ALIBABA CLOUD

阿里云

Web应用防火墙

监控与告警

文档版本：20220422

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
Courier字体	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.告警设置

网站接入WAF防护后，您可以通过告警设置，使WAF在网站请求流量中检测到攻击事件、异常流量时向您发送告警通知，帮助您及时掌握业务的安全状态。

前提条件

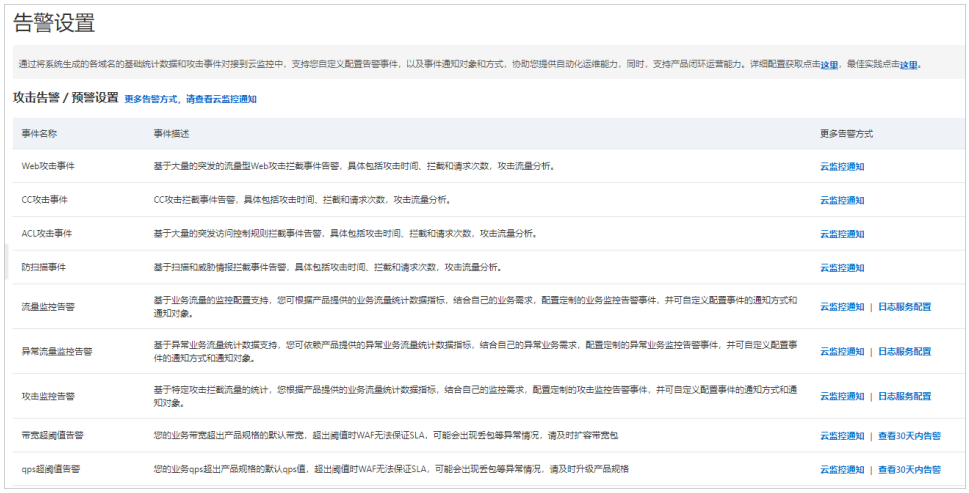
- 已将网站接入WAF防护。相关操作，请参见使用教程。
- （可选）已开通WAF日志服务并为WAF防护的网站域名开启了日志采集。相关操作，请参见步骤1：开通WAF日志服务、步骤2：开启日志采集。

  WAF默认支持通过阿里云云监控服务向您提供业务监控与告警设置。云监控可以对WAF攻击事件、WAF监控指标设置监控和告警规则。如果云监控支持的WAF监控指标无法满足您的需求，您可以使用日志服务自定义更丰富的WAF告警设置。

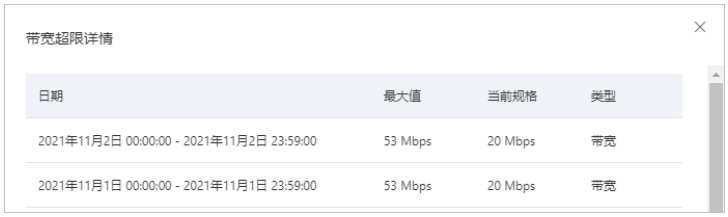
  如需使用日志服务告警配置，必须满足该前提条件。

操作步骤

1. 登录Web应用防火墙控制台。
2. <新版防护引擎>在顶部菜单栏，选择Web应用防火墙实例的资源组和地域（**中国内地**、**非中国内地**）。
3. 在左侧导航栏，选择**系统管理 > 告警设置**。
4. 在**告警设置**页面，根据要告警的事件类型，选择对应的告警方式。



告警方式	说明
云监控通知 通过云监控的报警服务功能创建WAF报警规则。	云监控（CloudMonitor）是一项针对阿里云资源和互联网应用进行监控的服务。云监控报警服务支持监控云产品的事件和业务指标。 **云监控通知**支持监控**告警设置**页面罗列的所有WAF事件，包括**Web攻击事件**、**CC攻击事件**、**ACL攻击事件**、**防扫描事件**、**流量监控告警**、**异常流量监控告警**、**攻击监控告警**、**带宽超阈值告警**、**QPS超阈值告警**。 单击**云监控通知**，页面将跳转到云监控控制台的报警规则设置页面，您可以在该页面设置报警规则。具体设置方法，请参见使用云监控设置WAF监控与告警。

告警方式	说明
日志服务配置 通过WAF日志服务的日志告警功能创建WAF报警规则。	WAF日志服务帮助您采集并存储WAF防护域名的请求日志，供您进行查询与分析。您可以基于日志查询与分析结果，为您关注的数据指标自定义告警规则。 **日志服务配置**支持监控多种组合业务指标，灵活度更高，适合定制化的告警场景，但使用复杂度也更高。 单击**日志服务配置**，页面将跳转到WAF日志服务页面，您可以在该页面查询与分析WAF日志并自定义告警。关于WAF日志告警的配置方法，请参见快速设置日志告警。关于WAF日志告警的配置示例，请参见概述。
查看30天内告警 查看WAF业务带宽或QPS超限的告警详情。	如果网站业务的正常业务流量超出WAF实例的业务带宽或QPS规格，WAF会自动在控制台顶部为您展示告警通知（如下图所示）并通过短信和邮件向您的阿里云账号联系人发送告警通知。  告警设置页面支持查看近30内的告警详情。您可以在**带宽超阈值告警**、**QPS超阈值告警**后单击**查看30天内告警**，查看业务带宽或QPS超限的告警详情（如下图所示）。  更多信息，请参见WAF业务带宽。

2.支持的监控类型和业务指标

本文介绍了使用阿里云云监控服务对网站业务设置监控和报警通知的相关内容，具体包括业务可用性监控、攻击事件监控、业务指标监控。

背景信息

WAF已集成了阿里云云监控服务，支持对您的业务站点、接入WAF防护的网站域名上发生的攻击事件和访问请求指标进行实时监控并报警。

云监控（CloudMonitor）是一项针对阿里云资源和互联网应用进行监控的服务。云监控为您提供系统事件的报警功能。您可以通过设置报警规则，使云监控在检测到系统事件时，通过邮件、短信、钉钉等方式向指定联系人发送通知或设置报警回调，使您第一时间知晓严重事件并及时进行处理，形成线上自动化运维闭环。

支持监控的站点指标

云监控可以模拟真实用户访问的探测请求，监控全国各省市运营商网络终端用户到您服务站点的访问情况，及时发现异常。

站点监控支持监控的业务指标如下表所示。建议您在使用站点监控设置时，覆盖所有支持的指标。

监控指标	重要级别	功能	设置方法
ECS性能监控	重要	对ECS的CPU使用率、内存使用率、磁盘空间使用率、带宽使用率进行监控。	设置ECS实例报警
SLB性能监控	重要	对SLB连接数、带宽使用率、PPS进行监控。	设置负载均衡报警规则
OSS性能监控	重要	对OSS的系统基本运行状态、性能以及计量等方面的数据指标进行监控。	监控服务概览
HTTP/HTTPS	重要	对指定的URL和IP地址进行HTTP或HTTPS探测。	站点监控由云监控服务提供。由于站点监控功能不涉及与WAF相关的操作，您只需登录阿里云账号，即可参照以下帮助文档进行操作： • 创建站点监控任务 • 修改站点监控任务 • 查看站点监控任务
PING	重要	对指定的URL和IP地址进行ICMP ping网络质量探测。	
TCP	重要	对指定的端口进行TCP存活探测。	
UDP	按需	对指定的端口进行UDP存活探测。	
DNS	按需	对指定的域名进行DNS探测。	
POP3	按需	对指定的URL和IP地址进行POP3探测。	
SMTP	按需	对指定的URL和IP地址进行SMTP探测。	

监控指标	重要级别	功能	设置方法
FTP	按需	对指定的URL和IP地址进行FTP探测。	

支持监控的攻击事件类型

云监控支持对接入WAF防护的网站域名上发生的Web攻击、CC攻击、扫描攻击、访问控制事件进行监控和报警。您可以根据事件的严重等级，设置以短信、邮件、钉钉等方式接收通知或设置报警回调。关于如何配置攻击事件监控和告警，请参见设置WAF攻击事件监控与告警。

> **注意** 攻击事件监控仅对已接入WAF防护的网站域名生效。您需要先完成网站接入，再配置相关的报警规则。关于网站接入的具体操作，请参见添加域名。

支持监控的WAF报警事件如下表所示。

事件名称	含义	类型	状态取值	事件等级
waf_event_aclattack	访问控制事件	acl	start、end	CRITICAL
waf_event_ccattack	CC攻击事件	cc	start、end	CRITICAL
waf_event_webattack	Web攻击事件	web	start、end	CRITICAL
waf_event_webscan	防扫描事件	webscan	start、end	CRITICAL

支持监控的WAF业务指标

云监控支持对接入WAF防护的网站域名的系统请求数据指标设置异常监控和告警。支持自定义指标异常的判断方法，并设置通过短信、邮件、钉钉等接收通知或设置报警回调。关于如何配置WAF业务指标监控和报警，请参见设置WAF业务指标监控报警。

> **注意** 业务指标监控仅对已接入WAF防护的网站域名生效。您需要先完成网站接入，再配置相关的报警规则。关于网站接入的具体操作，请参见添加域名。

支持监控的WAF监控类型如下表所示。

监控项	维度	单位	指标含义	备注
4XX占比	域名	%	每分钟4XX状态码的占比（不包含405）	报警信息以小数形式呈现
5XX占比	域名	%	每分钟5XX状态码的占比	报警信息以小数形式呈现
访问控制拦截量（5m）	域名	个	近5分钟内精准访问控制拦截量	无

监控项	维度	单位	指标含义	备注
访问控制拦截占比（5m）	域名	%	近5分钟内精准访问控制拦截占总请求量的占比	报警信息以小数形式呈现
CC防护拦截量（5m）	域名	个	近5分钟内CC安全防护拦截量	无
CC防护拦截占比（5m）	域名	%	近5分钟内CC安全防护拦截占总请求量的占比	报警信息以小数形式呈现
Web攻击拦截量（5m）	域名	个	近5分钟内Web应用攻击防护拦截量	无
Web攻击拦截占比（5m）	域名	%	近5分钟内Web应用攻击防护拦截占总请求量的占比	报警信息以小数形式呈现
QPS	域名	个/秒	QPS	无
QPS环比增长率	域名	%	每分钟QPS的环比增长率	报警信息以百分比形式呈现
QPS环比下降率	域名	%	每分钟QPS的环比下降率	报警信息以百分比形式呈现

3.使用云监控设置WAF监控与告警

WAF已集成了阿里云云监控服务，您可以在云监控中配置WAF的攻击事件报警通知规则。本文介绍了如何使用云监控服务配置WAF监控与告警。

前提条件

已将网站业务接入WAF进行防护。相关操作，请参见添加域名。

支持监控的WAF指标和事件

关于云监控支持监控的WAF指标和事件，请参见支持的监控类型和业务指标。

设置告警联系人

设置告警联系人后，您配置的WAF监控和告警信息会发送给告警联系人。告警联系人需及时查看告警通知信息，并对告警进行相应的处理。

您可以登录云监控控制台，创建报警联系人、创建报警联系组、批量添加报警联系人到报警联系组。详细介绍，请参见创建报警联系人或报警联系组。

设置WAF攻击事件监控与告警

设置WAF攻击事件监控与告警后，当WAF防护的域名发生了Web攻击和CC攻击等威胁时，云监控将根据您设置的报警事件等级和报警接收方式，发送报警通知。关于支持的详细监控类型和字段说明，请参见支持监控的攻击事件类型。

您可以登录云监控控制台，选择WAF相关参数，创建WAF攻击事件报警规则。详细介绍，请参见创建系统事件报警规则。

成功创建规则后，当已接入WAF防护的域名上发生攻击事件时，报警规则中设置的联系人将会收到相关报警通知。

您也可以在云监控查询近期发生的WAF监控事件。

设置WAF业务指标监控报警

设置WAF业务指标监控报警规则后，您可以及时获知已接入WAF防护的域名上的业务指标异常情况（例如，QPS环比下降、异常响应码占比突增、攻击拦截量突增等），并在发生异常时第一时间进行处理，以便尽快恢复业务。关于支持监控的WAF业务指标类型，请参见支持监控的WAF业务指标。

