CDN

用户指南

用户指南

控制台介绍

快速开始

Alibaba Cloud CDN(内容分发网络)，建立并覆盖在承载网之上、由分布在不同区域的边缘节点服务器群组成的分布式网络，替代传统以WEB Server为中心的数据传输模式。CDN控制台可以帮助您完成添加CDN加速域名、刷新缓存等配置任务，也提供了实时数据分析的资源监控服务等。本文档主要介绍CDN控制台入门。

CDN运行概况总览

登录到 CDN控制台 后，首页展示的就是当前账户下CDN运行概况总览情况，主要包括：

1. 计费类型展示及变更
2. 关键数据展示区，包含运行域名数、当月总流量等信息
3. 本月概览数据模块
    i. 加速域名产生的带宽峰值信息
    ii. 按照下行流量累计值排名的Top4加速域名
    iii. 访问加速资源的用户区域分布占比
    iv. 用户访问加速资源的实时缓存命中率

注：本月指自然月。

可以通过左侧的导航栏，完成相关的功能设置以及数据浏览：

功能	简述
域名管理	添加加速域名、管理或删除已有加速域名，并可以对加速域名基本信息和配置信息进行变更
监控	包含四部分，流量监控、用户访问监控、数据分析、安全防护
刷新	提供URL刷新和目录刷新两种方式
支出	查看各类服务费用支出情况
日志	日志下载、日志存储（即将上线）、云报表
工具	链路诊断工具、IP查询

业务类型

类型1：图片小文件加速

应用场景介绍

网站或者应用的静态内容分发，例如各种类型的图像文件，html文件、flash动画、css、javascript文件等。适用于各种门户网站、电子商务类网站、新闻资讯类站点或应用、政府/企业官网站点、娱乐游戏类站点或应用等。

操作步骤

步骤 1. 添加加速域名

请参考 快速入门，注意选择业务类型为：**图片小文件加速**

步骤 2. 域名配置

域名添加完成后，需要根据您的业务选择合适的功能对加速域名进行配置，当前所有域名配置为可选，鉴于“图片小文件”加速，推荐设置如下功能

推荐配置

- HTTPS安全加速，仅需开启安全加速模式后上传加速域名证书/私钥，并支持对证书进行查看、停用、启用、编辑操作，了解 证书格式说明。
- 缓存配置，可针对不同“目录路径”和“文件名后缀”的资源进行缓存服务器行为的设置，用户可自定义指定资源内容的缓存过期时间规则。
- 访问控制相关设置，可以保证分发内容安全，防止盗链或者恶意请求造成不必要流量损失。
    - Refer防盗链
    - IP黑名单
- 性能优化相关设置，智能压缩分发内容、忽略URL参数提升缓存命中率。
    - 页面优化
    - 智能压缩
    - 过滤参数
- 更多功能请浏览 CDN功能列表。

类型2：大文件下载加速

应用场景介绍

网站或者应用的静态大文件分发，例如游戏安装包.apk文件、应用更新文件.rar、补丁程序文件、音视频文件等相对较大的文件。适用于下载类站点和音视频的应用

操作步骤

步骤 1. 添加加速域名

请参考 快速入门，注意选择业务类型为：**大文件下载加速**

步骤 2. 域名配置

域名添加完成后，需要根据您的业务选择合适的功能对加速域名进行配置，当前所有域名配置为可选，鉴于“大文件下载”加速，推荐设置如下功能

推荐配置

- HTTPS安全加速，仅需开启安全加速模式后上传加速域名证书/私钥，并支持对证书进行查看、停用、启用、编辑操作，了解 证书格式说明。

- 缓存配置，可针对不同“目录路径”和“文件名后缀”的资源进行缓存服务器行为的设置，用户可自定义指定资源内容的缓存过期时间规则。
- 访问控制相关设置，可以保证分发内容安全，防止盗链或者恶意请求造成不必要流量损失。
    - Refer防盗链
    - IP黑名单
- Range回源，开启该功能，可以减少回源流量消耗，并且提升资源响应时间。
- URL预热 ，将源站的内容主动预热到L2 Cache节点上，用户首次访问可直接命中缓存，缓解源站压力。
- 更多功能请浏览 域名配置概览

类型3：视音频点播加速

应用场景介绍

各类视音频站点，如影视类视频网站、在线教育类视频网站、新闻类视频站点、短视频社交类网站以及音频类相关站点和应用。

操作步骤

步骤 1. 添加加速域名

请参考 快速入门，注意选择业务类型为：**视音频点播加速**

步骤 2. 域名配置

域名添加完成后，需要根据您的业务选择合适的功能对加速域名进行配置，当前所有域名配置为可选，鉴于“视音频点播”加速，推荐设置如下功能

推荐配置

- HTTPS安全加速，仅需开启安全加速模式后上传加速域名证书/私钥，并支持对证书进行查看、停用、启用、编辑操作，了解 证书格式说明。
- 缓存配置，可针对不同“目录路径”和“文件名后缀”的资源进行缓存服务器行为的设置，用户可自定义指定资源内容的缓存过期时间规则。
- 访问控制相关设置，可以保证分发内容安全，防止盗链或者恶意请求造成不必要流量损失。
    - 鉴权设置，URL鉴权功能是通过阿里云CDN加速节点与客户资源站点配合实现的

一种更为安全可靠的源站资源防盗方法，能有效保护用户源站资源。
    - Refer防盗链
    - IP黑名单
- Range回源，开启该功能，可以减少回源流量消耗，并且提升资源响应时间。
- 拖拽播放，开启即支持视音频点播的随机拖拽播放功能
- URL预热 ，将源站的内容主动预热到L2 Cache节点上，用户首次访问可直接命中缓存，缓解源站压力。
- 更多功能请浏览 域名配置概览

类型4：直播流媒体加速

应用场景介绍

为各类视频直播平台提供高性能稳定直播技术支持，如交互性在线教育网站、游戏竞技类直播站点、个人秀场直播、事件类和垂直行业的直播平台等。当前支持RTMP，HLS，FLV三种格式直播内容加速

操作步骤

步骤 1：添加加速域名

请参考 快速入门，注意选择业务类型为：**直播流媒体加速**

注意：该业务类型不支持用户自定义直播中心服务器，统一采用阿里云CDN直播中心源站地址：**live-origin.alivecdn.com**

步骤 2：推流说明

推流地址

```
rtmp://video-center.alivecdn.com/app-name/video-name?vhost=test.example.com
```

控制台上的位置：在【域名管理】选择用于直播的加速域名进行【配置】，进入配置页：

说明

- 默认推流数限制为20个
- video-center.alivecdn.com是直播中心服务器域名，暂不支持自定义
- app-name是应用名称，支持自定义：**字母、数字、下划线组合，不要用特殊字符**，可以更改，不能超过255个字符
- video-name是流名称，支持自定义：**字母、数字、下划线组合，不要用特殊字符**，可以更改，不能超过255个字符
- vhost参数是最终在边缘节点播放的域名，即你的加速域名（如示例中：test.example.com）。

步骤 3：播流说明

- 根据上述中心推的流，边缘支持三种方式读：

方式	URL
RTMP	rtmp://test.example.com/app-name/video-name
FLV	http://test.example.com/app-name/video-name.flv
M3U8	http://test.example.com/app-name/video-name.m3u8

- 控制台上的位置如下所示：

步骤 4：域名配置

域名添加完成后，需要根据您的业务选择合适的功能对加速域名进行配置，当前所有域名配置为可选，鉴于“直播流媒体”加速，推荐设置如下功能

鉴权设置，URL鉴权功能是通过阿里云CDN加速节点与客户资源站点配合实现的一种更为安全可靠的源站资源防盗方法，能有效保护用户源站资源。

注意：

- 目前采用推流播流采用同一套鉴权方案
- 只有进行鉴权配置后，该加速域名才能正常进行推流和播流，当前直播业务类型**仅支持A类型鉴权方式**
- 推流和播流地址需要分别进行鉴权签名计算，每一个签名都是严格按照URL计算的，故不可使用推流URL计算得到的签名应用到播流地址，同理每一种播流地址都会对应不同的鉴权计算结果
- 计算签名时的URL无需携带参数，例如计算推流鉴权签名时，无需携带?vhost=test.yourcompany.com
- 举例如下

操作步骤	**内容**
资源URL	rtmp://video-center.alivecdn.com/app-name/video-name
鉴权设置	鉴权方式：A方式 鉴权Key：test123 有效时间3600s
推流地址	rtmp://video-center.alivecdn.com/app-name/video-name?auth_key=1449030595-0-0-dee5f3819d7b62a9830ee2913caf111c&vhost=test.example.com
播流地址（以FLV格式为例）	http://test.example.com/app-name/video-name.flv?auth_key=1449030834-0-0-5e1c604710241001fd7a367bc96a17b7

- Notify_URL设置，流状态实时反馈，通过HTTP接口向用户服务器发送GET请求，将视频流推送成功，断流成功的状态实时反馈给用户，用户服务器通过200响应返回接口返回结果，默认返回 1 表示接收成功；0代表接收失败；

类型5：全站加速

应用场景介绍

全站加速**融合了动态加速和静态加速**，突破以往的单项加速，通过简单的配置即可智能区分动静态请求，实现整站加速。 全站加速适用于各行业动静态内容混合、含较多动态资源请求（如asp、jsp、php等格式的文件）的站点：

- 场景1：丰富和复杂的动态内容降低了页面加载速度，影响用户体验。
- 场景2：单线源站、突发流量、网络拥塞等导致页面延迟和内容交付失败。
- 场景3：游戏类客户，动态内容实时通信高并发，传统通信协议无法满足性能需求。
- 场景4：源站负载分配不均，突发访问造成的源站压力。
- 场景5：国内运营商环境复杂，网站被劫持，站点内容遭篡改，仅使用HTTP协议传输可能会有动态内容泄露风险，需要寻求更安全高效的网络链路和内容分发途径。

