解释： 匹配特征时，指定开始的位置 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-612 Port

参数	参数类型	描述
port_type	Integer	参数解释： 端口操作类型 约束限制： 不涉及 取值范围： -1：Any 0：包含 1：排除 默认取值： 不涉及
ports	String	参数解释： 端口 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

状态码：400

表 4-613 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询项目ID为3d84b4755b39476997c9f7b5a95c25ed，防火墙ID为d6e0a4d0-8b16-47fe-98b6-e1b0a7a14788，防护对象ID为64b8978e-20c0-4b43-b178-885e3cbb380d，自定义IPS规则id为c059d198-5996-46dd-9ed8-83167e6babe8的IPS规则详情

https://{Endpoint}/v1/3d84b4755b39476997c9f7b5a95c25ed/ips/custom-rule/c059d198-5996-46dd-9ed8-83167e6babe8?fw_instance_id=d6e0a4d0-8b16-47fe-98b6-e1b0a7a14788&object_id=64b8978e-20c0-4b43-b178-885e3cbb380d

响应示例

状态码：200

OK

```
{
  "data": {
    "action": 1,
    "affected_os": 0,
    "attack_type": 1,
    "config_status": 2,
    "contents": [ {
      "content": "apitest",
      "depth": 65535,
      "is_hex": false,
      "is_ignore": true,
      "is_uri": false,
      "offset": 0,
      "relative_position": 0
    } ],
    "direction": -1,
    "dst_port": {
      "port_type": -1
    },
    "ips_cfw_id": "c059d198-5996-46dd-9ed8-83167e6babe8",
    "ips_id": "350020",
    "ips_name": "api验证可删可改",
    "protocol": 9,
    "severity": 0,
    "software": 0,
    "src_port": {
      "port_type": -1
    }
  }
}
```

```
}  
}
```

状态码：400

Bad Request

```
{  
  "error_code": "CFW.00200003",  
  "error_msg": "参数错误"  
}
```

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 ShowCustomerIpsInfoSolution {  
  
    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();  
        ShowCustomerIpsInfoRequest request = new ShowCustomerIpsInfoRequest();  
        request.withIpsCfwId("{ips_cfw_id}");  
        try {  
            ShowCustomerIpsInfoResponse response = client.showCustomerIpsInfo(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 = ShowCustomerIpsInfoRequest()  
        request.ips_cfw_id = "{ips_cfw_id}"  
        response = client.show_customer_ips_info(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.ShowCustomerIpsInfoRequest{  
    request.IpsCfwId = "{ips_cfw_id}"  
    response, err := client.ShowCustomerIpsInfo(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

错误码

请参见[错误码](#)。

4.8.15 更新自定义 IPS 规则

功能介绍

更新自定义IPS规则

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/ips/custom-rule/{ips_cfw_id}

表 4-614 路径参数

参数	是否必选	参数类型	描述
ips_cfw_id	是	String	参数解释： cfw侧自定义IPS规则id 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-615 请求 Body 参数

参数	是否必选	参数类型	描述
action_type	是	Integer	参数解释： 自定义IPS规则执行动作,仅更新自定义IPS规则场景下需要设置，其他场景无需设置 约束限制： 不涉及 取值范围： 0：只记录日志 1：重置/拦截 默认取值： 不涉及

参数	是否必选	参数类型	描述
affected_os	是	Integer	参数解释： 影响操作系统 约束限制： 不涉及 取值范围： 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 默认取值： 不涉及

参数	是否必选	参数类型	描述
attack_type	是	Integer	参数解释： 攻击类型 约束限制： 不涉及 取值范围： 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 默认取值： 不涉及
contents	是	Array of IpsContent objects	参数解释： 匹配IPS攻击的报文内容 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
direction	是	Integer	参数解释： 方向 约束限制： 不涉及 取值范围： -1：Any 0：客户端到服务端 1：服务端到客户端 默认取值： 不涉及
dst_port	是	Port object	参数解释： 端口信息 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
object_id	否	String	参数解释： 防护对象ID，字段废弃 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	是否必选	参数类型	描述
ips_name	是	String	参数解释： ips规则名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
protocol	是	Integer	参数解释： 协议类型 约束限制： 不涉及 取值范围： 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: ORACLE 默认取值： 不涉及

参数	是否必选	参数类型	描述
severity	是	Integer	参数解释： 严重程度 约束限制： 不涉及 取值范围： 0：致命 1：高危 2：中危 3：低危 默认取值： 不涉及

参数	是否必选	参数类型	描述
software	是	Integer	参数解释： 影响软件 约束限制： 不涉及 取值范围： 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: TREND_MICRO

参数	是否必选	参数类型	描述
			33: VMWARE 34: WORD_PRESS 35: Others 默认取值: 不涉及
src_port	是	Port object	参数解释: 端口信息 约束限制: 不涉及 取值范围: 不涉及 默认取值: 不涉及

表 4-616 IpsContent

参数	是否必选	参数类型	描述
content	否	String	参数解释: 内容 约束限制: 不涉及 取值范围: 不涉及 默认取值: 不涉及
depth	否	Integer	参数解释: 匹配特征时，截止匹配的位置 约束限制: 不涉及 取值范围: 不涉及 默认取值: 不涉及

参数	是否必选	参数类型	描述
is_hex	否	Boolean	参数解释： 报文内容是否为十六进制 约束限制： 不涉及 取值范围： true： 是十六进制 false： 不是十六进制 默认取值： 不涉及
is_ignore	否	Boolean	参数解释： 是否忽略大小写 约束限制： 不涉及 取值范围： true： 忽略 false： 不忽略 默认取值： 不涉及
is_uri	否	Boolean	参数解释： 是否匹配URL中跟“内容”一致的字段 约束限制： 不涉及 取值范围： true： 匹配 false： 不匹配 默认取值： 不涉及
offset	否	Integer	参数解释： 匹配特征时开始的位置 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
relative_position	否	Integer	参数解释： 匹配特征时，指定开始的位置 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-617 Port

参数	是否必选	参数类型	描述
port_type	否	Integer	参数解释： 端口操作类型 约束限制： 不涉及 取值范围： -1：Any 0：包含 1：排除 默认取值： 不涉及
ports	否	String	参数解释： 端口 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-618 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 更新自定义IPS规则响应数据 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-619 data

参数	参数类型	描述
id	String	参数解释： 防火墙id 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

状态码：400

表 4-620 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

更新项目id为3d84b4755b39476997c9f7b5a95c25ed，ips_cfw_id为cd74e74a-310f-4558-a4bf-da4215c820f8的自定义IPS规则

```
https://{Endpoint}/v1/3d84b4755b39476997c9f7b5a95c25ed/ips/custom-rule/cd74e74a-310f-4558-a4bf-da4215c820f8
```

```
{
  "fw_instance_id": "d6e0a4d0-8b16-47fe-98b6-e1b0a7a14788",
  "ips_name": "test_操作系统linux",
  "direction": -1,
  "src_port": {
    "port_type": -1
  },
  "dst_port": {
    "port_type": -1
  },
  "contents": [ {
    "content": "ab",
    "is_hex": false,
    "is_ignore": false,
    "is_uri": false,
    "relative_position": 0,
    "offset": 0,
    "depth": 65535
  } ],
  "action_type": 1,
  "severity": 0,
  "attack_type": 1,
  "software": 0,
  "affected_os": 1,
  "protocol": 9
}
```

响应示例

状态码：200

OK

```
{
  "data": {
    "id": "64b8978e-20c0-4b43-b178-885e3cbb380d"
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

SDK 代码示例

SDK代码示例如下。

Java

更新项目id为3d84b4755b39476997c9f7b5a95c25ed，ips_cfw_id为cd74e74a-310f-4558-a4bf-da4215c820f8的自定义IPS规则

```
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 UpdateCustomerIpsSolution {

    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();
        UpdateCustomerIpsRequest request = new UpdateCustomerIpsRequest();
        request.withIpsCfwId("{ips_cfw_id}");
        CustomerIpsSaveDto body = new CustomerIpsSaveDto();
        Port srcPortbody = new Port();
        srcPortbody.withPortType(-1);
        Port dstPortbody = new Port();
        dstPortbody.withPortType(-1);
        List<IpsContent> listbodyContents = new ArrayList<>();
        listbodyContents.add(
            new IpsContent()
                .withContent("ab")
                .withDepth(65535)
                .withIsHex(false)
                .withIsIgnore(false)
                .withIsUri(false)
                .withOffset(0)
                .withRelativePosition(0)
        );
        body.withSrcPort(srcPortbody);
        body.withSoftware(0);
        body.withSeverity(0);
        body.withProtocol(9);
        body.withIpsName("test_操作系统linux");
        body.withFwInstanceId("d6e0a4d0-8b16-47fe-98b6-e1b0a7a14788");
        body.withDstPort(dstPortbody);
        body.withDirection(-1);
        body.withContents(listbodyContents);
        body.withAttackType(1);
        body.withAffectedOs(CustomerIpsSaveDto.AffectedOsEnum.NUMBER_1);
        body.withActionType(CustomerIpsSaveDto.ActionTypeEnum.NUMBER_1);
        request.withBody(body);
        try {
            UpdateCustomerIpsResponse response = client.updateCustomerIps(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3d84b4755b39476997c9f7b5a95c25ed, ips_cfw_id为cd74e74a-310f-4558-a4bf-da4215c820f8的自定义IPS规则

```
# 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 = UpdateCustomerIpsRequest()
        request.ips_cfw_id = "{ips_cfw_id}"
        srcPortbody = Port(
            port_type=-1
        )
        dstPortbody = Port(
            port_type=-1
        )
        listContentsbody = [
            IpsContent(
                content="ab",
                depth=65535,
                is_hex=False,
                is_ignore=False,
                is_uri=False,
                offset=0,
                relative_position=0
            )
        ]
        request.body = CustomerIpsSaveDto(
            src_port=srcPortbody,
            software=0,
            severity=0,
            protocol=9,
```

```
ips_name="test_操作系统linux",
fw_instance_id="d6e0a4d0-8b16-47fe-98b6-e1b0a7a14788",
dst_port=dstPortbody,
direction=-1,
contents=listContentsbody,
attack_type=1,
affected_os=1,
action_type=1
)
response = client.update_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

更新项目id为3d84b4755b39476997c9f7b5a95c25ed，ips_cfw_id为cd74e74a-310f-4558-a4bf-da4215c820f8的自定义IPS规则

```
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.UpdateCustomerIpsRequest{}
    request.IpsCfwId = "{ips_cfw_id}"
    portTypeSrcPort := int32(-1)
    srcPortbody := &model.Port{
        PortType: &portTypeSrcPort,
    }
    portTypeDstPort := int32(-1)
    dstPortbody := &model.Port{
        PortType: &portTypeDstPort,
    }
    contentContents := "ab"
    depthContents := int32(65535)
    isHexContents := false
    isIgnoreContents := false
    isUriContents := false
```



```
offsetContents:= int32(0)
relativePositionContents:= int32(0)
var listContentsbody = []model.IpsContent{
    {
        Content: &contentContents,
        Depth: &depthContents,
        IsHex: &isHexContents,
        IsIgnore: &isIgnoreContents,
        IsUri: &isUriContents,
        Offset: &offsetContents,
        RelativePosition: &relativePositionContents,
    },
}
request.Body = &model.CustomerIpsSaveDto{
    SrcPort: srcPortbody,
    Software: int32(0),
    Severity: int32(0),
    Protocol: int32(9),
    IpsName: "test_操作系统linux",
    FwInstanceId: "d6e0a4d0-8b16-47fe-98b6-e1b0a7a14788",
    DstPort: dstPortbody,
    Direction: int32(-1),
    Contents: listContentsbody,
    AttackType: int32(1),
    AffectedOs: model.GetCustomerIpsSaveDtoAffectedOsEnum().E_1,
    ActionType: model.GetCustomerIpsSaveDtoActionTypeEnum().E_1,
}
response, err := client.UpdateCustomerIps(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

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 4-621 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-622 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 4-623 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 4-624 响应 Body 参数

参数	参数类型	描述
data	LogConfigDto object	日志配置数据

表 4-625 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
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"  
  }  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cfw.v1.region.CfwRegion;  
import com.huaweicloud.sdk.cfw.v1.*;  
import com.huaweicloud.sdk.cfw.v1.model.*;  
  
public class ListLogConfigSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        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();  
        ListLogConfigRequest request = new ListLogConfigRequest();  
        try {  
            ListLogConfigResponse response = client.listLogConfig(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        }
```

```
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.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 = ListLogConfigRequest()
        response = client.list_log_config(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    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.ListLogConfigRequest{}
response, err := client.ListLogConfig(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
```

更多

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

状态码

状态码	描述
200	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 4-626 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-627 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 4-628 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 4-629 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

参数	是否必选	参数类型	描述
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 4-630 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.withLtsLogGroupld("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.getErrMsg());
}
}
```

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.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 = 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 4-631 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-632 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 4-633 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 4-634 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 4-635 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 查询防火墙日志

功能介绍

查询防火墙日志

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/cfw/{fw_instance_id}/logs

表 4-636 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-637 请求 Body 参数

参数	是否必选	参数类型	描述
filters	否	Array of Filter objects	参数解释： 过滤条件 约束限制： 不涉及 取值范围： 1-1024 默认取值： 不涉及
limit	是	Integer	参数解释： 每页显示个数 约束限制： 不涉及 取值范围： 1-1024 默认取值： 不涉及
offset	否	Integer	参数解释： 偏移量 约束限制： 第一页为空，其他页不为空 取值范围： 相对于上一页的偏移量 默认取值： 不涉及
log_id	否	String	参数解释： 文档ID 约束限制： 第一页为空，其他页不为空 取值范围： 上一次查询最后一条数据的 log_id 默认取值： 不涉及

参数	是否必选	参数类型	描述
next_date	否	Long	参数解释： 下个日期 约束限制： 第一页为空，其他页不为空 取值范围： 查询流量日志时为上一次查询最后一条数据的end_time 查询访问控制日志时为上一次查询最后一条数据的hit_time 查询访问控制日志时为上一次查询最后一条数据的event_time 查询URL日志时为上一次查询最后一条数据的hit_time 默认取值： 不涉及
start_time	是	Long	参数解释： 开始时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
end_time	是	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及

参数	是否必选	参数类型	描述
log_type	是	String	参数解释： 日志类型 约束限制： 不涉及 取值范围： internet为南北向日志、nat为nat场景日志，vpc为东西向日志，vgw为vgw场景日志 默认取值： 不涉及
type	是	String	参数解释： 日志类型 约束限制： 不涉及 取值范围： attack为攻击日志、acl 访问控制日志，flow 流量日志，url url 日志 默认取值： 不涉及

表 4-638 Filter

参数	是否必选	参数类型	描述
field	是	String	参数解释： 日志字段，如src_ip 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
values	否	Array of strings	参数解释： 值 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
operator	是	String	参数解释： 操作符 约束限制： 不涉及 取值范围： equal 等于 not_equal 不等于 contain 包含 starts_with 以开始 默认取值： 不涉及

响应参数

状态码：200

表 4-639 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 日志 取值范围： 不涉及

表 4-640 data

参数	参数类型	描述
limit	Integer	参数解释： 条数 取值范围： 不涉及
records	Array of LogVO objects	参数解释： 日志 取值范围： 不涉及
total	Long	参数解释： 记录总数 取值范围： 不涉及

表 4-641 LogVO

参数	参数类型	描述
app	String	参数解释： 应用 取值范围： 不涉及
bytes	Double	参数解释： 流字节数，流量日志字段 取值范围： 不涉及
direction	String	参数解释： 会话方向 取值范围： out2in 外到内访问 in2out 内到外访问
dst_host	String	参数解释： 流字节数，访问控制日志和流量日志中存在 取值范围： 目的网址

参数	参数类型	描述
dst_ip	String	参数解释： 目的IP 取值范围： 不涉及
dst_port	Integer	参数解释： 目的端口 取值范围： 不涉及
end_time	Long	参数解释： 会话结束时间，流量日志字段 取值范围： 不涉及
log_id	String	参数解释： 日志ID，用于分页查询 取值范围： 不涉及
packets	Double	参数解释： 流包数，流量日志字段 取值范围： 不涉及
protocol	String	参数解释： 协议 取值范围： 不涉及
src_ip	String	参数解释： 源Ip 取值范围： 不涉及
src_port	Integer	参数解释： 源端口 取值范围： 不涉及
start_time	Long	参数解释： 会话开始时间，流量日志字段 取值范围： 不涉及

参数	参数类型	描述
dst_region_id	AnyType	参数解释： 目的地区ID 取值范围： 不涉及
dst_region_name	String	参数解释： 目的地区名称 取值范围： 不涉及
dst_province_id	String	参数解释： 目的省份ID 取值范围： 不涉及
dst_province_name	String	参数解释： 目的省份名称 取值范围： 不涉及
dst_city_id	String	参数解释： 目的城市ID 取值范围： 不涉及
dst_city_name	String	参数解释： 目的城市名称 取值范围： 不涉及
src_region_id	String	参数解释： 源地区ID 取值范围： 不涉及
src_region_name	String	参数解释： 源地区名称 取值范围： 不涉及
src_province_id	String	参数解释： 源省份ID 取值范围： 不涉及

参数	参数类型	描述
src_province_name	String	参数解释： 源省份名称 取值范围： 不涉及
src_city_id	String	参数解释： 源城市ID 取值范围： 不涉及
src_city_name	String	参数解释： 源城市名称 取值范围： 不涉及
vgw_id	String	参数解释： 虚拟网关ID 取值范围： 不涉及
sctp_verification_tag	Long	参数解释： sctp验证标签，流量日志字段 取值范围： 不涉及
sctp_is_handshake_flow	String	参数解释： sctp握手流，流量日志字段 取值范围： 不涉及
qos_rule_id	String	参数解释： qos规则ID，流量日志、访问控制日志存在 取值范围： 不涉及
qos_rule_name	String	参数解释： qos规则名称，流量日志、访问控制日志存在 取值范围： 不涉及

参数	参数类型	描述
qos_channel_id	String	参数解释： qos通道ID，流量日志字段 取值范围： 不涉及
qos_channel_name	String	参数解释： qos通道名称，流量日志字段 取值范围： 不涉及
qos_drop_packets	Double	参数解释： qos丢包数，流量日志字段 取值范围： 不涉及
qos_drop_bytes	Double	参数解释： qos丢弃字节，流量日志字段 取值范围： 不涉及
qos_rule_type	Integer	参数解释： qos规则类型，流量日志、访问控制日志存在 取值范围： 不涉及
qos_channel_type	Integer	参数解释： qos通道类型，流量日志字段 取值范围： 不涉及
action	String	参数解释： 动作，攻击日志、访问控制日志、URL日志存在 取值范围： 不涉及
url	String	参数解释： url，URL日志字段 取值范围： 不涉及

参数	参数类型	描述
hit_time	Long	参数解释： 命中时间，访问控制日志、URL日志存在 取值范围： 不涉及
rule_id	String	参数解释： 规则ID，访问控制日志、URL日志存在 取值范围： 不涉及
rule_name	String	参数解释： 规则名称，访问控制日志、URL日志存在 取值范围： 不涉及
rule_type	Integer	参数解释： 规则类型，访问控制日志和URL日志字段 取值范围： 不涉及
attack_rule	String	参数解释： 规则类型，攻击日志字段 取值范围： 不涉及
attack_rule_id	String	参数解释： 攻击规则ID，攻击日志字段 取值范围： 不涉及
attack_type	String	参数解释： 攻击类型，攻击日志字段 取值范围： 不涉及
event_time	Long	参数解释： 事件时间，攻击日志字段 取值范围： 不涉及
level	String	参数解释： 攻击等级，攻击日志字段 取值范围： 不涉及

参数	参数类型	描述
packet	String	参数解释： 规则载荷，攻击日志字段 取值范围： 不涉及
source	String	参数解释： 攻击来源，攻击日志字段 取值范围： 不涉及
real_ip	String	参数解释： 真实IP，攻击日志字段 取值范围： 不涉及
tag	Integer	参数解释： 标签类型，攻击日志字段 取值范围： 1 WAF回源IP

状态码：400

表 4-642 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

项目ID为a16df7cf1d094befa6bbc72cbf51e93a，防火墙ID为fcd04edd-428a-4631-bef5-46a924293cca，时间范围为1751952647737到1751963447737，查询数量1000条，过滤条件为目的IP为方向为100.85.219.117，查询互联网边界的流量日志

```
https://{Endpoint}/v1/a16df7cf1d094befa6bbc72cbf51e93a/cfw/fcd04edd-428a-4631-bef5-46a924293cca/logs
```

```
{
  "limit" : 1000,
  "filters" : [ {
    "field" : "dst_ip",
    "operator" : "equal",
    "values" : [ "100.85.219.117" ]
  } ],
  "start_time" : 1751952647737,
  "end_time" : 1751963447737,
  "log_type" : "internet",
  "type" : "flow"
}
```

响应示例

状态码：200

OK

```
{
  "data" : {
    "limit" : 1000,
    "records" : [ {
      "app" : "DNS",
      "bytes" : 87,
      "direction" : "in2out",
      "dst_ip" : "100.85.219.117",
      "dst_port" : 53,
      "end_time" : 1751963431000,
      "log_id" : "73861",
      "packets" : 1,
      "protocol" : "UDP",
      "src_ip" : "100.93.2.30",
      "src_port" : 55637,
      "start_time" : 1751963369000
    } ],
    "total" : 1
  }
}
```

状态码：400

Bad Request

```
{
  "error_code" : "CFW.00200003",
  "error_msg" : "参数错误"
}
```

SDK 代码示例

SDK代码示例如下。

Java

项目ID为a16df7cf1d094befa6bbc72cbf51e93a，防火墙ID为fcd04edd-428a-4631-bef5-46a924293cca，时间范围为1751952647737到1751963447737，查询数量1000条，过滤条件为目的IP为方向为100.85.219.117，查询互联网边界的流量日志

```
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 ListLogsSolution {

    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();
        ListLogsRequest request = new ListLogsRequest();
        request.withFwInstanceId("{fw_instance_id}");
        ListLogsRequestBody body = new ListLogsRequestBody();
        List<String> listFiltersValues = new ArrayList<>();
        listFiltersValues.add("100.85.219.117");
        List<Filter> listbodyFilters = new ArrayList<>();
        listbodyFilters.add(
            new Filter()
                .withField("dst_ip")
                .withValues(listFiltersValues)
                .withOperator(Filter.OperatorEnum.fromValue("equal"))
        );
        body.withType(ListLogsRequestBody.TypeEnum.fromValue("flow"));
        body.withLogType(ListLogsRequestBody.LogTypeEnum.fromValue("internet"));
        body.withEndTime(1751963447737L);
        body.withStartTime(1751952647737L);
        body.withLimit(1000);
        body.withFilters(listbodyFilters);
        request.withBody(body);
        try {
            ListLogsResponse response = client.listLogs(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a16df7cf1d094befa6bbc72cbf51e93a，防火墙ID为fcd04edd-428a-4631-bef5-46a924293cca，时间范围为1751952647737到1751963447737，查询数量1000条，过滤条件为目的IP为方向为100.85.219.117，查询互联网边界的流量日志

```
# 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 = ListLogsRequest()
        request.fw_instance_id = "{fw_instance_id}"
        listValuesFilters = [
            "100.85.219.117"
        ]
        listFiltersbody = [
            Filter(
                field="dst_ip",
                values=listValuesFilters,
                operator="equal"
            )
        ]
        request.body = ListLogsRequestBody(
            type="flow",
            log_type="internet",
            end_time=1751963447737,
            start_time=1751952647737,
            limit=1000,
            filters=listFiltersbody
        )
        response = client.list_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