您可以登录云监控控制台，选择WAF相关参数，创建WAF业务指标监控报警规则。详细介绍，请参见创建报警规则。

成功创建规则后，当Web应用防火墙监控指标满足报警条件时，报警规则中指定的联系人组会收到报警通知。

高级自定义监控

您可以使用日志服务配置自定义业务指标监控和报警。详细介绍，请参见使用日志服务设置监控与告警。

4.使用日志服务自定义WAF监控与告警

4.1. 概述

本实践基于阿里云日志服务的告警功能，为接入Web应用防火墙（WAF）并开启了日志服务的业务配置自定义监控图表和告警服务，适用于企业级和个人用户在使用WAF时对业务整体流量和安全状态进行监控和告警。

使用流程

本实践的操作环节包括以下任务。

步骤	说明
步骤1：创建WAF日志分析仪表盘	使用Web应用防火墙日志服务发起查询/分析后，您可以依据当前查询语句创建一个仪表盘。仪表盘默认包含当前查询语句对应的图表。
步骤2：配置日志图表	创建Web应用防火墙日志分析仪表盘后，您可以在仪表盘中编辑/删除已有日志图表或通过复制创建新的日志图表。
步骤3：配置日志告警	创建Web应用防火墙日志分析仪表盘后，您可以在仪表盘中配置日志告警。日志告警必须关联仪表盘中已有的日志图表，并使用关联图表中的参数设置告警触发条件。日志告警支持自定义告警信息发送模板。

配置范例

本实践提供了13个日志图表和告警配置范例供您参考，分别是4XX比例（忽略拦截数据）、5XX比例异常告警、QPS异常告警、QPS突增告警、QPS突降告警、5分钟内ACL拦截情况告警、5分钟内WAF拦截情况告警、5分钟内CC拦截情况告警、5分钟内防扫描拦截情况、5分钟内单IP攻击量预警、5分钟内单IP攻击域名数量告警、5分钟平均时延情况、UID维度流量突降告警场景。

建议您在熟悉日志图表（步骤2）和告警配置（步骤3）后，再参见WAF日志告警配置案例添加图表并在添加图表的过程中直接配置告警。

关于在告警配置中用到的监控指标以及监控指标的阈值设置建议，请参见常用监控指标。

关于围绕监控指标进行查询/分析时用到的SQL查询语句，请参见查询与分析语句。

4.2. 步骤1：创建WAF日志分析仪表盘

使用Web应用防火墙（WAF）日志服务发起查询/分析后，您可以依据当前查询语句创建一个仪表盘。仪表盘默认包含当前查询语句对应的图表。

前提条件

- 域名已接入Web应用防火墙进行防护。更多信息，请参见修改域名DNS。
- 域名已开启Web应用防火墙日志服务。更多信息，请参见步骤1：开通WAF日志服务。

操作步骤

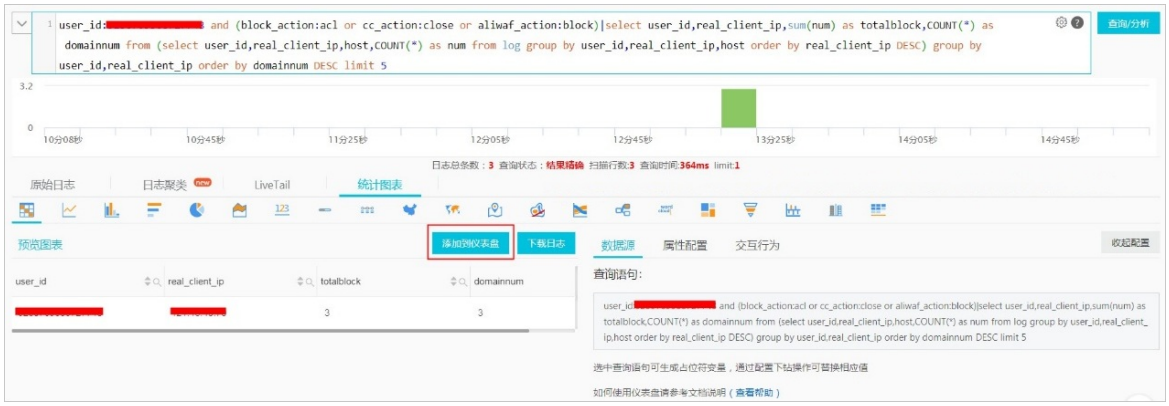
1. 登录Web应用防火墙控制台。
2. 进入日志服务高级管理页面。
   i. 在页面上方选择地域（**中国内地**、**海外地区**）。

ii. 在左侧导航栏单击**日志管理 > 日志服务**。

iii. 在**日志服务**页面右上角，单击**高级管理**。

iv. 在弹出的对话框中，单击**确定**。

3. 在**Project列表**中，定位到要操作的Web应用防火墙日志项目，单击Project名称。
4. 输入查询语句，并单击**查询/分析**。
5. 查询结束后，单击**统计图表**下的**添加到仪表盘**。



6. 在**添加到仪表盘**对话框，完成以下配置，并单击**确定**。

配置项	说明
操作类型	选择**新建仪表盘**。
Dashboard名称	为仪表盘命名。
图表名称	为当前查询语句对应的图表命名。

执行结果

成功创建仪表盘后，页面跳转到新建的仪表盘。仪表盘默认包含步骤4使用的查询语句对应的图表，您可以根据需要在仪表盘编辑当前图表或添加更多的图表。

后续步骤

步骤2：配置日志图表

4.3. 步骤2：配置日志图表

创建Web应用防火墙（WAF）日志分析仪表盘后，您可以在仪表盘中编辑、删除已有日志图表或通过复制创建新的日志图表。

前提条件

已创建日志分析仪表盘。更多信息，请参见步骤1：创建WAF日志分析仪表盘。

背景信息

本实践提供了13个默认的图表配置范例，具体请参见WAF日志告警配置案例。

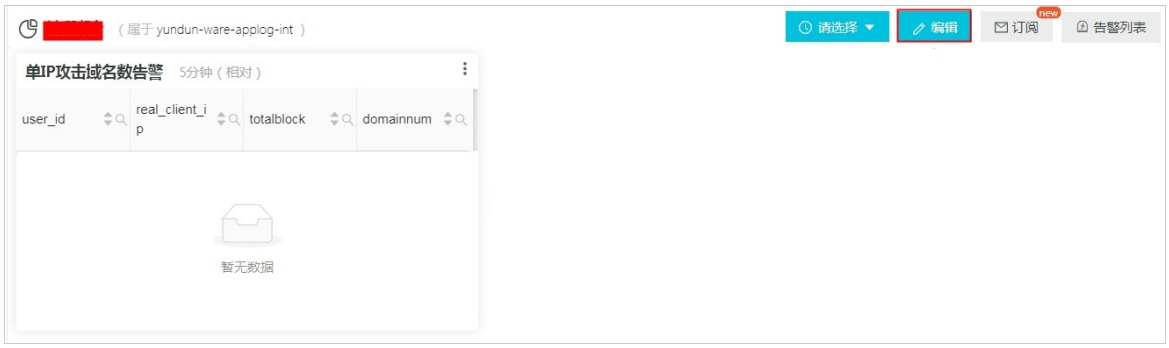
建议您先熟悉告警配置步骤，再参考范例添加图表并在添加图表的过程中直接配置告警。更多信息，请参见步骤3：配置日志告警。

操作步骤

1. 进入自定义的Web应用防火墙日志分析仪表盘。

   具体操作请参见步骤1：创建WAF日志分析仪表盘。

2. 单击仪表盘右上角的**编辑**。



   仪表盘切换到编辑模式。

3. 在仪表盘编辑模式下，根据需要编辑、删除当前仪表盘中已有图表或通过复制添加新的图表。

   > **说明** 您可以先通过复制添加图表，然后再编辑图表的配置。通过该方式在仪表盘里添加多个图表，实现多样化的数据展示以及告警配置。

   - 通过复制添加新的图表

a. 定位到要复制的图表，将光标悬置在图表右上角的选项图标（⋮）上，并单击**复制**。

成功复制图表后，当前图表旁边出现一个相同的图表。

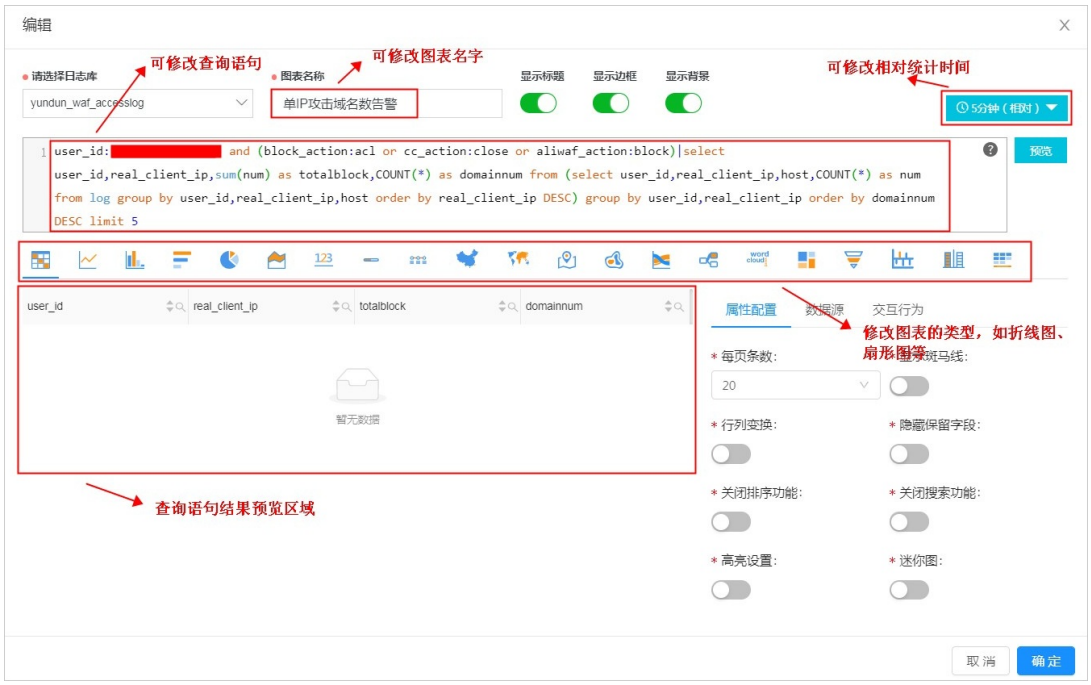
b. 用光标拖动复制生成的图表到仪表盘上的合适位置。

- 编辑已有图表

a. 定位到要编辑的图表，将光标悬置在图表右上角的选项图标（⋮）上，并单击**编辑**。

b. 在**编辑**页面，根据需要修改当前图表的配置，例如**图表名称**、SQL查询语句、相对统计时间、图表类型等，并单击**确定**。

> **说明** 如果您修改了SQL查询语句，则必须单击**预览**，由系统自动检查语句的正确性后才可以单击**确定**。如果SQL查询语句有问题，您会收到报错信息，这时**确定**按钮不可操作。只有将SQL查询语句修改正确后，您才可以单击**确定**。



- 删除已有图表

  定位到要删除的图表，将光标悬置在图表右上角的选项图标（⋮）上，并单击**删除**。

后续步骤

步骤3：配置日志告警

4.4. 步骤3：配置日志告警

创建Web应用防火墙（WAF）日志分析仪表盘后，您可以在仪表盘中配置日志告警。日志告警必须关联仪表盘中已有的日志图表，并使用关联图表中的参数设置告警触发条件。日志告警支持自定义告警信息发送模板。