针对以上各种场景，阿里云CDN全站加速提供：

- 动静分离加速，动态内容采用智能路由、传输协议优化和链路复用技术，静态内容采用边缘缓存，提升整站资源加载速度。
- 实时探测及平滑跨网技术稳定高效处理高流量负载，提供全天候全网可用性。
- 回源负载均衡、多源主备、连接复用和有序回源技术降低源站压力和故障风险。
- 全链路HTTPS安全加速、防盗链、IP限流等保证源站安全。
- 自定义设置动静规则、缓存规则并配备全景信息监控和告警功能。

> 注意：**全站加速默认纯动态加速**，即所有动静态请求都通过最优路由回源获取资源，可通过配置指定静态文件类型或路径，实现智能区分动静态资源，静态资源缓存在边缘节点上，动态资源使用动态加速，达到最快的加速效果。

操作步骤

步骤 1. 添加加速域名

请参考 快速入门，注意选择业务类型为：**全站加速**

步骤 2. 域名配置

域名添加完成后，全站加速默认使用的是 **纯动态加速**，需通过 **配置动静态加速规则** 指定静态文件，指定的静态文件将使用静态加速，缓存在CDN节点上，达到更好的加速效果。具体配置方法如下：

动静态加速规则设置：

- 静态文件类型
- 静态URI设置
- 静态路径设置
- 特殊header头设置
- 动态协议跟随回源

推荐配置：

- HTTPS安全加速，仅需开启安全加速模式后上传加速域名证书/私钥，并支持对证书进行查看、停用、启用、编辑操作，了解 证书格式说明。
- 访问控制相关设置，可以保证分发内容安全，防止盗链或者恶意请求造成不必要流量损失。
    - Refer防盗链
    - IP黑名单
- 性能优化相关设置，智能压缩分发内容、忽略URL参数提升缓存命中率。
    - 页面优化
    - 智能压缩
    - 过滤参数
- 更多功能请浏览 CDN功能列表。

计费规则

- CDN全站加速计费项为“**基础费用**”+“**请求数费用**”。其中“**基础费用**”是根据CDN服务所选择的“按峰值带宽”或“按流量”计费的基础费用。“**请求数费用**”包含动态HTTP请求数、动态HTTPS请求数和静态HTTPS请求数，分别按照单价按日计费。
- 计费详情请看 全站加速计费规则。

类型6：移动加速

概述

1. 前 言

移动加速(Mobile Accelerator)是阿里云针对**移动应用**推出的**动静态全网加速**产品，旨在依托阿里云遍布全国的CDN节点，海量带宽网络等优越的基础设施资源，以及使用智能域名解析、无线协议优化、内容动态压缩、运营商级别优化等技术，为开发者提供更快、更稳定的网络接入能力，有效提升移动应用的可用性及用户体验。

2. 功能特性

移动加速服务将主要通过以下几种技术手段来实现移动应用网络加速：

协议优化：采用深度优化定制的私有协议替换传统的HTTP协议，收获多路复用、请求头压缩、请求优先级支持等收益，防止内容劫持现象发生。同时我们也为云加速服务终端与加速节点间长连复用，最小化TCP的建连开销，提高连接利用率和请求响应速度；

链路优化：以阿里云遍布全国的优质边缘节点，海量的带宽资源为基础设施，结合HTTPDNS智能路由精准的调度，实现快速选路，就近接入；云加速节点会缓存热点内容，大大提高访问效率；云加速节点和ECS间搭建专线进行链路加速，如果您已经在使用阿里云ECS作为服务后端，加速结果更是锦上添花；

全站加速：支持HTTP/HTTPS的动态和静态请求的全站加速

3. 下载并安装SDK

移动加速通用版SDK提供 iOS和Android两个版本，支持动态加速域名列表管理，首次安装后即可对所有移动加速域名进行全网提速，可在CDN控制台管理移动加速域名的状态和配置，查看SDK开发指南

- iOS SDK开发指南
- Android SDK开发指南
- 控制台使用说明

移动加速开通说明

一、开通服务

1.1 申请“移动加速”资格

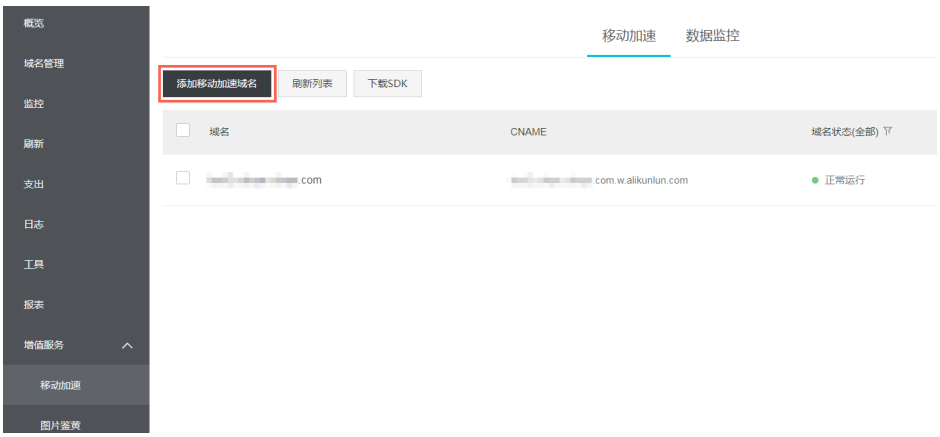
目前CDN移动加速已全面公测，可在移动加速产品介绍页申请使用资格：

1.2 开通后，进入移动加速控制台

在CDN控制台左侧导航栏的【增值服务】里可进入【移动加速】的控制台：

二、添加加速域名

2.1 点击“添加移动加速域名”：



2.2 在添加域名菜单里面，选择要加速的域名:

2.3 如果没有域名可选择，先添加域名：

三、移动端SDK集成

SDK集成参考文档：Android SDK开发指南 和 iOS SDK开发指南。

配置样例

1. 配置目标

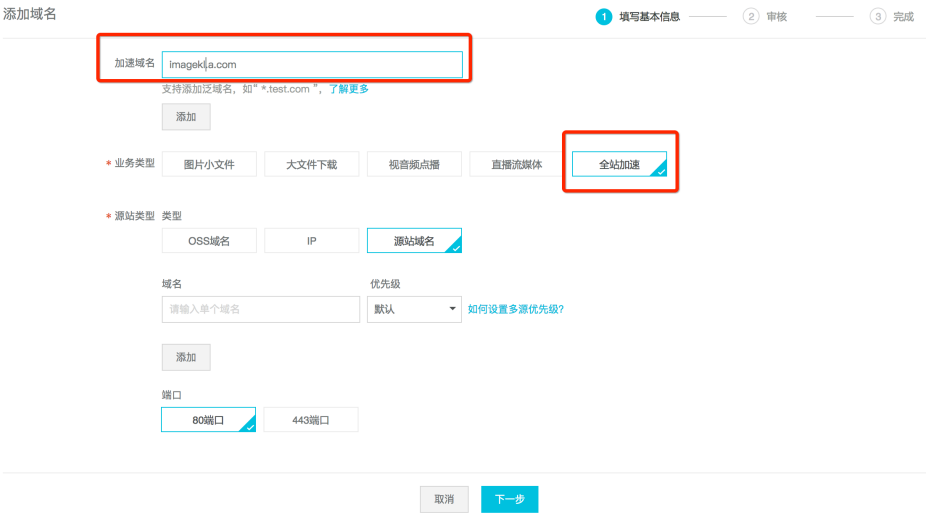
1.1、配置加速域名，用于接入CDN移动加速系统；

1.2、保证流量逐步接入移动加速。

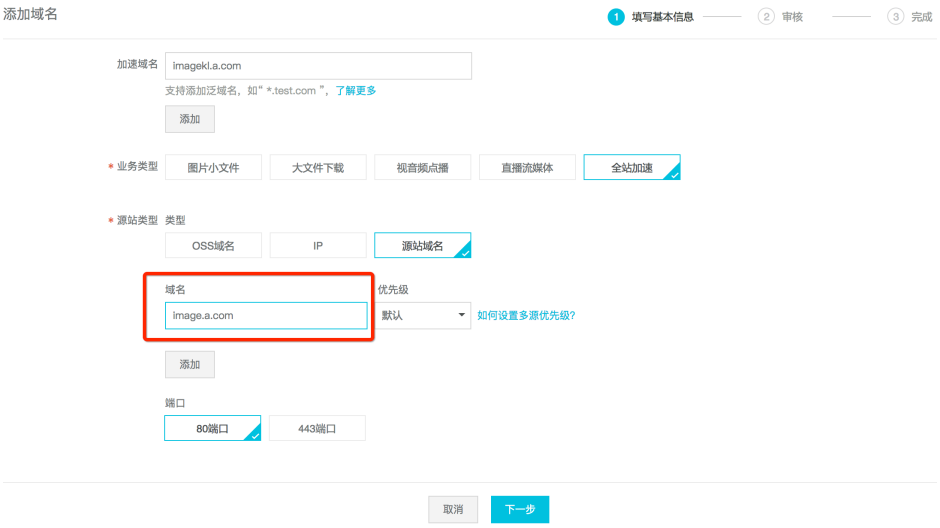
2. 配置方案

假设待加速域名为image.a.com。

2.1、新申请一个CDN移动加速域名（ 假设是imagekl.a.com ），配置到CDN控制台，业务类型选择**全站加速**。



2.2、配置源站类型为**源站域名**，回源域名是image.a.com；



2.3、 上述配置完成后，CDN会为加速域名分配一个CNAME域名，如imagekl.a.com.w.kunlunpi.com（请以CDN实际分配的CNAME域名为准），修改imagekl.a.com的DNS权威服务器配置，将CDN分配的imagekl.a.com.w.kunlunpi.com添加成imagekl.a.com域名的CNAME记录。