项目ID为a16df7cf1d094befa6bbc72cbf51e93a，防火墙ID为fcd04edd-428a-4631-bef5-46a924293cca，时间范围为1751952647737到1751963447737，查询数量1000条，过滤条件为目的IP为方向为100.85.219.117，查询互联网边界的流量日志

```
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.ListLogsRequest{}
    request.FwInstanceId = "{fw_instance_id}"
    var listValuesFilters = []string{
        "100.85.219.117",
    }
    var listFiltersbody = []model.Filter{
        {
            Field: "dst_ip",
            Values: &listValuesFilters,
            Operator: model.GetFilterOperatorEnum().EQUAL,
        },
    }
    request.Body = &model.ListLogsRequestBody{
        Type: model.GetListLogsRequestBodyTypeEnum().FLOW,
        LogType: model.GetListLogsRequestBodyLogTypeEnum().INTERNET,
        EndTime: int64(1751963447737),
        StartTime: int64(1751952647737),
        Limit: int32(1000),
        Filters: &listFiltersbody,
    }
    response, err := client.ListLogs(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

错误码

请参见[错误码](#)。

4.9.5 导出防火墙日志

功能介绍

导出防火墙日志

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/cfw/{fw_instance_id}/logs/export

表 4-643 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-644 请求 Body 参数

参数	是否必选	参数类型	描述
filters	否	Array of Filter objects	参数解释： 过滤条件 约束限制： 不涉及 取值范围： 1-1024 默认取值： 不涉及
start_time	是	Long	参数解释： 开始时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及

参数	是否必选	参数类型	描述
end_time	是	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
log_type	是	String	参数解释： 日志类型 约束限制： 不涉及 取值范围： internet为南北向日志、nat为nat场景日志，vpc为东西向日志，vgw为vgw场景日志 默认取值： 不涉及
type	是	String	参数解释： 日志类型 约束限制： 不涉及 取值范围： attack为攻击日志、acl 访问控制日志，flow 流量日志，url url 日志 默认取值： 不涉及
time_zone	否	String	参数解释： 时区 约束限制： 不涉及 取值范围： "GMT+08:00" 默认取值： 不涉及

表 4-645 Filter

参数	是否必选	参数类型	描述
field	是	String	参数解释： 日志字段，如src_ip 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
values	否	Array of strings	参数解释： 值 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
operator	是	String	参数解释： 操作符 约束限制： 不涉及 取值范围： equal 等于 not_equal 不等于 contain 包含 starts_with 以开始 默认取值： 不涉及

响应参数

状态码：200

表 4-646 响应 Body 参数

参数	参数类型	描述
-	File	

状态码：400

表 4-647 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

项目ID为a16df7cf1d094befa6bbc72cbf51e93a，防火墙ID为fcd04edd-428a-4631-bef5-46a924293cca，时间范围为1751958412875到1751969212875，过滤条件为目的IP为方向为100.85.219.117，目的端口为55637，导出互联网边界的流量日志

```
https://{Endpoint}/v1/a16df7cf1d094befa6bbc72cbf51e93a/cfw/fcd04edd-428a-4631-bef5-46a924293cca/logs/export
{
  "filters": [ {
    "field": "dst_ip",
    "operator": "equal",
    "values": [ "100.85.219.117" ]
  }, {
    "field": "src_port",
    "operator": "equal",
    "values": [ "55637" ]
  } ],
  "start_time": 1751958412875,
  "end_time": 1751969212875,
  "type": "flow",
  "log_type": "internet",
  "time_zone": "GMT%2B08:00"
}
```

响应示例

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

SDK 代码示例

SDK代码示例如下。

Java

项目ID为a16df7cf1d094befa6bbc72cbf51e93a，防火墙ID为fcd04edd-428a-4631-bef5-46a924293cca，时间范围为1751958412875到1751969212875，过滤条件为目的IP为方向为100.85.219.117，目的端口为55637，导出互联网边界的流量日志

```
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 ExportLogsSolution {

    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();
        ExportLogsRequest request = new ExportLogsRequest();
        request.withFwInstanceId("{fw_instance_id}");
        ExportLogsRequestBody body = new ExportLogsRequestBody();
        List<String> listFiltersValues = new ArrayList<>();
        listFiltersValues.add("55637");
        List<String> listFiltersValues1 = new ArrayList<>();
        listFiltersValues1.add("100.85.219.117");
        List<Filter> listbodyFilters = new ArrayList<>();
        listbodyFilters.add(
            new Filter()
                .withField("dst_ip")
                .withValues(listFiltersValues1)
                .withOperator(Filter.OperatorEnum.fromValue("equal"))
        );
        listbodyFilters.add(
            new Filter()
                .withField("src_port")
                .withValues(listFiltersValues)
                .withOperator(Filter.OperatorEnum.fromValue("equal"))
        );
        body.withTimeZone("GMT%2B08:00");
        body.withType(ExportLogsRequestBody.TypeEnum.fromValue("flow"));
        body.withLogType(ExportLogsRequestBody.LogTypeEnum.fromValue("internet"));
        body.withEndTime(1751969212875L);
        body.withStartTime(1751958412875L);
        body.withFilters(listbodyFilters);
        request.withBody(body);
        try {
```

```
        ExportLogsResponse response = client.exportLogs(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a16df7cf1d094bfa6bbc72cbf51e93a，防火墙ID为fcd04edd-428a-4631-bef5-46a924293cca，时间范围为1751958412875到1751969212875，过滤条件为目的IP为方向为100.85.219.117，目的端口为55637，导出互联网边界的流量日志

```
# 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 = ExportLogsRequest()
        request.fw_instance_id = "{fw_instance_id}"
        listValuesFilters = [
            "55637"
        ]
        listValuesFilters1 = [
            "100.85.219.117"
        ]
        listFiltersbody = [
            Filter(
                field="dst_ip",
                values=listValuesFilters1,
                operator="equal"
            ),
            Filter(
                field="src_port",
                values=listValuesFilters,
                operator="equal"
            )
        ]
```



```
request.body = ExportLogsRequestBody(  
    time_zone="GMT%2B08:00",  
    type="flow",  
    log_type="internet",  
    end_time=1751969212875,  
    start_time=1751958412875,  
    filters=listFiltersbody  
)  
response = client.export_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

项目ID为a16df7cf1d094bfa6bbc72cbf51e93a，防火墙ID为fcd04edd-428a-4631-bef5-46a924293cca，时间范围为1751958412875到1751969212875，过滤条件为目的IP为方向为100.85.219.117，目的端口为55637，导出互联网边界的流量日志

```
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.ExportLogsRequest{  
        request.FwInstanceId = "{fw_instance_id}"  
        var listValuesFilters = []string{  
            "55637",  
        }  
        var listValuesFilters1 = []string{  
            "100.85.219.117",  
        }  
        var listFiltersbody = []model.Filter{  
            {  
                Field: "dst_ip",  
                Values: &listValuesFilters1,  
                Operator: model.GetFilterOperatorEnum().EQUAL,  
            },  
        },  
    }
```

```
{
    Field: "src_port",
    Values: &listValuesFilters,
    Operator: model.GetFilterOperatorEnum().EQUAL,
},
},
timeZoneExportLogsRequestBody:= "GMT%2B08:00"
request.Body = &model.ExportLogsRequestBody{
    TimeZone: &timeZoneExportLogsRequestBody,
    Type: model.GetExportLogsRequestBodyTypeEnum().FLOW,
    LogType: model.GetExportLogsRequestBodyLogTypeEnum().INTERNET,
    EndTime: int64(1751969212875),
    StartTime: int64(1751958412875),
    Filters: &listFiltersbody,
}
response, err := client.ExportLogs(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

错误码

请参见[错误码](#)。

4.9.6 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 4-648 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-649 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	参数解释： 日志类型 约束限制： 不涉及 取值范围： internet为南北向日志、nat为nat场景日志，vpc为东西向日志，vgw为vgw场景日志 默认取值： 不涉及
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 4-650 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 4-651 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 4-652 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 4-653 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 4-654 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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, 错误的请求

```
{
  "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.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.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，错误的请求
401	Unauthorized，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到
500	Internal Server Error

错误码

请参见[错误码](#)。

4.9.7 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 4-655 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-656 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
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，首页时为空，非首页时不为空
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
log_type	否	String	参数解释： 日志类型 约束限制： 不涉及 取值范围： internet为南北向日志、nat为nat场景日志，vpc为东西向日志，vgw为vgw场景日志 默认取值： 不涉及
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 4-657 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 4-658 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 4-659 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 4-660 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

参数	参数类型	描述
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
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 4-661 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释: 错误码 取值范围: 不涉及
error_msg	String	参数解释: 错误描述 取值范围: 不涉及

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, 错误的请求

```
{
  "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.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 = 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, 错误的请求
401	Unauthorized, 错误的请求
403	Forbidden, 禁止访问
404	Not Found, 网页未找到
500	Internal Server Error

错误码

请参见[错误码](#)。

4.9.8 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 4-662 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID, 可以通过调用API获取, 也可以从控制台获取。 项目ID获取方式

表 4-663 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

参数	是否必选	参数类型	描述
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
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

参数	是否必选	参数类型	描述
attack_type	否	String	参数解释： 入侵事件类型 约束限制： 不涉及 取值范围： Access Control：访问控制 Vulnerability scanning：漏洞扫描 Email attack：邮件攻击 Vulnerability Attack：漏洞攻击 Web attack：Web攻击 password attack：密码攻击 Hijacking attack：劫持攻击 Protocol exception：协议异常 Trojan horse：特洛伊木马 worms：蠕虫 Buffer Overflow：缓冲区溢出 Hacking tools：黑客工具 Spyware：间谍软件 DDoS flooding：DDoS泛洪 Application-layer DDoS attacks：应用层DDoS攻击 Other suspicious behavior：其他可疑行为 Suspicious DNS activity：可疑DNS活动 Phishing：网络钓鱼 Spam：垃圾邮件 Others：其他攻击 默认取值： 不涉及
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

参数	是否必选	参数类型	描述
dst_host	否	String	目标主机
log_type	否	String	参数解释： 日志类型 约束限制： 不涉及 取值范围： internet为南北向日志、nat为nat场景日志，vpc为东西向日志，vgw为vgw场景日志 默认取值： 不涉及
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 4-664 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 4-665 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 4-666 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 4-667 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

参数	参数类型	描述
attack_type	String	参数解释： 入侵事件类型 约束限制： 不涉及 取值范围： Access Control：访问控制 Vulnerability scanning：漏洞扫描 Email attack：邮件攻击 Vulnerability Attack：漏洞攻击 Web attack：Web攻击 password attack：密码攻击 Hijacking attack：劫持攻击 Protocol exception：协议异常 Trojan horse：特洛伊木马 worms：蠕虫 Buffer Overflow：缓冲区溢出 Hacking tools：黑客工具 Spyware：间谍软件 DDoS flooding：DDoS泛洪 Application-layer DDoS attacks：应用层DDoS攻击 Other suspicious behavior：其他可疑行为 Suspicious DNS activity：可疑DNS活动 Phishing：网络钓鱼 Spam：垃圾邮件 Others：其他攻击 默认取值： 不涉及
attack_rule	String	攻击规则
level	String	威胁等级，包括CRITICAL、HIGH、MEDIUM、LOW
source	String	来源
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

参数	参数类型	描述
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
dst_city_name	String	目的城市名称

表 4-668 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 4-669 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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" : "+hZUzMhV+hY/AaHMCABFKABpXPNAADAGof1kVe6QZF
+UMcTQH5B0wdaz888+uoAYAOVyNQAAAQEICjrmikVb9JLCR0VUIC9uaWNUJtlwcG9ydHMIkMvVHJpJTZFaX
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..\u001F.t.>..."
}, {
  "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, 错误的请求

```
{
  "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.cfww.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfww.v1.*;
import com.huaweicloud.sdk.cfww.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.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 huaweicloudsdkcfww.v1.region.cfww_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfww.v1 import *

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

```
ak = 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, 错误的请求
401	Unauthorized, 错误的请求
403	Forbidden, 禁止访问
404	Not Found, 网页未找到
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 4-670 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-671 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 4-672 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 4-673 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 4-674 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 4-675 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.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 = 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 4-676 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-677 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 4-678 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 4-679 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 4-680 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 4-681 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 4-682 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 4-683 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 4-684 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-685 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 4-686 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 4-687 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 4-688 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 4-689 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.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrMsg());
    }
}
```

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 4-690 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-691 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 4-692 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 4-693 响应 Body 参数

参数	参数类型	描述
data	CaptureResultUrlVO object	抓包结果url

表 4-694 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 4-695 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 4-696 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.cfww.v1.region.CfwRegion;
import com.huaweicloud.sdk.cfww.v1.*;
import com.huaweicloud.sdk.cfww.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.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 huaweicloudsdkcfww.v1.region.cfww_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfww.v1 import *

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

```
ak = 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 4-697 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-698 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 4-699 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 4-700 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 4-701 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 4-702 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();
        }
    }
}
```

```
        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 4-703 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。可通过 项目ID获取方式 获取

表 4-704 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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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 4-705 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 4-706 响应 Body 参数

参数	参数类型	描述
data	AntiVirusVO object	反病毒开关数据

表 4-707 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，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到

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 4-708 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。可通过 项目ID获取方式 获取

表 4-709 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 4-710 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 4-711 请求 Body 参数

参数	是否必选	参数类型	描述
anti_virus_status	是	Integer	参数解释： 反病毒开关状态 约束限制： 不涉及 取值范围： 0：开启 1：关闭 默认取值： 不涉及
object_id	是	String	防护对象ID

响应参数

状态码：200

表 4-712 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 4-713 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，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到

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 4-714 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。可通过 项目ID获取方式 获取

表 4-715 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，type为0时，object_id为互联网边界防护对象ID，type为1时，object_id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
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0，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
engine_type	否	Integer	废弃字段

请求参数

表 4-716 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 4-717 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 4-718 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 4-719 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();  
        }  
    }  
}
```

```
        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，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到

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 4-720 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。可通过 项目ID获取方式 获取

表 4-721 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 4-722 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 4-723 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
scan_protocol_configs	是	Array of ScanProtocolConfig objects	扫描协议配置列表

表 4-724 ScanProtocolConfig

参数	是否必选	参数类型	描述
action	是	Integer	反病毒动作，0：观察 1：拦截 2：禁用

参数	是否必选	参数类型	描述
protocol_type	是	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 4-725 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 4-726 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 listScanProtocolConfigsboby = []model.ScanProtocolConfig{
        {
            Action: &actionScanProtocolConfigs,
            ProtocolType: &protocolTypeScanProtocolConfigs,
        },
    }
    objectIdAntiVirusRuleDto:= "1b90f031-0c7b-4f25-95e2-b6d9940d269e"
    request.Body = &model.AntiVirusRuleDto{
        ScanProtocolConfigs: &listScanProtocolConfigsboby,
        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，错误的请求
403	Forbidden，禁止访问

状态码	描述
404	Not Found，网页未找到

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 4-727 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。可通过 项目ID获取方式 获取

表 4-728 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 4-729 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 4-730 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 4-731 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.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 = 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，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到

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 4-732 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。可通过 项目ID获取方式 获取

表 4-733 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 4-734 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 4-735 请求 Body 参数

参数	是否必选	参数类型	描述
alarm_time_period	是	Integer	告警周期，0：全天，1：8时到22时
alarm_type	是	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：异常外联告警

参数	是否必选	参数类型	描述
enable_status	是	Integer	告警状态 0：失效; 1：生效
frequency_count	是	Integer	告警触发频次
frequency_time	是	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	是	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1时，severity为2；当type为2时，severity固定为3
topic_urn	是	String	告警urn
username	是	String	用户名称，为cfw

响应参数

状态码：200

表 4-736 响应 Body 参数

参数	参数类型	描述
data	ResponseData object	通用返回体

表 4-737 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",
  "username" : "cfw"
}
```

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.withUsername("cfw");
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.getErrMsg());
}
}
```

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()
```

```
        username="cfw",
        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{
        usernameUpdateAttackLogAlarmConfigDto:= "cfw"
        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{
```

```
Username: &usernameUpdateAttackLogAlarmConfigDto,
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，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到

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 4-738 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。可通过 项目ID获取方式 获取

表 4-739 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 4-740 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 4-741 响应 Body 参数

参数	参数类型	描述
tags	Array of TagValue objects	tag列表

表 4-742 TagValue

参数	参数类型	描述
key	String	参数解释： 标签键 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
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.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 = 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
401	Unauthorized，错误的请求
403	Forbidden，禁止访问

状态码	描述
404	Not Found，网页未找到

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 4-743 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，可以通过调用API获取，也可以从控制台获取。可通过 项目ID获取方式 获取

表 4-744 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 4-745 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 4-746 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 4-747 ResourceTag

参数	参数类型	描述
key	String	参数解释： 标签键 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
value	String	参数解释： 标签值 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	参数类型	描述
update_time	String	参数解释： 标签更新时间 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

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.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 = 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，错误的请求
403	Forbidden，禁止访问
404	Not Found，网页未找到

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 4-748 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，可以通过调用API获取，也可以从控制台获取。可通过 项目ID获取方式 获取

表 4-749 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 4-750 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 4-751 请求 Body 参数

参数	是否必选	参数类型	描述
tags	否	Array of CreateTag objects	参数解释： 创建防火墙标签列表 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-752 CreateTag

参数	是否必选	参数类型	描述
key	否	String	参数解释： 标签键 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
value	否	String	参数解释： 标签值 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-753 响应 Body 参数

参数	参数类型	描述
data	String	

状态码：201

Created

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.getHttpStatusCode());
            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

状态码	描述
201	Created
401	Unauthorized, 错误的请求
403	Forbidden, 禁止访问
404	Not Found, 网页未找到

错误码

请参见[错误码](#)。

4.14 VPC 间防火墙管理

4.14.1 更新 VPC 间防火墙防护状态

功能介绍

更新VPC间防火墙防护状态

调用方法

请参见[如何调用API](#)。

URI

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

表 4-754 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-755 请求 Body 参数

参数	是否必选	参数类型	描述
object_id	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
status	是	Integer	参数解释： VPC边界防护状态 约束限制： 不涉及 取值范围： 0- 开启防护、1- 关闭防护 默认取值： 不涉及

响应参数

状态码：200

表 4-756 响应 Body 参数

参数	参数类型	描述
data	ChangeEastWestFirewallStatusResponseData object	

表 4-757 ChangeEastWestFirewallStatusResponseData

参数	参数类型	描述
id	String	参数解释： VPC边界防护对象id 取值范围： 不涉及

状态码：400

表 4-758 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

项目ID为a16df7cf1d094befa6bbc72cbf51e93a，防火墙ID为af050cf8-8ad8-4c12-a027-d7a5784e0f64，VPC边界防护对象ID为37ccee40-a967-4eb7-bd38-cbe1cddc787b，VPC间防火墙开启防护

```
https://{Endpoint}/v1/a16df7cf1d094befa6bbc72cbf51e93a/firewall/east-west/protect
{
  "object_id" : "37ccee40-a967-4eb7-bd38-cbe1cddc787b",
  "status" : 0
}
```

响应示例

状态码：200

OK

```
{
  "data" : {
    "id" : "000a598a-9470-4643-9c8b-8a45dadb462e"
  }
}
```

状态码：400

Bad Request

```
{
  "error_code" : "CFW.00200003",
  "error_msg" : "参数错误"
}
```

SDK 代码示例

SDK代码示例如下。

Java

项目ID为a16df7cf1d094befa6bbc72cbf51e93a，防火墙ID为af050cf8-8ad8-4c12-a027-d7a5784e0f64，VPC边界防护对象ID为37ccee40-a967-4eb7-bd38-cbe1cddc787b，VPC间防火墙开启防护

```
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.*;

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_0);
        body.withObjectId("37ccee40-a967-4eb7-bd38-cbe1cddc787b");
        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

项目ID为a16df7cf1d094befa6bbc72cbf51e93a，防火墙ID为af050cf8-8ad8-4c12-a027-d7a5784e0f64，VPC边界防护对象ID为37ccee40-a967-4eb7-bd38-cbe1cddc787b，VPC间防火墙开启防护

```
# 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 = ChangeEastWestFirewallStatusRequest()
        request.body = ChangeProtectStatusRequestBody(
            status=0,
            object_id="37ccee40-a967-4eb7-bd38-cbe1cddc787b"
        )
        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

项目ID为a16df7cf1d094befa6bbc72cbf51e93a，防火墙ID为af050cf8-8ad8-4c12-a027-d7a5784e0f64，VPC边界防护对象ID为37ccee40-a967-4eb7-bd38-cbe1cddc787b，VPC间防火墙开启防护

```
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.ChangeEastWestFirewallStatusRequest{}
    request.Body = &model.ChangeProtectStatusRequestBody{
        Status: model.GetChangeProtectStatusRequestBodyStatusEnum().E_0,
        ObjectID: "37ccee40-a967-4eb7-bd38-cbe1cddc787b",
    }
    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	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.14.2 查询 VPC 边界防火墙使用的引流 VPC 信息

功能介绍

查询VPC边界防火墙使用的引流VPC信息

调用方法

请参见[如何调用API](#)。

URI

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

表 4-759 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-760 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-761 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 查询VPC间防火墙使用引流VPC返回体 取值范围： 不涉及

表 4-762 data

参数	参数类型	描述
inspection_vpc_list	Array of InspectionVpcInfo objects	参数解释： 使用引流VPC列表 取值范围： 不涉及

表 4-763 InspectionVpcInfo

参数	参数类型	描述
inspection_vpc_id	String	参数解释： 引流VPC 取值范围： 不涉及
name	String	参数解释： 引流VPC 取值范围： 不涉及

状态码：400

表 4-764 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

项目ID为a16df7cf1d094befa6bbc72cbf51e93a，防火墙ID为af050cf8-8ad8-4c12-a027-d7a5784e0f64，查询VPC间防火墙使用的引流VPC信息

https://{Endpoint}/v1/a16df7cf1d094befa6bbc72cbf51e93a/firewall/east-west/inspection-vpc?
fw_instance_id=af050cf8-8ad8-4c12-a027-d7a5784e0f64

响应示例

状态码：200

OK

```
{
  "data": {
    "inspection_vpc_list": [ {
      "inspection_vpc_id": "e76c22b9-0f97-4113-92e3-963846b5dd03",
      "name": "vpc-test-1"
    } ]
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

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 ShowEwAssociatedVpcSolution {

    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();
        ShowEwAssociatedVpcRequest request = new ShowEwAssociatedVpcRequest();
        try {
            ShowEwAssociatedVpcResponse response = client.showEwAssociatedVpc(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 = ShowEwAssociatedVpcRequest()
    response = client.show_ew_associated_vpc(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.ShowEwAssociatedVpcRequest{}
    response, err := client.ShowEwAssociatedVpc(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

错误码

请参见[错误码](#)。

4.14.3 查询 VPC 间防火墙使用的企业路由器信息

功能介绍

查询VPC间防火墙使用的企业路由器信息

调用方法

请参见[如何调用API](#)。

URI

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

表 4-765 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-766 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-767 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 查询VPC间防火墙使用的企业路由器信息返回体 取值范围： 不涉及

表 4-768 data

参数	参数类型	描述
er_list	Array of ErInfo objects	参数解释： 企业路由器列表 取值范围： 不涉及

表 4-769 ErInfo

参数	参数类型	描述
er_id	String	参数解释： 企业路由器ID 取值范围： 不涉及
name	String	参数解释： 企业路由器名称 取值范围： 不涉及

状态码：400

表 4-770 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

无

响应示例

状态码：200

OK

```
{
  "data": {
    "er_list": [ {
      "er_id": "290d6c67-9ef6-40a3-a1b1-5048e6fd423c",
      "name": "test"
    } ]
  }
}
```

状态码：400

Bad Request

```
{
  "error_code" : "CFW.00200003",
  "error_msg" : "参数错误"
}
```

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 ShowEwAssociatedErSolution {

    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();
        ShowEwAssociatedErRequest request = new ShowEwAssociatedErRequest();
        try {
            ShowEwAssociatedErResponse response = client.showEwAssociatedEr(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 = ShowEwAssociatedErRequest()
        response = client.show_ew_associated_er(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.ShowEwAssociatedErRequest{}
    response, err := client.ShowEwAssociatedEr(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

错误码

请参见[错误码](#)。

4.14.4 获取私网网段的信息

功能介绍

东西向私网网段查询接口

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/firewall/{fw_instance_id}/east-west/private-network-segments

表 4-771 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-772 响应 Body 参数

参数	参数类型	描述
data	ListPrivateNetworkSegmentsVO object	参数解释： 私网网段信息列表 取值范围： 不涉及

表 4-773 ListPrivateNetworkSegmentsVO

参数	参数类型	描述
private_network_list	Array of PrivateNetworkSegmentVO objects	参数解释： 私网网段列表 取值范围： 不涉及
total	Integer	参数解释： 总条数 取值范围： 不涉及
quota	Integer	参数解释： 私网网段配额 取值范围： 不涉及
support_az_list	Array of strings	参数解释： 支持的可用区列表 取值范围： 不涉及

表 4-774 PrivateNetworkSegmentVO

参数	参数类型	描述
conf_id	String	参数解释： 私网网段ID，更新私网网段时需要 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	参数类型	描述
protect_network	String	参数解释： 私网网段，格式为：IP/Mask 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
az_info	String	参数解释： 私网网段的可用区信息，如AZ1 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
subnet_name	String	参数解释： 子网名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

状态码：400

表 4-775 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及

参数	参数类型	描述
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询东西向私网网段列表，项目ID为176d761dc156471c9c9b227376594160，防火墙ID为48e074e4-3b1a-448b-a17f-2fd2a12df1d2f

https://{Endpoint}v2/176d761dc156471c9c9b227376594160/firewall/48e074e4-3b1a-448b-a17f-2fd2a12df1d2/east-west/private-network-segments

响应示例

状态码：200

私网网段信息列表

```
{
  "data": {
    "private_network_list": [ {
      "conf_id": "4",
      "protect_network": "100.64.0.0/10"
    }, {
      "conf_id": "1",
      "protect_network": "10.0.0.0/8",
      "az_info": "AZ1",
      "subnet_name": "test"
    }, {
      "conf_id": "2",
      "protect_network": "172.16.0.0/12"
    }, {
      "conf_id": "3",
      "protect_network": "192.168.0.0/16"
    } ],
    "total": 4,
    "quota": 32,
    "support_az_list": [ "AZ1", "AZ3" ]
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00000008",
  "error_msg": "系统内部错误"
}
```