前提条件

已创建日志分析仪表盘。更多信息，请参见步骤1：创建WAF日志分析仪表盘。

背景信息

本实践提供了13个默认的告警配置范例，具体请参见WAF日志告警配置案例。

建议您先熟悉图表配置步骤，再参考范例添加图表并在添加图表的过程中直接配置告警和通知方式。更多信息，请参见步骤2：配置日志图表。

操作步骤

1. 进入自定义的Web应用防火墙日志分析仪表盘。

   具体操作请参见步骤1：创建WAF日志分析仪表盘。

2. 在仪表盘右上角，选择**告警列表 > 新建**。

3. 在**创建告警**页面，完成以下**告警配置**，并单击**下一步**。


创建告警
告警配置
通知
* 告警名称
单IP攻击域名数告警
10/64
* 关联图表
0
图表名称
单IP攻击域名数告警
查询语句
user_id:[redacted] and (block_action:acl or cc_action:close or aliwaf_action:block) |select user_id,real_client_ip,sum(num) as totalblock,COUNT(*) as domainnum from (select user_id,real_client_ip,host,COUNT(*) as num from log group by user_id,real_client_ip,host order by real_client_ip DESC) group by user_id,real_client_ip order by domainnum DESC limit 5
查询区间
5分钟（相对）
1
添加
* 检查频率
固定间隔
1
分钟
* 触发条件
$0.domainnum>=10
支持加(+)、减(-)、乘(*)、除(/)、取模(%)5种基础运算符和大于（>）、大于等于（>=）、小于（<）、小于等于（<=）、等于（==）、不等于（!=）、正则匹配（=~）、正则不匹配（!~）8种比较运算符。帮助文档
高级选项
* 触发通知阈值
1
* 通知间隔
5分钟
下一步
取消

配置项	说明
告警名称	

配置项	说明
关联图表	设置告警中关联的图表。 设置关联图表时，**查询区间**为服务端每次执行查询时，读取的数据时间范围，支持相对时间与整点时间。例如，执行时间点为14:30:06，设置**查询区间**为15分钟（相对），则查询区间为 14:15:06- 14:30:06；设置**查询区间**为15分钟（整点时间），则查询区间为：14:15:00- 14:30:00。 需要添加多个图表时，只需单击**添加**并设置即可。最多支持关联三个图表。图表名称前的编号为该图表在告警中的编号，您可以在触发条件中通过编号指定关联的图表。
频率	
触发条件	判断告警是否触发的条件表达式，满足该条件时会根据**执行间隔**和**通知间隔**发送告警通知。 图表默认从0开始编号，在触发条件里用 `$0` 表示第一个图表。例如，您可以设置 `$0.domainnum>=10` ，表示第一个图表中 `domainnum` 字段值大于等于10时触发告警。 多个条件之间使用 `&&` 连接，表示逻辑与的关系，即必须同时满足；使用 `||` 连接，表示逻辑或的关系，即满足其中一个即可。 **说明** 更多告警条件表达式语法请参见告警条件表达式语法。
高级选项	
触发通知阈值	
通知间隔	

说明 **触发通知阈值**、**通知间隔**、检查**频率**三个条件配合使用，表示日志系统按照设置的检查频率去检查触发条件是否满足，并在**通知间隔**内达到**触发通知阈值**次数时推送告警信息。

4. 在**创建告警**页面，完成**通知**设置，并单击**提交**。

日志服务支持多种常用的告警通知方式，例如**短信**、**语音**、**邮件**、**WebHook+钉钉机器人**等。您必须先在**通知列表**右侧选择要使用的通知方式，然后完成具体配置。支持选择并配置多种通知方式。

- 短信告警：设置接收告警的**手机号码**和**发送内容**。发送内容中可以指定告警字段。单击**查看全部变量**了解各字段的含义。

- 语音告警：设置接收告警的**手机号码**和**发送内容**。

- 邮件告警：设置接收告警的**收件人**邮箱地址、告警邮件的**主题**和**发送内容**。

- WebHook+钉钉机器人：设置接收告警的钉钉群机器人的webhook地址（**请求地址**）和**发送内容**。

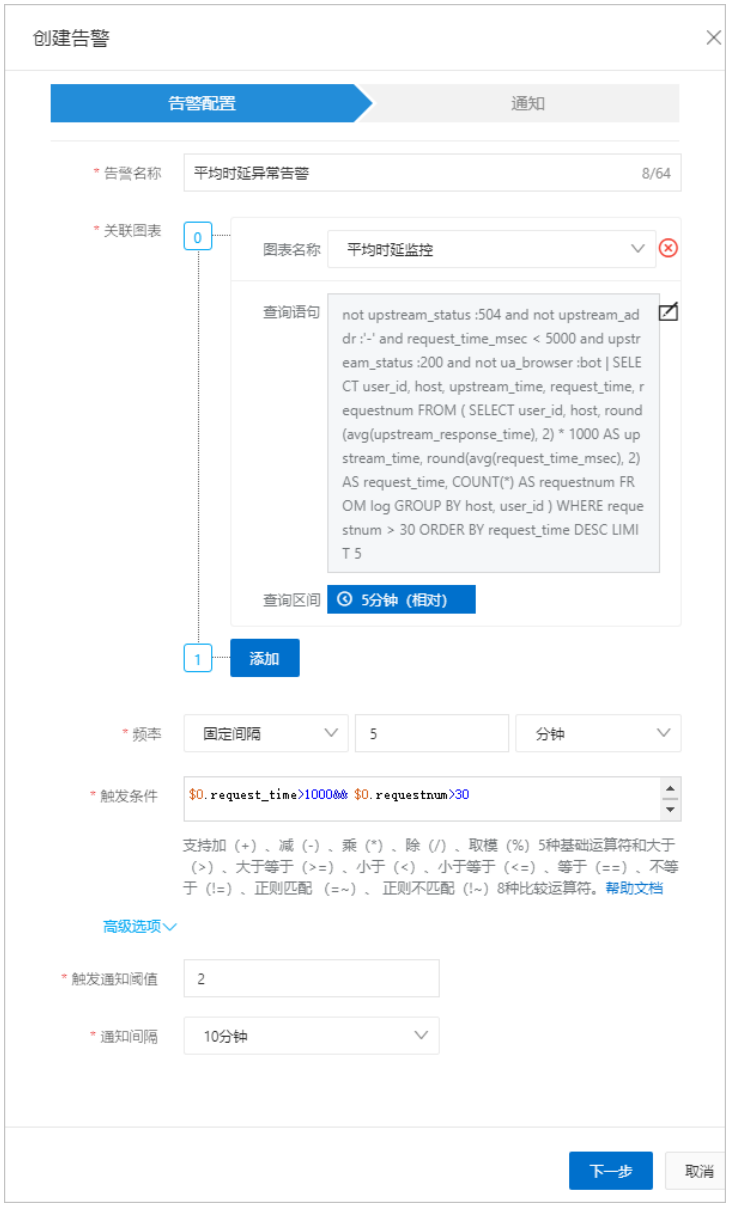
5. 重复步骤2~步骤4，创建更多的告警配置。

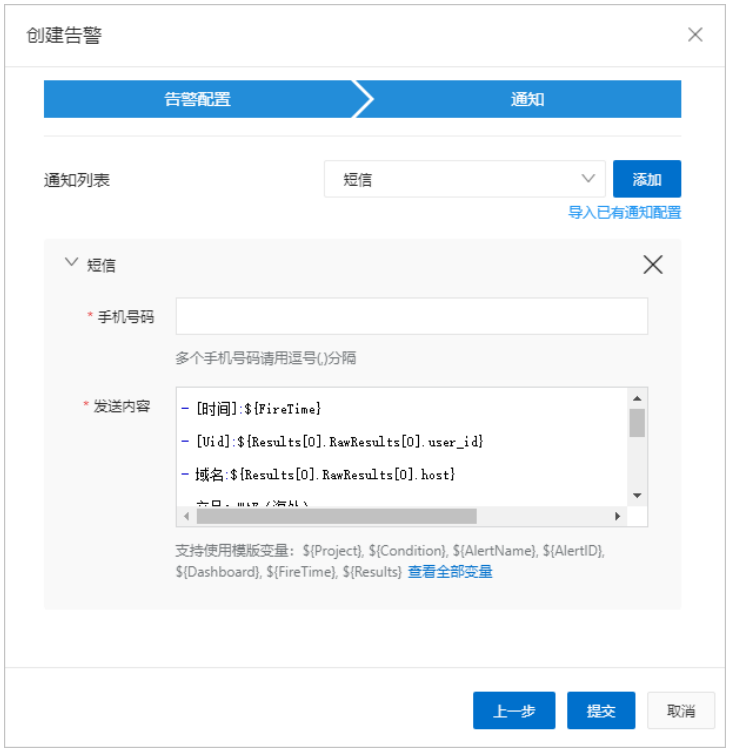
4.5. WAF日志告警配置案例

本文提供了典型的Web应用防火墙（WAF）日志查询与分析告警配置案例。您可以参考本文提供的告警配置参数，在自定义WAF日志仪表盘中添加监控图表及配置告警。

> **注意** 本文以旧版日志服务告警配置为例，介绍相关配置参数。如果您已升级使用了新版日志服务告警，请结合本文提供的查询语句及告警参数建议，并参见快速设置日志告警来完成相关配置。

旧版日志服务告警的配置参数如下图所示。


创建告警
告警配置
通知
* 告警名称
平均时延异常告警
8/64
* 关联图表
0
图表名称
平均时延监控
查询语句
not upstream_status :504 and not upstream_addr :'-' and request_time_msec < 5000 and upstream_status :200 and not ua_browser :bot | SELECT user_id, host, upstream_time, request_time, requestnum FROM (SELECT user_id, host, round(avg(upstream_response_time), 2) * 1000 AS upstream_time, round(avg(request_time_msec), 2) AS request_time, COUNT(*) AS requestnum FROM log GROUP BY host, user_id) WHERE requestnum > 30 ORDER BY request_time DESC LIMIT 5
查询区间
5分钟（相对）
1
添加
* 频率
固定间隔
5
分钟
* 触发条件
$0.request_time>1000&& $0.requestnum>30
支持加（+）、减（-）、乘（*）、除（/）、取模（%）5种基础运算符和大于（>）、大于等于（>=）、小于（<）、小于等于（<=）、等于（==）、不等于（!=）、正则匹配（=~）、正则不匹配（!~）8种比较运算符。帮助文档
高级选项
* 触发通知阈值
2
* 通知间隔
10分钟
下一步
取消



4XX比例异常告警

告警参数配置建议：

- **图表名称**：4XX比例（忽略拦截数据）
- **查询语句**：

```
user_id :您的阿里云账号ID
and not real_client_ip :被拦截的请求IP |
SELECT
  user_id,
  host AS "域名",
  Rate_2XX AS "2XX比例",
  Rate_3XX AS "3XX比例",
  Rate_4XX AS "4XX比例",
  Rate_5XX AS "5XX比例",
  countall AS "aveQPS",
  status_2XX,
  status_3XX,
  status_4XX,
  status_5XX,
  countall
FROM(
    SELECT
      user_id,
      host,
      round(
        round(status_2XX * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_2XX,
      round(
        round(status_3XX * 1.0000 / countall, 4) * 100,
```

```
        2
      ) AS Rate_3XX,
      round(
        round (status_4XX * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_4XX,
      round(
        round(status_5XX * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_5XX,
      status_2XX,
      status_3XX,
      status_4XX,
      status_5XX,
      countall
    FROM(
        SELECT
          user_id,
          host,
          count_if(
            status >= 200
            and status < 300
          ) AS status_2XX,
          count_if(
            status >= 300
            and status < 400
          ) AS status_3XX,
          count_if(
            status >= 400
            and status < 500
            and status <> 444
            and status <> 405
          ) AS status_4XX,
          count_if(
            status >= 500
            and status < 600
          ) AS status_5XX,
          COUNT(*) AS countall
        FROM          log
        GROUP BY
          host,
          user_id
      )
  )
WHERE
  countall > 120
ORDER BY
  Rate_4XX DESC
LIMIT
  5
```

该图表包含以下字段：`aveQPS`、`2XX比例`、`3XX比例`、`4XX比例`、`5XX比例`，分别表示域名QPS和各类型响应状态码的占比。其中，`4XX比例`不包含WAF拦截的CC攻击和Web攻击等造成的444和405状态码，以便只展示因业务自身原因造成的状态码变化。在设置告警触发条件时，您可以自由组合上述字段。例如，`aveQPS>10 && 2XX比例<60`表示在设定的统计时间内，指定域名的QPS达到10以上且2XX比例小于60%。