2.4、 配置动静态加速规则

- 添加域名后，点击配置，进入加速域名配置界面，如下图所示：

- 移动加速相关的配置项为：动静态加速规则，配置规则如下：
    - 静态文件类型
        - 请求URL后缀名匹配时，为静态加速请求。
        - 例：配置为.jpg,.txt,.html，请求 /1.png、/1/2/3.txt、、/a.html均为静态加速请求。
    - 静态URI设置：
        - 请求URL和配置内容完全匹配时，为静态加速请求。
        - 例：/1/2/3.jpg，请求 /1/2/3.jpg为静态加速请求，其余为动态加速请求。
    - 静态路径设置：
        - 请求URL和配置内容正则匹配（仅支持*正则匹配，*匹配0个或多个任意字符）时，为静态加速请求。
        - 例：/*.py，请求 /a/b/c.py为静态加速请求，/a/b/c.py/d为动态加速请求。

3. 验证方法

上述四个配置步骤完成后，等待2.4 DNS权威服务器解析结果生效后，可以通过以下两步验证CDN移动加速配置是否已经生效。

3.1、 ping imagekl.a.com 确认该域名已经解析到阿里CDN；

3.2、 访问测试对象确认能正确返回。

至此，CDN控制台的相关配置工作已经完成，可以进入后续的SDK集成调试阶段。

4. 备注

配置过程中注意事项请参考概述。

iOS SDK开发指南

本文档介绍了MAC iOS SDK的使用方式。

集成前可参考：移动加速 iOS Demo。

1. SDK集成

- 指定Master仓库和阿里云仓库：

```
source 'https://github.com/CocoaPods/Specs.git'
source 'https://github.com/aliyun/aliyun-specs.git'
```

- 添加依赖：

```
pod 'AlicloudMAC', '~> 1.0.0'
```

2. SDK使用

移动加速SDK内部Log查看Tips：可通过过滤字段[MAC查看。

2.1 获取加速示例并初始化

- AppKey和AppSecret可在 App列表页 获取。

```
AlicloudMACService *service = [AlicloudMACService sharedInstance];
[service initWithAppKey:@"******" appSecret:@"******" callback:^(BOOL res, NSError *error) {
if (res) {
NSLog(@"MAC SDK init success.");
} else {
NSLog(@"MAC SDK init failed, error: %@", error);
```

```
}
}];
```

2.2 加速请求配置

若原生网络请求基于 NSURLConnection 或者 NSURLSession（session对象通过sharedSession:获取）发出，SDK可自动拦截原生网络请求，走到加速链路。

若原生网络请求基于 NSURLSession（session对象配置有自定义 NSURLSessionConfiguration），需注册移动加速的 MACURLProtocol，如下所示：

```
NSURLSessionConfiguration *configuration = [NSURLSessionConfiguration defaultSessionConfiguration];
configuration.protocolClasses = @[ [MACURLProtocol class] ];
NSURLSession *session = [NSURLSession sessionWithConfiguration:configuration];
```

- 移动加速SDK是通过注册 NSURLProtocol 拦截网络请求，需要注意 NSURLProtocol 的注册顺序。多个 NSURLProtocol 注册后，网络请求拦截为注册的相反顺序。移动加速 MACURLProtocol 的注册时机为SDK初始化时，调用停止和重启接口时，分别为注销和重新注册 MACURLProtocol。
- 示例：

```
[[AlicloudMACService sharedInstance] initWithAppKey:testAppKey appSecret:testAppSecret callback:^(BOOL res,
NSError *error) {
if (res) {
/* HookURLProtocol注册在SDK初始化之后，因此HookURLProtocol先拦截到网络请求 */
[NSURLProtocol registerClass:[HookURLProtocol class]];
}
}
```

2.3 自定义降级策略

- 用户可设置降级策略，满足降级条件的网络请求，降级走原生网络库链路。
- 基于下述接口配置：

```
- (void)setDegradationPolicy:(id<MACDegradationDelegate>)delegate;
```

2.4 停止和重启移动加速

- 调用2.1所示的初始化接口，并按照2.2完成配置后，原生网络请求可自动被拦截，走到加速链路。
- 调用停止接口，停止网络请求拦截。

```
/**
停止移动加速
```

```
*/
- (void)stop:(MACCallbackHandler)callback;
```

- 调用重启接口，重新恢复网络请求拦截。

```
/**
重启移动加速
*/
- (void)restart:(MACCallbackHandler)callback;
```

2.5 如何查看网络请求是否加速成功？

- 打开移动加速SDK Log。
- SDK初始化成功后，发出网络请求，可看到如下日志：

```
[MACURLProtocol]-[I]: URL: [https://xxx.xxx.com/xx], accelerate type: [2]
[MACURLProtocol]-[D]: Start loading request: <NSMutableURLRequest: 0xxxxxxxxxxxx> { URL:
https://xxx.xxx.com/xx }
```

- 网络请求结束后，可查看到如下日志，
    - request result
        - 1：网络请求成功
        - 0：网络请求失败
    - accelerate result
        - 1：网络请求加速成功
        - 0：网络请求加速失败

```
[MACACCSNetworkRequest]-[I]: [https://xxx.xxx.com/xx] request result: [1], accelerate result: [1]
```

3. API接口

```
/**
降级策略定义
*/
@protocol MACDegradationDelegate <NSObject>

- (BOOL)shouldDegrade:(NSString *)hostName;

@end

/**
SDK回调Handler定义

@param res 回调结果
*/
```

```
typedef void (^MACCallbackHandler)(CallbackResult *res);

/**
SDK初始化并开启移动加速

@param appKey AppKey
@param appSecret AppSecret
@param callback 回调
*/
- (void)initWithAppKey:(NSString *)appKey
appSecret:(NSString *)appSecret
callback:(MACCallbackHandler)callback;

/**
设置自定义降级策略

@param delegate 降级策略
*/
- (void)setDegradationPolicy:(id<MACDegradationDelegate>)delegate;

/**
停止移动加速
*/
- (void)stop:(MACCallbackHandler)callback;

/**
重启移动加速
*/
- (void)restart:(MACCallbackHandler)callback;

/**
日志开关

@param enabled YES: 打开; NO: 关闭（默认）
*/
- (void)setLogEnabled:(BOOL)enabled;
```

4. 示例

```
/**
初始化MAC SDK
*/
- (void)initMACSDK {
AlicloudMACService *service = [AlicloudMACService sharedInstance];
[service setDegradationPolicy:(id)self];
[service initWithAppKey:@"******" appSecret:@"******" callback:^(BOOL res, NSError *error) {
if (res) {
NSLog(@"MAC SDK init success.");
} else {
NSLog(@"MAC SDK init failed, error: %@", error);
}
}];
}
```

```
/**
自定义降级策略

@param url 请求URL
@return YES: 降级到原生网络库; NO: 不降级
*/
- (BOOL)shouldDegrade:(NSURL *)url {
/* 若请求Host为a.b.com，降级走原生网络库 */
if ([[url host] isEqualToString:@"a.b.com"]) {
return YES;
}
return NO;
}

static NSURLSession *_session;

/**
发网络请求
*/
- (void)sendNetworkReqeust {
static dispatch_once_t onceToken;
dispatch_once(&onceToken, ^{
/* 若基于NSURLSession发网络请求并配置SessionConfiguration，需要注册MACURLProtocol */
if (!_session) {
NSURLSessionConfiguration *configuration = [NSURLSessionConfiguration defaultSessionConfiguration];
configuration.protocolClasses = @[ [MACURLProtocol class] ];
_session = [NSURLSession sessionWithConfiguration:configuration];
}
});
NSURL *url = [NSURL URLWithString:@"xxxxxx"];
NSURLRequest *request = [NSURLRequest requestWithURL:url];
NSURLSessionDataTask *task = [_session dataTaskWithRequest:request completionHandler:^(NSData * _Nullable
data, NSURLResponse * _Nullable response, NSError * _Nullable error) {
if (error) {
NSLog(@"Error: %@", error);
return;
}
NSLog(@"Content: %@", [[NSString alloc] initWithData:data encoding:NSUTF8StringEncoding]);
}];
[task resume];
}
```

Android SDK开发指南

- 3.支持的版本
- 4.API
- 5.最佳实践 5.1 初始化5.2 构建请求对象5.3 同步请求过程5.4 异步请求过程如何判断加速是否成功

1.前言

本文旨在介绍MAC Android SDK的接入步骤和使用方法

2.安装

2.1 配置maven仓库

build.gradle添加阿里云maven仓库

```
allprojects {
repositories {
maven {
url "http://maven.aliyun.com/nexus/content/repositories/releases"
}
}
}
```

2.2 配置gradle依赖

```
dependencies {
compile 'com.aliyun.ams:alicloud-android-mac:1.0.0'
}
```

目前MAC android sdk只支持arm架构，建议用真机进行测试

2.3 Manifest配置

2.3.1 添加组件

```
        <service
android:name="anetwork.channel.aidl.NetworkService"
android:exported="false">
<intent-filter>
<action android:name="anetwork.channel.aidl.IRemoteNetworkGetter" />
</intent-filter>
</service>
```

2.3.2 添加权限

```
    <uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
```

2.4 Proguard配置

```
-keep class com.aliyun.ams.** {*;}
-keep public class org.android.spdy.** {*;}

-dontwarn com.alibaba.**
-dontwarn com.taobao.**
-dontwarn anetwork.channel.**
-dontwarn org.android.**
```

3.支持的版本

mac sdk支持的android最小版本为10

```
minSdkVersion 10
```

4.API

- MacClient

MacClient主要用来发起请求Request和得到响应Response，使用方法请参考最佳实践：

```
public final class MacClient {

// 用于MAC sdk的初始化
public static void init(MacConfig config);

// 根据输入的Request获得一个Call对象
public Call newCall(Request request);

// MacClient的Builder
public static final class Builder {
public MacClient build();
}
}
```