状态码

状态码	描述
200	私网网段信息列表

状态码	描述
400	Bad Request

错误码

请参见[错误码](#)。

4.14.5 更新私网网段

功能介绍

更新私网网段的REST接口，主要特性：

- 根据私网网段ID，更新对应的私网网段信息

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/firewall/{fw_instance_id}/east-west/private-network-segments

表 4-776 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-777 请求 Body 参数

参数	是否必选	参数类型	描述
conf_id	否	String	参数解释： 私网网段ID，更新私网网段时需要 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
protect_network	是	String	参数解释： 私网网段，格式为：IP/Mask 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
az_info	否	String	参数解释： 私网网段的可用区信息，如AZ1 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
subnet_name	否	String	参数解释： 子网名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-778 响应 Body 参数

参数	参数类型	描述
data	String	参数解释： 防火墙ID 取值范围： 不涉及

状态码：400

表 4-779 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及

参数	参数类型	描述
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

更新东西向私网网段，项目ID为176d761dc156471c9c9b227376594160，防火墙ID为48e074e4-3b1a-448b-a17f-2fd2a12df1d2f，更新配置ID为1，更新私网地址段为10.0.0.0/8

```
https://{Endpoint}v2/176d761dc156471c9c9b227376594160/firewall/48e074e4-3b1a-448b-a17f-2fd2a12df1d2/east-west/private-network-segments
{
  "az_info" : "AZ1",
  "protect_network" : "10.0.0.0/8",
  "subnet_name" : "test",
  "conf_id" : "1"
}
```

响应示例

状态码：200

更新成功，返回防火墙实例ID

```
{
  "data" : "48e074e4-3b1a-448b-a17f-2fd2a12df1d2"
}
```

状态码：400

Bad Request

```
{
  "error_code" : "CFW.00000008",
  "error_msg" : "系统内部错误"
}
```

状态码

状态码	描述
200	更新成功，返回防火墙实例ID
400	Bad Request

错误码

请参见[错误码](#)。

4.14.6 创建私网网段

功能介绍

添加私网网段的接口，添加私网网段后该网段的流量将引流到VPC防火墙防护

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/firewall/{fw_instance_id}/east-west/private-network-segments/batch-create

表 4-780 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-781 请求 Body 参数

参数	是否必选	参数类型	描述
private_network_segments	是	Array of PrivateNetworkSegmentVO objects	参数解释： 私网网段的信息列表，用于东西向VPC防护引流 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-782 PrivateNetworkSegmentVO

参数	是否必选	参数类型	描述
conf_id	否	String	参数解释： 私网网段ID，更新私网网段时需要 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
protect_network	是	String	参数解释： 私网网段，格式为：IP/Mask 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
az_info	否	String	参数解释： 私网网段的可用区信息，如AZ1 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
subnet_name	否	String	参数解释： 子网名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-783 响应 Body 参数

参数	参数类型	描述
data	String	参数解释： 防火墙ID 取值范围： 不涉及

状态码：400

表 4-784 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及

参数	参数类型	描述
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

创建东西向私网网段，项目ID为176d761dc156471c9c9b227376594160，防火墙ID为48e074e4-3b1a-448b-a17f-2fd2a12df1d2f，可用区为cn-north-7a，私网地址段为10.0.0.0/8

```
https://{Endpoint}v2/176d761dc156471c9c9b227376594160/firewall/48e074e4-3b1a-448b-a17f-2fd2a12df1d2/east-west/private-network-segments/batch-create
{
  "private_network_segments": [ {
    "protect_network": "10.0.0.0/8",
    "subnet_name": "test",
    "az_info": "cn-north-7a"
  } ]
}
```

响应示例

状态码：200

更新私网网段成功，返回防火墙实例ID

```
{
  "data": "48e074e4-3b1a-448b-a17f-2fd2a12df1d2"
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00000008",
  "error_msg": "系统内部错误"
}
```

状态码

状态码	描述
200	更新私网网段成功，返回防火墙实例ID
400	Bad Request

错误码

请参见[错误码](#)。

4.14.7 删除私网网段信息

功能介绍

删除私网网段的接口，根据用户输入的私网网段conf_id，删除对应ID的私网网段

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/firewall/{fw_instance_id}/east-west/private-network-segments/batch-delete

表 4-785 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-786 请求 Body 参数

参数	是否必选	参数类型	描述
conf_ids	是	Array of strings	参数解释： 删除的私网网段ID列表，可以通过调用[获取私网网段的信息]获得，通过返回值中的data.private_network_list.conf_id获得 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-787 响应 Body 参数

参数	参数类型	描述
data	String	参数解释： 防火墙ID 取值范围： 不涉及

状态码：400

表 4-788 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及

参数	参数类型	描述
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

删除东西向私网网段，项目ID为176d761dc156471c9c9b227376594160，防火墙ID为48e074e4-3b1a-448b-a17f-2fd2a12df1d2f，配置ID为b3316c64-31e7-4f23-b01e-1ed6900df53b

```
https://{Endpoint}v2/176d761dc156471c9c9b227376594160/firewall/48e074e4-3b1a-448b-a17f-2fd2a12df1d2/east-west/private-network-segments/batch-delete?object_id=000a598a-9470-4643-9c8b-8a45dadb462e

{
  "conf_ids": [ "b3316c64-31e7-4f23-b01e-1ed6900df53b" ]
}
```

响应示例

状态码：200

删除私网网段成功，返回防火墙示例ID

```
{
  "data": "48e074e4-3b1a-448b-a17f-2fd2a12df1d2"
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00000008",
  "error_msg": "系统内部错误"
}
```

状态码

状态码	描述
200	删除私网网段成功，返回防火墙示例ID
400	Bad Request

错误码

请参见[错误码](#)。

4.15 日志分析

4.15.1 查询流量趋势

功能介绍

查询流量趋势，包括南北向、EIP、东西向的流量趋势

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/cfw/logs/traffic-trend

表 4-789 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-790 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
range	否	Integer	参数解释： 时间范围 约束限制： 不涉及 取值范围： 0为近一时 1近一天 2近七天 默认取值： 不涉及
log_type	是	String	参数解释： 日志类型 约束限制： 不涉及 取值范围： internet为南北向日志、nat为nat场景日志，vpc为东西向日志，vgw为vgw场景日志 默认取值： 不涉及
start_time	否	Long	参数解释： 开始时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及

参数	是否必选	参数类型	描述
end_time	否	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
vgw_id	否	Array of strings	参数解释： VGW ID 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
agg_type	是	String	参数解释： 聚合类型 约束限制： 不涉及 取值范围： avg为平均值 max为最大值 默认取值： 不涉及
ip	否	Array of strings	参数解释： IP地址列表，查看IP的流量趋势 约束限制： 不涉及 取值范围： IP地址 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-791 响应 Body 参数

参数	参数类型	描述
data	Array of TrafficTrendVO objects	参数解释： 流量趋势数据 取值范围： 不涉及

表 4-792 TrafficTrendVO

参数	参数类型	描述
agg_time	Long	参数解释： 聚合时间点 取值范围： 不涉及
bps	Double	参数解释： bps 取值范围： 不涉及
in_bps	Double	参数解释： 入方向bps 取值范围： 不涉及
out_bps	Double	参数解释： 出方向bps 取值范围： 不涉及

状态码：400

表 4-793 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询项目ID为a16df7cf1d094befa6bbc72cbf51e93a防火墙ID为c598d612-8112-465c-bffd-863d4ce85a50时间范围为7天的vpc间流量趋势，计算方式为最大值

https://{Endpoint}/v1/a16df7cf1d094befa6bbc72cbf51e93a/cfw/logs/traffic-trend?
fw_instance_id=c598d612-8112-465c-bffd-863d4ce85a50&range=2&log_type=vpc&agg_type=max

响应示例

状态码：200

OK

```
{
  "data": [ {
    "agg_time": 1750299480000,
    "bps": 0
  } ]
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200007",
  "error_msg": "时间范围错误"
}
```

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 ShowTrafficTrendSolution {

    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();
        ShowTrafficTrendRequest request = new ShowTrafficTrendRequest();
        try {
            ShowTrafficTrendResponse response = client.showTrafficTrend(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 = ShowTrafficTrendRequest()
    response = client.show_traffic_trend(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.ShowTrafficTrendRequest{}
    response, err := client.ShowTrafficTrend(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

状态码	描述
400	Bad Request

错误码

请参见[错误码](#)。

4.15.2 查询访问日志统计信息

功能介绍

获取访问日志的TOP统计信息，如TOP命中规则等

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/cfw/logs/access-top

表 4-794 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-795 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
range	否	Integer	参数解释： 时间范围 约束限制： 不涉及 取值范围： 0为近一时 1近一天 2近七天 默认取值： 不涉及
direction	否	String	参数解释： 会话方向 约束限制： 不涉及 取值范围： in2out为出云方向 out2in为入云方向 默认取值： 不涉及
start_time	否	Long	参数解释： 开始时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及

参数	是否必选	参数类型	描述
end_time	否	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
vgw_id	否	Array of strings	参数解释： VGW ID 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
log_type	否	String	参数解释： 日志类型 约束限制： 不涉及 取值范围： internet为南北向日志、nat为nat场景日志，vpc为东西向日志，vgw为vgw场景日志 默认取值： 不涉及

参数	是否必选	参数类型	描述
item	是	String	参数解释： 聚合类型 约束限制： 不涉及 取值范围： strategy_hit_info 策略的命中趋势 strategy_dashboard 策略命中概览 top_deny_rule TOP命中拦截策略 dst_ip TOP拦截目的IP src_ip TOP拦截源IP dst_port TOP拦截端口 dst_region TOP拦截目的地区 src_region TOP拦截源地区 默认取值： 不涉及
rule_id	否	Array of strings	参数解释： 规则ID列表 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-796 响应 Body 参数

参数	参数类型	描述
data	AccessTopVO object	

表 4-797 AccessTopVO

参数	参数类型	描述
deny_count	Long	参数解释： 阻断次数 取值范围： 不涉及
deny_top_one_acl_id	String	参数解释： Top命中的阻断策略ID 取值范围： 不涉及
deny_top_one_acl_name	String	参数解释： Top命中的阻断策略名称 取值范围： 不涉及
hit_count	Long	参数解释： 命中次数 取值范围： 不涉及
in2out_deny_dst_ip_list	Array of AccessTopMemberVO objects	参数解释： TOP出云阻断目的IP列表 取值范围： 不涉及
in2out_deny_dst_port_list	Array of AccessTopMemberVO objects	参数解释： TOP出云阻断端口列表 取值范围： 不涉及
in2out_deny_dst_region_list	Array of AccessTopMemberVO objects	参数解释： TOP出云阻断目的地区列表 取值范围： 不涉及
in2out_deny_src_ip_list	Array of AccessTopMemberVO objects	参数解释： TOP出云阻断源IP列表 取值范围： 不涉及

参数	参数类型	描述
out2in_deny_dst_ip_list	Array of AccessTopMemberVO objects	参数解释： TOP入云阻断目的IP列表 取值范围： 不涉及
out2in_deny_dst_port_list	Array of AccessTopMemberVO objects	参数解释： TOP入云阻断目的端口列表 取值范围： 不涉及
out2in_deny_src_ip_list	Array of AccessTopMemberVO objects	参数解释： TOP入云阻断源IP列表 取值范围： 不涉及
out2in_deny_src_port_list	Array of AccessTopMemberVO objects	参数解释： TOP入云阻断源端口列表 取值范围： 不涉及
out2in_deny_src_region_list	Array of AccessTopMemberVO objects	参数解释： TOP入云阻断源地区列表 取值范围： 不涉及
permit_count	Long	参数解释： 放行次数 取值范围： 不涉及
permit_top_one_acl_id	String	参数解释： 高频命中的放行策略ID 取值范围： 不涉及
permit_top_one_acl_name	String	参数解释： 高频命中的放行策略名称 取值范围： 不涉及
records	Array of AccessTopStatisticsVO objects	参数解释： 命中趋势 取值范围： 不涉及

参数	参数类型	描述
top_deny_rule_list	Array of AccessTopMemberVO objects	参数解释： TOP阻断规则列表 取值范围： 不涉及

表 4-798 AccessTopStatisticsVO

参数	参数类型	描述
agg_time	Long	参数解释： 聚合时间 取值范围： 不涉及
deny_access_top_counts	Long	参数解释： 阻断数量 取值范围： 不涉及
permit_access_top_counts	Long	参数解释： 放行数量 取值范围： 不涉及
total_access_top_counts	Long	参数解释： 命中次数 取值范围： 不涉及

表 4-799 AccessTopMemberVO

参数	参数类型	描述
count	String	参数解释： 次数 取值范围： 不涉及
item	String	参数解释： 项 取值范围： 不涉及

参数	参数类型	描述
name	String	参数解释： 项名称 取值范围： 不涉及

状态码：400

表 4-800 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询项目ID为a16df7cf1d094befa6bbc72cbf51e93a防火墙ID为c598d612-8112-465c-bffd-863d4ce85a50时间范围为1小时的策略概览

https://{Endpoint}/v1/a16df7cf1d094befa6bbc72cbf51e93a/cfw/logs/access-top?fw_instance_id=a7df0f6c-da03-4511-ad0b-b17b589ff0ec&item=strategy_dashboard&range=0

响应示例

状态码：200

OK

```
{
  "data": {
    "deny_count": 27,
    "deny_top_one_acl_id": "1ceb7988-b11b-4f1e-b7f1-1a807e9e89f3",
    "deny_top_one_acl_name": "deny",
    "hit_count": 4404,
    "permit_count": 4377,
    "permit_top_one_acl_id": "d1de06f0-dfe1-46e5-a50d-81a423b157a8",
    "permit_top_one_acl_name": "23333"
  }
}
```

状态码：400

Bad Request

```
{
  "error_code" : "CFW.00200007",
  "error_msg" : "时间范围错误"
}
```

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 ShowAccessTopSolution {

    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();
        ShowAccessTopRequest request = new ShowAccessTopRequest();
        try {
            ShowAccessTopResponse response = client.showAccessTop(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 = ShowAccessTopRequest()
        response = client.show_access_top(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.ShowAccessTopRequest{}
    response, err := client.ShowAccessTop(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

错误码

请参见[错误码](#)。

4.15.3 查询攻击统计

功能介绍

根据防火墙攻击日志，查询攻击统计信息

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/cfw/logs/attack-statistic

表 4-801 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-802 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
range	否	Integer	参数解释： 时间范围 约束限制： 不涉及 取值范围： 0为近一时 1近一天 2近七天 默认取值： 不涉及

参数	是否必选	参数类型	描述
log_type	是	String	参数解释： 日志类型 约束限制： 不涉及 取值范围： internet为南北向日志、nat为nat场景日志，vpc为东西向日志，vgw为vgw场景日志 默认取值： 不涉及
direction	否	String	参数解释： 会话方向 约束限制： 不涉及 取值范围： in2out为出云方向 out2in为入云方向 默认取值： 不涉及
start_time	否	Long	参数解释： 开始时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
end_time	否	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及

参数	是否必选	参数类型	描述
vgw_id	否	Array of strings	参数解释： VGW ID 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
item	是	String	参数解释： 聚合类型 约束限制： 不涉及 取值范围： dst TOP攻击目的统计 src TOP攻击来源统计 默认取值： 不涉及
size	否	Integer	参数解释： 聚合条数 约束限制： 不涉及 取值范围： 0到100条 默认取值： 50

请求参数

无

响应参数

状态码：200

表 4-803 响应 Body 参数

参数	参数类型	描述
data	Array of AttackStatisticRespBody objects	参数解释： 攻击日志统计 取值范围： 不涉及

表 4-804 AttackStatisticRespBody

参数	参数类型	描述
apps	Array of TopInfo objects	参数解释： 应用列表 取值范围： 不涉及
associated_name	String	参数解释： 绑定资源名称 取值范围： 不涉及
associated_type	String	参数解释： 绑定资源类型 取值范围： PORT：IPV4云服务器 IPV6_PORT：IPV6云服务器
attack_count	Long	参数解释： 攻击次数 取值范围： 不涉及

参数	参数类型	描述
attack_type	String	参数解释： 攻击类型 取值范围： Access Control：访问控制 Vulnerability scanning：漏洞扫描 Email attack：邮件攻击 Vulnerability Attack：漏洞攻击 Web attack：Web攻击 password attack：密码攻击 Hijacking attack：劫持攻击 Protocol exception：协议异常 Trojan horse：特洛伊木马
deny_count	Long	参数解释： 拦截次数 取值范围： 不涉及
dst_ports	Array of TopInfo objects	参数解释： 目的端口列表 取值范围： 不涉及
ip	String	参数解释： IP地址 取值范围： 不涉及
latest_time	Long	参数解释： 最近攻击时间 取值范围： 不涉及
region_id	String	参数解释： 地区ID 取值范围： 不涉及
region_name	String	参数解释： 地区名称 取值范围： 不涉及

参数	参数类型	描述
src_type	String	参数解释： 攻击来源类型 取值范围： 不涉及
vgw_id	String	参数解释： vgw Id 取值范围： 不涉及

表 4-805 TopInfo

参数	参数类型	描述
count	Long	参数解释： 次数 取值范围： 不涉及
item	String	参数解释： 项 取值范围： 不涉及
item_id	String	参数解释： 项ID 取值范围： 不涉及

状态码：400

表 4-806 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及

参数	参数类型	描述
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询项目ID为a16df7cf1d094befa6bbc72cbf51e93a防火墙ID为a7df0f6c-da03-4511-ad0b-b17b589ff0ec时间范围为1小时的TOP攻击来源统计

```
https://{Endpoint}/v1/a16df7cf1d094befa6bbc72cbf51e93a/cfw/logs/attack-statistic?  
fw_instance_id=a7df0f6c-da03-4511-ad0b-b17b589ff0ec&log_type=internet&item=src&range=0
```

响应示例

状态码：200

OK

```
{  
  "data": [ {  
    "apps": [ {  
      "count": 23,  
      "item": "UDP-ANY"  
    } ],  
    "attack_count": 23,  
    "attack_type": "Vulnerability Exploit Attack",  
    "deny_count": 0,  
    "dst_ports": [ {  
      "count": 23,  
      "item": "9034"  
    } ],  
    "ip": "110.41.54.9",  
    "latest_time": 1750675436000,  
    "region_id": "US",  
    "src_type": "out2in"  
  } ]  
}
```

状态码：400

Bad Request

```
{  
  "error_code": "CFW.00200007",  
  "error_msg": "时间范围错误"  
}
```

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 ListAttackStatisticSolution {

    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();
        ListAttackStatisticRequest request = new ListAttackStatisticRequest();
        try {
            ListAttackStatisticResponse response = client.listAttackStatistic(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 = ListAttackStatisticRequest()
    response = client.list_attack_statistic(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.ListAttackStatisticRequest{}
    response, err := client.ListAttackStatistic(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

错误码

请参见[错误码](#)。

4.15.4 查询攻击趋势

功能介绍

查询攻击趋势

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/cfw/logs/trend-attack

表 4-807 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-808 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
range	否	Integer	参数解释： 时间范围 约束限制： 不涉及 取值范围： 0为近一时 1近一天 2近七天 默认取值： 不涉及
log_type	是	String	参数解释： 日志类型 约束限制： 不涉及 取值范围： internet为南北向日志、nat为nat场景日志，vpc为东西向日志，vgw为vgw场景日志 默认取值： 不涉及
start_time	否	Long	参数解释： 开始时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及

参数	是否必选	参数类型	描述
end_time	否	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
vgw_id	否	Array of strings	参数解释： VGW ID 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-809 响应 Body 参数

参数	参数类型	描述
data	Array of AttackTrendResp Body objects	参数解释： 攻击趋势 取值范围： 不涉及

表 4-810 AttackTrendRespBody

参数	参数类型	描述
deny_count	Long	参数解释： 阻断次数 取值范围： 不涉及
permit_count	Long	参数解释： 放行次数 取值范围： 不涉及
time	Long	参数解释： 聚合时间 取值范围： 不涉及

状态码：400

表 4-811 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询项目ID为a16df7cf1d094befa6bbc72cbf51e93a防火墙ID为a7df0f6c-da03-4511-ad0b-b17b589ff0ec时间范围为1小时的互联网边界的攻击趋势

https://{Endpoint}/v1/a16df7cf1d094befa6bbc72cbf51e93a/cfw/logs/trend-attack?fw_instance_id=a7df0f6c-da03-4511-ad0b-b17b589ff0ec&log_type=internet&range=0

响应示例

状态码：200

OK

```
{
  "data": [ {
    "deny_count": 0,
    "permit_count": 0,
    "time": 1750680060000
  } ]
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200007",
  "error_msg": "时间范围错误"
}
```

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 ShowAttackTrendSolution {

    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();
        ShowAttackTrendRequest request = new ShowAttackTrendRequest();
        try {
            ShowAttackTrendResponse response = client.showAttackTrend(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 = ShowAttackTrendRequest()
        response = client.show_attack_trend(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.ShowAttackTrendRequest{}
response, err := client.ShowAttackTrend(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

错误码

请参见[错误码](#)。

4.15.5 查询攻击概览

功能介绍

查询攻击概览

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/cfw/logs/total-attack

表 4-812 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-813 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
range	否	Integer	参数解释： 时间范围 约束限制： 不涉及 取值范围： 0为近一时 1近一天 2近七天 默认取值： 不涉及

参数	是否必选	参数类型	描述
log_type	是	String	参数解释： 日志类型 约束限制： 不涉及 取值范围： internet为南北向日志、nat为nat场景日志，vpc为东西向日志，vgw为vgw场景日志 默认取值： 不涉及
start_time	否	Long	参数解释： 开始时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
end_time	否	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
vgw_id	否	Array of strings	参数解释： VGW ID 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-814 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 攻击统计信息 取值范围： 不涉及

表 4-815 data

参数	参数类型	描述
attack_count	Long	参数解释： 攻击次数 取值范围： 不涉及
deny_count	Long	参数解释： 阻断次数 取值范围： 不涉及
permit_count	Long	参数解释： 放行次数 取值范围： 不涉及
risk_ports	Long	参数解释： 风险端口 取值范围： 不涉及

状态码：400

表 4-816 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询项目ID为a16df7cf1d094befa6bbc72cbf51e93a防火墙ID为a7df0f6c-da03-4511-ad0b-b17b589ff0ec时间范围为1小时的互联网边界的安全概览

https://{Endpoint}/v1/a16df7cf1d094befa6bbc72cbf51e93a/cfw/logs/total-attack?fw_instance_id=a7df0f6c-da03-4511-ad0b-b17b589ff0ec&range=0&log_type=internet

响应示例

状态码：200

OK

```
{
  "data": {
    "attack_count": 0,
    "deny_count": 0,
    "permit_count": 0,
    "risk_ports": 0
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200007",
  "error_msg": "时间范围错误"
}
```

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 ShowAttackTotalSolution {

    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();
        ShowAttackTotalRequest request = new ShowAttackTotalRequest();
        try {
            ShowAttackTotalResponse response = client.showAttackTotal(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 = ShowAttackTotalRequest()
    response = client.show_attack_total(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.ShowAttackTotalRequest{}
    response, err := client.ShowAttackTotal(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

错误码

请参见[错误码](#)。

4.15.6 查询日志数量

功能介绍

统计日志数量，如统计风险IP的数量

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/logs/count

表 4-817 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-818 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
range	否	Integer	参数解释： 时间范围 约束限制： 不涉及 取值范围： 0为近一时 1近一天 2近七天 默认取值： 不涉及
start_time	否	Long	参数解释： 开始时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
end_time	否	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及

参数	是否必选	参数类型	描述
vgw_id	否	Array of strings	参数解释： VGW ID 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
item	是	String	参数解释： 聚合类型 约束限制： 不涉及 取值范围： risk_ip 访问风险IP数量 risk_host 访问风险域名数量 open_port 开放端口的数量 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-819 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 聚合时间点 取值范围： 不涉及

表 4-820 data

参数	参数类型	描述
count	Long	参数解释： 聚合时间点 取值范围： 不涉及

状态码：400

表 4-821 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询项目ID为a16df7cf1d094befa6bbc72cbf51e93a防火墙ID为dc3f5b8e-efed-4e31-9ed0-785f342c51fc时间范围从1751450695602到17514542956027的风险IP数量

https://{Endpoint}/v1/a16df7cf1d094befa6bbc72cbf51e93a/logs/count?fw_instance_id=dc3f5b8e-efed-4e31-9ed0-785f342c51fc&item=risk_ip&start_time=1751450695602&end_time=1751454295602

响应示例

状态码：200

ShowLogsCountResp

```
{
  "data": {
    "count": 0
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200007",
}
```

```
"error_msg": "时间范围错误"  
}
```