- **查询区间**：5分钟（相对）
- **频率**：固定间隔5分钟
- **触发条件**：`$0.countall>3000&& $0.4XX比例>80`
- **触发通知阈值**：2次
- **通知间隔**：10分钟
- **发送内容**：

```
- [时间]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- 域名:${Results[0].RawResults[0].域名}
- 产品: WAF
- 最近5分钟内总请求数:${Results[0].RawResults[0].countall}
- 2XX比例:${Results[0].RawResults[0].2XX比例} %
- 3XX比例:${Results[0].RawResults[0].3XX比例} %
- 4XX比例:${Results[0].RawResults[0].4XX比例} %
- 5XX比例:${Results[0].RawResults[0].5XX比例} %
```

5XX比例异常告警

告警参数配置建议：

- **图表名称**：5XX比例
- **查询语句**：

```
user_id :您的阿里云账号ID
and not real_client_ip :被拦截的请求IP |
select
  user_id,
  host AS "域名",
  Rate_2XX AS "2XX比例",
  Rate_3XX AS "3XX比例",
  Rate_4XX AS "4XX比例",
  Rate_5XX AS "5XX比例",
  countall AS "相对时间内访问量",
  status_2XX,
  status_3XX,
  status_4XX,
  status_5XX,
  countall
FROM(
    SELECT
      user_id,
      host,
      round(
        round(status_2XX * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_2XX,
```

```
      round(
        round(status_3XX * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_3XX,
      round(
        round (status_4XX * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_4XX,
      round(
        round(status_5XX * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_5XX,
      status_2XX,
      status_3XX,
      status_4XX,
      status_5XX,
      countall
    FROM(
        SELECT
          user_id,
          host,
          count_if(
            status >= 200
            and status < 300
          ) AS status_2XX,
          count_if(
            status >= 300
            and status < 400
          ) AS status_3XX,
          count_if(
            status >= 400
            and status < 500
          ) AS status_4XX,
          count_if(
            status >= 500
            and status < 600
          ) AS status_5XX,
          COUNT(*) AS countall
        FROM          log
        GROUP BY
          host,
          user_id
      )
  )
WHERE
  countall > 120
ORDER BY
  Rate_5XX DESC
LIMIT
  5
```

- **查询区间**：5分钟（相对）
- **频率**：固定间隔5分钟

- **触发条件**：`$0.countall>3000&& $0.5XX比例>80`
- **触发通知阈值**：2次
- **通知间隔**：10分钟
- **发送内容**：

```
- [时间]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- 域名:${Results[0].RawResults[0].域名}
- 产品: WAF
- 最近5分钟内总请求数:${Results[0].RawResults[0].countall}
- 2XX比例:${Results[0].RawResults[0].2XX比例} %
- 3XX比例:${Results[0].RawResults[0].3XX比例} %
- 4XX比例:${Results[0].RawResults[0].4XX比例} %
- 5XX比例:${Results[0].RawResults[0].5XX比例} %
```

QPS异常告警

告警参数配置建议：

- **图表名称**：QPS TOP 5
- **查询语句**：

```
user_id :您的阿里云账号ID
and not real_client_ip :被拦截的请求IP |
SELECT
  user_id,
  host,
  Rate_2XX,
  Rate_3XX,
  Rate_4XX,
  Rate_5XX,
  countall / 60 as "aveQPS",
  status_2XX,
  status_3XX,
  status_4XX,
  status_5XX,
  countall
FROM(
    SELECT
      user_id,
      host,
      round(
        round(status_2XX * 1.0000 / countall, 4) * 100,
        2
      ) as Rate_2XX,
      round(
        round(status_3XX * 1.0000 / countall, 4) * 100,
        2
      ) as Rate_3XX,
      round(
        round (status_4XX * 1.0000 / countall, 4) * 100,
        2
      ) as Rate_4XX,
      round(
```

```
        round(status_5XX * 1.0000 / countall, 4) * 100,
        2
      ) as Rate_5XX,
      status_2XX,
      status_3XX,
      status_4XX,
      status_5XX,
      countall
    FROM(
        SELECT
          user_id,
          host,
          count_if(
            status >= 200
            and status < 300
          ) as status_2XX,
          count_if(
            status >= 300
            and status < 400
          ) as status_3XX,
          count_if(
            status >= 400
            and status < 500
            and status <> 444
            and status <> 405
          ) as status_4XX,
          count_if(
            status >= 500
            and status < 600
          ) as status_5XX,
          COUNT(*) as countall
        FROM          log
        GROUP BY
          host,
          user_id
      )
  )
WHERE
  countall > 120
ORDER BY
  aveQPS DESC
LIMIT
  5
```

- **查询区间**：1分钟（相对）
- **频率**：固定间隔1分钟
- **触发条件**：`$0.aveQPS>=50`
- **触发通知阈值**：1次
- **通知间隔**：5分钟
- **发送内容**：

```
- [时间]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- 域名：${Results[0].RawResults[0].host}
- 产品：WAF
- 过去1分钟平均QPS：${Results[0].RawResults[0].aveQPS}
- 响应码 2xx_rate :${Results[0].RawResults[0].Rate_2XX}%
- 响应码 3xx_rate :${Results[0].RawResults[0].Rate_3XX}%
- 响应码 4xx_rate :${Results[0].RawResults[0].Rate_4XX}%
- 响应码 5xx_rate :${Results[0].RawResults[0].Rate_5XX}%
```

QPS突增告警

告警参数配置建议：

- **图表名称**：QPS突增监控
- **查询语句**：

```
user_id :您的阿里云账号ID |
SELECT
  t1.user_id,
  t1.now1mQPS,
  t1.past1mQPS,
  in_ratio,
  t1.host,
  t2.Rate_2XX,
  Rate_3XX,
  Rate_4XX,
  Rate_5XX,
  aveQPS
FROM  (
    (
      SELECT
        user_id,
        round(c [1] / 60, 0) AS now1mQPS,
        round(c [2] / 60, 0) AS past1mQPS,
        round(
          round(c [1] / 60, 0) / round(c [2] / 60, 0) * 100 -100,
          0
        ) AS in_ratio,
        host
      FROM        (
          SELECT
            compare(t, 60) AS c,
            host,
            user_id
          FROM            (
              SELECT
                COUNT(*) AS t,
                host,
                user_id
              FROM                log
              GROUP by
                host,
                user_id
```

```
        )
      GROUP by
        host,
        user_id
    )
  WHERE
    c [3] > 1.1
    and (
      c [1] > 180
      or c [2] > 180
    )
) t1
JOIN (
  SELECT
    user_id,
    host,
    Rate_2XX,
    Rate_3XX,
    Rate_4XX,
    Rate_5XX,
    countall / 60 AS "aveQPS",
    status_2XX,
    status_3XX,
    status_4XX,
    status_5XX,
    countall
  FROM        (
      SELECT
        user_id,
        host,
        round(
          round(status_2XX * 1.0000 / countall, 4) * 100,
          2
        ) AS Rate_2XX,
        round(
          round(status_3XX * 1.0000 / countall, 4) * 100,
          2
        ) AS Rate_3XX,
        round(
          round(status_4XX * 1.0000 / countall, 4) * 100,
          2
        ) AS Rate_4XX,
        round(
          round(status_5XX * 1.0000 / countall, 4) * 100,
          2
        ) AS Rate_5XX,
        status_2XX,
        status_3XX,
        status_4XX,
        status_5XX,
        countall
      FROM            (
          SELECT
            user_id,
            host
```

```
                host,
                count_if(
                  status >= 200
                  and status < 300
                ) AS status_2XX,
                count_if(
                  status >= 300
                  and status < 400
                ) AS status_3XX,
                count_if(
                  status >= 400
                  and status < 500
                  and status <> 444
                  and status <> 405
                ) AS status_4XX,
                count_if(
                  status >= 500
                  and status < 600
                ) AS status_5XX,
                COUNT(*) AS countall
              FROM                log
              GROUP BY
                host,
                user_id
            )
        )
      WHERE
        countall > 1
    ) t2 on t1.host = t2.host
  )
ORDER BY
  in_ratio DESC
LIMIT
  5
```

- **查询区间**：1分钟（相对）
- **频率**：固定间隔1分钟
- **触发条件**：`$0.now1mqps>50&& $0.in_ratio>300`
- **触发通知阈值**：1次
- **通知间隔**：5分钟
- **发送内容**：

```
- [时间]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- 域名：${Results[0].RawResults[0].host}
- 产品：WAF
- 过去1分钟平均QPS：${Results[0].RawResults[0].now1mqps}
- QPS突增率:${Results[0].RawResults[0].in_ratio}%
- 响应码 2xx_Rate :${Results[0].RawResults[0].rate_2xx}%
- 响应码 3xx_rate :${Results[0].RawResults[0].Rate_3XX}%
- 响应码 4xx_rate :${Results[0].RawResults[0].Rate_4XX}%
- 响应码 5xx_rate :${Results[0].RawResults[0].Rate_5XX}%
```

QPS突降告警

- **图表名称**：QPS突降监控
- **查询语句**：

```
user_id :您的阿里云账号ID |
SELECT
  t1.user_id,
  t1.now1mQPS,
  t1.past1mQPS,
  de_ratio,
  t1.host,
  t2.Rate_2XX,
  Rate_3XX,
  Rate_4XX,
  Rate_5XX,
  aveQPS
FROM  (
    (
      SELECT
        user_id,
        round(c [1] / 60, 0) AS now1mQPS,
        round(c [2] / 60, 0) AS past1mQPS,
        round(
          100-round(c [1] / 60, 0) / round(c [2] / 60, 0) * 100,
          2
        ) AS de_ratio,
        host
      FROM        (
          SELECT
            compare(t, 60) AS c,
            host,
            user_id
          FROM            (
              SELECT
                COUNT(*) AS t,
                host,
                user_id
              FROM                log
              GROUP BY
                host,
                user_id
            )
          GROUP BY
            host,
            user_id
        )
      WHERE
        c [3] < 0.9
        AND (
          c [1] > 180
          or c [2] > 180
        )
    ) t1
```

```
    JOIN (
      SELECT
        user_id,
        host,
        Rate_2XX,
        Rate_3XX,
        Rate_4XX,
        Rate_5XX,
        countall / 60 AS "aveQPS",
        status_2XX,
        status_3XX,
        status_4XX,
        status_5XX,
        countall
      FROM        (
          SELECT
            user_id,
            host,
            round(
              round(status_2XX * 1.0000 / countall, 4) * 100,
              2
            ) AS Rate_2XX,
            round(
              round(status_3XX * 1.0000 / countall, 4) * 100,
              2
            ) AS Rate_3XX,
            round(
              round(status_4XX * 1.0000 / countall, 4) * 100,
              2
            ) AS Rate_4XX,
            round(
              round(status_5XX * 1.0000 / countall, 4) * 100,
              2
            ) AS Rate_5XX,
            status_2XX,
            status_3XX,
            status_4XX,
            status_5XX,
            countall
          FROM            (
              SELECT
                user_id,
                host,
                count_if(
                  status >= 200
                  and status < 300
                ) AS status_2XX,
                count_if(
                  status >= 300
                  and status < 400
                ) AS status_3XX,
                count_if (
                  status >= 400
                  and status < 500
```

```
                and status <> 444
                and status <> 405
              ) AS status_4XX,
              count_if(
                status >= 500
                and status < 600
              ) AS status_5XX,
              COUNT(*) AS countall
            FROM                log
            GROUP BY
              host,
              user_id
          )
        )
      WHERE
        countall > 1
    ) t2 on t1.host = t2.host
  )
ORDER BY
  de_ratio DESC
LIMIT
  5
```

该图表中包含 `now1mpqs`（当前一分钟的平均QPS）、`past1mqps`（过去一分钟的平均QPS）、`de_ratio`（QPS下降率）、`host` 等字段，您可以根据需要使用这些字段设置告警条件。

- **查询区间**：1分钟（相对）
- **频率**：固定间隔1分钟
- **触发条件**：`$0.now1mqps>10&& $0.de_ratio>50`
- **触发通知阈值**：2次
- **通知间隔**：5分钟
- **发送内容**：

```
- [时间]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- 域名：${Results[0].RawResults[0].host}
- 产品：WAF（海外）
- 过去1分钟平均QPS：${Results[0].RawResults[0].now1mqps}
- QPS突降率:${Results[0].RawResults[0].de_ratio}%
- 响应码 2xx_rate :${Results[0].RawResults[0].rate_2xx}%
- 响应码 3xx_rate :${Results[0].RawResults[0].Rate_3XX}%
- 响应码 4xx_rate :${Results[0].RawResults[0].Rate_4XX}%
- 响应码 5xx_rate :${Results[0].RawResults[0].Rate_5XX}%
```