- MacConfig

MAC sdk在初始化时需要传入全局配置MacConfig，使用方法请参考最佳实践-初始化:

```
public final class MacConfig {

// MacConfig的Builder
public static final class Builder {
// 设置Context
public Builder context(Context context);

// 设置appKey
public Builder appKey(String appKey);

// 设置appSecret
public Builder appSecret(String appSecret);

// 创建MacConfig对象
public MacConfig build();
}
}
```

- Request

Request表示一个HTTP请求，每一个Request包含一个URL、method、请求header和body，使用方法请参考最佳实践-构建请求对象：

```
public final class Request {

// 返回URL
public String url();

// 返回method，默认为Get
public String method();

// 返回请求头部
public Map<String, String> headers();

// 返回请求body
public byte[] body();

// Request的Builder
public static final class Builder {
// 设置URL
public Builder url(String url);
// 设置method
public Builder method(String method, byte[] body);
// 设置header
public Builder headers(Map<String, String> headers);
// 添加header
public Builder addHeader(String name, String value);
// 移除header
public Builder removeHeader(String name);
// 构建Request对象
public Request build();
}
}
```

- Response

Response表示一个Request的响应，每一个Response包含状态码、响应头部以及响应body：

```
public final class Response {
// 返回状态码
public int code();
// 返回响应头部
public Map<String, String> headers();
// 返回响应body
public byte[] body();
// 返回请求是否成功
public boolean isSuccessful();
}
```

- Callback

MAC sdk允许用户使用异步Callback的方式，正常时返回Response，异常时返回MacException，使用方法请参考**最佳实践-异步请求过程**：

```
public interface Callback {
// 正常时返回Response
void onResponse(Call call, Response response);

// 异常时返回MacException
void onFailure(Call call, MacException exception);
}
```

5.最佳实践

5.1 初始化

第一步：调用MacConfig.init方法，设置AppKey，AppSecret，Context，建议在Application.onCreate时调用：

```
public class DemoApplication extends Application {

@Override
protected void attachBaseContext(Context base) {
super.attachBaseContext(base);
}

@Override
public void onCreate() {
super.onCreate();

// 初始化MacConfig
```

```
MacConfig config = new MacConfig.Builder()
.context(this)
.appKey(APP_KEY)
.appSecret(APP_SECRET)
.build();

MacClient.init(config);
}
}
```

其中，AppKey和AppSecret可在 App列表页 获取。

第二步：构造MacClient对象，通过该对象来进行网络操作：

```
// 构造MacClient对象
MacClient client = new MacClient.Builder().build();
```

5.2 构建请求对象

请求对象Request可以设置url，header，method等，其中method默认为Get方法：

```
Request req = new Request.Builder()
.url(url)
.headers(headers)
.addHeader("User-Agent", "Your UA")
.method("POST", body)
.build();
```

5.3 同步请求过程

下面为移动加速的同步请求示例，**使用时请确保同步请求方法在后台线程中执行**:

```
new Thread(new Runnable() {
@Override
public void run() {
Response rsp = null;
try {
rsp = client.newCall(req).execute();
} catch (MacException e) {
e.printStackTrace();
}

if (rsp != null) {
int statusCode = rsp.code();
byte[] data = rsp.body();
Log.d(TAG, "[DemoActivity] execute statusCode: " + statusCode + " data: " + new String(data));
}
}
}).start();
```

5.4 异步请求过程

下面为移动加速的异步请求示例，**使用时请确保异步请求方法在后台线程中执行**:

```
new Thread(new Runnable() {
@Override
public void run() {
client.newCall(req).enqueue(new Callback() {
@Override
public void onResponse(Call call, Response response) {
int statusCode = response.code();
byte[] data = response.body();

Log.d(TAG, "[DemoActivity] onResponse statusCode: " + statusCode + " data: " + new String(data));
}

@Override
public void onFailure(Call call, MacException e) {
Log.d(TAG, e.getMessage(), e);
}
});
}
}).start();
```

如何判断加速是否成功

过滤和查看tag为mac的日志，例如控制台通过adb logcat -s mac来过滤

请求成功后可以看到类似日志：

```
[DHandler] url: https://xxx/xxx.html AccSuccess: 1 reqSuccess: 1
```

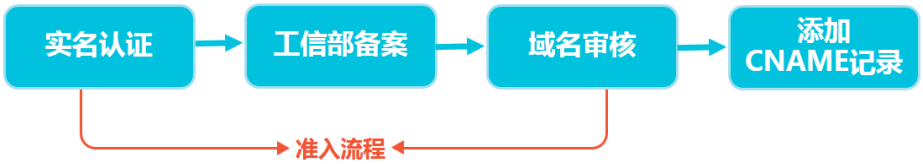
其中，AccSuccess为1表示加速成功，reqSuccess为1表示请求成功

域名准入标准

为您的域名加速之前，请先了解以下规则和标准。

CDN加速域名准入标准

准入与生效流程



1. 实名认证：请登录阿里云官网完成。。
2. **在工信部完成备案**：推荐接入阿里云备案。
3. 域名审核：加速域名的源站内容，您可以选择保存于ECS或OSS。如源站内容不在阿里云，接入前请联系人工审核。
4. 添加CNAME记录：将您的域名指向CDN生成的CNAME域名，即在DNS服务商处为您的域名添加CNAME记录，请参考如何配置CNAME。

注意:

- 如果你的源站部署在ECS上，请关注ECS带宽；建议您的带宽至少为你整体业务量的20%。
- 源站安全软件设置中，请确保CDN缓存节点可访问源站。
- 请确保CDN加速服务停止后，所有请求都将回源。
- 添加完成配置后，你得到的CNAME域名不能直接访问，只能使用CNAME访问。
- 对于大文件，不建议使用range：0~无穷大。

域名审核标准

所有接入CDN的域名都要经过审核。CDN目前不支持接入的加速域名类型**包括但不限于**:

- 无法正常访问或内容不含有任何实质信息
- 游戏私服类
- 传奇类游戏、纸牌类游戏
- 盗版软件等无版权下载网站
- P2P类金融网站
- 彩票类网站
- 违规医院和药品类网站
- 涉黄、涉毒、涉赌等
- 自动超时拒绝：您的域名因不符合CDN接入规则而拒绝，请您查看之前的反馈结果，合规后可再行申请提交审核。

属于以上违规内容的加速域名被攻击或者恶意下载导致的费用损失，阿里云CDN将不承担任何责任，全部损失将由您自行承担。

- 对于您已接入阿里云CDN的域名，会进行定期复审。如发现以上任何一种违规行为，系统将立即中止该域名的CDN加速，同时中止您所有域名的CDN服务。

- 若您的域名加速被**“无法正常访问或内容不含有任何实质信息”**理由拒绝，且您的业务又是合规业务，您可以开启一个工单，将网站的业务截图内容（截图包含该域名）通过工单发送。工单单独审核后

，会告知您第二次的审核结果。

数量限制

数量	限制数量
域名	每个阿里云账户下，最多支持加速 **20个** 域名。
IP源站	每个加速域名的默认IP源站数量限制为 **10个** IP地址。
缓存刷新类操作	**URL刷新**：2000条/日/每账户。 **目录刷新**：100个/日/每账户

如有大量域名加速需求，请提工单申请特殊支持。

加速域名回收规则

如果您的加速域名...	系统会...	如需继续使用CDN加速，您需要...
超过90天没有任何访问流量 （包含处于“正常运行”状态）	自动停用该域名 仍保存该加速域名相关记录	启用加速域名。
处于“停用”状态超过120天 （包含“审核未通过”状态）	自动删除该域名相关记录	重新添加域名。

CDN功能列表

CDN 功能列表

HTTPS安全加速

项目	说明	默认值
HTTPS安全加速	提供全链路HTTPS安全加速方案，仅需开启安全加速模式后上传加速域名证书/私钥，并支持对证书进行查看、停用、启用、编辑操作	未开启
强制跳转	加速域名开启“HTTPS安全加速”的前提下，支持自定义设置，将用户的原请求方式进行强制	未开启

	跳转	

回源设置

项目	说明	默认值
回源 host	指定回源的 host 域名，提供三种选项：加速域名、源站域名、自定义域名	加速域名
协议跟随回源	开启该功能后，回源使用协议和客户端访问资源的协议保持一致	未开启

缓存设置

项目	说明	默认值
缓存过期时间	自定义指定资源内容的缓存过期时间规则	未开启
设置HTTP头	可设置http请求头，目前提供9个http请求头参数可供自行定义取值	未开启
自定义404页面	提供三种选项：默认404、公益404、自定义404	默认404

访问控制

项目	说明	默认值
Refer防盗链	用户可以通过配置访问的 referer 黑白名单来对访问者身份进行识别和过滤	未开启
鉴权配置	URL鉴权方式保护用户源站资源	未开启
IP黑名单	用户可以通过配置访问的 IP黑名单来对访问者身份进行识别和过滤	未开启

性能优化

项目	说明	默认值
页面优化	压缩与去除页面中无用的空行、回车等内容，有效缩减页面大小	未开启
智能压缩	支持多种内容格式的智能压缩，有效减少用户传输内容的大小	未开启

过滤参数	勾选后，回源会去除 url 中？之后的参数	未开启

视频相关设置

项目	说明	默认值
range回源	指客户端通知源站服务器只返回指定范围的部分内容，对于较大文件的分发加速有很大帮助	未开启
拖拽播放	开启即支持视音频点播的随机拖拽播放功能	未开启
Notify_URL	【直播适用】流状态实时信息回调，可以及时通知用户推流或断流操作结果	未开启