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 ShowLogsCountSolution {  
  
    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();  
        ShowLogsCountRequest request = new ShowLogsCountRequest();  
        try {  
            ShowLogsCountResponse response = client.showLogsCount(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 = ShowLogsCountRequest()
        response = client.show_logs_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.ShowLogsCountRequest{}
    response, err := client.ShowLogsCount(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	ShowLogsCountResp
400	Bad Request

错误码

请参见[错误码](#)。

4.15.7 查询流量日志统计详情

功能介绍

查询流量日志统计详情，如统计某个源IP访问详情

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/cfw/logs/flow-detail

表 4-822 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-823 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
range	否	Integer	参数解释： 时间范围 约束限制： 不涉及 取值范围： 0为近一时 1近一天 2近七天 默认取值： 不涉及

参数	是否必选	参数类型	描述
log_type	是	String	参数解释： 日志类型 约束限制： 不涉及 取值范围： internet为南北向日志、nat为nat场景日志，vpc为东西向日志，vgw为vgw场景日志 默认取值： 不涉及
direction	否	String	参数解释： 会话方向 约束限制： 不涉及 取值范围： in2out为出云方向 out2in为入云方向 默认取值： 不涉及
start_time	否	Long	参数解释： 开始时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
end_time	否	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及

参数	是否必选	参数类型	描述
vgw_id	否	Array of strings	参数解释： VGW ID 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
asset_type	否	String	参数解释： IP类型 约束限制： 不涉及 取值范围： public 公网IP private 私网IP open_port 默认取值： 不涉及
item	是	String	参数解释： 聚合类型 约束限制： 不涉及 取值范围： src_ip 源IP dst_ip 目的IP dst_port 目的端口 protocol 协议 dst_host 目的域名 app 应用 dst_region_name 目的地区 src_region_name 源地区 默认取值： 不涉及

参数	是否必选	参数类型	描述
value	是	String	参数解释： 统计对象 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-824 响应 Body 参数

参数	参数类型	描述
data	FlowAnalysisVO object	

表 4-825 FlowAnalysisVO

参数	参数类型	描述
app_count	Long	参数解释： 应用统计 取值范围： 不涉及
bytes	Double	参数解释： 字节数 取值范围： 不涉及
dst_ip_count	Long	参数解释： 目的IP计数 取值范围： 不涉及

参数	参数类型	描述
dst_port_count	Long	参数： 目的端口计数 取值范围： 不涉及
end_time	Long	参数解释： 结束时间 取值范围： 不涉及
records	Array of SessionVO objects	参数解释： TOP会话详情 取值范围： 不涉及
request_byte	Double	参数解释： 请求字节数 取值范围： 不涉及
response_byte	Double	参数解释： 响应字节数 取值范围： 不涉及
sessions	Long	参数解释： 会话次数 取值范围： 不涉及
src_ip_count	Long	参数解释： 源IP计数 取值范围： 不涉及
start_time	Long	参数解释： 开始时间 取值范围： 不涉及

表 4-826 SessionVO

参数	参数类型	描述
app	String	参数解释： 应用 取值范围： 不涉及
bytes	Double	参数解释： 字节数 取值范围： 不涉及
dst_associate_instance_type	String	参数解释： 目的IP关联资产类型 取值范围： 不涉及
dst_device_name	String	参数解释： 目的IP关联资产名称 取值范围： 不涉及
dst_ip	String	参数解释： 目的IP 取值范围： 不涉及
dst_port	String	参数解释： 目的端口 取值范围： 不涉及
dst_host	String	参数解释： 目的域名 取值范围： 不涉及
dst_region_id	String	参数解释： 目的regionID 取值范围： 不涉及

参数	参数类型	描述
dst_region_name	String	参数解释： 目的地区 取值范围： 不涉及
end_time	Long	参数解释： 结束时间 取值范围： 不涉及
protocol	String	参数解释： 协议 取值范围： 不涉及
request_byte	Double	参数解释： 请求字节数 取值范围： 不涉及
response_byte	Double	参数解释： 响应字节数 取值范围： 不涉及
sessions	Long	参数解释： 会话数量 取值范围： 不涉及
src_associate_instance_type	String	参数解释： 源IP关联资产类型 取值范围： 不涉及
src_device_name	String	参数解释： 源IP关联资产名称 取值范围： 不涉及
src_ip	String	参数解释： 源IP 取值范围： 不涉及

参数	参数类型	描述
src_region_id	String	参数解释： 源地区 ID 取值范围： 不涉及
src_region_name	String	参数解释： 源地区 取值范围： 不涉及
start_time	Long	参数解释： 会话开始时间 取值范围： 不涉及

状态码：400

表 4-827 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询项目ID为a16df7cf1d094befa6bbc72cbf51e93a，防火墙ID为c598d612-8112-465c-bffd-863d4ce85a50，时间范围为7天，方向为出云方向，目的IP为100.93.4.158的流量日志统计详情

```
https://{Endpoint}/v1/a16df7cf1d094befa6bbc72cbf51e93a/cfw/logs/flow-detail?  
fw_instance_id=c598d612-8112-465c-  
bffd-863d4ce85a50&log_type=internet&direction=in2out&item=dst_ip&range=2&value=100.93.4.158
```

响应示例

状态码：200

OK

```
{
  "data": {
    "app_count": 1,
    "bytes": 10219,
    "dst_port_count": 1,
    "end_time": 1751355533000,
    "records": [ {
      "app": "HTTP",
      "bytes": 10219,
      "dst_ip": "100.93.12.160",
      "dst_port": "8080",
      "end_time": 1751355533000,
      "protocol": "TCP",
      "request_byte": 5357,
      "response_byte": 4862,
      "sessions": 11,
      "src_associate_instance_type": "PORT",
      "src_device_name": "ecs_172",
      "src_ip": "100.93.2.30",
      "start_time": 1751355362000
    } ],
    "request_byte": 5357,
    "response_byte": 4862,
    "sessions": 11,
    "src_ip_count": 1,
    "start_time": 1751355362000
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200007",
  "error_msg": "时间范围错误"
}
```

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 ShowFlowDetailSolution {

    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();
ShowFlowDetailRequest request = new ShowFlowDetailRequest();
try {
    ShowFlowDetailResponse response = client.showFlowDetail(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 = ShowFlowDetailRequest()
        response = client.show_flow_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.ShowFlowDetailRequest{}
    response, err := client.ShowFlowDetail(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

错误码

请参见[错误码](#)。

4.15.8 查询流量 TOP 统计

功能介绍

查询流量TOP统计

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/cfw/logs/flow-top

表 4-828 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-829 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	是否必选	参数类型	描述
range	否	Integer	参数解释： 时间范围 约束限制： 不涉及 取值范围： 0为近一时 1近一天 2近七天 默认取值： 不涉及
log_type	是	String	参数解释： 日志类型 约束限制： 不涉及 取值范围： internet为南北向日志、nat为nat场景日志，vpc为东西向日志，vgw为vgw场景日志 默认取值： 不涉及
direction	否	String	参数解释： 会话方向 约束限制： 不涉及 取值范围： in2out为出云方向 out2in为入云方向 默认取值： 不涉及
start_time	否	Long	参数解释： 开始时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及

参数	是否必选	参数类型	描述
end_time	否	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
vgw_id	否	Array of strings	参数解释： VGW ID 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
asset_type	否	String	参数解释： IP类型 约束限制： 不涉及 取值范围： public 公网IP private 私网IP open_port 默认取值： 不涉及

参数	是否必选	参数类型	描述
item	是	Array of strings	参数解释： 聚合类型 约束限制： 不涉及 取值范围： src_ip 源IP dst_ip 目的IP dst_port 目的端口 protocol 协议 dst_host 目的域名 app 应用 dst_region_name 目的地区 src_region_name 源地区 默认取值： 不涉及
order	是	String	参数解释： 排序条件 约束限制： 不涉及 取值范围： record 会话条数 byte 会话流量 默认取值： 5
size	否	Integer	参数解释： 聚合条数 约束限制： 不涉及 取值范围： 0到10条 默认取值： 5

请求参数

无

响应参数

状态码：200

表 4-830 响应 Body 参数

参数	参数类型	描述
data	Map<String,Array< ItemVO >>	参数解释： TOP统计信息 取值范围： 不涉及

表 4-831 ItemVO

参数	参数类型	描述
key	String	参数解释： 聚合项 取值范围： 不涉及
name	String	参数解释： 聚合项名称 取值范围： 不涉及
value	Long	参数解释： 统计值 取值范围： 不涉及

状态码：400

表 4-832 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及

参数	参数类型	描述
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询项目ID为a16df7cf1d094befa6bbc72cbf51e93a，防火墙ID为c598d612-8112-465c-bffd-863d4ce85a50，时间范围为7天，方向为出云方向，按照会话流量排序，TOP访问域名

```
https://{Endpoint}/v1/a16df7cf1d094befa6bbc72cbf51e93a/cfw/logs/flow-top?fw_instance_id=c598d612-8112-465c-bffd-863d4ce85a50&log_type=internet&order=byte&item=dst_host&range=2&direction=in2out
```

响应示例

状态码：200

OK

```
{
  "data": {
    "dst_host": [ {
      "key": "www.test.com",
      "value": 10219
    } ]
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200007",
  "error_msg": "时间范围错误"
}
```

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 ShowFlowTopSolution {

    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();
        ShowFlowTopRequest request = new ShowFlowTopRequest();
        try {
            ShowFlowTopResponse response = client.showFlowTop(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 = ShowFlowTopRequest()
        response = client.show_flow_top(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.ShowFlowTopRequest{}
    response, err := client.ShowFlowTop(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

错误码

请参见[错误码](#)。

4.15.9 查询流量日志统计

功能介绍

查询流量日志统计

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/cfw/logs/flow-statistic

表 4-833 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-834 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
range	否	Integer	参数解释： 时间范围 约束限制： 不涉及 取值范围： 0为近一时 1近一天 2近七天 默认取值： 不涉及
log_type	是	String	参数解释： 日志类型 约束限制： 不涉及 取值范围： internet为南北向日志、nat为nat场景日志，vpc为东西向日志，vgw为vgw场景日志 默认取值： 不涉及
direction	否	String	参数解释： 会话方向 约束限制： 不涉及 取值范围： in2out为出云方向 out2in为入云方向 默认取值： 不涉及

参数	是否必选	参数类型	描述
start_time	否	Long	参数解释： 开始时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
end_time	否	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
vgw_id	否	Array of strings	参数解释： VGW ID 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
asset_type	否	String	参数解释： IP类型 约束限制： 不涉及 取值范围： public 公网IP private 私网IP open_port 默认取值： 不涉及

参数	是否必选	参数类型	描述
item	是	String	参数解释： 聚合类型 约束限制： 不涉及 取值范围： src_ip 源IP dst_ip 目的IP dst_port 目的端口 protocol 协议 dst_host 目的域名 app 应用 dst_region_name 目的地区 src_region_name 源地区 risk_ip 风险IP risk_host 风险域名 open_port 开放端口 默认取值： 不涉及
size	否	Integer	参数解释： 聚合条数 约束限制： 不涉及 取值范围： 0到10条 默认取值： 5

请求参数

无

响应参数

状态码：200

表 4-835 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 流量日志统计 取值范围： 不涉及

表 4-836 data

参数	参数类型	描述
records	Array of FlowDetailsVO objects	参数解释： 统计记录 取值范围： 不涉及
total	Long	参数解释： 总数 取值范围： 不涉及

表 4-837 FlowDetailsVO

参数	参数类型	描述
apps	Array of ItemVO objects	参数解释： 应用统计 取值范围： 不涉及
associate_instance_type	String	参数解释： 关联资产类型 取值范围： 不涉及
device_name	String	参数解释： 关联资产名称 取值范围： 不涉及

参数	参数类型	描述
item	String	参数解释： 聚合项 取值范围： 不涉及
last_time	Long	参数解释： 最近访问时间 取值范围： 不涉及
agg_start_time	Long	参数解释： 聚合开始时间 取值范围： 不涉及
agg_end_time	Long	参数解释： 聚合结束时间 取值范围： 不涉及
ports	Array of ItemVO objects	参数解释： 端口统计 取值范围： 不涉及
region	String	参数解释： 地区 取值范围： 不涉及
request_byte	Double	参数解释： 请求字节数 取值范围： 不涉及
response_byte	Double	参数解释： 响应字节数 取值范围： 不涉及
sessions	Long	参数解释： 会话数量 取值范围： 不涉及

参数	参数类型	描述
tags	String	参数解释： 标签 取值范围： 不涉及
src_ip	Array of ItemVO objects	参数解释： 源IP 取值范围： 不涉及
dst_ip	Array of ItemVO objects	参数解释： 目的IP 取值范围： 不涉及
protocol	String	参数解释： 协议 取值范围： 不涉及

表 4-838 ItemVO

参数	参数类型	描述
key	String	参数解释： 聚合项 取值范围： 不涉及
name	String	参数解释： 聚合项名称 取值范围： 不涉及
value	Long	参数解释： 统计值 取值范围： 不涉及

状态码：400

表 4-839 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询项目ID为a16df7cf1d094befa6bbc72cbf51e93a，防火墙ID为c598d612-8112-465c-bffd-863d4ce85a50，时间范围为7天，方向为出云方向，外联域名统计信息

```
https://{Endpoint}/v1/a16df7cf1d094befa6bbc72cbf51e93a/cfw/logs/flow-statistic?fw_instance_id=c598d612-8112-465c-bffd-863d4ce85a50&log_type=internet&direction=in2out&item=dst_host&range=2&size=1000
```

响应示例

状态码：200

OK

```
{
  "data": {
    "records": [ {
      "apps": [ {
        "key": "HTTP",
        "value": 11
      } ],
      "item": "www.test.com",
      "last_time": 175135533000,
      "ports": [ {
        "key": "8080",
        "value": 11
      } ],
      "request_byte": 5357,
      "response_byte": 4862,
      "sessions": 11
    } ],
    "total": 1
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200007",
  "error_msg": "时间范围错误"
}
```

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 ListFlowStatisticSolution {

    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();
        ListFlowStatisticRequest request = new ListFlowStatisticRequest();
        try {
            ListFlowStatisticResponse response = client.listFlowStatistic(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

```
# 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 = ListFlowStatisticRequest()
    response = client.list_flow_statistic(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.ListFlowStatisticRequest{}
    response, err := client.ListFlowStatistic(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

错误码

请参见[错误码](#)。

4.15.10 查询会话趋势

功能介绍

查询会话趋势

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/cfw/logs/flow-trend

表 4-840 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-841 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
range	否	Integer	参数解释： 时间范围 约束限制： 不涉及 取值范围： 0为近一时 1近一天 2近七天 默认取值： 不涉及
log_type	是	String	参数解释： 日志类型 约束限制： 不涉及 取值范围： internet为南北向日志、nat为nat场景日志，vpc为东西向日志，vgw为vgw场景日志 默认取值： 不涉及
direction	否	String	参数解释： 会话方向 约束限制： 不涉及 取值范围： in2out为出云方向 out2in为入云方向 默认取值： 不涉及

参数	是否必选	参数类型	描述
start_time	否	Long	参数解释： 开始时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
end_time	否	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
vgw_id	否	Array of strings	参数解释： VGW ID 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
asset_type	否	String	参数解释： IP类型 约束限制： 不涉及 取值范围： public 公网IP private 私网IP open_port 默认取值： 不涉及
ip	否	Array of strings	ips

参数	是否必选	参数类型	描述
vpc	否	Array of strings	vpcs

请求参数

无

响应参数

状态码：200

表 4-842 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 会话趋势统计 取值范围： 不涉及

表 4-843 data

参数	参数类型	描述
records	Array of FlowTrendVO objects	参数解释： 流量趋势数据 取值范围： 不涉及
total	Long	参数解释： 总数 取值范围： 不涉及

表 4-844 FlowTrendVO

参数	参数类型	描述
agg_time	Long	参数解释： 聚合时间点 取值范围： 不涉及

参数	参数类型	描述
in_bps	Double	参数解释： 入方向bps 取值范围： 不涉及
out_bps	Double	参数解释： 出方向bps 取值范围： 不涉及

状态码：400

表 4-845 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询项目ID为a16df7cf1d094befa6bbc72cbf51e93a，防火墙ID为c598d612-8112-465c-bffd-863d4ce85a50，时间范围为近1小时，VpcID为42562e8a-af8d-4a9a-91a4-c1d6c83ecfc5的会话趋势

```
https://{Endpoint}/v1/a16df7cf1d094befa6bbc72cbf51e93a/cfw/logs/flow-trend?  
fw_instance_id=c598d612-8112-465c-bffd-863d4ce85a50&range=0&vpc=42562e8a-af8d-4a9a-91a4-  
c1d6c83ecfc5&log_type=vpc
```

响应示例

状态码：200

OK

```
{  
  "data": {  
    "records": [ {  
      "agg_time": 1751522040000,  
      "in_bps": 0,  
      "out_bps": 0,  
      "vpc": "42562e8a-af8d-4a9a-91a4-c1d6c83ecfc5"  
    } ]  
  }  
}
```

```
"out_bps" : 0
}],
"total" : 60
}
}
```

状态码：400

Bad Request

```
{
  "error_code" : "CFW.00200007",
  "error_msg" : "时间范围错误"
}
```

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 ShowFlowTrendSolution {

    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();
        ShowFlowTrendRequest request = new ShowFlowTrendRequest();
        try {
            ShowFlowTrendResponse response = client.showFlowTrend(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 = ShowFlowTrendRequest()  
        response = client.show_flow_trend(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.ShowFlowTrendRequest{}  
response, err := client.ShowFlowTrend(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

错误码

请参见[错误码](#)。

4.15.11 查询攻击日志统计详情

功能介绍

查询攻击日志统计详情

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/cfw/logs/attack-detail

表 4-846 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-847 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
range	否	Integer	参数解释： 时间范围 约束限制： 不涉及 取值范围： 0为近一时 1近一天 2近七天 默认取值： 不涉及

参数	是否必选	参数类型	描述
log_type	是	String	参数解释： 日志类型 约束限制： 不涉及 取值范围： internet为南北向日志、nat为nat场景日志，vpc为东西向日志，vgw为vgw场景日志 默认取值： 不涉及
start_time	否	Long	参数解释： 开始时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
end_time	否	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
vgw_id	否	Array of strings	参数解释： VGW ID 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	是否必选	参数类型	描述
action	是	Integer	参数解释： 动作 约束限制： 不涉及 取值范围： 0 全部 1 拦截 默认取值： 不涉及
item	是	String	参数解释： 聚合类型 约束限制： 不涉及 取值范围： src_region_id TOP外部攻击来源地区 attack_type 攻击类型 in_src_ip TOP内部攻击来源IP out_src_ip TOP外部攻击来源IP dst_port TOP被攻击端口 dst_ip TOP攻击目的IP attack_rule TOP攻击规则 src_ip TOP攻击来源IP 默认取值： 不涉及
value	是	String	参数解释： 统计对象 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-848 响应 Body 参数

参数	参数类型	描述
data	AttackDetailVO object	

表 4-849 AttackDetailVO

参数	参数类型	描述
app_count	Long	参数解释： 应用数量 取值范围： 不涉及
attack_rule_count	Long	参数解释： 攻击规则数量 取值范围： 不涉及
attack_type_count	Long	参数解释： 攻击类型数量 取值范围： 不涉及
count	Long	参数解释： 攻击次数 取值范围： 不涉及
dst_ip_count	Long	参数解释： 目的IP数量 取值范围： 不涉及
dst_port_count	Long	参数解释： 攻击端口数量 取值范围： 不涉及

参数	参数类型	描述
end_time	Long	参数解释： 结束时间 取值范围： 不涉及
records	Array of AttackLog objects	参数解释： 攻击事件明细 取值范围： 不涉及
src_ip_count	Long	参数解释： 源IP数量 取值范围： 不涉及
start_time	Long	参数解释： 开始时间 取值范围： 不涉及
total	Long	参数解释： 总数 取值范围： 不涉及

表 4-850 AttackLog

参数	参数类型	描述
action	String	参数解释： 动作 取值范围： 不涉及
app	String	参数解释： 应用 取值范围： 不涉及
attack_rule	String	参数解释： 攻击规则 取值范围： 不涉及

参数	参数类型	描述
attack_rule_id	String	参数解释： 攻击规则ID 取值范围： 不涉及
attack_type	String	参数解释： 攻击类型 取值范围： 不涉及
direction	String	参数解释： 攻击方向 取值范围： 不涉及
dst_ip	String	参数解释： 目的IP 取值范围： 不涉及
dst_port	Integer	参数解释： 目的端口 取值范围： 不涉及
dst_region_id	String	参数解释： 目的地区Id 取值范围： 不涉及
dst_region_name	String	参数解释： 目的地区名称 取值范围： 不涉及
dst_province_id	String	参数解释： 目的省份ID 取值范围： 不涉及
dst_province_name	String	参数解释： 目的省份名称 取值范围： 不涉及

参数	参数类型	描述
dst_city_id	String	参数解释： 目的城市Id 取值范围： 不涉及
dst_city_name	String	参数解释： 目的城市名称 取值范围： 不涉及
event_time	Long	参数解释： 发生时间 取值范围： 不涉及
level	String	参数解释： 危险等级 取值范围： 不涉及
protocol	String	参数解释： 协议 取值范围： 不涉及
source	String	参数解释： 来源 取值范围： 不涉及
src_ip	String	参数解释： 源IP 取值范围： 不涉及
real_ip	String	参数解释： 真实IP 取值范围： 不涉及
tag	Integer	参数解释： 标签 取值范围： 不涉及

参数	参数类型	描述
src_port	Integer	参数解释： 源端口 取值范围： 不涉及
src_region_id	String	参数解释： 源地区Id 取值范围： 不涉及
src_region_name	String	参数解释： 源地区名称 取值范围： 不涉及
src_province_id	String	参数解释： 源省份Id 取值范围： 不涉及
src_province_name	String	参数解释： 源省份名称 取值范围： 不涉及
src_city_id	String	参数解释： 源城市Id 取值范围： 不涉及
src_city_name	String	参数解释： 源城市 取值范围： 不涉及
vgw_id	String	参数解释： VGW Id 取值范围： 不涉及

状态码：400

表 4-851 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

```
https://{Endpoint}/v1/7db2c6e2934046dd8c5a996ed4780c5b/cfw/logs/attack-detail?fw_instance_id=a7df0f6c-da03-4511-ad0b-b17b589ff0ec&log_type=internet&range=2&item=src_region_id&action=0&value=US
```

查询项目ID为7db2c6e2934046dd8c5a996ed4780c5b，防火墙ID为a7df0f6c-da03-4511-ad0b-b17b589ff0ec，时间范围为7天，外部攻击来源地区为US的 攻击日志统计详情

响应示例

状态码：200
OK

```
{
  "data": {
    "app_count": 1,
    "attack_rule_count": 1,
    "attack_type_count": 1,
    "count": 1,
    "dst_ip_count": 1,
    "dst_port_count": 1,
    "end_time": 1751014701000,
    "records": [ {
      "action": "permit",
      "app": "UDP-ANY",
      "attack_rule": "Realtek Jungle SDK Command Injection Vulnerability (CVE-2021-35394)",
      "attack_rule_id": "806310",
      "attack_type": "Vulnerability Exploit Attack",
      "dst_ip": "121.37.223.24",
      "dst_port": 9034,
      "dst_region_id": "CN",
      "dst_region_name": "中国",
      "dst_province_id": "GD",
      "dst_province_name": "Guangdong",
      "dst_city_id": "Guangzhou",
      "dst_city_name": "Guangzhou",
      "event_time": 1751014701000,
      "level": "CRITICAL",
      "protocol": "UDP",
      "source": "predefined",
      "src_ip": "92.112.125.103",
      "tag": -1,
      "src_port": 43533,
      "src_region_id": "US",
```

```
{
  "src_region_name": "美国"
}],
"src_ip_count": 1,
"start_time": 1751014701000,
"total": 1
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200007",
  "error_msg": "时间范围错误"
}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.15.12 查询攻击日志 TOP 统计

功能介绍

查询攻击日志TOP统计

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/cfw/logs/top-attack

表 4-852 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-853 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
range	否	Integer	参数解释： 时间范围 约束限制： 不涉及 取值范围： 0为近一时 1近一天 2近七天 默认取值： 不涉及

参数	是否必选	参数类型	描述
log_type	是	String	参数解释： 日志类型 约束限制： 不涉及 取值范围： internet为南北向日志、nat为nat场景日志，vpc为东西向日志，vgw为vgw场景日志 默认取值： 不涉及
start_time	否	Long	参数解释： 开始时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
end_time	否	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
vgw_id	否	Array of strings	参数解释： VGW ID 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	是否必选	参数类型	描述
action	是	Integer	参数解释： 动作 约束限制： 不涉及 取值范围： 0 全部 1 拦截 默认取值： 不涉及
item	是	Array of strings	参数解释： 聚合类型 约束限制： 不涉及 取值范围： src_region_id TOP外部攻击来源地区 attack_type 攻击类型 in_src_ip TOP内部攻击来源IP out_src_ip TOP外部攻击来源IP dst_port TOP被攻击端口 dst_ip TOP攻击目的IP attack_rule TOP攻击规则 src_ip TOP攻击来源IP level TOP威胁等级 默认取值： 不涉及
size	否	Integer	参数解释： 聚合条数 约束限制： 不涉及 取值范围： 0到10条 默认取值： 5