5分钟内ACL拦截情况告警

告警参数配置建议：

- **图表名称**：ACL规则拦截量
- **查询语句**：

```
user_id :您的阿里云账号ID |
SELECT
  user_id,
  host,
  count_if(
    final_plugin = 'waf'
    AND final_action = 'block'
  ) AS "规则防护引擎拦截量",
  count_if(
    final_plugin = 'cc'
    AND final_action = 'block'
  ) AS "CC拦截量",
  count_if(
    final_plugin = 'acl'
    AND final_action = 'block'
  ) AS "ACL拦截量",
  count_if(
    final_plugin = 'antiscan'
    AND final_action = 'block'
  ) AS "扫描防护拦截量",
  count_if(
    (final_plugin = 'waf'
    AND final_action = 'block')
    OR (final_plugin = 'cc'
    AND final_action = 'block')
    OR (final_plugin = 'acl'
    AND final_action = 'block')
    OR (final_plugin = 'antiscan'
    AND final_action = 'block')
  ) AS totalblock
GROUP BY
  host,
  user_id
HAVING
  (
    "ACL拦截量" >= 0
    AND "规则防护引擎拦截量" >= 0
    AND "CC拦截量" >= 0
    AND "扫描防护拦截量" >= 0
    AND totalblock > 10
  )
ORDER BY
  "ACL拦截量" DESC
LIMIT
  5
```

- **查询区间**：5分钟（相对）
- **频率**：固定间隔5分钟
- **触发条件**：`$0.totalblock>=500&&($0.ACL拦截量>=500)`
- **触发通知阈值**：1次
- **通知间隔**：5分钟
- **发送内容**：

```
- [时间]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- 域名:${Results[0].RawResults[0].host}
- 产品: WAF
- 最近5分钟内拦截总量:${Results[0].RawResults[0].totalblock}
- ACL拦截量:${Results[0].RawResults[0].ACL拦截量}
- 规则防护引擎拦截量:${Results[0].RawResults[0].规则防护引擎拦截量}
- CC拦截量:${Results[0].RawResults[0].CC拦截量}
- 扫描防护拦截量:${Results[0].RawResults[0].扫描防护拦截量}
```

5分钟内规则防护引擎拦截情况告警

告警参数配置建议：

- **图表名称**：规则防护引擎拦截量
- **查询语句**：

```
user_id :您的阿里云账号ID |
SELECT
  user_id,
  host,
  count_if(
    final_plugin = 'waf'
    AND final_action = 'block'
  ) AS "规则防护引擎拦截量",
  count_if(
    final_plugin = 'cc'
    AND final_action = 'block'
  ) AS "CC拦截量",
  count_if(
    final_plugin = 'acl'
    AND final_action = 'block'
  ) AS "ACL拦截量",
  count_if(
    final_plugin = 'antiscan'
    AND final_action = 'block'
  ) AS "扫描防护拦截量",
  count_if(
    (final_plugin = 'waf'
    AND final_action = 'block')
    OR (final_plugin = 'cc'
    AND final_action = 'block')
    OR (final_plugin = 'acl'
    AND final_action = 'block')
    OR (final_plugin = 'antiscan'
    AND final_action = 'block')
  ) AS totalblock
GROUP BY
  host,
  user_id
HAVING
  (
    "ACL拦截量" >= 0
    AND "规则防护引擎拦截量" >= 0
    AND "CC拦截量" >= 0
    AND "扫描防护拦截量" >= 0
    AND totalblock > 10
  )
ORDER BY
  "规则防护引擎拦截量" DESC
LIMIT
  5
```

- **查询区间**：5分钟（相对）
- **频率**：固定间隔5分钟
- **触发条件**：`$0.totalblock>=500&&($0.规则防护引擎拦截量>=500)`
- **触发通知阈值**：1次
- **通知间隔**：5分钟
- **发送内容**：

```
- [时间]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- 域名:${Results[0].RawResults[0].host}
- 产品: WAF
- 最近5分钟内拦截总量:${Results[0].RawResults[0].totalblock}
- ACL拦截量:${Results[0].RawResults[0].ACL拦截量}
- 规则防护引擎拦截量:${Results[0].RawResults[0].规则防护引擎拦截量}
- CC拦截量:${Results[0].RawResults[0].CC拦截量}
- 扫描防护拦截量:${Results[0].RawResults[0].扫描防护拦截量}
```

5分钟内CC拦截情况告警

告警参数配置建议：

- **图表名称**：CC防护规则拦截量
- **查询语句**：

```
user_id :您的阿里云账号ID |
SELECT
  user_id,
  host,
  count_if(
    final_plugin = 'waf'
    AND final_action = 'block'
  ) AS "规则防护引擎拦截量",
  count_if(
    final_plugin = 'cc'
    AND final_action = 'block'
  ) AS "CC拦截量",
  count_if(
    final_plugin = 'acl'
    AND final_action = 'block'
  ) AS "ACL拦截量",
  count_if(
    final_plugin = 'antiscan'
    AND final_action = 'block'
  ) AS "扫描防护拦截量",
  count_if(
    (final_plugin = 'waf'
    AND final_action = 'block')
    OR (final_plugin = 'cc'
    AND final_action = 'block')
    OR (final_plugin = 'acl'
    AND final_action = 'block')
    OR (final_plugin = 'antiscan'
    AND final_action = 'block')
  ) AS totalblock
GROUP BY
  host,
  user_id
HAVING
  (
    "ACL拦截量" >= 0
    AND "规则防护引擎拦截量" >= 0
    AND "CC拦截量" >= 0
    AND "扫描防护拦截量" >= 0
    AND totalblock > 10
  )
ORDER BY
  "CC拦截量" DESC
LIMIT
  5
```

- **查询区间**：5分钟（相对）
- **频率**：固定间隔5分钟
- **触发条件**：`$0.totalblock>=500&&($0.CC拦截量>=500)`
- **触发通知阈值**：1次
- **通知间隔**：5分钟
- **发送内容**：

```
- [时间]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- 域名:${Results[0].RawResults[0].host}
- 产品: WAF
- 最近5分钟内拦截总量:${Results[0].RawResults[0].totalblock}
- ACL拦截量:${Results[0].RawResults[0].ACL拦截量}
- 规则防护引擎拦截量:${Results[0].RawResults[0].规则防护引擎拦截量}
- CC拦截量:${Results[0].RawResults[0].CC拦截量}
- 扫描防护拦截量:${Results[0].RawResults[0].扫描防护拦截量}
```

5分钟内扫描拦截情况告警

告警参数配置建议：

- **图表名称**：扫描防护拦截量
- **查询语句**：

```
user_id :您的阿里云账号ID |
SELECT
  user_id,
  host,
  count_if(
    final_plugin = 'waf'
    AND final_action = 'block'
  ) AS "规则防护引擎拦截量",
  count_if(
    final_plugin = 'cc'
    AND final_action = 'block'
  ) AS "CC拦截量",
  count_if(
    final_plugin = 'acl'
    AND final_action = 'block'
  ) AS "ACL拦截量",
  count_if(
    final_plugin = 'antiscan'
    AND final_action = 'block'
  ) AS "扫描防护拦截量",
  count_if(
    (final_plugin = 'waf'
    AND final_action = 'block')
    OR (final_plugin = 'cc'
    AND final_action = 'block')
    OR (final_plugin = 'acl'
    AND final_action = 'block')
    OR (final_plugin = 'antiscan'
    AND final_action = 'block')
  ) AS totalblock
GROUP BY
  host,
  user_id
HAVING
  (
    "ACL拦截量" >= 0
    AND "规则防护引擎拦截量" >= 0
    AND "CC拦截量" >= 0
    AND "扫描防护拦截量" >= 0
    AND totalblock > 10
  )
ORDER BY
  "扫描防护拦截量" DESC
LIMIT
  5
```

- **查询区间**：5分钟（相对）
- **频率**：固定间隔5分钟
- **触发条件**：`$0.totalblock>=500&&($0.扫描防护拦截量>=500)`
- **触发通知阈值**：1次
- **通知间隔**：5分钟
- **发送内容**：

```
- [时间]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- 域名:${Results[0].RawResults[0].host}
- 产品：WAF（海外）
- 最近5分钟内拦截总量:${Results[0].RawResults[0].totalblock}
- ACL拦截量:${Results[0].RawResults[0].ACL拦截量}
- 规则防护引擎拦截量:${Results[0].RawResults[0].规则防护引擎拦截量}
- CC拦截量:${Results[0].RawResults[0].CC拦截量}
- 扫描防护拦截量:${Results[0].RawResults[0].扫描防护拦截量}
```

单IP攻击量预警

告警参数配置建议：

- **图表名称**：单IP攻击量
- **查询语句**：

```
user_id :您的阿里云账号ID |
SELECT
  user_id,
  real_client_ip,
  concat(
    'ACL拦截量:',
    cast(aclblock AS varchar(10)),
    '  ',
    '规则防护引擎拦截量:',
    cast(wafblock AS varchar(10)),
    '
',
    'CC拦截量:',
    cast(aclblock AS varchar(10))
  ) AS blockNum,
  totalblock,
  allRequest
FROM  (
    SELECT
      user_id,
      real_client_ip,
      count_if(
        final_plugin = 'acl'
        AND final_action = 'block'
      ) AS aclblock,
      count_if(
        final_plugin = 'waf'
        AND final_action = 'block'
      ) AS wafblock,
      count_if(
        final_plugin = 'cc'
        AND final_action = 'block'
      ) AS ccblock,
      count_if(
        (
          final_plugin = 'acl'
          AND final_action = 'block'
```

```
        )
        OR (
          final_plugin = 'waf'
          AND final_action = 'block'
        )
        OR (
          final_plugin = 'cc'
          AND final_action = 'block'
        )
      ) AS totalblock,
      COUNT(*) AS allRequest
    FROM      log
    GROUP BY
      user_id,
      real_client_ip
    HAVING
      totalblock > 1
    ORDER BY
      totalblock DESC
    LIMIT
      5
  )
```

该图表中包含 `real_client_ip`（攻击IP）、`blockNum`（包含 `ACL拦截量`、`规则防护引擎拦截量`、`CC拦截量` 等数据）、`totalblock`（总拦截请求数）、`allRequest`（总请求数）字段，您可以根据需要使用这些字段设置告警条件。

- **查询区间**：5分钟（相对）
- **频率**：固定间隔5分钟
- **触发条件**：`$0.totalblock >=500`
- **触发通知阈值**：1次
- **通知间隔**：5分钟
- **发送内容**：

```
- [时间]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- 产品:WAF
- 最近5分钟内单IP攻击排行Top3:
- ${Results[0].RawResults[0].real_client_ip}  (${Results[0].RawResults[0].blockNum})
- ${Results[0].RawResults[1].real_client_ip}  (${Results[0].RawResults[1].blockNum})
- ${Results[0].RawResults[2].real_client_ip}  (${Results[0].RawResults[2].blockNum})
```

单IP攻击域名数量告警

告警参数配置建议：

- **图表名称**：单IP攻击域名数量
- **查询语句**：

```
user_id :您的阿里云账号ID
and not upstream_status :504
and not upstream_addr :'-'
and request_time_msec < 5000
and upstream_status :200
and not ua_browser :bot |
SELECT
  user_id,
  host,
  upstream_time,
  request_time,
  requestnum
FROM  (
    SELECT
      user_id,
      host,
      round(avg(upstream_response_time), 2) * 1000 AS upstream_time,
      round(avg(request_time_msec), 2) AS request_time,
      COUNT(*) AS requestnum
    FROM      log
    GROUP BY
      host,
      user_id
  )
WHERE
  requestnum > 30
ORDER BY
  request_time DESC
LIMIT
  5
```

该图表中包含 `real_client_ip`（攻击IP）、`totalblock`（总拦截请求数）、`domainnum`（该IP攻击的域名数）等字段。在设置告警触发条件时，您可以自由组合上述字段来设置告警条件。例如，`totalblock>500&& domainnum>5` 表示某IP在对应时间内总攻击量达到500，并且攻击域名数多于5个。