其他设置

项目	说明	默认值
设置httpDNS	httpDNS是域名解析服务，通过HTTP协议直接访问阿里云CDN的服务器	未开启

设置httpDNS

功能简介

- 传统的DNS解析是通过访问运营商Local DNS获得解析结果，这种方式容易引发域名劫持、域名解析错误、流量跨网等问题， 从而导致网站无法访问或访问缓慢。
- httpDNS是域名解析服务，通过HTTP协议直接访问阿里云CDN的服务器，由于绕过了运营商的Local DNS，因此可以避免DNS劫持并获得实时精确的DNS解析结果。
- **原理：** 客户端发起请求，通过HTTP协议访问阿里云CDN指定httpDNS服务端，该服务端依托遍布各地的二级DNS节点解析域名，获得域名解析结果并最终返回给客户端。

httpDNS 接口

支持通过HTTP接口直接访问，访问方式如下

服务URL：

```
http://umc.danuoyi.alicdn.com/multi_dns_resolve
```

请求方法：POST

支持参数： client_ip=x.x.x.x 如果使用发起httpDNS请求的客户端IP，该参数可以忽略。

请求示例： 待解析的多个域名放到POST的body中，域名之间以空白分隔，空白可以是空格、TAB和换行符。

```
#curl 'http://umc.danuoyi.alicdn.com/multi_dns_resolve?client_ip=182.92.253.16
' -d 'd.tv.taobao.com'
```

返回格式： json 数据，解析后提取域名对应的ip，多个ip之间可以做轮询，需要遵循ttl进行缓存和过期。

```
{"dns":[{"host":"d.tv.taobao.com","ips":[{"ip":"115.238.23.240","spdy":0},{"ip":"115.238.23.250","spdy":0}],"ttl"
:300,"port":80}],"port":80}
```

多个域名请求事例：

请求示例

```
#curl 'http://umc.danuoyi.alicdn.com/multi_dns_resolve?client_ip=182.92.253.16
' -d 'd.tv.taobao.com vmtstvcdn.alicdn.com'
```

返回示例

```
{"dns":[{"host":"vmtstvcdn.alicdn.com","ips":[{"ip":"115.238.23.250","spdy":0},{"ip":"115.238.23.240","spdy
":0}],"ttl":300,"port":80},{"host":"d.tv.taobao.com","ips":[{"ip":"115.238.23.240","spdy":0},{"ip":"115.238.23.
250","spdy":0}],"ttl":300,"port":80}],"port":80}
```

内容回源设置

设置多源优先级与自定义端口

设置多源优先级

功能介绍

阿里云CDN支持三种类型回源域名，包括**oss回源域名**、**IP**和**自定义域名**。其中IP和自定义域名支持多IP或多域名设置，并支持用在多源站场景下，进行回源优先级设置。

当用户选择的回源源站类型为IP或自定义域名时，可设置多个源站，并为多源站设置优先级。添加多源站时，源站优先级为**“主”**和**“备”**，优先级为**主>备**。

用户100%回源流量都将首先回源优先级高的源站，如果某个源站健康检查连续3次都是失败的话，则100%的流量都将选择优先级第二的源站回源。如果主动健康检查成功的话，该源站就会重新标记为可用，恢复原来优先级。当所有源站的回源优先级一样时，cdn将自动轮询回源。

源站健康检查：实行主动四层健康检查机制，每5秒主动健康检查源站一次。

主要支持场景：主备方式切换源站

配置说明

进入CDN域名管理列表页，选择相应域名进入配置页面，可在回源设置里，设置【多源优先级功能】。

1. 从域名列表点击 **配置**，在域名配置页面打开源站设置功能。
2. 设置 **回源源站** 和 **优先级**。
    - 如果您选择的源站信息为 **IP** 或 **源站域名**，则您仍然按照外网流量标准进行计费。
    - 如果您选择的源站信息为 **OSS域名**，即从CDN回源OSS，则按照内网的价格计费。**OSS价格详情。**
    - **如果选择域名类型为“源站域名”，并设置了一个oss的域名，那么仍然按照外网流量价格计费。**
3. 设置完成后，点击 **确认**，设置成功。

您可以选择的回源端口类型为：80端口、443端口和自定义端口。

注意：

- 多源优先级的设置只支持IP和源站域名类型，OSS域名不支持多源优先级功能；您可以根据实际需求，选择适合自己的源站类型及设置合理的优先级
- 直播加速不支持源站设置

设置自定义端口

您可以在开通白名单后，设置自定义端口。自定义端口支持范围为0-65535。

- 当您的静态或动态协议设置为**跟随**时，无法设置自定义端口。
- 如果您通过OpenAPI，设置自己的回源协议为**跟随**，请确保您的回源协议和自定义端口均能正常使用。
- 当您通过端口设置了回源协议（**HTTP**或**HTTPS**）和自定义端口时，则无论您在控制台如何设置，回源都将按照端口的配置进行。

私有bucket回源授权

功能介绍

私有bucket回源授权是指若加速域名想要回源至该用户账号下标记为私有的bucket时，需要首先进行授权，授权成功并开启授权配置后，用户开启了私有bucket授权的域名有权限访问私有bucket

风险提示

- 授权成功并开启了对应域名的私有bucket功能，该加速域名可以访问您的私有bucket内的资源内容，开启该功能前，请根据实际的业务情况，谨慎决策；若您授权的私有bucket内容并不适合作为cdn加速域名的回源内容，请勿授权或者开启该功能。
- 您可以配合使用cdn提供的refer防盗链功能，鉴权等功能，有效保护您的资源安全。
- 若您的网站有攻击风险，请购买高防服务，请勿授权或开启私有bucket功能。

配置说明

- 如何开启私有bucket回源授权

域名配置 — 源站设置 — 点击 “私有bucket授权”

点击 “立即授权”

3. 授权成功，为该域名开启私有bucket回源配置，点击确定

4. 成功

- 如何关闭私有bucket回源授权

进入访问控制 — 角色管理

删除“AliyunCDNAccessingPrivateOSSRole”授权

3. 私有bucket授权删除成功

提示：若您的加速域名正在使用私有bucket做为源站进行回源，请不要关闭或删除私有bucket授权

协议跟随回源

功能介绍

开启该功能后，回源使用协议和客户端访问资源的协议保持一致，即如果客户端使用 HTTPS 方式请求资源，当节点上未缓存该资源时，会使用相同的 HTTPS 方式回源获取资源；同理类似 HTTP 协议的请求。

注意：

- 源站需要同时支持 80 端口和 443 端口，否则有可能会造成回源失败

配置说明

进入CDN域名管理列表页，选择相应域名进入配置页面，可在回源设置里，开启/关闭【协议跟随回源】功能。

回源HOST

功能介绍

自定义在CDN节点回源时所需访问的具体域名（如果您 一个IP源站 绑定了 多个域名/站点 的时候，就需设置回源Host 指定回到具体哪个域名，否则会回源失败）。

- **回源host** 为可选配置项，默认值为：
    - 如果源站是 IP类型，回源host默认加速域名。
    - 如果源站是 OSS源站类型，回源host默认是源站域名。
- 可选项分别是：加速域名、源站域名、自定义域名。

注意：目前不支持sni 回源。

配置引导

变更配置：CDN域名管理页—>选择域名进入配置页面—>回源设置，可修改回源host的配置。

源站和回源host的区别（**一个IP/主机** 是能绑定 **多个域名/站点**的，因此需要通过设置回源Host指定回源时回到哪个域名/站点）：

- 源站： 源站决定了回源时，请求到哪个IP
- 回源host：回源host决定回源请求访问到该IP上的哪个站点。（如果您 一个IP源站 绑定了 多个域名

/站点 的时候，就需设置 回源Host 指定回源到具体哪个域名，否则会回源失败）

例一
源站是域名源站为www.a.com 回源host设置为www.b.com
那么实际回源是请求到www.a.com 解析到的IP上对应的具体的站点：www.b.com

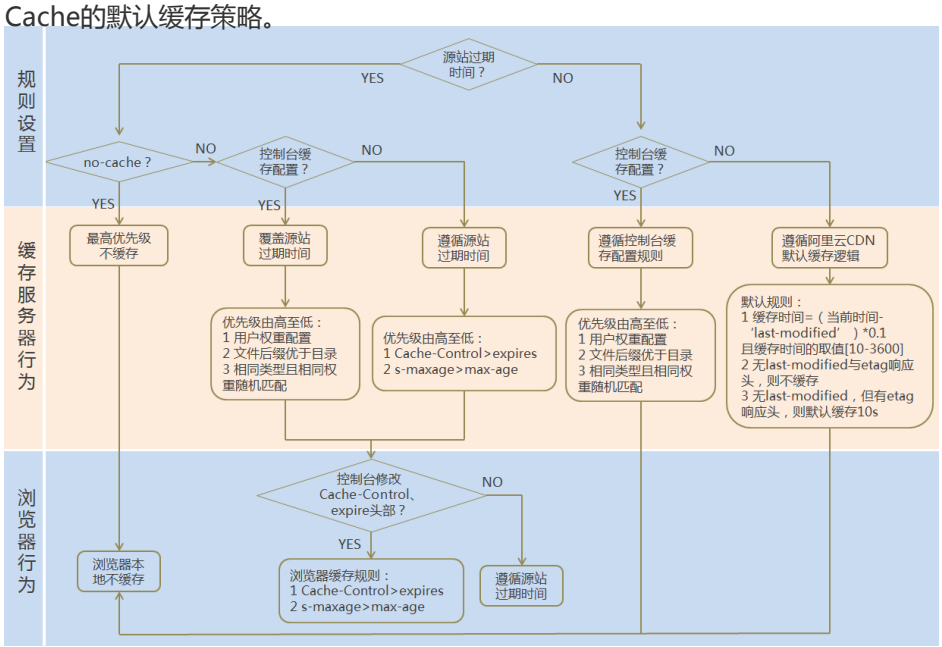
例二
源站是IP源站为1.1.1.1 回源host设置为www.b.com
那么实际回源的是1.1.1.1上对应的具体站点：www.b.com