请求参数

无

响应参数

状态码：200

表 4-854 响应 Body 参数

参数	参数类型	描述
data	AttackTopRespBody object	

表 4-855 AttackTopRespBody

参数	参数类型	描述
attack_rule	Array of TopInfo objects	参数解释： TOP攻击规则 取值范围： 不涉及
attack_type	Array of TopInfo objects	参数解释： TOP攻击类型 取值范围： 不涉及
dst_ip	Array of TopInfo objects	参数解释： TOP攻击目的IP 取值范围： 不涉及
dst_port	Array of TopInfo objects	参数解释： TOP被攻击端口 取值范围： 不涉及
in_src_ip	Array of TopInfo objects	参数解释： TOP内部攻击来源IP 取值范围： 不涉及
level	Array of TopInfo objects	参数解释： TOP威胁等级 取值范围： 不涉及

参数	参数类型	描述
out_src_ip	Array of TopInfo objects	参数解释： TOP外部攻击来源IP 取值范围： 不涉及
src_ip	Array of TopInfo objects	参数解释： TOP攻击来源IP 取值范围： 不涉及
src_region_id	Array of TopInfo objects	参数解释： 源地区ID 取值范围： 不涉及

表 4-856 TopInfo

参数	参数类型	描述
count	Long	参数解释： 次数 取值范围： 不涉及
item	String	参数解释： 项 取值范围： 不涉及
item_id	String	参数解释： 项ID 取值范围： 不涉及

状态码：400

表 4-857 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询项目ID为7db2c6e2934046dd8c5a996ed4780c5b防火墙ID为a7df0f6c-da03-4511-ad0b-b17b589ff0ec时间范围为7天的互联网边界的TOP攻击类型

https://{Endpoint}/v1/7db2c6e2934046dd8c5a996ed4780c5b/cfw/logs/top-attack?fw_instance_id=a7df0f6c-da03-4511-ad0b-b17b589ff0ec&item=attack_type&log_type=internet&action=0&range=2

响应示例

状态码：200

OK

```
{
  "data": {
    "attack_type": [ {
      "count": 51,
      "item": "Vulnerability Exploit Attack",
      "item_id": "Vulnerability Exploit Attack"
    } ]
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200007",
  "error_msg": "时间范围错误"
}
```

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

public class ShowAttackTopSolution {

    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();
        ShowAttackTopRequest request = new ShowAttackTopRequest();
        try {
            ShowAttackTopResponse response = client.showAttackTop(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

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcfwr.v1.region.cfwr_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfwr.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 = ShowAttackTopRequest()
    response = client.show_attack_top(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.ShowAttackTopRequest{}
    response, err := client.ShowAttackTop(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

错误码

请参见[错误码](#)。

4.15.13 查询访问控制统计详情

功能介绍

查询访问控制统计详情

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/cfw/logs/top-access-detail

表 4-858 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-859 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
range	否	Integer	参数解释： 时间范围 约束限制： 不涉及 取值范围： 0为近一时 1近一天 2近七天 默认取值： 不涉及
direction	否	String	参数解释： 会话方向 约束限制： 不涉及 取值范围： in2out为出云方向 out2in为入云方向 默认取值： 不涉及
start_time	否	Long	参数解释： 开始时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及

参数	是否必选	参数类型	描述
end_time	否	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
vgw_id	否	Array of strings	参数解释： VGW ID 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
log_type	否	String	参数解释： 日志类型 约束限制： 不涉及 取值范围： internet为南北向日志、nat为nat场景日志，vpc为东西向日志，vgw为vgw场景日志 默认取值： 不涉及

参数	是否必选	参数类型	描述
item	是	String	参数解释： 聚合类型 约束限制： 不涉及 取值范围： top_deny_rule 命中拦截策略 dst_ip 拦截目的IP src_ip 拦截源IP dst_port 拦截端口 dst_region 拦截目的地区 src_region 拦截源地区 默认取值： 不涉及
item_id	是	String	参数解释： 聚合对象 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-860 响应 Body 参数

参数	参数类型	描述
data	AccessDetailVO object	

表 4-861 AccessDetailVO

参数	参数类型	描述
dst_ip_count	Long	参数解释： 目的IP数量 取值范围： 不涉及
dst_port_count	Long	参数解释： 目的端口数量 取值范围： 不涉及
hit_count	Long	参数解释： 命中次数 取值范围： 不涉及
protocol_count	Long	参数解释： 协议数量 取值范围： 不涉及
recent_end_time	Long	参数解释： 结束时间 取值范围： 不涉及
recent_start_time	Long	参数解释： 开始时间 取值范围： 不涉及
record_total	Long	参数解释： 记录数量 取值范围： 不涉及
records	Array of AccessLogInfo objects	参数解释： 命中详情 取值范围： 不涉及

参数	参数类型	描述
rule_hit_count	Long	参数解释： 命中规则数 取值范围： 不涉及
src_ip_count	Long	参数解释： 源IP数量 取值范围： 不涉及

表 4-862 AccessLogInfo

参数	参数类型	描述
action	String	参数解释： 动作 取值范围： 不涉及
app	String	参数解释： 应用 取值范围： 不涉及
url	String	
dst_host	String	参数解释： 目的域名 取值范围： 不涉及
dst_ip	String	参数解释： 目的IP 取值范围： 不涉及
dst_port	Integer	参数解释： 目的端口 取值范围： 不涉及

参数	参数类型	描述
dst_region_id	String	参数解释： 目的地区ID 取值范围： 不涉及
dst_region_name	String	参数解释： 目的地区名称 取值范围： 不涉及
dst_province_id	String	参数解释： 目的省份ID 取值范围： 不涉及
dst_province_name	String	参数解释： 目的省份名称 取值范围： 不涉及
dst_city_id	String	参数解释： 目的城市ID 取值范围： 不涉及
dst_city_name	String	参数解释： 目的城市名称 取值范围： 不涉及
hit_time	Long	参数解释： 命中时间 取值范围： 不涉及
log_id	String	
protocol	String	参数解释： 协议 取值范围： 不涉及

参数	参数类型	描述
rule_id	String	参数解释： 规则ID 取值范围： 不涉及
rule_name	String	参数解释： 规则名称 取值范围： 不涉及
rule_type	Integer	
src_ip	String	参数解释： 源IP 取值范围： 不涉及
src_port	Integer	参数解释： 源端口 取值范围： 不涉及
src_region_id	String	参数解释： 源地区ID 取值范围： 不涉及
src_region_name	String	参数解释： 源地区名称 取值范围： 不涉及
src_province_id	String	参数解释： 源省份ID 取值范围： 不涉及
src_province_name	String	参数解释： 源省份名称 取值范围： 不涉及

参数	参数类型	描述
src_city_id	String	参数解释： 源城市ID 取值范围： 不涉及
src_city_name	String	参数解释： 源城市名称 取值范围： 不涉及
vgw_id	String	参数解释： VGW ID 取值范围： 不涉及
qos_rule_id	String	
qos_rule_name	String	
qos_rule_type	Integer	

状态码：400

表 4-863 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询项目ID为7db2c6e2934046dd8c5a996ed4780c5b防火墙ID为a7df0f6c-da03-4511-ad0b-b17b589ff0ec时间范围为1小时，IP为59.39.0.152的出云拦截详情

v1/7db2c6e2934046dd8c5a996ed4780c5b/cfw/logs/top-access-detail?fw_instance_id=a7df0f6c-da03-4511-ad0b-b17b589ff0ec&item=dst_ip&range=0&direction=in2out&item_id=59.39.0.152

响应示例

状态码：200

OK

```
{
  "data": {
    "dst_ip_count": 1,
    "dst_port_count": 1,
    "hit_count": 3,
    "protocol_count": 1,
    "recent_end_time": 1751545553000,
    "recent_start_time": 1751545469000,
    "record_total": 3,
    "records": [ {
      "action": "deny",
      "app": "HTTP",
      "dst_ip": "59.39.0.152",
      "dst_port": 80,
      "dst_region_id": "CN",
      "dst_region_name": "中国",
      "dst_province_id": "GD",
      "dst_province_name": "Guangdong",
      "dst_city_id": "Foshan",
      "dst_city_name": "Foshan",
      "hit_time": 1751545498000,
      "protocol": "TCP",
      "rule_id": "1ceb7988-b11b-4f1e-b7f1-1a807e9e89f3",
      "rule_name": "deny",
      "src_ip": "121.37.223.24",
      "src_port": 39948,
      "src_region_id": "CN",
      "src_region_name": "中国",
      "src_province_id": "GD",
      "src_province_name": "Guangdong",
      "src_city_id": "Guangzhou",
      "src_city_name": "Guangzhou"
    } ],
    "rule_hit_count": 1,
    "src_ip_count": 1
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200007",
  "error_msg": "时间范围错误"
}
```

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 ShowAccessDetailSolution {

    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();
        ShowAccessDetailRequest request = new ShowAccessDetailRequest();
        try {
            ShowAccessDetailResponse response = client.showAccessDetail(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 = ShowAccessDetailRequest()
    response = client.show_access_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.ShowAccessDetailRequest{}
    response, err := client.ShowAccessDetail(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

状态码	描述
400	Bad Request

错误码

请参见[错误码](#)。

4.16 多账号管理

4.16.1 查询账号列表

功能介绍

查询账号列表

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/system/multi-account/accounts

表 4-864 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-865 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
limit	是	Integer	参数解释： 查询返回记录的数量限制 约束限制： 不涉及 取值范围： 1-1024 默认取值： 不涉及
offset	是	Integer	参数解释： 偏移量，表示查询该偏移量后面的记录 约束限制： 不涉及 取值范围： 0 - 1024 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-866 响应 Body 参数

参数	参数类型	描述
data	PageDataAccountVO object	

表 4-867 PageDataAccountVO

参数	参数类型	描述
limit	Integer	参数解释： 数量限制 取值范围： 不涉及
offset	Integer	参数解释： 偏移量 取值范围： 不涉及
records	Array of AccountVO objects	参数解释： 账号列表 取值范围： 不涉及
total	Long	参数解释： 总数 取值范围： 不涉及

表 4-868 AccountVO

参数	参数类型	描述
account_id	String	参数解释： 账号ID 取值范围： 不涉及
account_name	String	参数解释： 账号名称 取值范围： 不涉及

参数	参数类型	描述
eip_count_protected	Integer	参数解释： 防护的EIP数量 取值范围： 不涉及
eip_count_total	Integer	参数解释： EIP总数 取值范围： 不涉及
eip_count_unprotected	Integer	参数解释： 未开启防护的EIP数量 取值范围： 不涉及
organization_id	String	参数解释： 组织ID 取值范围： 不涉及
project_id	String	参数解释： 项目ID 取值范围： 不涉及
project_name	String	参数解释： 项目名称 取值范围： 不涉及

状态码：400

表 4-869 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及

参数	参数类型	描述
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询账号列表，项目ID为176d761dc156471c9c9b227376594160，防火墙ID为ea0e700f-7ae6-47c1-a8d0-ed4b1acd8e8e，数量限制为50，偏移量为0

```
https://{Endpoint}/v1/176d761dc156471c9c9b227376594160/system/multi-account/accounts?limit=50&offset=0&fw_instance_id=ea0e700f-7ae6-47c1-a8d0-ed4b1acd8e8e
```

响应示例

状态码：200

OK

```
{
  "data": {
    "limit": 50,
    "offset": 0,
    "records": [ {
      "account_id": "09bb24e6f280d23d0f9fc0104b901480",
      "account_name": "SCC_CNFW_c00617156",
      "eip_count_protected": 0,
      "eip_count_total": 36,
      "eip_count_unprotected": 36,
      "organization_id": "o-r0b4bh2jcr1ke7yn70uhwlp5d0ktwib",
      "project_id": "09bb24e6fe80d23d2fa2c010b53b418c",
      "project_name": "cn-north-7"
    } ],
    "total": 1
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

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 ListAccountsSolution {

    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();
        ListAccountsRequest request = new ListAccountsRequest();
        try {
            ListAccountsResponse response = client.listAccounts(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 = ListAccountsRequest()
    response = client.list_accounts(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.ListAccountsRequest{}
    response, err := client.ListAccounts(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

错误码

请参见[错误码](#)。

4.16.2 批量添加账号

功能介绍

批量添加账号

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/system/multi-account/accounts

表 4-870 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-871 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-872 请求 Body 参数

参数	是否必选	参数类型	描述
accounts	否	Array of AccountInfo objects	参数解释： 账号列表 约束限制： 不涉及 默认取值： 不涉及

表 4-873 AccountInfo

参数	是否必选	参数类型	描述
account_id	是	String	参数解释： 账号ID，可以通过调用[查询组织账号列表接口]获得，通过返回值中的data.id获得 约束限制： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-874 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 多账号响应信息 取值范围： 不涉及

表 4-875 data

参数	参数类型	描述
id	String	参数解释： 防火墙ID 取值范围： 不涉及
name	String	参数解释： 防火墙名称 取值范围： 不涉及

状态码：400

表 4-876 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

批量添加账号，项目ID为176d761dc156471c9c9b227376594160，防火墙ID为ea0e700f-7ae6-47c1-a8d0-ed4b1acd8e8e，组织ID为o-r0b4bh2jcr1ke7yn70uhwlp5d0ktwisb，账号ID为09bb24e6f280d23d0f9fc0104b901480

```
https://{Endpoint}/v1/176d761dc156471c9c9b227376594160/system/multi-account/accounts?fw_instance_id=ea0e700f-7ae6-47c1-a8d0-ed4b1acd8e8e
```

```
{
  "accounts" : [ {
    "account_id" : "09bb24e6f280d23d0f9fc0104b901480"
  } ]
}
```

响应示例

状态码：200

OK

```
{
  "data" : {
    "id" : "ea0e700f-7ae6-47c1-a8d0-ed4b1acd8e8e",
    "name" : "CFW-9809-1"
  }
}
```

状态码：400

Bad Request

```
{
  "error_code" : "CFW.00200003",
  "error_msg" : "参数错误"
}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.16.3 批量移除账号

功能介绍

批量移除账号

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/system/multi-account/batch-delete

表 4-877 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-878 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-879 请求 Body 参数

参数	是否必选	参数类型	描述
accounts	否	Array of AccountInfo objects	参数解释： 账号列表 约束限制： 不涉及 默认取值： 不涉及

表 4-880 AccountInfo

参数	是否必选	参数类型	描述
account_id	是	String	参数解释： 账号ID，可以通过调用[查询组织账号列表接口]获得，通过返回值中的data.id获得 约束限制： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-881 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 多账号响应信息 取值范围： 不涉及

表 4-882 data

参数	参数类型	描述
id	String	参数解释： 防火墙ID 取值范围： 不涉及
name	String	参数解释： 防火墙名称 取值范围： 不涉及

状态码：400

表 4-883 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

批量移除账号，项目ID为176d761dc156471c9c9b227376594160，防火墙ID为ea0e700f-7ae6-47c1-a8d0-ed4b1acd8e8e，组织ID为o-r0b4bh2jcr1ke7yn70uhwlp5d0ktwisb，账号ID为09bb24e6f280d23d0f9fc0104b901480

```
https://{Endpoint}/v1/176d761dc156471c9c9b227376594160/system/multi-account/batch-delete?fw_instance_id=ea0e700f-7ae6-47c1-a8d0-ed4b1acd8e8e

{
  "accounts": [ {
    "account_id": "09bb24e6f280d23d0f9fc0104b901480"
  } ]
}
```

响应示例

状态码：200

OK

```
{
  "data": {
    "id": "ea0e700f-7ae6-47c1-a8d0-ed4b1acd8e8e",
    "name": "CFW-name"
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.16.4 开启多账号管理

功能介绍

开启多账号管理

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/system/multi-account/enable

表 4-884 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-885 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-886 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 响应信息 取值范围： 不涉及

表 4-887 data

参数	参数类型	描述
id	String	参数解释： 防火墙ID 取值范围： 不涉及
name	String	参数解释： 防火墙名称 取值范围： 不涉及
trust_service_status	Integer	参数解释： 云防火墙可信服务状态 取值范围： 1 已开启

状态码：400

表 4-888 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

开启多账号管理，项目ID为176d761dc156471c9c9b227376594160，防火墙ID为ea0e700f-7ae6-47c1-a8d0-ed4b1acd8e8e

```
https://{Endpoint}/v1/176d761dc156471c9c9b227376594160/system/multi-account/enable?fw_instance_id=ea0e700f-7ae6-47c1-a8d0-ed4b1acd8e8e
```

响应示例

状态码：200

OK

```
{
  "data": {
    "id": "ea0e700f-7ae6-47c1-a8d0-ed4b1acd8e8e",
    "name": "CFW-9809-1"
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

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 EnableMultiAccountSolution {

    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();
EnableMultiAccountRequest request = new EnableMultiAccountRequest();
try {
    EnableMultiAccountResponse response = client.enableMultiAccount(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.valueOf("<YOUR REGION>")) \
        .build()

    try:
        request = EnableMultiAccountRequest()
        response = client.enable_multi_account(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.EnableMultiAccountRequest{}
    response, err := client.EnableMultiAccount(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

错误码

请参见[错误码](#)。

4.16.5 查询组织账号列表

功能介绍

查询组织账号列表

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/system/multi-account/organization-accounts

表 4-889 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-890 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	是否必选	参数类型	描述
parent_id	否	String	参数解释： 父节点（根或组织单元）的唯一标识符（ID） 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
limit	否	Integer	参数解释： 查询返回记录的数量限制 约束限制： 不涉及 取值范围： 1-2000 默认取值： 2000
marker	否	String	参数解释： 分页标记 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-891 响应 Body 参数

参数	参数类型	描述
data	Array of OrganizationAccountInfo objects	参数解释： 查询组织账号响应 取值范围： 不涉及
marker	String	参数解释： 分页标记 取值范围： 不涉及
total	Integer	参数解释： 总数 取值范围： 不涉及

表 4-892 OrganizationAccountInfo

参数	参数类型	描述
delegated	Boolean	参数解释： 是否已添加 取值范围： 不涉及
id	String	参数解释： 账号的唯一标识符 取值范围： 不涉及
name	String	参数解释： 账号名称 取值范围： 不涉及
org_type	String	参数解释： 组织节点类型 取值范围： account

参数	参数类型	描述
parent_id	String	参数解释： 父节点（根或组织单元）的唯一标识符（ID） 取值范围： 不涉及
urn	String	参数解释： 账号的统一资源名称 取值范围： 不涉及

状态码：400

表 4-893 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询组织账号，项目ID为176d761dc156471c9c9b227376594160，防火墙ID为ea0e700f-7ae6-47c1-a8d0-ed4b1acd8e8e，组织ID为o-r0b4bh2jcr1ke7yn70uhwlp5d0ktwisb，账号ID为09bb24e6f280d23d0f9fc0104b901480

https://{Endpoint}/v1/176d761dc156471c9c9b227376594160/system/multi-account/organization-accounts?fw_instance_id=ea0e700f-7ae6-47c1-a8d0-ed4b1acd8e8e

响应示例

状态码：200

OK

```
{
  "data": [{
    "delegated": false,
    "id": "01074115e5d14ee7b4351f22f2deb501",
```

```
"name" : "SCC_CNFW_c00617156_test1",
"org_type" : "account",
"urn" : "organizations::52d4c977dc834434920af54060d74af4:account:o-
r0b4bh2jcr1ke7yn70uhwlp5d0ktwisb/01074115e5d14ee7b4351f22f2deb501"
}]
}
```

状态码：400

Bad Request

```
{
  "error_code" : "CFW.00200003",
  "error_msg" : "参数错误"
}
```

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 ListOrganizationAccountsSolution {

    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();
        ListOrganizationAccountsRequest request = new ListOrganizationAccountsRequest();
        try {
            ListOrganizationAccountsResponse response = client.listOrganizationAccounts(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 = ListOrganizationAccountsRequest()
        response = client.list_organization_accounts(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.ListOrganizationAccountsRequest{}  
response, err := client.ListOrganizationAccounts(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

错误码

请参见[错误码](#)。

4.16.6 查询组织结构

功能介绍

查询组织结构

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/system/multi-account/organization-tree

表 4-894 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-895 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
parent_id	否	String	参数解释： 父节点（根或组织单元）的唯一标识符（ID） 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
limit	否	Integer	参数解释： 查询返回记录的数量限制 约束限制： 不涉及 取值范围： 1-2000 默认取值： 2000
marker	否	String	参数解释： 分页标记 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-896 响应 Body 参数

参数	参数类型	描述
data	Array of OrganizationAccountInfo objects	参数解释： 查询组织账号响应 取值范围： 不涉及
marker	String	参数解释： 分页标记 取值范围： 不涉及

参数	参数类型	描述
total	Integer	参数解释： 总数 取值范围： 不涉及

表 4-897 OrganizationAccountInfo

参数	参数类型	描述
delegated	Boolean	参数解释： 是否已添加 取值范围： 不涉及
id	String	参数解释： 账号的唯一标识符 取值范围： 不涉及
name	String	参数解释： 账号名称 取值范围： 不涉及
org_type	String	参数解释： 组织节点类型 取值范围： account
parent_id	String	参数解释： 父节点（根或组织单元）的唯一标识符（ID） 取值范围： 不涉及
urn	String	参数解释： 账号的统一资源名称 取值范围： 不涉及

状态码：400

表 4-898 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询组织结构，项目ID为176d761dc156471c9c9b227376594160，防火墙ID为ea0e700f-7ae6-47c1-a8d0-ed4b1acd8e8e，组织单元ID为ou-xuyegm9ace9gqucy2sdlw821r22s2xo0

https://{Endpoint}/v1/176d761dc156471c9c9b227376594160/system/multi-account/organization-tree?fw_instance_id=ea0e700f-7ae6-47c1-a8d0-ed4b1acd8e8e&parent_id=ou-xuyegm9ace9gqucy2sdlw821r22s2xo0

响应示例

状态码：200

OK

```
{
  "data": [ {
    "delegated": false,
    "id": "ou-5685cxqjfngoh898df28jm736x4ugvb2",
    "name": "test1",
    "org_type": "unit",
    "parent_id": "ou-xuyegm9ace9gqucy2sdlw821r22s2xo0",
    "urn": "organizations::52d4c977dc834434920af54060d74af4:ou:o-r0b4bh2jcr1ke7yn70uhwlp5d0ktwisb/ou-5685cxqjfngoh898df28jm736x4ugvb2"
  } ],
  "total": 1
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

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

public class ListOrganizationTreeSolution {

    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();
        ListOrganizationTreeRequest request = new ListOrganizationTreeRequest();
        try {
            ListOrganizationTreeResponse response = client.listOrganizationTree(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 huaweicloudsdkcfww.v1.region.cfww_region import CfwRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcfww.v1 import *

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

```
ak = 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 = ListOrganizationTreeRequest()
    response = client.list_organization_tree(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.ListOrganizationTreeRequest{}
    response, err := client.ListOrganizationTree(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

错误码

请参见[错误码](#)。

4.17 时间表管理

4.17.1 查询时间表列表

功能介绍

查询时间表列表

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/schedules

表 4-899 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-900 Query 参数

参数	是否必选	参数类型	描述
object_id	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
name	否	String	参数解释： 时间表名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
desc	否	String	参数解释： 时间表描述 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
limit	是	Integer	参数解释： 查询返回记录的数量限制 约束限制： 不涉及 取值范围： 1-1024 默认取值： 不涉及
offset	是	Integer	参数解释： 偏移量，表示查询该偏移量后面的记录 约束限制： 不涉及 取值范围： 0 - 1024 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-901 响应 Body 参数

参数	参数类型	描述
data	PageDataScheduleVO object	参数解释： 时间表响应信息 取值范围： 不涉及

表 4-902 PageDataScheduleVO

参数	参数类型	描述
limit	Integer	参数解释： 数量限制 取值范围： 不涉及
offset	Integer	参数解释： 偏移量 取值范围： 不涉及
records	Array of ScheduleVO objects	参数解释： 时间表列表 取值范围： 不涉及
total	Long	参数解释： 总数 取值范围： 不涉及

表 4-903 ScheduleVO

参数	参数类型	描述
schedule_id	String	参数解释： 时间表ID 取值范围： 不涉及
name	String	参数解释： 时间表名称 取值范围： 不涉及
description	String	参数解释： 时间表描述 取值范围： 不涉及

参数	参数类型	描述
ref_count	Integer	参数解释： 引用次数 取值范围： 不涉及
periodic	Array of periodic objects	参数解释： 周期计划 取值范围： 不涉及
absolute	absolute object	参数解释： 绝对时间计划 取值范围： 不涉及