- **查询区间**：5分钟（相对）
- **频率**：固定间隔1分钟
- **触发条件**：`$0.domainnum>=10`
- **触发通知阈值**：1次
- **通知间隔**：5分钟
- **发送内容**：

```
- [时间]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- 产品：WAF
- 攻击IP:${Results[0].RawResults[0].real_client_ip}
- 攻击的域名数:${Results[0].RawResults[0].domainnum}
- 最近5分钟总攻击请求数:${Results[0].RawResults[0].totalblock}
- 请及时关注处理
```

5分钟平均时延异常告警

告警参数配置建议：

- **图表名称**：平均时延监控
- **查询语句**：

```
user_id :您的阿里云账号ID
and not upstream_status :504
and not upstream_addr :'-'
and request_time_msec < 5000
and upstream_status :200
and not ua_browser :bot |
SELECT
  user_id,
  host,
  upstream_time,
  request_time,
  requestnum
FROM  (
    SELECT
      user_id,
      host,
      round(avg(upstream_response_time), 2) * 1000 AS upstream_time,
      round(avg(request_time_msec), 2) AS request_time,
      COUNT(*) AS requestnum
    FROM      log
    GROUP BY
      host,
      user_id
  )
WHERE
  requestnum > 30
ORDER BY
  request_time DESC
LIMIT
  5
```

- **查询区间**：5分钟（相对）
- **频率**：固定间隔5分钟
- **触发条件**：`$0.request_time>1000&& $0.requestnum>30`
- **触发通知阈值**：2次
- **通知间隔**：10分钟
- **发送内容**：

```
- [时间]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- 域名:${Results[0].RawResults[0].host}
- 产品：WAF（海外）
- [触发条件]:${condition}
- 最近5分钟延时情况TOP 3（毫秒）
- Host1:${Results[0].RawResults[0].host} Delay_time:${Results[0].RawResults[0].upstream_time}
- Host2:${Results[0].RawResults[1].host} Delay_time:${Results[0].RawResults[1].upstream_time}
- Host3:${Results[0].RawResults[2].host} Delay_time:${Results[0].RawResults[2].upstream_time}
```

流量突降告警

告警参数配置建议：

- **图表名称**：流量突降监控
- **查询语句**：

```
user_id :您的阿里云账号ID |
SELECT
  t1.user_id,
  t1.now1mQPS,
  t1.past1mQPS,
  de_ratio,
  t2.Rate_2XX,
  Rate_3XX,
  Rate_4XX,
  Rate_5XX,
  aveQPS
FROM  (
    (
      SELECT
        user_id,
        round(c [1] / 60, 0) AS now1mQPS,
        round(c [2] / 60, 0) AS past1mQPS,
        round(
          100-round(c [1] / 60, 0) / round(c [2] / 60, 0) * 100,
          2
        ) AS de_ratio
      FROM        (
          SELECT
            compare(t, 60) AS c,
            user_id
          FROM            (
              SELECT
                COUNT(*) AS t,
                user_id
              FROM                log
              GROUP BY
                user_id
            )
          GROUP BY
```

```
            user_id
        )
      WHERE
        c [3] < 0.9
        AND (
          c [1] > 180
          or c [2] > 180
        )
    ) t1
    JOIN (
      SELECT
        user_id,
        Rate_2XX,
        Rate_3XX,
        Rate_4XX,
        Rate_5XX,
        countall / 60 AS "aveQPS",
        status_2XX,
        status_3XX,
        status_4XX,
        status_5XX,
        countall
      FROM        (
          SELECT
            user_id,
            round(
              round(status_2XX * 1.0000 / countall, 4) * 100,
              2
            ) AS Rate_2XX,
            round(
              round(status_3XX * 1.0000 / countall, 4) * 100,
              2
            ) AS Rate_3XX,
            round(
              round(status_4XX * 1.0000 / countall, 4) * 100,
              2
            ) AS Rate_4XX,
            round(
              round(status_5XX * 1.0000 / countall, 4) * 100,
              2
            ) AS Rate_5XX,
            status_2XX,
            status_3XX,
            status_4XX,
            status_5XX,
            countall
          FROM            (
              SELECT
                user_id,
                count_if(
                  status >= 200
                  AND status < 300
                ) AS status_2XX,
                count_if(
```

```
                status >= 300
                AND status < 400
              ) AS status_3XX,
              count_if (
                status >= 400
                AND status < 500
                AND status <> 444
                AND status <> 405
              ) AS status_4XX,
              count_if(
                status >= 500
                AND status < 600
              ) AS status_5XX,
              COUNT(*) AS countall
            FROM                log
            GROUP BY
              user_id
          )
        )
      WHERE
        countall > 0
    ) t2 ON t1.user_id = t2.user_id
  )
ORDER BY
  de_ratio DESC
LIMIT
  5
```

- **查询区间**：1分钟（相对）
- **频率**：固定间隔1分钟
- **触发条件**：`$0.de_ratio>50&& $0.now1mqps>20`
- **触发通知阈值**：1次
- **通知间隔**：5分钟
- **发送内容**：

```
- [时间]:${FireTime}
- [UID]:${Results[0].RawResults[0].user_id}
- 产品：WAF
- 过去1分钟平均QPS：${Results[0].RawResults[0].now1mqps}
- [触发条件(突降率&QPS)]:${condition}
- QPS突降率:${Results[0].RawResults[0].de_ratio}%
- 响应码 2xx_rate :${Results[0].RawResults[0].rate_2xx}%
- 响应码 3xx_rate :${Results[0].RawResults[0].Rate_3XX}%
- 响应码 4xx_rate :${Results[0].RawResults[0].Rate_4XX}%
- 响应码 5xx_rate :${Results[0].RawResults[0].Rate_5XX}%
```

4.6. 常用监控指标

本文介绍了使用Web应用防火墙日志服务发起查询/分析时常用的监控指标及其含义。您可以将这些指标用于告警配置条件中，自定义监控业务的异常情况。本文也提供了在告警配置中建议使用的监控指标阈值和指标异常时的处理建议。

监控指标	释义	建议阈值	处理建议
200	服务器已成功处理请求，返回了请求的数据。	初始化正常业务时，200状态码的告警监控阈值可以配置为90%，具体根据实际业务情况调整。	如果发现低于监控比例，需要分析比例下降的原因，例如是否因为其他错误状态码比例增加。
request_time_msec	客户端请求到返回结果的请求耗时。	按实际业务请求所需耗时，设置合适的超时告警监控阈值。	如果发现域名请求耗时较长，需要检查客户端-WAF-源站整体网路链路质量，并排查源站响应状态是否正常。
upstream_response_time	请求回源时，源站返回数据的响应时间。		
ssl_handshake_time	HTTPS协议请求时，客户端与WAF的SSL握手时间。		
status:302 and block_action:tmd/status:200 and block_action:tmd	人机校验JS请求状态码，302表示触发默认策略，200表示触发自定义CC防护策略。	初始化时，建议配置5%~10%的告警阈值比例，后续运营期间可以根据业务拦截情况灵活调整。	• 如果达到告警阈值，建议分析是否受到CC攻击，根据攻击情况设置自定义规则。 • 检查服务器是否出现异常，如大量的5xx状态码、4xx状态码。
status:200 and block_action:antifraud	被数据风控规则拦截。		测试可用后再上线，如弹出率过高，说明场景可能有问题，建议联系阿里云研发团队进行确认。
status:404	服务器找不到请求的资源。		查询触发告警的IP。 • 如果是个例，则可能存在恶意用户遍历服务器资源。 • 如果是普遍存在，则需要确认服务器是否正常或者是否有文件丢失。
status:405	被Web应用防护规则或精准访问控制规则拦截。		通过全量日志分析拦截的规则、请求行为，判断是正常拦截还是误拦截。
status:444	被WAF CC自定义规则拦截。		• 如果达到告警阈值，建议分析是否受到CC攻击，根据攻击情况设置自定义规则。 • 如果不是攻击，而是API调用，则需要判断是否需要调整阈值或者单独放行固定服务器的调用。
status:499	客户端发起请求，服务端未返回数据，超过客户端设置的等待时间后，客户端主动断链，服务端返回给客户端该状态码。		• 检查源站是否异常，如响应缓慢，数据库存在大量慢查询。 • 存在攻击将源站资源占满。

监控指标	释义	建议阈值	处理建议
status:500	（Internal Server Error）服务器内部错误，无法完成请求。		建议检查源站处理资源负载、数据库等情况。
status:502	（Bad Gateway）错误网关， 服务器作为网关或代理，从上游服务器收到无效响应。一般由于回源网络质量变差、回源链路有访问控制拦截回源请求导致源站无响应。		• 建议检查回源网络链路、回源链路中间的访问控制策略、源站处理资源负载、数据库等情况。 • 检查源站是否拦截了WAF回源IP的请求。
status:503	（Service Unavailable）服务不可用，由于超载或停机维护，服务器目前无法使用。		建议检查源站是否异常。
status:504	（Gateway Timeout）网关超时，服务器作为网关或代理，但是没有及时从上游服务器收到请求。		根据以下可能的原因进行排查： • 服务器无法响应，负载过高。 • 源站丢弃请求没有reset。 • 协议通讯不成功。

4.7. 查询与分析语句

本文介绍了使用WAF日志服务配置常用业务指标监控及告警时，用到的日志查询与分析语句。

您可以使用WAF日志服务监控以下业务指标：

> **说明** 单击某个指标，查看对应的日志查询与分析语句。关于监控指标的更多信息，请参见常用监控指标。

- request_time_msec
- upstream_response_time
- status:200
- status:302 or 200 and final_plugin:'cc'
- status:200 and final_plugin:'antifraud'
- status:404
- status:405 and waf_action:'block'
- status:405 and final_plugin:'acl'
- status:444
- status:499
- status:500
- status:502
- status:503
- status:504

request_time_msec

指标释义：表示客户端从发起请求到获得返回结果的请求耗时。

```
* |
SELECT
  user_id,
  host,
  round(
    round(request_time_cnt * 1.0000 / countall, 4) * 100,
    2
  ) AS percent
FROM  (
    SELECT
      user_id,
      host,
      count_if(request_time_msec > 500) AS request_time_cnt,
      COUNT(*) AS countall
    FROM      log
    GROUP BY
      user_id,
      host
  )
GROUP BY
  user_id,
  host,
  percent
```

upstream_response_time

指标释义：表示WAF转发客户端请求到源站服务器，源站返回数据的响应时间。

```
* |
SELECT
  user_id,
  host,
  round(
    round(
      upstream_response_time_cnt * 1.0000 / countall,
      4
    ) * 100,
    2
  ) AS percent
FROM  (
    SELECT
      user_id,
      host,
      count_if(upstream_response_time > 500) AS upstream_response_time_cnt,
      COUNT(*) AS countall
    FROM      log
    GROUP BY
      user_id,
      host
  )
GROUP BY
  user_id,
  host,
  percent
```

status:200

指标释义：表示服务器已成功处理请求，返回了被请求的数据。

```
* |
SELECT
  user_id,
  host AS "域名",
  Rate_200 AS "200比例",
  Rate_302 AS "302比例",
  Rate_404 AS "404比例",
  Rate_405 AS "405比例",
  Rate_444 AS "444比例",
  Rate_499 AS "499比例",
  Rate_500 AS "500比例",
  Rate_502 AS "502比例",
  Rate_503 AS "503比例",
  Rate_504 AS "504比例",
  countall / 60 AS "aveQPS",
  status_200,
  status_302,
  status_404,
  status_405,
  status_444,
  status_499,
  status_500,
  status_502
```

```
  status_502,
  status_503,
  status_504,
  countall
FROM  (
    SELECT
      user_id,
      host,
      round(
        round(status_200 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_200,
      round(
        round(status_302 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_302,
      round(
        round (status_404 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_404,
      round(
        round (status_405 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_405,
      round(
        round (status_405 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_444,
      round(
        round (status_405 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_499,
      round(
        round(status_500 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_500,
      round(
        round(status_502 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_502,
      round(
        round(status_503 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_503,
      round(
        round(status_504 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_504,
      status_200,
      status_302,
      status_404,
      status_405,
      status_444,
      status_499,
      status_500,
```

```
      status_502,
      status_503,
      status_504,
      countall
    FROM      (
        SELECT
          user_id,
          host,
          count_if(status = 200) AS status_200,
          count_if(status = 302) AS status_302,
          count_if(status = 404) AS status_404,
          count_if(status = 405) AS status_405,
          count_if(status = 444) AS status_444,
          count_if(status = 499) AS status_499,
          count_if(status = 500) AS status_500,
          count_if(status = 502) AS status_502,
          count_if(status = 503) AS status_503,
          count_if(status = 504) AS status_504,
          COUNT(*) AS countall
        FROM          log
        GROUP BY
          user_id,
          host
      )
  )
WHERE
  countall > 120
ORDER BY
  Rate_200 DESC
LIMIT
  5
```

status:302 or 200 and final_plugin:'cc'