节点缓存设置

缓存配置

功能介绍

- 该功能可以针对不同“目录路径”和“文件名后缀”的资源进行缓存服务器行为的设置，用户可自定义指定资源内容的缓存过期时间规则。
- 支持用户自定义缓存策略优先级。
- Cache的默认缓存策略。



注意：

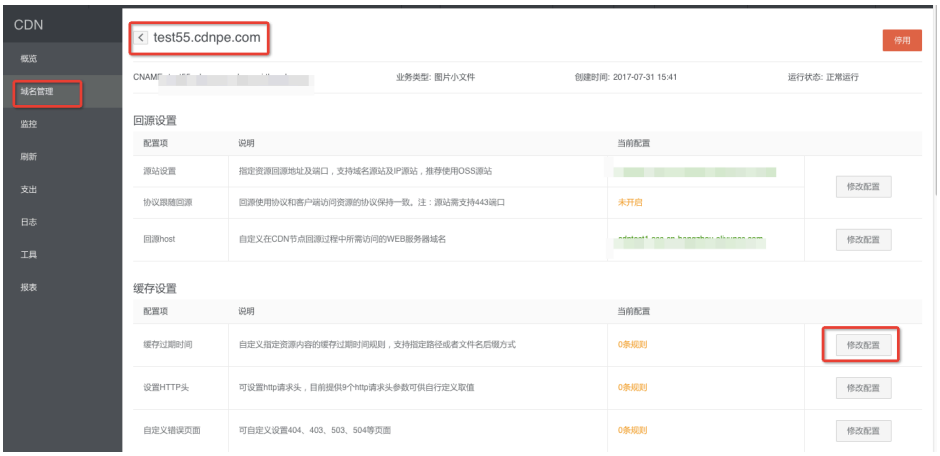
- 用于配置文件过期时间，在此配置的优先级会高于源站配置。如果源站未配置cache配置，支持按目录、文件后缀两种方式设置(支持设置完整路径缓存策略)。
- 了解详细CDN节点默认缓存策略。
- CDN的缓存是有可能由于热度较低被提前剔除出CDN节点的。

注意事项

1. 对于不经常更新的静态文件，建议将缓存时间设置为1个月以上（eg：图片类型，应用下载类型）；
2. 对于需要更新并且更新很频繁的静态文件，可以将缓存时间设置短些，视业务情况而定（eg：js,css等）；
3. 对于动态文件（eg：php | jsp | asp），建议设置缓存时间为0s，即不缓存；若动态文件例如php文件内容更新频率较低，推荐设置较短缓存时间；
4. 建议源站的内容不要使用同名更新，以版本号的方式方步，即采用img-v1.0.jpg、img-v2.1.jpg的命名方式。

配置引导

- CDN域名概览页—>选择域名进入管理页面—>【缓存配置】。



- 点击【修改配置】，可以管理缓存规则，添加、修改、删除。

- 点击【添加】，增加缓存规则，按目录或者按文件后缀。

- 举例：为加速域名 example.aliyun.com 设置三则缓存配置规则：
  - 缓存策略1：文件名后缀为jpg、png的所有资源 过期时间为1月，权重设置为90。
  - 缓存策略2：目录为/www/dir/aaa 过期时间为1小时，权重设置为70。
  - 缓存策略3：完整路径为/www/dir/aaa/example.php 过期时间为0s，权重设置为80。
  - 则这三个缓存策略的生效顺序是：策略1—>策略3—>策略2。

注：

- 权重可设置1-99数字越大，优先级越高，优先生效；
- 不推荐设置相同的权重，权重相同的两条缓存策略优先级随机。

自定义错误页面

功能介绍

客户可以自行定义状态码时返回的页面，优化用户体验。提供三种选项：默认页面、自定义页面

以返回码 404为例

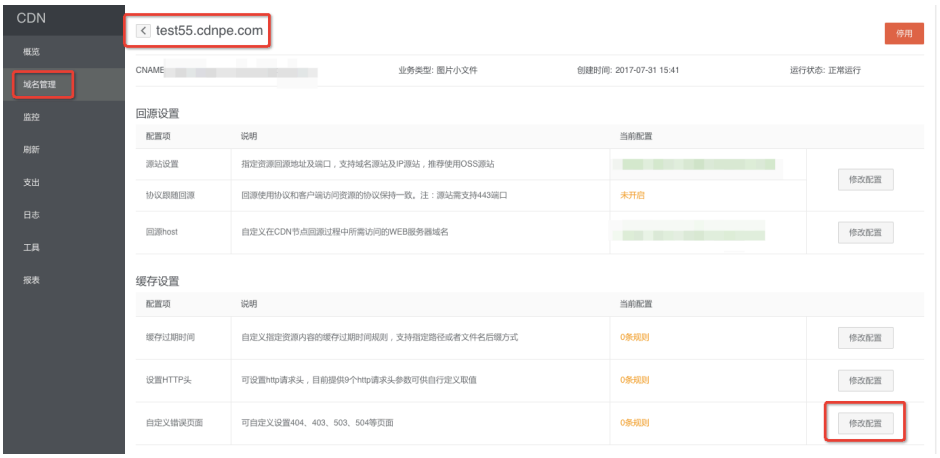
- 默认值：http 响应返回 404 时，服务器返回默认 404 Not Found页面
- 公益404，http 响应返回 404 时，将会跳转到实时更新的公益主题 404 页面，查看公益404页面
- 自定义404，http 响应返回 404 时，将会跳转到自行设计和编辑的 404 页面，需要自定义跳转页的完整URL地址

注意事项

- 公益 404 页面属于阿里云公益资源，不会造成用户的任何流量费用，完全免费
- 自定义页面属于个人资源，按照正常分发计费

配置引导

- CDN域名概览页—>选择域名进入域名配置页面—>设置【自定义错误页面】功能



- 点击【修改配置】，可以查看和管理当前自定义错误页面列表

- 点击【添加】，增加自定义返回码的页面内容

- 若选择“自定义 404”选项，将该页面资源如其他静态文件一样存储到源站域名下，并通过加速域名访问即可，只需填写完整的加速域名URL（包含http://）

  例如：加速域名为 exp.aliyun.com404页面为error404.html ，并将error404.html页面存储到源站中选择“自定义404” ，填写： http://exp.aliyun.com/error404.html 即可

设置HTTP头

功能介绍

- 可设置http响应头，目前提供9个http请求头参数可供自行定义取值，参数解释如下

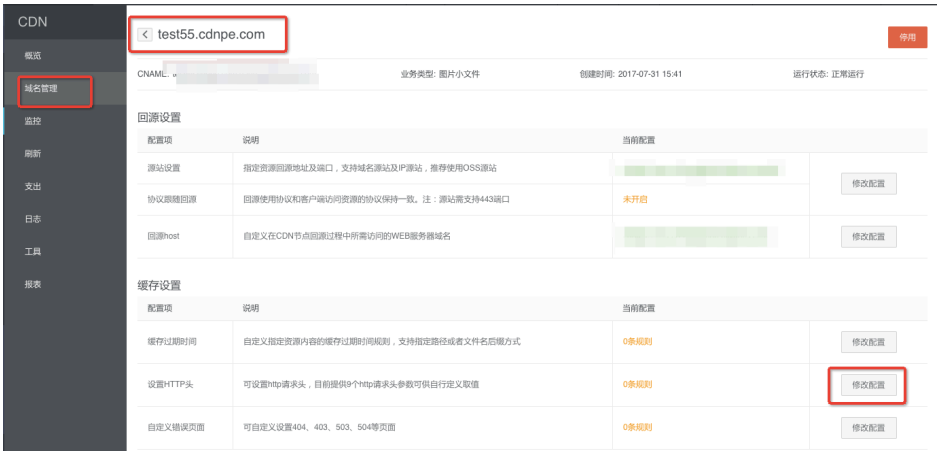
参数	解释
Content-Type	指定客户程序响应对象的内容类型
Cache-Control	指定客户程序请求和响应遵循的缓存机制
Content-Disposition	指定客户程序响应对象时激活文件下载设置默认的文件名
Content-Language	指定客户程序响应对象的语言
Expires	指定客户程序响应对象的过期时间
Access-Control-Allow-Origin	指定允许的跨域请求的来源
Access-Control-Allow-Methods	指定允许的跨域请求方法
Access-Control-Max-Age	指定客户程序对特定资源的预取请求返回结果的缓存时间
Access-Control-Expose-Headers	指定允许访问的自定义头信息

注意事项

- HTTP响应头的设置会影响该加速域名下所有资源的客户程序（例如浏览器）的响应行为，而不会影响缓存服务器的行为
- 目前仅支持这些http头参数取值设置，有其他HTTP头部设置需求，请提工单反馈
- Access-Control-Allow-Origin参数的取值，支持”*”(表示全部域名)或者完整域名例如：“www.aliyun.com”；目前不支持泛域名设置

配置引导

CDN域名概览页—>选择域名进入配置页面—>【设置HTTP头】



点击【修改配置】，可以管理当前http header的规则列表

点击【添加】，增加HTTP HEADER自定义设置

刷新缓存

URL 刷新

原理：通过提供文件URL的方式，强制CDN节点回源拉取最新的文件。

任务生效时间：5-10 分钟之内生效。

注意事项：

- 输入的 URL 必须带有 http://或者 https://
- 同一个 ID 每天最多只能预热刷新共 2000 个 URL。
- 提供批量刷新缓存的接口，详见 刷新缓存API。

目录刷新

原理：通过提供文件目录的方式，强制CDN节点回源拉取最新的文件。

任务生效时间：5-10 分钟之内生效。

注意事项：

- 一天最多提交 100 个刷新请求。
- 所输入内容，需以 http://或者 https://开始，以/结束。
- 提供批量刷新缓存的接口，详见 刷新缓存API。