表 4-904 periodic

参数	参数类型	描述
type	Integer	参数解释： 周期计划类型 取值范围： 0：每天 1：每周的某几天 2：每周的一段时间
start_time	String	参数解释： 周期计划开始时间 取值范围： 不涉及
end_time	String	参数解释： 周期计划结束时间 取值范围： 不涉及
week_mask	Array of integers	参数解释： 每周的某几天 取值范围： 不涉及

参数	参数类型	描述
start_week	Integer	参数解释： 每周周期计划开始日 取值范围： 不涉及
end_week	Integer	参数解释： 每周周期计划结束日 取值范围： 不涉及

表 4-905 absolute

参数	参数类型	描述
start_time	String	参数解释： 绝对计划开始时间 取值范围： 不涉及
end_time	String	参数解释： 绝对计划结束时间 取值范围： 不涉及

状态码：400

表 4-906 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

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的第一页时间表列表

```
https://{Endpoint}/v1/2349ba469daf4b7daf268bb0261d18b0/schedules?object_id=a37bb4eb-c49e-4e88-bf77-944a75b0ce8a&limit=10&offset=0
```

响应示例

状态码：200

OK

```
{
  "data": {
    "limit": 2,
    "offset": 0,
    "records": [ {
      "schedule_id": "4cf990c4-e71c-4540-8852-06693b76f15b",
      "name": "dfsasf",
      "description": "",
      "ref_count": 0,
      "periodic": [ ],
      "absolute": {
        "start_time": 1752909841782,
        "end_time": 1756365845000
      }
    }, {
      "schedule_id": "8e8bb0b6-bd4b-4970-affe-1fd277d7520f",
      "name": "fdsf",
      "description": "",
      "ref_count": 0,
      "periodic": [ {
        "type": 0,
        "start_time": "00:00:00",
        "end_time": "23:59:59"
      } ],
      "absolute": { }
    } ],
    "total": 2
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.17.2 创建时间表

功能介绍

创建时间表

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/schedule

表 4-907 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-908 请求 Body 参数

参数	是否必选	参数类型	描述
object_id	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
name	是	String	参数解释： 时间表名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
description	否	String	参数解释： 时间表描述 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
periodic	否	Array of periodic objects	参数解释： 周期计划 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
absolute	否	absolute object	参数解释： 绝对时间计划 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-909 periodic

参数	是否必选	参数类型	描述
type	是	Integer	参数解释： 周期计划类型 约束限制： 不涉及 取值范围： 0：每天 1：每周的某几天 2：每周的一段时间 默认取值： 不涉及

参数	是否必选	参数类型	描述
start_time	是	String	参数解释： 周期计划开始时间 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
end_time	是	String	参数解释： 周期计划结束时间 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
week_mask	否	Array of integers	参数解释： 每周的某几天 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
start_week	否	Integer	参数解释： 每周周期计划开始日 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
end_week	否	Integer	参数解释： 每周周期计划结束日 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-910 absolute

参数	是否必选	参数类型	描述
start_time	否	Long	参数解释： 绝对计划开始时间 约束限制： 不涉及 取值范围： 时间戳（毫秒级） 默认取值： 不涉及
end_time	否	Long	参数解释： 绝对计划结束时间 约束限制： 不涉及 取值范围： 时间戳（毫秒级） 默认取值： 不涉及

响应参数

状态码：200

表 4-911 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 时间表返回体 取值范围： 不涉及

表 4-912 data

参数	参数类型	描述
id	String	参数解释： 时间表ID 取值范围： 不涉及
name	String	参数解释： 名称 取值范围： 不涉及

状态码：400

表 4-913 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

创建时间表，项目ID为a16df7cf1d094befa6bbc72cbf51e93a，防护对象ID为24988c28-1357-4738-93aa-e6a120b1d0a0，周期计划为每天00:00:00到23:59:59，名称为test，描述为test

```
https://{Endpoint}/v1/a16df7cf1d094befa6bbc72cbf51e93a/schedule
{
  "object_id" : "24988c28-1357-4738-93aa-e6a120b1d0a0",
  "periodic" : [ {
    "type" : 0,
    "start_time" : "00:00:00",
    "end_time" : "23:59:59"
  } ],
  "name" : "test",
  "description" : "test"
}
```

响应示例

状态码：200

OK

```
{
  "data" : {
    "id" : "3ed296c8-96f7-4a6d-a1f8-746d4a8f867a",
    "name" : "test"
  }
}
```

状态码：400

Bad Request

```
{
  "error_code" : "CFW.00200003",
  "error_msg" : "参数错误"
}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.17.3 更新时间表

功能介绍

更新时间表

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/schedule/{schedule_id}

表 4-914 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
schedule_id	是	String	参数解释： 时间表ID，可以通过调用[查询时间表列表接口]获得，通过返回值中的data.records.schedule_id获得 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-915 请求 Body 参数

参数	是否必选	参数类型	描述
object_id	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
name	是	String	参数解释： 时间表名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
description	否	String	参数解释： 时间表描述 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
periodic	否	Array of periodic objects	参数解释： 周期计划 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
absolute	否	absolute object	参数解释： 绝对时间计划 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-916 periodic

参数	是否必选	参数类型	描述
type	是	Integer	参数解释： 周期计划类型 约束限制： 不涉及 取值范围： 0：每天 1：每周的某几天 2：每周的一段时间 默认取值： 不涉及

参数	是否必选	参数类型	描述
start_time	是	String	参数解释： 周期计划开始时间 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
end_time	是	String	参数解释： 周期计划结束时间 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
week_mask	否	Array of integers	参数解释： 每周的某几天 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
start_week	否	Integer	参数解释： 每周周期计划开始日 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
end_week	否	Integer	参数解释： 每周周期计划结束日 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-917 absolute

参数	是否必选	参数类型	描述
start_time	否	Long	参数解释： 绝对计划开始时间 约束限制： 不涉及 取值范围： 时间戳（毫秒级） 默认取值： 不涉及
end_time	否	Long	参数解释： 绝对计划结束时间 约束限制： 不涉及 取值范围： 时间戳（毫秒级） 默认取值： 不涉及

响应参数

状态码：200

表 4-918 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 时间表返回体 取值范围： 不涉及

表 4-919 data

参数	参数类型	描述
id	String	参数解释： 时间表ID 取值范围： 不涉及
name	String	参数解释： 名称 取值范围： 不涉及

状态码：400

表 4-920 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

更新时间表，项目ID为a16df7cf1d094befa6bbc72cbf51e93a，防护对象ID为24988c28-1357-4738-93aa-e6a120b1d0a0，时间表ID为3ed296c8-96f7-4a6d-a1f8-746d4a8f867a，周期计划为每天00:00:00到23:59:59，名称为test，描述为test

```
https://{Endpoint}/v1/a16df7cf1d094bfa6bbc72cbf51e93a/schedule/3ed296c8-96f7-4a6d-a1f8-746d4a8f867a

{
  "object_id" : "24988c28-1357-4738-93aa-e6a120b1d0a0",
  "periodic" : [ {
    "type" : 0,
    "start_time" : "00:00:00",
    "end_time" : "23:59:59"
  } ],
  "name" : "test",
  "description" : "test"
}
```

响应示例

状态码：200

OK

```
{
  "data" : {
    "id" : "3ed296c8-96f7-4a6d-a1f8-746d4a8f867a",
    "name" : "test"
  }
}
```

状态码：400

Bad Request

```
{
  "error_code" : "CFW.00200003",
  "error_msg" : "参数错误"
}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.17.4 删除时间表

功能介绍

删除时间表

调用方法

请参见[如何调用API](#)。

URI

DELETE /v1/{project_id}/schedule/{schedule_id}

表 4-921 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
schedule_id	是	String	参数解释： 时间表ID，可以通过调用[查询时间表列表接口]获得，通过返回值中的data.records.schedule_id获得 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-922 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 时间表返回体 取值范围： 不涉及

表 4-923 data

参数	参数类型	描述
id	String	参数解释： 时间表ID 取值范围： 不涉及
name	String	参数解释： 名称 取值范围： 不涉及

状态码：400

表 4-924 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

删除时间表，项目ID为a16df7cf1d094befa6bbbc72cbf51e93a，时间表ID为3ed296c8-96f7-4a6d-a1f8-746d4a8f867a

```
https://{Endpoint}/v1/a16df7cf1d094befa6bbc72cbf51e93a/schedule/3ed296c8-96f7-4a6d-a1f8-746d4a8f867a
```

响应示例

状态码：200

OK

```
{
  "data": {
    "id": "3ed296c8-96f7-4a6d-a1f8-746d4a8f867a",
    "name": "test"
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.17.5 批量删除时间表

功能介绍

批量删除时间表

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/schedules/batch-delete

表 4-925 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-926 请求 Body 参数

参数	是否必选	参数类型	描述
object_id	是	String	参数解释： 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。 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	是否必选	参数类型	描述
schedule_ids	是	Array of strings	参数解释： 时间表ID列表，可以通过调用[查询时间表列表接口]获得，通过返回值中的data.records.schedule_id获得 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

响应参数

状态码：200

表 4-927 响应 Body 参数

参数	参数类型	描述
data	Array of strings	参数解释： 被删除的时间表ID列表 取值范围： 不涉及

状态码：400

表 4-928 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

批量删除时间表，项目ID为a16df7cf1d094bfa6bbc72cbf51e93a，f防护对象ID为24988c28-1357-4738-93aa-e6a120b1d0a0，时间表ID包括94759e04-ece6-454c-8f43-c47b68ba4d29和cf7bcca4-41c3-4938-b2cf-9dd8b1e0689b

```
https://{Endpoint}/v1/a16df7cf1d094bfa6bbc72cbf51e93a/schedules/batch-delete
{
  "object_id": "24988c28-1357-4738-93aa-e6a120b1d0a0",
  "schedule_ids": [ "94759e04-ece6-454c-8f43-c47b68ba4d29", "cf7bcca4-41c3-4938-b2cf-9dd8b1e0689b" ]
}
```

响应示例

状态码：200

OK

```
{
  "data": [ "94759e04-ece6-454c-8f43-c47b68ba4d29", "cf7bcca4-41c3-4938-b2cf-9dd8b1e0689b" ]
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.18 安全报告

4.18.1 查询安全报告模板列表

功能介绍

查询安全报告模板列表

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/report-profile

表 4-929 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-930 Query 参数

参数	是否必选	参数类型	描述
category	否	String	参数解释： 报告类型 约束限制： 不涉及 取值范围： daily 日报 weekly 周报 custom 自定义报告 默认取值： 不涉及

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
limit	是	Integer	参数解释： 查询返回记录的数量限制 约束限制： 不涉及 取值范围： 1-1024 默认取值： 不涉及
offset	是	Integer	参数解释： 偏移量，表示查询该偏移量后面的记录 约束限制： 不涉及 取值范围： 0 - 1024 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-931 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 查询模板列表响应 取值范围： 不涉及

表 4-932 data

参数	参数类型	描述
limit	Integer	参数解释： 数量限制 取值范围： 不涉及
offset	Integer	参数解释： 偏移量 取值范围： 不涉及
records	Array of ReportProfileVO objects	参数解释： 模板列表 取值范围： 不涉及
total	Integer	参数解释： 总数 取值范围： 不涉及

表 4-933 ReportProfileVO

参数	参数类型	描述
profile_id	String	参数解释： 模板ID 取值范围： 不涉及

参数	参数类型	描述
name	String	参数解释： 模板名称 取值范围： 不涉及
category	String	参数解释： 模板类型 取值范围： daily 日报 weekly 周报 custom 自定义报告
status	Integer	参数解释： 启用状态 取值范围： 不涉及
report_id	String	参数解释： 最新的报告的ID 取值范围： 不涉及
last_time	Long	参数解释： 最新的报告的生成时间，毫秒级时间戳 取值范围： 不涉及

状态码：400

表 4-934 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询项目ID为eb7accd78ea845078275b2ad5280a109防火墙ID为8a996d43-75bd-48b4-b1b8-de0706b72b8f安全报告模板列表，偏移量为0，返回数量限制为10

```
https://{Endpoint}/v1/eb7accd78ea845078275b2ad5280a109/report-profile?fw_instance_id=8a996d43-75bd-48b4-b1b8-de0706b72b8f&offset=0&limit=10
```

响应示例

状态码：200

OK

```
{
  "data": {
    "limit": 10,
    "offset": 0,
    "records": [ {
      "profile_id": "53a9d352-4b35-4703-801a-0dbcfc306fc5",
      "name": "test",
      "category": "custom",
      "status": 1,
      "report_id": "2ff069b3-1ef8-4a6e-9aaf-d7308eeba826",
      "last_time": 1749814939598
    } ],
    "total": 1
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

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 ListReportProfilesSolution {

    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();
ListReportProfilesRequest request = new ListReportProfilesRequest();
try {
    ListReportProfilesResponse response = client.listReportProfiles(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

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 = ListReportProfilesRequest()
        response = client.list_report_profiles(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.ListReportProfilesRequest{}
    response, err := client.ListReportProfiles(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

错误码

请参见[错误码](#)。

4.18.2 创建安全报告模板

功能介绍

创建安全报告模板

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/report-profile

表 4-935 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-936 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-937 请求 Body 参数

参数	是否必选	参数类型	描述
category	是	String	参数解释： 报告类型 约束限制： 不涉及 取值范围： daily 日报 weekly 周报 custom 自定义报告 默认取值： 不涉及
description	否	String	参数解释： 描述信息 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
name	是	String	参数解释： 模板名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
send_period	否	Integer	参数解释： 发送时间，日报和周报需要设置 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
send_week_day	否	Integer	参数解释： 发送星期，周报需要设置 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
statistic_period	否	StatisticPeriod object	参数解释： 统计周期，自定义报告需要设置 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
status	否	Integer	参数解释： 启用状态 约束限制： 不涉及 取值范围： 0 关闭 1 启用 默认取值： 不涉及
topic_urn	是	String	参数解释： 通知群组 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
subscription_type	否	Integer	参数解释： 通知方式 约束限制： 不涉及 取值范围： 0 SMN通知方式 1 不需要通知 默认取值： 不涉及

表 4-938 StatisticPeriod

参数	是否必选	参数类型	描述
end_time	否	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
start_time	否	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及

响应参数

状态码：200

表 4-939 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 响应信息 取值范围： 不涉及

表 4-940 data

参数	参数类型	描述
id	String	参数解释： 模板ID 取值范围： 不涉及
name	String	参数解释： 模板名称 取值范围： 不涉及

状态码：400

表 4-941 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

创建安全报告模板，项目ID为eb7accd78ea845078275b2ad5280a109，防火墙ID为8a996d43-75bd-48b4-b1b8-de0706b72b8f，模板名称为test，类型为日报，发送的topic名称为test，urn为urn:smn:cn-north-4:eb7accd78ea845078275b2ad5280a109:test，发送时间为每天0点

```
https://{Endpoint}/v1/eb7accd78ea845078275b2ad5280a109/report-profile?
fw_instance_id=8a996d43-75bd-48b4-b1b8-de0706b72b8f

{
  "name" : "test",
  "category" : "daily",
  "topic_urn" : "urn:smn:cn-north-4:eb7accd78ea845078275b2ad5280a109:test",
  "send_period" : 0
}
```

响应示例

状态码：200

OK

```
{
  "data" : {
    "id" : "f9116386-4ef9-40b5-bc2e-bdd7c17b3650",
    "name" : "fgsd"
  }
}
```

状态码：400

Bad Request

```
{
  "error_code" : "CFW.00200003",
  "error_msg" : "参数错误"
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建安全报告模板，项目ID为eb7accd78ea845078275b2ad5280a109，防火墙ID为8a996d43-75bd-48b4-b1b8-de0706b72b8f，模板名称为test，类型为日报，发送的topic名称为test，urn为urn:smn:cn-north-4:eb7accd78ea845078275b2ad5280a109:test，发送时间为每天0点

```
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 CreateReportProfileSolution {

    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();
CreateReportProfileRequest request = new CreateReportProfileRequest();
CreateReportProfileDto body = new CreateReportProfileDto();
body.withTopicUrn("urn:smn:cn-north-4:eb7accd78ea845078275b2ad5280a109:test");
body.withSendPeriod(0);
body.withName("test");
body.withCategory(CreateReportProfileDto.CategoryEnum.fromValue("daily"));
request.withBody(body);
try {
    CreateReportProfileResponse response = client.createReportProfile(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

创建安全报告模板，项目ID为eb7accd78ea845078275b2ad5280a109，防火墙ID为8a996d43-75bd-48b4-b1b8-de0706b72b8f，模板名称为test，类型为日报，发送的topic名称为test，urn为urn:smn:cn-north-4:eb7accd78ea845078275b2ad5280a109:test，发送时间为每天0点

```
# 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 = CreateReportProfileRequest()
request.body = CreateReportProfileDto(
    topic_urn="urn:smn:cn-north-4:eb7accd78ea845078275b2ad5280a109:test",
    send_period=0,
    name="test",
    category="daily"
)
response = client.create_report_profile(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

创建安全报告模板，项目ID为eb7accd78ea845078275b2ad5280a109，防火墙ID为8a996d43-75bd-48b4-b1b8-de0706b72b8f，模板名称为test，类型为日报，发送的topic名称为test，urn为urn:smn:cn-north-4:eb7accd78ea845078275b2ad5280a109:test，发送时间为每天0点

```
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.CreateReportProfileRequest{}
    sendPeriodCreateReportProfileDto := int32(0)
    request.Body = &model.CreateReportProfileDto{
        TopicUrn: "urn:smn:cn-north-4:eb7accd78ea845078275b2ad5280a109:test",
        SendPeriod: &sendPeriodCreateReportProfileDto,
        Name: "test",
        Category: model.GetCreateReportProfileDtoCategoryEnum().DAILY,
    }
    response, err := client.CreateReportProfile(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

错误码

请参见[错误码](#)。

4.18.3 获取安全报告模板

功能介绍

获取安全报告模板

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/report-profile/{report_profile_id}

表 4-942 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
report_profile_id	是	String	参数解释： 安全报告模板ID，可以通过调用[查询安全报告模板列表接口]获得，通过返回值中的data.records.profile_id获得 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-943 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-944 响应 Body 参数

参数	参数类型	描述
data	ReportProfileInfoVO object	

表 4-945 ReportProfileInfoVO

参数	参数类型	描述
category	String	参数解释： 报告类型 取值范围： daily 日报 weekly 周报 custom 自定义报告
description	String	参数解释： 描述信息 取值范围： 不涉及
name	String	参数解释： 模板名称 取值范围： 不涉及
send_period	Integer	参数解释： 发送时间 取值范围： 不涉及
send_week_day	Integer	参数解释： 发送星期 取值范围： 不涉及

参数	参数类型	描述
statistic_period	StatisticPeriod object	参数解释： 统计周期，自定义报告需要设置 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
status	Integer	参数解释： 启用状态 取值范围： 0 关闭 1 启用
topic_name	String	参数解释： 通知主题名称 取值范围： 不涉及
topic_urn	String	参数解释： 通知主题urn 取值范围： 不涉及
subscription_type	Integer	参数解释： 通知方式 取值范围： 0 SMN通知方式 1 不需要通知

表 4-946 StatisticPeriod

参数	参数类型	描述
end_time	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及

参数	参数类型	描述
start_time	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及

状态码：400

表 4-947 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询安全报告模板详情，项目ID为eb7accd78ea845078275b2ad5280a109，防火墙ID为8a996d43-75bd-48b4-b1b8-de0706b72b8f，模板ID为de068699-a2d6-4f3d-8e14-c0ac16ff364e

https://{Endpoint}/v1/eb7accd78ea845078275b2ad5280a109/report-profile/de068699-a2d6-4f3d-8e14-c0ac16ff364e?fw_instance_id=8a996d43-75bd-48b4-b1b8-de0706b72b8f

响应示例

状态码：200

OK

```
{
  "data" : {
    "category" : "daily",
    "name" : "test",
    "send_period" : 16,
    "status" : 1,
    "topic_name" : "test",
```

```
"topic_urn" : "urn:smn:cn-north-4:eb7accd78ea845078275b2ad5280a109:test"
}
}
```

状态码：400

Bad Request

```
{
  "error_code" : "CFW.00200003",
  "error_msg" : "参数错误"
}
```

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 ShowReportProfileSolution {

    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();
        ShowReportProfileRequest request = new ShowReportProfileRequest();
        request.withReportProfileId("{report_profile_id}");
        try {
            ShowReportProfileResponse response = client.showReportProfile(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 = ShowReportProfileRequest()  
        request.report_profile_id = "{report_profile_id}"  
        response = client.show_report_profile(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.ShowReportProfileRequest{  
    request.ReportProfileId = "{report_profile_id}"  
}  
response, err := client.ShowReportProfile(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

错误码

请参见[错误码](#)。

4.18.4 更新安全报告模板

功能介绍

更新安全报告模板

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/report-profile/{report_profile_id}

表 4-948 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
report_profile_id	是	String	参数解释： 安全报告模板ID，可以通过调用[查询安全报告模板列表接口]获得，通过返回值中的data.records.profile_id获得 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-949 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-950 请求 Body 参数

参数	是否必选	参数类型	描述
description	否	String	参数解释： 模板描述 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
name	否	String	参数解释： 模板名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
send_period	否	Integer	参数解释： 发送时间，日报和周报需要设置 约束限制： 不涉及 取值范围： 0-23 默认取值： 不涉及
send_week_day	否	Integer	参数解释： 发送星期，周报需要设置 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
statistic_period	否	StatisticPeriod object	参数解释： 统计周期，自定义报告需要设置 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
status	否	Integer	参数解释： 启用状态 约束限制： 不涉及 取值范围： 0 关闭 1 启用 默认取值： 不涉及
topic_urn	否	String	参数解释： 通知群组 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
subscription_type	否	Integer	参数解释： 通知方式 约束限制： 不涉及 取值范围： 0 SMN通知方式 1 不需要通知 默认取值： 不涉及

表 4-951 StatisticPeriod

参数	是否必选	参数类型	描述
end_time	否	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
start_time	否	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及

响应参数

状态码：200

表 4-952 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 响应信息 取值范围： 不涉及

表 4-953 data

参数	参数类型	描述
id	String	参数解释： 模板ID 取值范围： 不涉及

参数	参数类型	描述
name	String	参数解释： 模板名称 取值范围： 不涉及

状态码：400

表 4-954 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

更新安全报告模板，项目ID为eb7accd78ea845078275b2ad5280a109，防火墙ID为8a996d43-75bd-48b4-b1b8-de0706b72b8f，模板名称为test，类型为日报，发送的topic名称为test，urn为urn:smn:cn-north-4:eb7accd78ea845078275b2ad5280a109:test，发送时间为每天16点

https://{Endpoint}/v1/eb7accd78ea845078275b2ad5280a109/report-profile/de068699-a2d6-4f3d-8e14-c0ac16ff364e?fw_instance_id=8a996d43-75bd-48b4-b1b8-de0706b72b8f

```
{
  "name": "test",
  "topic_urn": "urn:smn:cn-north-4:eb7accd78ea845078275b2ad5280a109:test",
  "send_period": 16
}
```

响应示例

状态码：200

OK

```
{
  "data": {
    "id": "de068699-a2d6-4f3d-8e14-c0ac16ff364e",
    "name": "test"
  }
}
```

状态码：400

Bad Request

```
{
  "error_code" : "CFW.00200003",
  "error_msg" : "参数错误"
}
```

SDK 代码示例

SDK代码示例如下。

Java

更新安全报告模板，项目ID为eb7accd78ea845078275b2ad5280a109，防火墙ID为8a996d43-75bd-48b4-b1b8-de0706b72b8f，模板名称为test，类型为日报，发送的topic名称为test，urn为urn:smn:cn-north-4:eb7accd78ea845078275b2ad5280a109:test，发送时间为每天16点

```
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 UpdateReportProfileSolution {

    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();
        UpdateReportProfileRequest request = new UpdateReportProfileRequest();
        request.withReportProfileId("{report_profile_id}");
        UpdateReportProfileDto body = new UpdateReportProfileDto();
        body.withTopicUrn("urn:smn:cn-north-4:eb7accd78ea845078275b2ad5280a109:test");
        body.withSendPeriod(16);
        body.withName("test");
        request.withBody(body);
        try {
            UpdateReportProfileResponse response = client.updateReportProfile(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        }
    }
}
```



```
        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

更新安全报告模板，项目ID为eb7accd78ea845078275b2ad5280a109，防火墙ID为8a996d43-75bd-48b4-b1b8-de0706b72b8f，模板名称为test，类型为日报，发送的topic名称为test，urn为urn:smn:cn-north-4:eb7accd78ea845078275b2ad5280a109:test，发送时间为每天16点

```
# 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 = UpdateReportProfileRequest()
        request.report_profile_id = "{report_profile_id}"
        request.body = UpdateReportProfileDto(
            topic_urn="urn:smn:cn-north-4:eb7accd78ea845078275b2ad5280a109:test",
            send_period=16,
            name="test"
        )
        response = client.update_report_profile(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