指标释义：表示请求触发了WAF的JavaScript人机校验策略。

```
* |
SELECT
  user_id,
  host AS "域名",
  Rate_200 AS "200比例",
  Rate_302 AS "302比例",
  Rate_404 AS "404比例",
  Rate_405 AS "405比例",
  Rate_444 AS "444比例",
  Rate_499 AS "499比例",
  Rate_500 AS "500比例",
  Rate_502 AS "502比例",
  Rate_503 AS "503比例",
  Rate_504 AS "504比例",
  countall / 60 AS "aveQPS",
  status_200,
  status_302,
  status_404
```

```
  status_404,
  status_405,
  status_444,
  status_499,
  status_500,
  status_502,
  status_503,
  status_504,
  countall
FROM  (
    SELECT
      user_id,
      host,
      round(
        round(status_200 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_200,
      round(
        round(status_302 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_302,
      round(
        round (status_404 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_404,
      round(
        round (status_405 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_405,
      round(
        round (status_405 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_444,
      round(
        round (status_405 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_499,
      round(
        round(status_500 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_500,
      round(
        round(status_502 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_502,
      round(
        round(status_503 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_503,
      round(
        round(status_504 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_504,
      status_200,
      status_302,
```

```
      status_404,
      status_405,
      status_444,
      status_499,
      status_500,
      status_502,
      status_503,
      status_504,
      countall
    FROM      (
        SELECT
          user_id,
          host,
          count_if(
            status = 200
            AND final_plugin = 'cc'
          ) AS status_200,
          count_if(
            status = 302
            AND final_plugin = 'cc'
          ) AS status_302,
          count_if(status = 404) AS status_404,
          count_if(status = 405) AS status_405,
          count_if(status = 444) AS status_444,
          count_if(status = 499) AS status_499,
          count_if(status = 500) AS status_500,
          count_if(status = 502) AS status_502,
          count_if(status = 503) as status_503,
          count_if(status = 504) AS status_504,
          COUNT(*) AS countall
        FROM          log
        GROUP BY
          user_id,
          host
      )
  )
WHERE
  countall > 120
ORDER BY
  Rate_200 DESC
LIMIT
  5
```

status:200 and final_plugin:'antifraud'

指标释义：表示请求被WAF的数据风控规则拦截。

```
* |
SELECT
  user_id,
  host AS "域名",
  Rate_200 AS "200比例",
  Rate_302 AS "302比例",
  Rate_404 AS "404比例",
```

```
  Rate_405 AS "405比例",
  Rate_444 AS "444比例",
  Rate_499 AS "499比例",
  Rate_500 AS "500比例",
  Rate_502 AS "502比例",
  Rate_503 AS "503比例",
  Rate_504 AS "504比例",
  countall / 60 AS "aveQPS",
  status_200,
  status_302,
  status_404,
  status_405,
  status_444,
  status_499,
  status_500,
  status_502,
  status_503,
  status_504,
  countall
FROM  (
    SELECT
      user_id,
      host,
      round(
        round(status_200 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_200,
      round(
        round(status_302 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_302,
      round(
        round (status_404 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_404,
      round(
        round (status_405 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_405,
      round(
        round (status_405 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_444,
      round(
        round (status_405 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_499,
      round(
        round(status_500 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_500,
      round(
        round(status_502 * 1.0000 / countall, 4) * 100,
        2
```

```
      ) AS Rate_502,
      round(
        round(status_503 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_503,
      round(
        round(status_504 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_504,
      status_200,
      status_302,
      status_404,
      status_405,
      status_444,
      status_499,
      status_500,
      status_502,
      status_503,
      status_504,
      countall
    FROM      (
        SELECT
          user_id,
          host,
          count_if(
            status = 200
            AND final_plugin = 'antifraud'
          ) AS status_200,
          count_if(status = 302) AS status_302,
          count_if(status = 404) AS status_404,
          count_if(status = 405) AS status_405,
          count_if(status = 444) AS status_444,
          count_if(status = 499) AS status_499,
          count_if(status = 500) AS status_500,
          count_if(status = 502) AS status_502,
          count_if(status = 503) AS status_503,
          count_if(status = 504) AS status_504,
          COUNT(*) AS countall
        FROM          log
        GROUP BY
          user_id,
          host
      )
  )
WHERE
  countall > 120
ORDER BY
  Rate_200 DESC
LIMIT
  5
```

status:404

指标释义：表示服务器找不到被请求的资源。

```
* |
SELECT
  user_id,
  host AS "域名",
  Rate_200 AS "200比例",
  Rate_302 AS "302比例",
  Rate_404 AS "404比例",
  Rate_405 AS "405比例",
  Rate_500 AS "500比例",
  Rate_502 AS "502比例",
  Rate_503 AS "503比例",
  Rate_504 AS "504比例",
  countall / 60 AS "aveQPS",
  status_200,
  status_302,
  status_404,
  status_405,
  status_500,
  status_502,
  status_503,
  status_504,
  countall
FROM  (
    SELECT
      user_id,
      host,
      round(
        round(status_200 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_200,
      round(
        round(status_302 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_302,
      round(
        round (status_404 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_404,
      round(
        round (status_405 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_405,
      round(
        round(status_500 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_500,
      round(
        round(status_502 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_502,
      round(
        round(status_503 * 1.0000 / countall, 4) * 100,
```

```
        2
      ) AS Rate_503,
      round(
        round(status_504 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_504,
      status_200,
      status_302,
      status_404,
      status_405,
      status_500,
      status_502,
      status_503,
      status_504,
      countall
    FROM      (
        SELECT
          user_id,
          host,
          count_if(status = 200) AS status_200,
          count_if(status = 302) AS status_302,
          count_if(status = 404) AS status_404,
          count_if(status = 405) AS status_405,
          count_if(status = 499) AS status_499,
          count_if(status = 500) AS status_500,
          count_if(status = 502) AS status_502,
          count_if(status = 503) AS status_503,
          count_if(status = 504) AS status_504,
          COUNT(*) AS countall
        FROM          log
        GROUP BY
          user_id,
          host
      )
  )
WHERE
  countall > 120
ORDER BY
  Rate_404 DESC
LIMIT
  5
```

status:405 and waf_action:'block'

指标释义：表示请求被WAF的规则防护引擎拦截。

```
* |
SELECT
  user_id,
  host AS "域名",
  Rate_200 AS "200比例",
  Rate_302 AS "302比例",
  Rate_404 AS "404比例",
  Rate_405 AS "405比例",
```

```
  Rate_444 AS "444比例",
  Rate_499 AS "499比例",
  Rate_500 AS "500比例",
  Rate_502 AS "502比例",
  Rate_503 AS "503比例",
  Rate_504 AS "504比例",
  countall / 60 AS "aveQPS",
  status_200,
  status_302,
  status_404,
  status_405,
  status_444,
  status_499,
  status_500,
  status_502,
  status_503,
  status_504,
  countall
FROM(
    SELECT
      user_id,
      host,
      round(
        round(status_200 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_200,
      round(
        round(status_302 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_302,
      round(
        round (status_404 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_404,
      round(
        round (status_405 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_405,
      round(
        round (status_405 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_444,
      round(
        round (status_405 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_499,
      round(
        round(status_500 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_500,
      round(
        round(status_502 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_502,
```

```
      round(
        round(status_503 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_503,
      round(
        round(status_504 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_504,
      status_200,
      status_302,
      status_404,
      status_405,
      status_444,
      status_499,
      status_500,
      status_502,
      status_503,
      status_504,
      countall
    FROM      (
        SELECT
          user_id,
          host,
          count_if(status = 200) AS status_200,
          count_if(status = 302) AS status_302,
          count_if(status = 404) AS status_404,
          count_if(
            status = 405
            and waf_action = 'block'
          ) AS status_405,
          count_if(status = 444) AS status_444,
          count_if(status = 499) AS status_499,
          count_if(status = 500) AS status_500,
          count_if(status = 502) AS status_502,
          count_if(status = 503) AS status_503,
          count_if(status = 504) AS status_504,
          COUNT(*) AS countall
        FROM          log
        GROUP BY
          user_id,
          host
      )
  )
WHERE
  countall > 120
ORDER BY
  Rate_405 DESC
LIMIT
  5
```

status:405 and final_plugin:'acl'

指标释义：表示请求被WAF的黑名单及自定义防护策略（ACL访问控制）规则拦截。

```
* |
SELECT
  user_id,
  host AS "域名",
  Rate_200 AS "200比例",
  Rate_302 AS "302比例",
  Rate_404 AS "404比例",
  Rate_405 AS "405比例",
  Rate_444 AS "444比例",
  Rate_499 AS "499比例",
  Rate_500 AS "500比例",
  Rate_502 AS "502比例",
  Rate_503 AS "503比例",
  Rate_504 AS "504比例",
  countall / 60 AS "aveQPS",
  status_200,
  status_302,
  status_404,
  status_405,
  status_444,
  status_499,
  status_500,
  status_502,
  status_503,
  status_504,
  countall
FROM(
    SELECT
      user_id,
      host,
      round(
        round(status_200 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_200,
      round(
        round(status_302 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_302,
      round(
        round (status_404 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_404,
      round(
        round (status_405 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_405,
      round(
        round (status_405 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_444,
      round(
        round (status_405 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_499,
      round(
```

```
      round(
        round(status_500 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_500,
      round(
        round(status_502 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_502,
      round(
        round(status_503 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_503,
      round(
        round(status_504 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_504,
      status_200,
      status_302,
      status_404,
      status_405,
      status_444,
      status_499,
      status_500,
      status_502,
      status_503,
      status_504,
      countall
    FROM      (
        SELECT
          user_id,
          host,
          count_if(status = 200) AS status_200,
          count_if(status = 302) AS status_302,
          count_if(status = 404) AS status_404,
          count_if(
            status = 405
            and final_plugin = 'acl'
          ) AS status_405,
          count_if(status = 444) AS status_444,
          count_if(status = 499) AS status_499,
          count_if(status = 500) AS status_500,
          count_if(status = 502) AS status_502,
          count_if(status = 503) AS status_503,
          count_if(status = 504) AS status_504,
          COUNT(*) AS countall
        FROM          log
        GROUP BY
          user_id,
          host
      )
  )
WHERE
  countall > 120
ORDER BY
  Rate_405 DESC
```

```
LIMIT
  5
```

status:444

指标释义：表示请求被WAF的CC安全防护规则拦截。

```
* |
select
  user_id,
  host AS "域名",
  Rate_200 AS "200比例",
  Rate_302 AS "302比例",
  Rate_404 AS "404比例",
  Rate_405 AS "405比例",
  Rate_444 AS "444比例",
  Rate_499 AS "499比例",
  Rate_500 AS "500比例",
  Rate_502 AS "502比例",
  Rate_503 AS "503比例",
  Rate_504 AS "504比例",
  countall / 60 AS "aveQPS",
  status_200,
  status_302,
  status_404,
  status_405,
  status_444,
  status_499,
  status_500,
  status_502,
  status_503,
  status_504,
  countall
FROM(
    SELECT
      user_id,
      host,
      round(
        round(status_200 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_200,
      round(
        round(status_302 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_302,
      round(
        round (status_404 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_404,
      round(
        round (status_405 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_405,
```

```
    round(
      round (status_405 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_444,
    round(
      round (status_405 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_499,
    round(
      round(status_500 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_500,
    round(
      round(status_502 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_502,
    round(
      round(status_503 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_503,
    round(
      round(status_504 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_504,
    status_200,
    status_302,
    status_404,
    status_405,
    status_444,
    status_499,
    status_500,
    status_502,
    status_503,
    status_504,
    countall
  FROM      (
      SELECT
        user_id,
        host,
        count_if(status = 200) AS status_200,
        count_if(status = 302) AS status_302,
        count_if(status = 404) AS status_404,
        count_if(status = 405) AS status_405,
        count_if(status = 444) AS status_444,
        count_if(status = 499) AS status_499,
        count_if(status = 500) AS status_500,
        count_if(status = 502) AS status_502,
        count_if(status = 503) AS status_503,
        count_if(status = 504) AS status_504,
        COUNT(*) AS countall
      FROM          log
      GROUP BY
        user_id,
        host
    )
```

```
  )
WHERE
  countall > 120
ORDER BY
  Rate_444 DESC
LIMIT
  5
```