URL 预热

原理：将指定的内容主动预热到CDN的L2节点上，用户首次访问即可直接命中缓存，降低源站压力。

任务生效时间：5-10 分钟之内生效。

注意事项：

- 输入的 URL 必须带有 http://或https://

- 同一个 ID 每天最多只能预热共 500 个 URL。
- 资源预热完成时间将取决于用户提交预热文件的数量、文件大小、源站带宽情况、网络状况等诸多因素。
- 提供批量预热资源的接口，详见 资源预热API。

进度查看

- 可在CDN控制台 **刷新—>操作记录**，查看资源刷新或预热的进度。
- 阿里云CDN提供查询进度的API： 查询刷新预热进度

高级设置

带宽封顶

带宽封顶

功能介绍

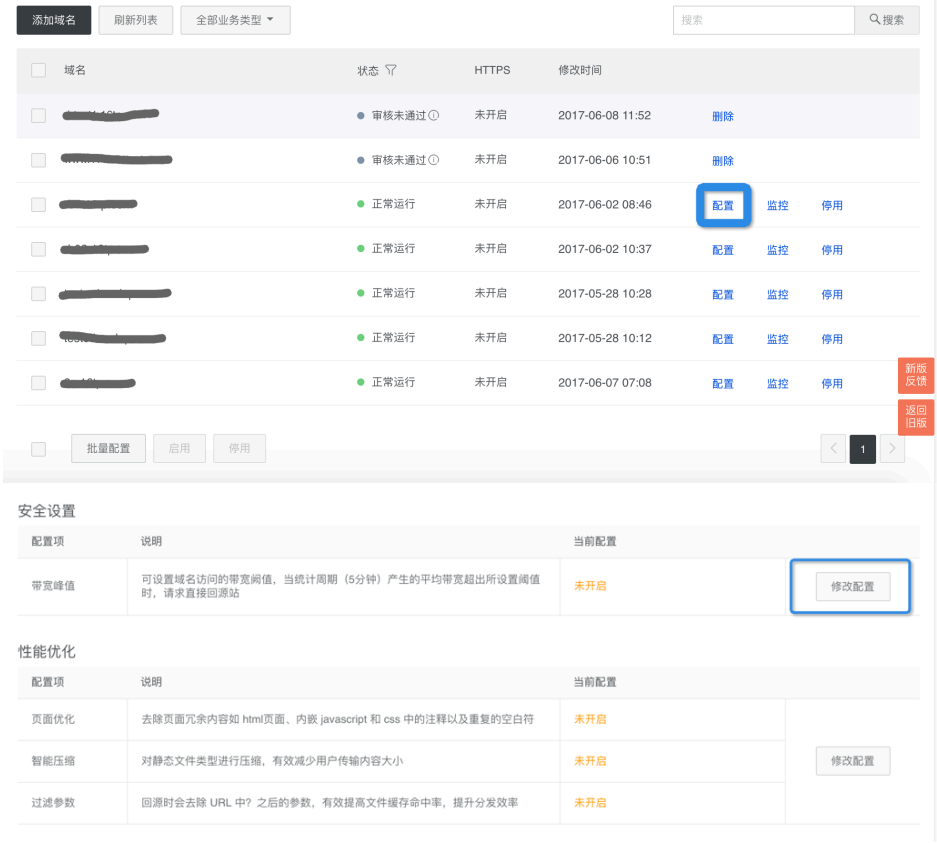
- 带宽封顶功能是指当统计周期（5分钟）产生的平均带宽超出所设置的带宽最大值时，为了保护您的域名安全，此时域名会自动下线，所有的请求会回到源站。此时CDN将停止加速服务，避免异常流量带来的非日常消费。域名下线后，你可以在控制台重新启动域名。

注意：带宽封顶的功能，泛域名暂不支持，设置后不会生效

- RAM子账号需云监控授权后使用，请授权AliyunCloudMonitorFullAccess策略组

如何开启带宽封顶功能

域名列表点击“配置”后，在选中域名配置页面找到“安全设置”，点击“修改配置”



开启带宽封顶功能，带宽单位支持Mbps, Gbps, Tbps（注，带宽进制为1000）

带宽封顶功能成功开启

您可以根据域名的实际使用情况，选择开启或者关闭带宽封顶功能

注意事项

- 开启带宽封顶功能后，您的业务会受到带宽封顶的限制而触发下线，为了不影响您的域名业务，建议您合理评估，谨慎设置带宽峰值

访问控制设置

防盗链

功能介绍

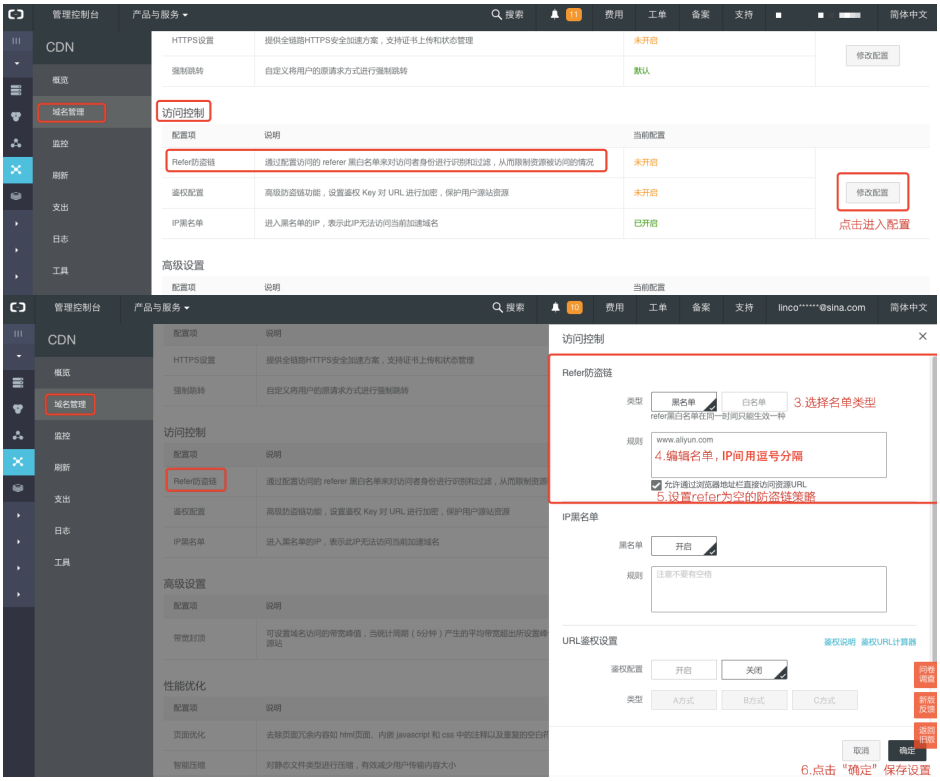
- 防盗链功能基于 HTTP 协议支持的 Referer 机制，通过 referer 跟踪来源，对来源进行识别和判断，用户可以通过配置访问的 referer 黑白名单来对访问者身份进行识别和过滤，从而限制 CDN 资源被访问的情况
- 目前防盗链功能支持黑名单或白名单机制，访客对资源发起请求后，请求到达 CDN 节点，CDN节点会根据用户预设的防盗链黑名单或白名单，对访客的身份进行过滤，符合规则可以顺利请求到资源；若不符合规则，该访客请求被禁止，返回403响应码。

注意事项

- 可选配置，默认不启用
- 开启功能，选择编辑refer黑名单或者白名单，黑白名单互斥，同一时间只支持一种方式
- 支持设置是否允许空 Referer 字段访问CDN资源。（即允许通过浏览器地址栏直接访问资源URL）
- 配置后会自动添加泛域名支持，例如填写a.com，最终配置生效的是*.a.com，所有子级域名都会生效

配置引导

CDN域名概览页—>进入域名管理页面—>选择需要设置的域名—>点击配置



鉴权配置

URL鉴权功能

概述

URL鉴权功能旨在保护用户站点的内容资源不被非法站点下载盗用。采用防盗链方法添加 referer 黑、白名单方式可以解决部分盗链问题，但是，由于 referer 内容可以伪造，referer 防盗链方式还不能很好的保护站点资源，因此采用URL鉴权方式保护用户源站资源更为安全有效。

原理

URL鉴权功能是通过阿里云CDN加速节点与客户资源站点配合实现的一种更为安全可靠的源站资源防盗方法。由CDN客户站点提供给用户加密 URL（包含权限验证信息），用户使用加密后的 URL 向加速节点发起请求，加速节点对加密 URL 中的权限信息进行验证以判断请求的合法性，对合法请求给予正常响应，拒绝非法请求，从而有效保护CDN客户站点资源。

URL鉴权方式

阿里云CDN 兼容并支持A、B、C三种鉴权方式，用户可以根据自己的业务情况，选择合适的鉴权方式，来实现对源站资源的有效保护

A鉴权方法

原理说明

用户访问加密 URL 构成

```
http://DomainName/Filename?auth_key=timestamp-rand-uid-md5hash
```

鉴权字段描述

- PrivateKey 字段用户可以自行设置
- 有效时间1800s是指，用户访问客户源服务器时间超过自定义失效时间(timestamp字段指定)的1800s后，该鉴权失效；例如用户设置访问时间2020-08-15 15:00:00，链接真正失效时间是2020-08-15 15:30:00

字段	描述
timestamp	失效时间，整形正数，固定长度10，1970年1月1日以来的秒数。用来控制失效时间，10位整数，有效时间1800s
rand	随机数，建议使用UUID (不能包含中划线”-“，如: 477b3bbc253f467b8def6711128c7bec格式)
uid	暂未使用（设置成0即可)
md5hash	通过md5算法计算出的验证串，数字和小写英文字母混合0-9a-z，固定长度32