更新安全报告模板，项目ID为eb7accd78ea845078275b2ad5280a109，防火墙ID为8a996d43-75bd-48b4-b1b8-de0706b72b8f，模板名称为test，类型为日报，发送的topic名称为test，urn为urn:smn:cn-north-4:eb7accd78ea845078275b2ad5280a109:test，发送时间为每天16点

```
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.UpdateReportProfileRequest{}
    request.ReportProfileId = "{report_profile_id}"
    topicUrnUpdateReportProfileDto:= "urn:smn:cn-north-4:eb7accd78ea845078275b2ad5280a109:test"
    sendPeriodUpdateReportProfileDto:= int32(16)
    nameUpdateReportProfileDto:= "test"
    request.Body = &model.UpdateReportProfileDto{
        TopicUrn: &topicUrnUpdateReportProfileDto,
        SendPeriod: &sendPeriodUpdateReportProfileDto,
        Name: &nameUpdateReportProfileDto,
    }
    response, err := client.UpdateReportProfile(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

错误码

请参见[错误码](#)。

4.18.5 删除安全报告模板

功能介绍

删除安全报告模板

调用方法

请参见[如何调用API](#)。

URI

DELETE /v1/{project_id}/report-profile/{report_profile_id}

表 4-955 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
report_profile_id	是	String	参数解释： 安全报告模板ID，可以通过调用[查询安全报告模板列表接口]获得，通过返回值中的data.records.profile_id获得 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-956 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-957 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 响应信息 取值范围： 不涉及

表 4-958 data

参数	参数类型	描述
id	String	参数解释： 模板ID 取值范围： 不涉及

参数	参数类型	描述
name	String	参数解释： 模板名称 取值范围： 不涉及

状态码：400

表 4-959 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

删除安全报告模板，项目ID为eb7accd78ea845078275b2ad5280a109，防火墙ID为8a996d43-75bd-48b4-b1b8-de0706b72b8f，模板ID为de068699-a2d6-4f3d-8e14-c0ac16ff364e

```
https://{Endpoint}/v1/eb7accd78ea845078275b2ad5280a109/report-profile/f9116386-4ef9-40b5-bc2e-bdd7c17b3650?fw_instance_id=8a996d43-75bd-48b4-b1b8-de0706b72b8f
```

响应示例

状态码：200

OK

```
{
  "data": {
    "id": "f9116386-4ef9-40b5-bc2e-bdd7c17b3650",
    "name": "test"
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
}
```

```
"error_msg": "参数错误"
}
```

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 DeleteReportProfileSolution {

    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();
        DeleteReportProfileRequest request = new DeleteReportProfileRequest();
        request.withReportProfileId("{report_profile_id}");
        try {
            DeleteReportProfileResponse response = client.deleteReportProfile(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 = DeleteReportProfileRequest()
        request.report_profile_id = "{report_profile_id}"
        response = client.delete_report_profile(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.DeleteReportProfileRequest{}
    request.ReportProfileId = "{report_profile_id}"
    response, err := client.DeleteReportProfile(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

错误码

请参见[错误码](#)。

4.18.6 查询安全报告发送历史

功能介绍

查询安全报告发送历史

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/report/history/{report_profile_id}

表 4-960 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
report_profile_id	是	String	参数解释： 安全报告模板ID，可以通过调用[查询安全报告模板列表接口]获得，通过返回值中的data.records.profile_id获得 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-961 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	是否必选	参数类型	描述
limit	是	Integer	参数解释： 查询返回记录的数量限制 约束限制： 不涉及 取值范围： 1-1024 默认取值： 不涉及
offset	是	Integer	参数解释： 偏移量，表示查询该偏移量后面的记录 约束限制： 不涉及 取值范围： 0 - 1024 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-962 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 查询报告发送历史数据 取值范围： 不涉及

表 4-963 data

参数	参数类型	描述
limit	Integer	参数解释： 数量限制 取值范围： 不涉及
offset	Integer	参数解释： 偏移量 取值范围： 不涉及
records	Array of ReportHistoryVO objects	参数解释： 发送历史列表 取值范围： 不涉及
total	Integer	参数解释： 总数 取值范围： 不涉及

表 4-964 ReportHistoryVO

参数	参数类型	描述
report_id	String	参数解释： 报告ID 取值范围： 不涉及
send_time	Long	参数解释： 发送时间 取值范围： 不涉及

参数	参数类型	描述
statistic_period	StatisticPeriod object	参数解释： 统计周期，自定义报告需要设置 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-965 StatisticPeriod

参数	参数类型	描述
end_time	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
start_time	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及

状态码：400

表 4-966 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询安全报告模板发送历史，项目ID为eb7accd78ea845078275b2ad5280a109，防火墙ID为8a996d43-75bd-48b4-b1b8-de0706b72b8f，模板ID为de068699-a2d6-4f3d-8e14-c0ac16ff364e，偏移量为0，返回数量限制为50

https://{Endpoint}/v1/eb7accd78ea845078275b2ad5280a109/report/history/de068699-a2d6-4f3d-8e14-c0ac16ff364e?fw_instance_id=8a996d43-75bd-48b4-b1b8-de0706b72b8f&limit=50&offset=0

响应示例

状态码：200

OK

```
{
  "data": {
    "limit": 0,
    "offset": 0,
    "records": [ {
      "report_id": "b5e342d5-6232-41a1-bed4-162965d43bb8",
      "send_time": 1751963401402,
      "statistic_period": {
        "end_time": 1751903999999,
        "start_time": 1751817600000
      }
    }
  ],
  "total": 27
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.18.7 查询安全报告

功能介绍

查询安全报告

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/report/{report_id}

表 4-967 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

参数	是否必选	参数类型	描述
report_id	是	String	参数解释： 安全报告ID，可以通过调用[查询安全报告发送历史接口]获得，通过返回值中的data.records.report_id获得 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-968 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
report_profile_id	是	String	参数解释： 安全报告模板ID 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-969 响应 Body 参数

参数	参数类型	描述
data	FirewallReport object	

表 4-970 FirewallReport

参数	参数类型	描述
attack_info	AttackReport object	参数解释： 安全事件详情 取值范围： 不涉及
category	String	参数解释： 报告类型 取值范围： 不涉及
internet_firewall	InternetReport object	参数解释： 互联网边界防火墙 取值范围： 不涉及
send_time	Long	参数解释： 发送时间 取值范围： 不涉及
statistic_period	StatisticPeriod object	参数解释： 统计范围 取值范围： 不涉及
vpc_firewall	VpcReport object	参数解释： VPC边界防火墙 取值范围： 不涉及

表 4-971 AttackReport

参数	参数类型	描述
dst_ip	Array of ItemVO objects	参数解释： TOP攻击目的IP 取值范围： 不涉及
ips_mode	Integer	参数解释： 入侵防御状态 取值范围： 不涉及
level	Array of ItemVO objects	参数解释： 攻击等级分布 取值范围： 不涉及
rule	Array of ItemVO objects	参数解释： TOP攻击规则 取值范围： 不涉及
src_ip	Array of ItemVO objects	参数解释： TOP来源IP 取值范围： 不涉及
trend	Array of TrendVO objects	参数解释： 攻击趋势 取值范围： 不涉及
type	Array of ItemVO objects	参数解释： TOP攻击分布 取值范围： 不涉及

表 4-972 InternetReport

参数	参数类型	描述
eip	Eip object	参数解释： EIP防护情况 取值范围： 不涉及
in2out	In2Out object	参数解释： 出云流量 取值范围： 不涉及
out2in	Out2in object	参数解释： 入云流量 取值范围： 不涉及
overview	Overview object	参数解释： 概览 取值范围： 不涉及
traffic_trend	Array of TrendVO objects	参数解释： 流量趋势 取值范围： 不涉及

表 4-973 Eip

参数	参数类型	描述
protected	ChangedVO object	参数解释： 防护情况 取值范围： 不涉及
total	Integer	参数解释： EIP数量 取值范围： 不涉及

表 4-974 In2Out

参数	参数类型	描述
dst_host	Array of ItemVO objects	参数解释： TOP访问域名 取值范围： 不涉及
dst_ip	Array of ItemVO objects	参数解释： TOP访问目的IP 取值范围： 不涉及
dst_port	Array of ItemVO objects	参数解释： TOP访问端口 取值范围： 不涉及
src_ip	Array of ItemVO objects	参数解释： TOP访问源IP 取值范围： 不涉及

表 4-975 Out2in

参数	参数类型	描述
dst_ip	Array of ItemVO objects	参数解释： TOP访问目的IP 取值范围： 不涉及
dst_port	Array of ItemVO objects	参数解释： TOP开放端口 取值范围： 不涉及
src_ip	Array of ItemVO objects	参数解释： TOP访问源IP 取值范围： 不涉及

表 4-976 StatisticPeriod

参数	参数类型	描述
end_time	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及
start_time	Long	参数解释： 结束时间 约束限制： 不涉及 取值范围： 毫秒级时间戳 默认取值： 不涉及

表 4-977 VpcReport

参数	参数类型	描述
app	Array of ItemVO objects	参数解释： TOP应用数量 取值范围： 不涉及
dst_ip	Array of ItemVO objects	参数解释： TOP访问目的IP 取值范围： 不涉及
overview	Overview object	参数解释： 概览 取值范围： 不涉及

参数	参数类型	描述
src_ip	Array of ItemVO objects	参数解释: TOP访问源IP 取值范围: 不涉及
traffic_trend	Array of TrendVO objects	参数解释: 流量趋势 取值范围: 不涉及
vpc	Vpc object	参数解释: vpc防护统计 取值范围: 不涉及

表 4-978 Overview

参数	参数类型	描述
access_policies	AccessPolicy object	参数解释: 访问控制策略 取值范围: 不涉及
assets	ChangedVO object	参数解释: 资产数量 取值范围: 不涉及
attack_event	AttackEvent object	参数解释: 威胁事件 取值范围: 不涉及
traffic_peak	TrendVO object	参数解释: 流量峰值 取值范围: 不涉及

表 4-979 AccessPolicy

参数	参数类型	描述
changed	Integer	参数解释： 变化数量 取值范围： 不涉及
eip	Integer	参数解释： EIP访问控制策略 取值范围： 不涉及
nat	Integer	参数解释： NAT访问控制策略 取值范围： 不涉及
total	Integer	参数解释： 总数 取值范围： 不涉及

表 4-980 AttackEvent

参数	参数类型	描述
changed	Integer	参数解释： 变化数量 取值范围： 不涉及
deny	Integer	参数解释： 阻断数量 取值范围： 不涉及
total	Integer	参数解释： 总数 取值范围： 不涉及

表 4-981 TrendVO

参数	参数类型	描述
agg_time	Long	参数解释： 聚合时间 取值范围： 不涉及
bps	Double	参数解释： 带宽 取值范围： 不涉及
deny	Long	参数解释： 阻断数量 取值范围： 不涉及
in_bps	Double	参数解释： 入方向bps 取值范围： 不涉及
out_bps	Double	参数解释： 出方向bps 取值范围： 不涉及
permit	Long	参数解释： 允许数量 取值范围： 不涉及

表 4-982 ItemVO

参数	参数类型	描述
key	String	参数解释： 聚合项 取值范围： 不涉及

参数	参数类型	描述
name	String	参数解释： 聚合项名称 取值范围： 不涉及
value	Long	参数解释： 统计值 取值范围： 不涉及

表 4-983 Vpc

参数	参数类型	描述
protected	ChangedVO object	参数解释： 防护统计 取值范围： 不涉及
total	Integer	参数解释： 总数 取值范围： 不涉及

表 4-984 ChangedVO

参数	参数类型	描述
changed	Integer	参数解释： 变化数量 取值范围： 不涉及
total	Integer	参数解释： 变化数量 取值范围： 不涉及
value	Integer	参数解释： 数量 取值范围： 不涉及

状态码：400

表 4-985 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

项目ID为eb7accd78ea845078275b2ad5280a109，防火墙ID为8a996d43-75bd-48b4-b1b8-de0706b72b8f，模板ID为e8f43868-d6e7-43a7-aa7f-dda0292f4785，报告ID为b07861af-3468-4d4e-b39b-30c9ca516888，查询安全报告

https://{Endpoint}/v1/eb7accd78ea845078275b2ad5280a109/report/b07861af-3468-4d4e-b39b-30c9ca516888?report_profile_id=e8f43868-d6e7-43a7-aa7f-dda0292f4785&fw_instance_id=8a996d43-75bd-48b4-b1b8-de0706b72b8f

响应示例

状态码：200

OK

```
{
  "data": {
    "attack_info": {
      "dst_ip": [ {
        "key": "120.52.95.246",
        "value": 47
      } ],
      "ips_mode": 1,
      "level": [ {
        "key": "CRITICAL",
        "value": 119
      } ],
      "rule": [ {
        "key": "Realtek Jungle SDK Command Injection Vulnerability (CVE-2021-35394)",
        "value": 30
      } ],
      "src_ip": [ {
        "key": "120.46.56.58",
        "value": 66
      } ],
      "trend": [ {
        "agg_time": 1748966700000,
        "deny": 2,
```

```
"permit" : 1
}],
"type" : [ {
  "key" : "Vulnerability Exploit Attack",
  "value" : 30
}]
},
"category" : "custom",
"internet_firewall" : {
  "eip" : {
    "protected" : {
      "value" : 3
    },
    "total" : 5
  },
  "in2out" : {
    "dst_host" : [ {
      "key" : "repo.huaweicloud.com",
      "value" : 58511700
    } ],
    "dst_port" : [ {
      "key" : "22",
      "value" : 12062960684
    } ],
    "src_ip" : [ {
      "key" : "172.18.0.90",
      "value" : 12062961860
    } ]
  },
  "out2in" : {
    "dst_ip" : [ {
      "key" : "120.46.56.58",
      "value" : 41828474
    } ],
    "dst_port" : [ {
      "key" : "23",
      "value" : 925712
    } ],
    "src_ip" : [ {
      "key" : "114.116.217.114",
      "value" : 4008768
    } ]
  },
  "overview" : {
    "access_policies" : {
      "eip" : 18,
      "nat" : 105,
      "total" : 123
    },
    "assets" : {
      "total" : 5
    },
    "attack_event" : {
      "deny" : 30,
      "total" : 119
    },
    "traffic_peak" : {
      "in_bps" : 2381284,
      "out_bps" : 3977
    }
  },
  "traffic_trend" : [ {
    "agg_time" : 1748966400000,
    "bps" : 0,
    "in_bps" : 3238,
    "out_bps" : 808
  } ]
},
"send_time" : 1749698298872,
```

```
"statistic_period": {
  "end_time": 1749657599999,
  "start_time": 1748966400000
},
"vpc_firewall": {
  "app": [ {
    "key": "SSH",
    "value": 33901440538
  }, {
    "key": "PING",
    "value": 784
  } ],
  "dst_ip": [ {
    "key": "172.17.0.192",
    "value": 33900674752
  }, {
    "key": "172.18.0.90",
    "value": 766570
  } ],
  "overview": {
    "access_policies": {
      "total": 10
    },
    "assets": {
      "total": 17
    },
    "attack_event": {
      "deny": 0,
      "total": 0
    },
    "traffic_peak": {
      "bps": 711475757
    }
  },
  "src_ip": [ {
    "key": "172.18.0.90",
    "value": 33900674752
  }, {
    "key": "172.17.0.192",
    "value": 766570
  } ],
  "traffic_trend": [ {
    "agg_time": 1748966400000,
    "bps": 0,
    "in_bps": 0,
    "out_bps": 0
  } ],
  "vpc": {
    "protected": {
      "value": 2
    },
    "total": 17
  }
}
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```

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 ShowFirewallReportSolution {

    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();
        ShowFirewallReportRequest request = new ShowFirewallReportRequest();
        request.withReportId("{report_id}");
        try {
            ShowFirewallReportResponse response = client.showFirewallReport(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 = ShowFirewallReportRequest()
    request.report_id = "{report_id}"
    response = client.show_firewall_report(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    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.ShowFirewallReportRequest{}
    request.ReportId = "{report_id}"
    response, err := client.ShowFirewallReport(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

错误码

请参见[错误码](#)。

4.19 IPS 白名单管理

4.19.1 更新 IPS 白名单

功能介绍

更新IPS白名单

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/cfw/{fw_instance_id}/ips/whitelist/{list_id}

表 4-986 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
list_id	是	String	参数解释 白名单id 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

请求参数

表 4-987 请求 Body 参数

参数	是否必选	参数类型	描述
source_addresses	是	String	参数解释： 白名单源地址 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
dest_address	是	String	参数解释： 目的地址 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
ip_version	是	Integer	参数解释： IP类型 约束限制： 不涉及 取值范围： 4：表示IP类型为IPv4 6：表示IP类型为IPv6 默认取值： 不涉及
name	是	String	参数解释： 白名单名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
ips_id	否	String	参数解释： ips规则id，不填写时默认为全部ips规则 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
effective_scopes	是	Array of strings	参数解释： 生效范围：NAT EIP VPC HCS场景中还有VGW信息 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-988 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释 返回数据 取值范围： 不涉及

表 4-989 data

参数	参数类型	描述
ips_whitelist_id	String	参数解释 白名单id 取值范围： 不涉及

状态码：400

表 4-990 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

更新ID为4bf99009-e595-4cd4-bde1-e88c18861ec5的IPS白名单

```
https://{Endpoint}/v1/eefb11b22d5944f2aa067954516df1cf/cfw/175572ed-67e8-4837-9776-e2e07080ea45/ips/whitelist/4bf99009-e595-4cd4-bde1-e88c18861ec5
{
  "name" : "test-002",
  "ip_version" : 4,
  "source_address" : "8.8.8.0/32",
  "dest_address" : "8.8.8.0/32",
  "effective_scopes" : [ "EIP" ]
}
```

响应示例

状态码：200

响应数据

```
{ }
```

状态码

状态码	描述
200	响应数据
400	Bad Request

错误码

请参见[错误码](#)。

4.19.2 查询 IPS 白名单列表详情

功能介绍

查询IPS白名单列表详情

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/cfw/{fw_instance_id}/ips/whitelist

表 4-991 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-992 Query 参数

参数	是否必选	参数类型	描述
source_addresses	否	String	参数解释： 白名单源地址 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
dest_address	否	String	参数解释： 目的地址 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
name	否	String	参数解释： 白名单名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
limit	是	Integer	参数解释： 每页大小 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
offset	是	Integer	参数解释： 偏移量 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-993 响应 Body 参数

参数	参数类型	描述
data	ListIpsWhiteLists VO object	参数解释： IPS白名单响应体 取值范围： 不涉及

表 4-994 ListIpsWhiteListsVO

参数	参数类型	描述
limit	Integer	参数解释： 每页数量 取值范围： 不涉及
offset	Integer	参数解释： 偏移量 取值范围： 不涉及

参数	参数类型	描述
total	Long	参数解释： 总数 取值范围： 不涉及
records	Array of IpsWhiteListVO objects	参数解释： IPS白名单列表 取值范围： 不涉及

表 4-995 IpsWhiteListVO

参数	参数类型	描述
source_address	String	参数解释： 白名单源地址 取值范围： 不涉及
dest_address	String	参数解释： 目的地址 取值范围： 不涉及
name	String	参数解释： 白名单名称 取值范围： 不涉及
list_id	String	参数解释： IPS白名单ID 取值范围： 不涉及
effective_scopes	Array of strings	参数解释： 生效范围：NAT EIP VPC HCS场景中还有VGW信息 取值范围： 不涉及

参数	参数类型	描述
ips_id	String	参数解释： 生效范围：IPS规则ID，all代表所有IPS 取值范围： 不涉及
ip_version	Integer	参数解释： IP类型 取值范围： 4：表示IP类型为IPv4 6：表示IP类型为IPv6
description	String	参数解释： 白名单描述 取值范围： 不涉及

状态码：400

表 4-996 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询项目ID为eefb11b22d5944f2aa067954516df1cf，防火墙实例ID为175572ed-67e8-4837-9776-e2e07080ea45的IPS白名单列表

https://{Endpoint}/v1/eefb11b22d5944f2aa067954516df1cf/cfw/175572ed-67e8-4837-9776-e2e07080ea45/ips/whitelist?limit=1024&offset=0

响应示例

状态码：200

返回值

```
{
  "data": {
    "limit": 1024,
    "offset": 0,
    "total": 2,
    "records": [ {
      "source_address": "0.0.0.0/0",
      "dest_address": "0.0.0.0/0",
      "name": "test-001",
      "list_id": "76fad07b-8304-486f-b802-943be05a3122",
      "effective_scopes": [ "EIP" ],
      "ip_version": 4
    }, {
      "source_address": "8.8.8.0/32",
      "dest_address": "8.8.8.0/32",
      "name": "test-002",
      "list_id": "4bf99009-e595-4cd4-bde1-e88c18861ec5",
      "effective_scopes": [ "EIP" ],
      "ip_version": 4
    } ]
  }
}
```

状态码

状态码	描述
200	返回值
400	Bad Request

错误码

请参见[错误码](#)。

4.19.3 批量添加 IPS 白名单

功能介绍

批量添加IPS白名单

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/cfw/{fw_instance_id}/ips/whitelist/batch-create

表 4-997 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-998 请求 Body 参数

参数	是否必选	参数类型	描述
ips_white_list_dto_list	是	Array of IpsWhiteListDto objects	参数解释： 添加的IPS白名单列表 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-999 IpsWhiteListDto

参数	是否必选	参数类型	描述
source_addresses	是	String	参数解释： 源地址 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
dest_address	是	String	参数解释： 目的地址 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
ip_version	是	Integer	参数解释： IP类型 约束限制： 不涉及 取值范围： 4：表示IP类型为IPv4 6：表示IP类型为IPv6 默认取值： 不涉及
name	是	String	参数解释： 白名单名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
ips_id	否	String	参数解释： IPS规则ID，all代表所有IPS 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
effective_scopes	否	Array of strings	参数解释： 生效范围：NAT EIP VPC HCS场景中还有VGW信息 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-1000 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 创建白名单DATA 取值范围： 不涉及

表 4-1001 data

参数	参数类型	描述
list_ids	Array of strings	参数解释： 创建白名单ID列表 取值范围： 不涉及

状态码：400

表 4-1002 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

创建IPS白名单，其中IPS白名单名称为test-001，IP类型为IPv4，源IP为8.8.8.0/32，目的IP为8.8.8.0/32，生效范围为EIP

```
https://{Endpoint}/v1/eefb11b22d5944f2aa067954516df1cf/cfw/175572ed-67e8-4837-9776-e2e07080ea45/ips/whitelist/batch-create

{
  "ips_white_list_dto_list" : [ {
    "name" : "test-001",
    "ip_version" : 4,
    "source_address" : "8.8.8.0/32",
    "dest_address" : "8.8.8.0/32",
    "effective_scopes" : [ "EIP" ]
  } ]
}
```

响应示例

状态码：200

```
{
  "data" : {
    "list_ids" : [ "76fad07b-8304-486f-b802-943be05a3122" ]
  }
}
```

状态码

状态码	描述
200	
400	Bad Request

错误码

请参见[错误码](#)。

4.19.4 批量删除 IPS 白名单

功能介绍

批量删除IPS白名单

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/cfw/{fw_instance_id}/ips/whitelist/batch-delete

表 4-1003 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-1004 请求 Body 参数

参数	是否必选	参数类型	描述
list_ids	是	Array of strings	参数解释： 要删除的白名单ID列表 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

响应参数

状态码：200

表 4-1005 响应 Body 参数

参数	参数类型	描述
data	data object	参数解释： 删除的白名单DATA 取值范围： 不涉及

表 4-1006 data

参数	参数类型	描述
list_ids	Array of strings	参数解释： 删除的白名单ID列表 取值范围： 不涉及

状态码：400

表 4-1007 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

删除ID为76fad07b-8304-486f-b802-943be05a3122的IPS白名单数据

```
https://{Endpoint}/v1/eefb11b22d5944f2aa067954516df1cf/cfw/175572ed-67e8-4837-9776-e2e07080ea45/ips/whitelist/batch-delete
{
  "list_ids": [ "76fad07b-8304-486f-b802-943be05a3122" ]
}
```

响应示例

状态码：200

响应数据

```
{
  "data": {
    "list_ids": [ "76fad07b-8304-486f-b802-943be05a3122" ]
  }
}
```

状态码

状态码	描述
200	响应数据
400	Bad Request

错误码

请参见[错误码](#)。

4.20 TLS 证书管理

4.20.1 导入 TLS 证书

功能介绍

导入TLS证书

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/certificate

表 4-1008 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-1009 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

表 4-1010 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	参数解释： 证书名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
content	是	String	参数解释： 证书内容 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
private_key	是	String	参数解释： 证书私钥 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	是否必选	参数类型	描述
key	是	String	参数解释： 密码 约束限制： 不涉及 取值范围： 长度为3-16个字符 至少包含以下字符中的2种：大写字母、小写字母、数字、特殊字符`~!@#\$%^&*()-_+= [];:","<.>/? 和空格。 默认取值： 不涉及

响应参数

状态码：200

OK

状态码：400

表 4-1011 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

导入TLS证书到项目ID为eb7accd78ea845078275b2ad5280a109防火墙ID为8a996d43-75bd-48b4-b1b8-de0706b72b8f

https://{Endpoint}/v1/28f403ddd3f141daa6e046e85cb15519/certificate?fw_instance_id=d06dad70-b3be-4480-86bc-b3d139e9938b

响应示例

状态码：400

Bad Request