status:499

指标释义：表示服务器超时未返回客户端请求的数据，客户端主动断链。服务器返回给客户端499状态码。

```
* |
SELECT
  user_id,
  host AS "域名",
  Rate_200 AS "200比例",
  Rate_302 AS "302比例",
  Rate_404 AS "404比例",
  Rate_405 AS "405比例",
  Rate_444 AS "444比例",
  Rate_499 AS "499比例",
  Rate_500 AS "500比例",
  Rate_502 AS "502比例",
  Rate_503 AS "503比例",
  Rate_504 AS "504比例",
  countall / 60 AS "aveQPS",
  status_200,
  status_302,
  status_404,
  status_405,
  status_444,
  status_499,
  status_500,
  status_502,
  status_503,
  status_504,
  countall
FROM(
    SELECT
      user_id,
      host,
      round(
        round(status_200 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_200,
      round(
        round(status_302 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_302,
      round(
        round (status_404 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_404
```

```
    ) AS Rate_404,
    round(
      round (status_405 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_405,
    round(
      round (status_405 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_444,
    round(
      round (status_405 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_499,
    round(
      round(status_500 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_500,
    round(
      round(status_502 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_502,
    round(
      round(status_503 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_503,
    round(
      round(status_504 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_504,
    status_200,
    status_302,
    status_404,
    status_405,
    status_444,
    status_499,
    status_500,
    status_502,
    status_503,
    status_504,
    countall
  FROM      (
      SELECT
        user_id,
        host,
        count_if(status = 200) AS status_200,
        count_if(status = 302) AS status_302,
        count_if(status = 404) AS status_404,
        count_if(status = 405) AS status_405,
        count_if(status = 444) AS status_444,
        count_if(status = 499) AS status_499,
        count_if(status = 500) AS status_500,
        count_if(status = 502) AS status_502,
        count_if(status = 503) AS status_503,
        count_if(status = 504) AS status_504,
        COUNT(*) AS countall
```

```
        FROM          log
        GROUP BY
          user_id,
          host
      )
  )
WHERE
  countall > 120
ORDER BY
  Rate_499 DESC
LIMIT
  5
```

status:500

指标释义：服务器内部错误（Internal Server Error），表示服务器无法完成请求。

```
* |
SELECT
  user_id,
  host AS "域名",
  Rate_200 AS "200比例",
  Rate_302 AS "302比例",
  Rate_404 AS "404比例",
  Rate_405 AS "405比例",
  Rate_444 AS "444比例",
  Rate_499 AS "499比例",
  Rate_500 AS "500比例",
  Rate_502 AS "502比例",
  Rate_503 AS "503比例",
  Rate_504 AS "504比例",
  countall / 60 AS "aveQPS",
  status_200,
  status_302,
  status_404,
  status_405,
  status_444,
  status_499,
  status_500,
  status_502,
  status_503,
  status_504,
  countall
FROM(
    SELECT
      user_id,
      host,
      round(
        round(status_200 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_200,
      round(
        round(status_302 * 1.0000 / countall, 4) * 100,
        2
```

```
      2
    ) AS Rate_302,
    round(
      round (status_404 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_404,
    round(
      round (status_405 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_405,
    round(
      round (status_405 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_444,
    round(
      round (status_405 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_499,
    round(
      round(status_500 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_500,
    round(
      round(status_502 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_502,
    round(
      round(status_503 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_503,
    round(
      round(status_504 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_504,
    status_200,
    status_302,
    status_404,
    status_405,
    status_444,
    status_499,
    status_500,
    status_502,
    status_503,
    status_504,
    countall
  FROM      (
      SELECT
        user_id,
        host,
        count_if(status = 200) AS status_200,
        count_if(status = 302) AS status_302,
        count_if(status = 404) AS status_404,
        count_if(status = 405) AS status_405,
        count_if(status = 444) AS status_444,
        count_if(status = 499) AS status_499,
```

```
          count_if(status = 500) AS status_500,
          count_if(status = 502) AS status_502,
          count_if(status = 503) AS status_503,
          count_if(status = 504) AS status_504,
          COUNT(*) AS countall
        FROM          log
        GROUP BY
          user_id,
          host
      )
  )
WHERE
  countall > 120
ORDER BY
  Rate_500 DESC
LIMIT
  5
```

status:502

指标释义：错误网关（Bad Gateway），表示服务器作为网关或代理，从上游服务器收到无效响应。一般由于回源网络质量变差、回源链路有访问控制策略拦截回源请求等，导致源站无响应。

```
* |
SELECT
  user_id,
  host AS "域名",
  Rate_200 AS "200比例",
  Rate_302 AS "302比例",
  Rate_404 AS "404比例",
  Rate_405 AS "405比例",
  Rate_444 AS "444比例",
  Rate_499 AS "499比例",
  Rate_500 AS "500比例",
  Rate_502 AS "502比例",
  Rate_503 AS "503比例",
  Rate_504 AS "504比例",
  countall / 60 AS "aveQPS",
  status_200,
  status_302,
  status_404,
  status_405,
  status_444,
  status_499,
  status_500,
  status_502,
  status_503,
  status_504,
  countall
FROM(
    SELECT
      user_id,
      host,
      round(
```

```
    round(
      round(status_200 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_200,
    round(
      round(status_302 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_302,
    round(
      round (status_404 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_404,
    round(
      round (status_405 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_405,
    round(
      round (status_405 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_444,
    round(
      round (status_405 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_499,
    round(
      round(status_500 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_500,
    round(
      round(status_502 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_502,
    round(
      round(status_503 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_503,
    round(
      round(status_504 * 1.0000 / countall, 4) * 100,
      2
    ) AS Rate_504,
    status_200,
    status_302,
    status_404,
    status_405,
    status_444,
    status_499,
    status_500,
    status_502,
    status_503,
    status_504,
    countall
  FROM      (
      SELECT
        user_id,
        host,
```

```
          count_if(status = 200) AS status_200,
          count_if(status = 302) AS status_302,
          count_if(status = 404) AS status_404,
          count_if(status = 405) AS status_405,
          count_if(status = 444) AS status_444,
          count_if(status = 499) AS status_499,
          count_if(status = 500) AS status_500,
          count_if(status = 502) AS status_502,
          count_if(status = 503) AS status_503,
          count_if(status = 504) AS status_504,
          COUNT(*) AS countall
        FROM          log
        GROUP BY
          user_id,
          host
      )
  )
WHERE
  countall > 120
ORDER BY
  Rate_502 DESC
LIMIT
  5
```

status:503

指标释义：服务不可用（Service Unavailable），表示由于超载或停机维护，服务器目前无法使用。

```
* |
SELECT
  user_id,
  host as "域名",
  Rate_200 as "200比例",
  Rate_302 as "302比例",
  Rate_404 as "404比例",
  Rate_405 as "405比例",
  Rate_444 as "444比例",
  Rate_499 as "499比例",
  Rate_500 as "500比例",
  Rate_502 as "502比例",
  Rate_503 as "503比例",
  Rate_504 as "504比例",
  countall / 60 as "aveQPS",
  status_200,
  status_302,
  status_404,
  status_405,
  status_444,
  status_499,
  status_500,
  status_502,
  status_503,
  status_504,
  countall
```

```
  countall
FROM(
    SELECT
      user_id,
      host,
      round(
        round(status_200 * 1.0000 / countall, 4) * 100,
        2
      ) as Rate_200,
      round(
        round(status_302 * 1.0000 / countall, 4) * 100,
        2
      ) as Rate_302,
      round(
        round (status_404 * 1.0000 / countall, 4) * 100,
        2
      ) as Rate_404,
      round(
        round (status_405 * 1.0000 / countall, 4) * 100,
        2
      ) as Rate_405,
      round(
        round (status_405 * 1.0000 / countall, 4) * 100,
        2
      ) as Rate_444,
      round(
        round (status_405 * 1.0000 / countall, 4) * 100,
        2
      ) as Rate_499,
      round(
        round(status_500 * 1.0000 / countall, 4) * 100,
        2
      ) as Rate_500,
      round(
        round(status_502 * 1.0000 / countall, 4) * 100,
        2
      ) as Rate_502,
      round(
        round(status_503 * 1.0000 / countall, 4) * 100,
        2
      ) as Rate_503,
      round(
        round(status_504 * 1.0000 / countall, 4) * 100,
        2
      ) as Rate_504,
      status_200,
      status_302,
      status_404,
      status_405,
      status_444,
      status_499,
      status_500,
      status_502,
      status_503,
      status_504,
```

```
      countall
    FROM      (
        SELECT
          user_id,
          host,
          count_if(status = 200) as status_200,
          count_if(status = 302) as status_302,
          count_if(status = 404) as status_404,
          count_if(status = 405) as status_405,
          count_if(status = 444) as status_444,
          count_if(status = 499) as status_499,
          count_if(status = 500) as status_500,
          count_if(status = 502) as status_502,
          count_if(status = 503) as status_503,
          count_if(status = 504) as status_504,
          COUNT(*) as countall
        FROM          log
        GROUP BY
          user_id,
          host
      )
  )
WHERE
  countall > 120
ORDER BY
  Rate_503 DESC
LIMIT
  5
```

status:504

指标释义：网关超时（Gateway Timeout），表示服务器作为网关或代理，没有及时从上游服务器收到请求。

```
* |
SELECT
  user_id,
  host AS "域名",
  Rate_200 AS "200比例",
  Rate_302 AS "302比例",
  Rate_404 AS "404比例",
  Rate_405 AS "405比例",
  Rate_444 AS "444比例",
  Rate_499 AS "499比例",
  Rate_500 AS "500比例",
  Rate_502 AS "502比例",
  Rate_503 AS "503比例",
  Rate_504 AS "504比例",
  countall / 60 AS "aveQPS",
  status_200,
  status_302,
  status_404,
  status_405,
  status_444,
```

```
  status_499,
  status_500,
  status_502,
  status_503,
  status_504,
  countall
FROM(
    SELECT
      user_id,
      host,
      round(
        round(status_200 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_200,
      round(
        round(status_302 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_302,
      round(
        round (status_404 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_404,
      round(
        round (status_405 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_405,
      round(
        round (status_405 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_444,
      round(
        round (status_405 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_499,
      round(
        round(status_500 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_500,
      round(
        round(status_502 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_502,
      round(
        round(status_503 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_503,
      round(
        round(status_504 * 1.0000 / countall, 4) * 100,
        2
      ) AS Rate_504,
      status_200,
      status_302,
      status_404,
      status_405,
```

```
      status_444,
      status_499,
      status_500,
      status_502,
      status_503,
      status_504,
      countall
    FROM      (
        SELECT
          user_id,
          host,
          count_if(status = 200) AS status_200,
          count_if(status = 302) AS status_302,
          count_if(status = 404) AS status_404,
          count_if(status = 405) AS status_405,
          count_if(status = 444) AS status_444,
          count_if(status = 499) AS status_499,
          count_if(status = 500) AS status_500,
          count_if(status = 502) AS status_502,
          count_if(status = 503) AS status_503,
          count_if(status = 504) AS status_504,
          COUNT(*) AS countall
        FROM          log
        GROUP BY
          user_id,
          host
      )
  )
WHERE
  countall > 120
ORDER BY
  Rate_504 DESC
LIMIT
  5
```