```
{
  "error_code" : "CFW.00200003",
  "error_msg" : "参数错误"
}
```

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.20.2 获取 TLS 证书

功能介绍

获取TLS证书

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/certificate

表 4-1012 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID，用于明确项目归属，配置后可通过该ID查询项目下资产，可以从调API处获取，也可以从控制台获取。 项目ID获取方式 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

表 4-1013 Query 参数

参数	是否必选	参数类型	描述
fw_instance_id	是	String	参数解释： 防火墙ID，用户创建防火墙实例后产生的唯一ID，配置后可区分不同防火墙，可通过 防火墙ID获取方式 获取 约束限制： 不涉及 取值范围： 32位UUID 默认取值： 不涉及

请求参数

无

响应参数

状态码：200

表 4-1014 响应 Body 参数

参数	参数类型	描述
data	ListTlsCertificateRespData object	参数解释： 获取TLS证书响应DATA 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-1015 ListTlsCertificateRespData

参数	参数类型	描述
name	String	参数解释： 证书名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
key	String	参数解释： 证书密钥 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
certificate_info_s	Array of CertificateInfo objects	参数解释： 证书信息 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-1016 CertificateInfo

参数	参数类型	描述
issuing_unit	IssuingUnit object	参数解释： 证书发布单位 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
validity_period	ValidityPeriod object	参数解释： 有效期信息 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-1017 IssuingUnit

参数	参数类型	描述
common_name	String	参数解释： 名称 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
organization	String	参数解释： 组织 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

参数	参数类型	描述
organization_unit	String	参数解释： 组织单位 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

表 4-1018 ValidityPeriod

参数	参数类型	描述
issuing_date	Long	参数解释： 颁发日期 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及
expiration_date	Long	参数解释： 有效日期 约束限制： 不涉及 取值范围： 不涉及 默认取值： 不涉及

状态码：400

表 4-1019 响应 Body 参数

参数	参数类型	描述
error_code	String	参数解释： 错误码 取值范围： 不涉及
error_msg	String	参数解释： 错误描述 取值范围： 不涉及

请求示例

查询项目ID为5de88e3e18124fa99675fb51ceceaa31,防火墙ID为ef053720-decd-4205-a923-91785d627203的证书列表

https://{Endpoint}/v1/5de88e3e18124fa99675fb51ceceaa31/certificate?fw_instance_id=ef053720-decd-4205-a923-91785d627203

响应示例

状态码：200

OK

```
{
  "data": {
    "name": "AV-CERT",
    "key": "Ga**a@11**",
    "certificate_info_s": [ {
      "issuing_unit": {
        "common_name": "192.168.1.100"
      },
      "validity_period": {
        "issuing_date": 1751133252000,
        "expiration_date": 1782669252000
      }
    }
  ]
}
```

状态码：400

Bad Request

```
{
  "error_code": "CFW.00200003",
  "error_msg": "参数错误"
}
```


状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.21 流量过滤

4.21.1 导入 IP 黑名单用于流量过滤

功能介绍

此接口用来导入IP黑名单，IP列表保存在request的body中，IP列表支持的格式如下：

单个IP地址，例如：100.1.1.10

连续的IP地址段，例如：80.1.1.3-80.1.1.30

掩码格式的网段，例如：6.6.6.0/24

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/ptf/ip-blacklist/import

表 4-1020 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-1021 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

请求参数

表 4-1022 请求 Body 参数

参数	是否必选	参数类型	描述
add_type	否	Integer	IP黑名单导入的方式，0表示增量导入，在原来的基础上追加；1表示全量导入，新导入的IP黑名单会覆盖已有的IP黑名单
ip_blacklist	否	String	IP列表，当前支持不同的IP地址之间使用“,”、“ ”、“;”、“\r\n”、“\n”、“\t”等分隔符进行分隔。
effect_scope	否	Array of integers	生效范围，1表示生效范围为eip, 2表示生效范围为nat, [1 2]表示 生效范围为eip和nat

响应参数

状态码：200

表 4-1023 响应 Body 参数

参数	参数类型	描述
data	Object	导入数据

请求示例

全量方式导入一个IP黑名单，项目id为eefb11b22d5944f2aa067954516df1cf，防护对象 id为812f478c-940e-4e00-824f-7287250d76dc，

生效范围为EIP，IP列表为
“100.1.1.10\n80.1.1.3-80.1.1.30\n6.6.6.0/24\n2001:db8::2-2001:db8::200\n3001:db8::2\n4001:db8::2/64”，v4格式和v6格式的ip可以一起导入，ip之间使用\n分隔。

```
https://{Endpoint}/v1/eefb11b22d5944f2aa067954516df1cf/ptf/ip-blacklist/import?fw_instance_id=812f478c-940e-4e00-824f-7287250d76dc

ImportIpBlacklistRequestBody:
{
  "add_type": 1,
  "ip_blacklist":
    "100.1.1.10\n80.1.1.3-80.1.1.30\n6.6.6.0/24\n2001:db8::2-2001:db8::200\n3001:db8::2\n4001:db8::2/64",
  "effect_scope": [
    1
  ]
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

全量方式导入一个IP黑名单，项目id为eefb11b22d5944f2aa067954516df1cf，防护对象 id为812f478c-940e-4e00-824f-7287250d76dc，

生效范围为EIP，IP列表为

“100.1.1.10\n80.1.1.3-80.1.1.30\n6.6.6.0/24\n2001:db8::2-2001:db8::200\n3001:db8::2\n4001:db8::2/64”，v4格式和v6格式的ip可以一起导入，ip之间使用\n分隔。

```
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 ImportIpBlacklistSolution {

    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();
        ImportIpBlacklistRequest request = new ImportIpBlacklistRequest();
        try {
            ImportIpBlacklistResponse response = client.importIpBlacklist(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

全量方式导入一个IP黑名单，项目id为eefb11b22d5944f2aa067954516df1cf，防护对象 id为812f478c-940e-4e00-824f-7287250d76dc，

生效范围为EIP，IP列表为

“100.1.1.10\n80.1.1.3-80.1.1.30\n6.6.6.0/24\n2001:db8::2-2001:db8::200\n3001:db8::2\n4001:db8::2/64”，v4格式和v6格式的ip可以一起导入，ip之间使用\n分隔。

```
# 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 = ImportIpBlacklistRequest()
        response = client.import_ip_blacklist(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

全量方式导入一个IP黑名单，项目id为eefb11b22d5944f2aa067954516df1cf，防护对象 id为812f478c-940e-4e00-824f-7287250d76dc，

生效范围为EIP，IP列表为

“100.1.1.10\n80.1.1.3-80.1.1.30\n6.6.6.0/24\n2001:db8::2-2001:db8::200\n3001:db8::2\n4001:db8::2/64”，v4格式和v6格式的ip可以一起导入，ip之间使用\n分隔。

```
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.ImportIpBlacklistRequest{}
response, err := client.ImportIpBlacklist(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	导入成功的响应，此响应体在成功的情况下字段都不赋值
401	鉴权认证失败
403	不允许操作
500	内部错误

错误码

请参见[错误码](#)。

4.21.2 获取导入的 IP 黑名单列表信息

功能介绍

获取防火墙实例中已经导入的IP黑名单信息，标准版防火墙只会显示一条EIP的记录，专业版防火墙可能显示EIP、NAT或EIP和NAT的记录，根据导入的情况确定。

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/ptf/ip-blacklist

表 4-1024 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-1025 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
limit	是	Integer	在分页查询的情况下，每页查询的记录条数，范围为1-1024
offset	是	Integer	数据查询的偏移量，在分页查询的时候使用，指定查询记录的起始位置，必须为数字，取值范围为大于等于0

请求参数

无

响应参数

状态码：200

表 4-1026 响应 Body 参数

参数	参数类型	描述
data	PageDataIpBlacklistsVo object	查询IP黑名单列表信息的HTTP响应

表 4-1027 PageDataIpBlacklistsVo

参数	参数类型	描述
limit	Integer	一次查询返回的记录条数，调用接口时赋值

参数	参数类型	描述
offset	Integer	本次查询结果返回后下次查询的偏移
records	Array of IpBlacklistVO objects	查询出的IP黑名单列表信息
total	Integer	防火墙实例中IP黑名单的总数

表 4-1028 IpBlacklistVO

参数	参数类型	描述
name	String	IP黑名单的文件名，对应导出时的文件名
effect_scope	Array of integers	IP黑名单的生效范围，1表示EIP，2表示NAT
import_status	Integer	IP黑名单的导入状态，有三种状态：2：生成中、1：成功、0：失败。
import_time	Long	IP黑名单的导入时间
error_message	String	导入失败的错误信息

请求示例

查询项目ID为eefb11b22d5944f2aa067954516df1cf，防火墙实例ID为812f478c-940e-4e00-824f-7287250d76dc的防火墙上已经导入的IP黑名单列表信息。

```
https://{Endpoint}/v1/eefb11b22d5944f2aa067954516df1cf/ptf/ip-blacklist?fw_instance_id=812f478c-940e-4e00-824f-7287250d76dc&limit=50&offset=0
```

响应示例

状态码：200

```
查询成功的示例：
{
  "data": {
    "limit": 50,
    "offset": 0,
    "records": [
      {
        "name": "ip-blacklist-eip.txt",
        "effect_scope": [
          1
        ],
        "import_status": 1,
        "import_time": 1740995060000,
        "error_message": "time out"
      }
    ],
    "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 ListIpBlacklistSolution {

    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();
        ListIpBlacklistRequest request = new ListIpBlacklistRequest();
        try {
            ListIpBlacklistResponse response = client.listIpBlacklist(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

```
# 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 = ListIpBlacklistRequest()
    response = client.list_ip_blacklist(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.ListIpBlacklistRequest{}
    response, err := client.ListIpBlacklist(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	
401	Unauthorized，错误的请求
403	Forbidden，禁止访问

错误码

请参见[错误码](#)。

4.21.3 删除已经导入的 IP 黑名单

功能介绍

删除流量过滤功能下已经导入的IP黑名单，指定生效范围进行删除。标准版的墙只会存在生效范围为EIP的IP黑名单，专业版的墙会存在生效范围为EIP和NAT的IP黑名单。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v1/{project_id}/ptf/ip-blacklist

表 4-1029 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-1030 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

请求参数

表 4-1031 请求 Body 参数

参数	是否必选	参数类型	描述
effect_scope	否	Array of integers	指定生效范围，1指定生效范围为EIP进行删除,2指定生效范围为NAT进行删除，1,2生效范围为EIP和NAT进行删除

响应参数

状态码：200

表 4-1032 响应 Body 参数

参数	参数类型	描述
data	Object	导入数据

请求示例

删除项目ID为eefb11b22d5944f2aa067954516df1cf，防火墙实例ID为812f478c-940e-4e00-824f-7287250d76dc上的IP黑名单，指定生效范围为EIP进行删除。

```
https://{Endpoint}/v1/eefb11b22d5944f2aa067954516df1cf/ptf/ip-blacklist?fw_instance_id=812f478c-940e-4e00-824f-7287250d76dc
{
  "effect_scope": [ 1 ]
}
```

响应示例

状态码：200

删除IP黑名单成功的响应，删除成功的情况下，此响应体内所有字符都为空。

```
{}
```

SDK 代码示例

SDK代码示例如下。

Java

删除项目ID为eefb11b22d5944f2aa067954516df1cf，防火墙实例ID为812f478c-940e-4e00-824f-7287250d76dc上的IP黑名单，指定生效范围为EIP进行删除。

```
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 DeletelpBlacklistSolution {

    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();
        DeletelpBlacklistRequest request = new DeletelpBlacklistRequest();
        IpBlacklistDeleteDto body = new IpBlacklistDeleteDto();
        List<Integer> listbodyEffectScope = new ArrayList<>();
        listbodyEffectScope.add(1);
        body.withEffectScope(listbodyEffectScope);
        request.withBody(body);
        try {
            DeletelpBlacklistResponse response = client.deletelpBlacklist(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eefb11b22d5944f2aa067954516df1cf，防火墙实例ID为812f478c-940e-4e00-824f-7287250d76dc上的IP黑名单，指定生效范围为EIP进行删除。

```
# 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 = DeleteIpBlacklistRequest()
        listEffectScopebody = [
            1
        ]
        request.body = IpBlacklistDeleteDto(
            effect_scope=listEffectScopebody
        )
        response = client.delete_ip_blacklist(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eefb11b22d5944f2aa067954516df1cf，防火墙实例ID为812f478c-940e-4e00-824f-7287250d76dc上的IP黑名单，指定生效范围为EIP进行删除。

```
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.DeletelPBlacklistRequest{}
var listEffectScopebody = []int32{
    int32(1),
}
request.Body = &model.IpBlacklistDeleteDto{
    EffectScope: &listEffectScopebody,
}
response, err := client.DeletelPBlacklist(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IP黑名单成功的响应，删除成功的情况下，此响应体内所有字符都为空。
401	Unauthorized，错误的请求
403	Forbidden，禁止访问

错误码

请参见[错误码](#)。

4.21.4 导出用于流量过滤的 IP 黑名单

功能介绍

指定IP黑名单的名字进行导出，当前只有两种文件名，在EIP生效的文件名为ip-blacklist-eip.txt，在 NAT生效的文件名为ip-blacklist-nat.txt。

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/ptf/ip-blacklist/export

表 4-1033 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。可通过 项目ID获取方式 获取

表 4-1034 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
name	是	String	IP黑名单的名字，如果要导出生效范围为EIP的IP黑名单，就指定名字为ip-blacklist-eip.txt，如果要导出生效范围为NAT的IP黑名单，就指定名字为ip-blacklist-nat.txt。

请求参数

无

响应参数

状态码：200

表 4-1035 响应 Header 参数

参数	参数类型	描述
Content-Disposition	String	指定响应中包含的文件的名称，当前返回的值为：attachment; filename=ip-blacklist.txt
Content-Type	String	参数解释： 响应的内容类型 约束限制： 不涉及 取值范围： text/plain 默认取值： 不涉及
Content-Length	Integer	文件的数据字节大小

表 4-1036 响应 Body 参数

参数	参数类型	描述
data	Object	导入数据

请求示例

从project id为eefb11b22d5944f2aa067954516df1cf、instance id为812f478c-940e-4e00-824f-7287250d76dc的防火墙上导出名字为ip-blacklist-eip.txt的IP黑名单文件

https://{Endpoint}/v1/eefb11b22d5944f2aa067954516df1cf/ptf/ip-blacklist/export?fw_instance_id=812f478c-940e-4e00-824f-7287250d76dc&name=ip-blacklist-eip.txt

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

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

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

    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();
        ExportIpBlacklistRequest request = new ExportIpBlacklistRequest();
        try {
            ExportIpBlacklistResponse response = client.exportIpBlacklist(request);
            System.out.println(response.toString());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```



```
    } 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 = ExportIpBlacklistRequest()
        response = client.export_ip_blacklist(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.ExportIpBlacklistRequest{}
response, err := client.ExportIpBlacklist(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	导出成功
401	Unauthorized，错误的请求
403	Forbidden，禁止访问

错误码

请参见[错误码](#)。

4.21.5 用于流量过滤的 IP 黑名单导入失败后进行重新导入

功能介绍

用于流量过滤的IP黑名单导入失败后，调用此接口进行重试。

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/ptf/ip-blacklist/retry

表 4-1037 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-1038 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
name	否	String	指定导入失败的IP黑名单的名字

请求参数

无

响应参数

状态码：200

表 4-1039 响应 Body 参数

参数	参数类型	描述
data	Object	导入数据

请求示例

```
https://{Endpoint}/v1/eefb11b22d5944f2aa067954516df1cf/ptf/ip-blacklist/retry?fw_instance_id=812f478c-940e-4e00-824f-7287250d76dc&name=ip-blacklist-eip.txt

请求：/v1/eefb11b22d5944f2aa067954516df1cf/ptf/ip-blacklist/retry?fw_instance_id=812f478c-940e-4e00-824f-7287250d76dc&name=ip-blacklist-eip.txt
响应： {}
```

响应示例

无

状态码

状态码	描述
200	重试操作的响应，在成功的情况下此结构内字段不赋值。

状态码	描述
401	Unauthorized，错误的请求
403	Forbidden，禁止访问

错误码

请参见[错误码](#)。

4.21.6 获取流量过滤功能的开关信息

功能介绍

流量过滤功能可以打开或者关闭，通过此接口可以获取当前的开关信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/ptf/ip-blacklist/switch

表 4-1040 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-1041 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

请求参数

无

响应参数

状态码：200

表 4-1042 响应 Body 参数

参数	参数类型	描述
data	IpBlacklistSwitchInfoVo object	包含IP黑名单功能的开关信息

表 4-1043 IpBlacklistSwitchInfoVo

参数	参数类型	描述
status	Integer	IP黑名单功能开关信息 1:enable 0:disable

请求示例

查询project id为eefb11b22d5944f2aa067954516df1cf、instance id为e494f486-fa6f-410e-9cfa-8edacfb28970的防火墙的IP黑名单开关信息，查询到的结果是功能开启。

https://{Endpoint}/v1/eefb11b22d5944f2aa067954516df1cf/ptf/ip-blacklist/switch?fw_instance_id=e494f486-fa6f-410e-9cfa-8edacfb28970

响应示例

状态码：200

返回流量过滤功能开关信息的响应体

```
{
  "data": {
    "status": 1
  }
}
```

状态码

状态码	描述
200	返回流量过滤功能开关信息的响应体
401	Unauthorized，错误的请求
403	Forbidden，禁止访问

错误码

请参见[错误码](#)。

4.21.7 开启或者关闭流量过滤的 IP 黑名单功能

功能介绍

开启或者关闭流量过滤功能，当前流量过滤是通过导入IP黑名单实现的。

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/ptf/ip-blacklist/switch

表 4-1044 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，可以通过调用API获取，也可以从控制台获取。 项目ID获取方式

表 4-1045 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

请求参数

表 4-1046 请求 Body 参数

参数	是否必选	参数类型	描述
status	是	Integer	打开或关闭IP黑名单功能 0: disable 1: enable

响应参数

状态码：200

表 4-1047 响应 Body 参数

参数	参数类型	描述
data	Object	导入数据

请求示例

关闭project id为eefb11b22d5944f2aa067954516df1cf、instance id为e494f486-fa6f-410e-9cfa-8edacfb28970上的IP黑名单功能

```
https://{Endpoint}/v1/eefb11b22d5944f2aa067954516df1cf/ptf/ip-blacklist/switch?fw_instance_id=e494f486-fa6f-410e-9cfa-8edacfb28970
```

请求body体:

```
{
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

关闭project id为eefb11b22d5944f2aa067954516df1cf、instance id为e494f486-fa6f-410e-9cfa-8edacfb28970上的IP黑名单功能

```
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 EnableIpBlacklistSolution {

    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();
        EnableIpBlacklistRequest request = new EnableIpBlacklistRequest();
        try {
            EnableIpBlacklistResponse response = client.enableIpBlacklist(request);
            System.out.println(response.toString());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
    } 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

关闭project id为eefb11b22d5944f2aa067954516df1cf、instance id为e494f486-fa6f-410e-9cfa-8edacfb28970上的IP黑名单功能

```
# 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 = EnableIpBlacklistRequest()
        response = client.enable_ip_blacklist(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

关闭project id为eefb11b22d5944f2aa067954516df1cf、instance id为e494f486-fa6f-410e-9cfa-8edacfb28970上的IP黑名单功能

```
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.EnableIpBlacklistRequest{}
    response, err := client.EnableIpBlacklist(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	打开或者关闭IP黑名单功能返回成功的响应，内部字段没有赋值。
401	Unauthorized，错误的请求
403	Forbidden，禁止访问

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。
201	Created	创建类的请求完全成功。
202	Accepted	已经接受请求，但未处理完成。
204	No Content	请求完全成功，同时HTTP响应不包含响应体。 在响应OPTIONS方法的HTTP请求时返回此状态码。

- 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。
405	Method Not Allowed	请求中带有该资源不支持的方法。 建议直接修改该请求，不要重试该请求。
409	Conflict	服务器在完成请求时发生冲突。 返回该状态码，表明客户端尝试创建的资源已经存在，或者由于冲突请求的更新操作不能被完成。

状态码	描述	说明
413	Request Entity Too Large	由于请求的实体过大，服务器无法处理，因此拒绝请求。为防止客户端的连续请求，服务器可能会关闭连接。如果只是服务器暂时无法处理，则会包含一个Retry-After的响应信息。
415	Unsupported Media Type	服务器无法处理请求附带的媒体格式。
429	Too Many Requests	表明请求超出了客户端访问频率的限制或者服务端接收到多于它能处理的请求。建议客户端读取相应的Retry-After首部，然后等待该首部指出的时间后再重试。
500	Internal Server Error	系统内部错误。
501	Not Implemented	服务器不支持请求的功能，无法完成请求。
503	Service Unavailable	被请求的服务无效。 建议直接修改该请求，不要重试该请求。

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.00109004	HTTP请求错误	HTTP请求错误	请稍后重试或联系技术支持
	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

模块	状态码	错误码	错误信息	描述	处理措施
	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
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

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存在
	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 0020	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.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.0020 0032	引擎不支持IPv6	引擎不支持IPv6	请联系技术支持
	400	CFW.0040 0007	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.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
更新ACL规则	400	CFW.0020 0005	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.0020 0005	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
	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

模块	状态码	错误码	错误信息	描述	处理措施
	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
	400	CFW.0070 0007	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.0070 0012	修改路由失败	修改路由失败	请联系技术支持
	400	CFW.0070 0015	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.0070 0016	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.0020 0016	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
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
	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

模块	状态码	错误码	错误信息	描述	处理措施
添加服务组	400	CFW.0020 0007	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.0020 0024	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.0040 0004	成员已存在	成员已存在	请删除一些服务组成员
	400	CFW.0090 0030	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.0020 0005	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.0020 0007	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.0020 0004	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.0020 0005	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.0040 0004	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.0020 0022	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.0020 0016	实例状态错误	实例状态错误	请联系技术支持
	400	CFW.0020 0032	引擎不支持IPv6	引擎不支持IPv6	请联系技术支持
	400	CFW.0040 0011	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.0040 0012	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

模块	状态码	错误码	错误信息	描述	处理措施
	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
	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。

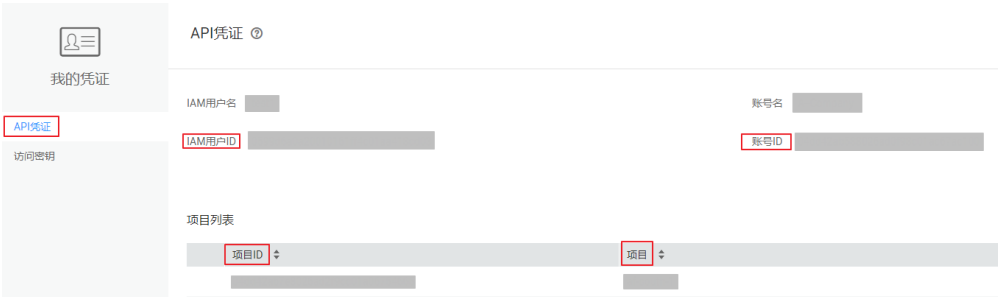
```
{
  "projects": [
    {
      "domain_id": "65382450e8f64ac0870cd180d14e684b",
      "is_domain": false,
      "parent_id": "65382450e8f64ac0870cd180d14e684b",
      "name": "xxxxxxx",
      "description": "",
      "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f897d6b99"
      },
      "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. [登录我的凭据控制台](#)。
 2. 单击用户名，在下拉列表中单击“基本信息”。
 3. 在基本信息页面单击“管理我的凭证”。
- 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CFW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的名称，在用户组详情页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

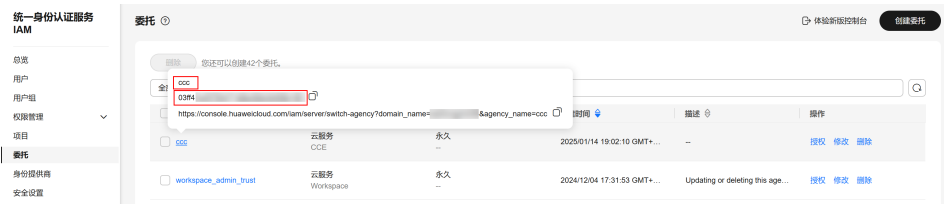


----结束

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
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

国家	编码
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
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

国家	编码
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
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

国家	编码
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
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

国家	编码
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
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

国家	编码
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
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

国家	编码
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
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