CDN服务器拿到请求后，首先会判断请求中的 timestamp 是否小于当前时间，如果小于，则认为过期失效并返回HTTP 403错误。如果 timestamp 大于当前时间，则构造出一个同样的字符串(参考以下sstring构造方式)。然后使用MD5算法算出 HashValue ，再和请求中带来的 md5hash 进行比对。比对结果一致，则认为鉴权通过，返回文件。否则鉴权失败，返回HTTP 403错误。

- HashValue 是通过以下字符串计算出来的：

```
sstring = "URI-Timestamp-rand-uid-PrivateKey"（URI是用户的请求对象相对地址，不包含参数，如/Filename）
HashValue = md5sum(sstring)
```

示例说明

通过 req_auth 请求对象:

```
http:// cdn.example.com/video/standard/1K.html
```

密钥设为：aliyuncdnexp1234 *(由用户自行设置)*

3. 鉴权配置文件失效日期为：2015年10月10日00:00:00,计算出来的秒数为1444435200

则CDN服务器会构造一个用于计算Hashvalue的签名字符串：

```
/video/standard/1K.html-1444435200-0-0-aliyuncdnexp1234"
```

CDN服务器会根据该签名字符串计算HashValue:

```
HashValue = md5sum("/video/standard/1K.html-1444435200-0-0-aliyuncdnexp1234") =
80cd3862d699b7118eed99103f2a3a4f
```

则请求时url为：

```
http://cdn.example.com/video/standard/1K.html?auth_key=1444435200-0-0-
80cd3862d699b7118eed99103f2a3a4f
```

计算出来的HashValue与用户请求中带的 md5hash = 80cd3862d699b7118eed99103f2a3a4f 值一致，于是鉴权通过

B鉴权方式

原理说明

用户访问加密 URL 格式

- 用户访问的 URL 如下:

```
http://DomainName/timestamp/md5hash/FileName
```

加密URL的构造:域名后跟生成URL的时间（精确到分钟）(timestamp)再跟md5值(md5hash)，最后拼接回源服务器的真实路径(FileName)，URL有效时间为1800s

- 当鉴权通过时，实际回源的URL是:

```
http://DomainName/FileName
```

鉴权字段描述

- 注意：PrivateKey 由CDN客户自行设置
- 有效时间1800s是指，用户访问客户源服务器时间超过自定义失效时间(timestamp字段指定)的1800s后，该鉴权失效；例如用户设置访问时间2020-08-15 15:00:00，链接真正失效时间是2020-08-15 15:30:00

字段	描述
DomainName	CDN客户站点的域名
timestamp	资源失效时间，作为URL的一部分，同时作为计算md5hash 的一个因子，格式为：YYYYMMDDHHMM ,有效时间1800s
md5hash	以timestamp、FileName和预先设定好的PrivateKey 共同做MD5获得的字符串，即md5(PrivateKey + timestamp + FileName)
FileName	实际回源访问的URL (注意，鉴权时候FileName要以/开头)

示例说明

回源请求对象:

```
http://cdn.example.com/4/44/44c0909bcfc20a01afaf256ca99a8b8b.mp3
```

密钥设为：aliyuncdnexp1234 *(用户自行设置)*

3. 用户访问客户源服务器时间为 201508150800（格式为： YYYYMMDDHHMM ）

则CDN服务器会构造一个用于计算 md5hash 的签名字符串:

```
aliyuncdnexp1234201508150800/4/44/44c0909bcfc20a01afaf256ca99a8b8b.mp3
```

服务器会根据该签名字符串计算 md5hash :

```
md5hash = md5sum("aliyuncdnexp1234201508150800/4/44/44c0909bcfc20a01afaf256ca99a8b8b.mp3") = 9044548ef1527deadafa49a890a377f0
```

请求CDN时url:

```
http://cdn.example.com/201508150800/9044548ef1527deadafa49a890a377f0/4/44/44c0909bcfc20a01afaf256ca99a8b8b.mp3
```

计算出来的 md5hash 与用户请求中带的 md5hash = 9044548ef1527deadafa49a890a377f0 值一致，于是鉴权通过

C鉴权方式

原理说明

用户访问加密 URL 格式

格式1

http://DomainName/{<md5hash>/<timestamp>}/FileName

格式2http://DomainName/FileName{&KEY1=<md5hash>&KEY2=<timestamp>}

- 花括号中的内容表示在标准的URL基础上添加的加密信息
- <md5hash> 是验证信息经过 MD5 加密后的字符串;
- <timestamp> 是未加密的字符串，以明文表示。固定长度10，1970年1月1日以来的秒数，表示为十六进制
- 采用格式一进行URL加密，例如

```
http://cdn.example.com/a37fa50a5fb8f71214b1e7c95ec7a1bd/55CE8100/test.flv
```

<md5hash> 为 a37fa50a5fb8f71214b1e7c95ec7a1bd<timestamp> 为 55CE8100

鉴权字段描述

- <md5hash> 部分字段描述

字段	描述
PrivateKey	干扰串，不同客户采用不同的干扰串
FileName	实际回源访问的URL (注意，鉴权时候path要以/开头)
time	用户访问源服务器时间，取 UNIX 时间，以十六进制数字字符表示。

- PrivateKey 取值 aliyuncdnexp1234
- FileName 取值 /test.flv
- time 取值 55CE8100

因此, md5hash 值为

```
md5hash = md5sum(aliyuncdnexp1234/test.flv55CE8100) = a37fa50a5fb8f71214b1e7c95ec7a1bd
```

明文: timestamp = 55CE8100

- 这样生成加密 URL:

格式一：

```
http://cdn.example.com/a37fa50a5fb8f71214b1e7c95ec7a1bd/55CE8100/test.flv
```

格式二：

```
http://cdn.example.com/test.flv?KEY1=a37fa50a5fb8f71214b1e7c95ec7a1bd&KEY2=55CE8100
```

示例说明

用户使用加密的 URL 访问加速节点,CDN服务器会先把加密串 1 提取出来, 并得到原始的 URL 的 <FileName> 部分，用户访问时间，然后按照定义的业务逻辑进行验证：

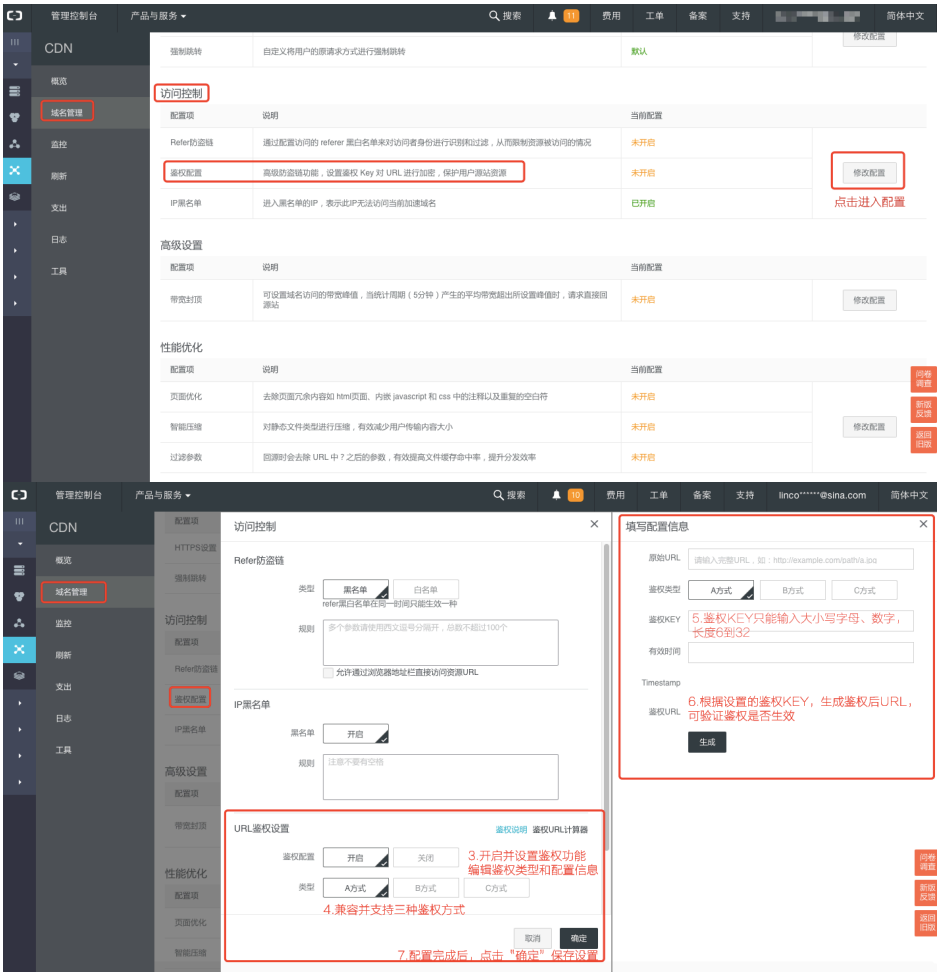
1. 使用原始的 URL 中的 <FileName> 部分,请求时间及 PrivateKey 进行 MD5 加密得到一个加密串2
2. 比较加密串 2 与加密串 1 是否一致，如果不一致则拒绝
3. 取加速节点服务器当前时间，并与从访问 URL 中所带的明文时间相减，判断是否超过设置的时限 t(时间域值 t 默认为 1800s)
4. 有效时间1800s是指，用户访问客户源服务器时间超过自定义时间的1800s后，该鉴权失效；例如用户设置访问时间2020-08-15 15:00:00，链接真正失效时间是2020-08-15 15:30:00
5. 时间差小于设置时限的为合法请求，CDN加速节点才会给予正常的响应，否则拒绝该请求，返回 http 403错误

URL鉴权代码示例

请查看**CDN周边工具**中 **鉴权代码示例** 文档

配置引导

CDN域名概览页—>进入域名管理页面—>选择需要设置的域名—>点击配置



IP黑名单

功能介绍

- 支持黑名单规则，添加了黑名单的IP，表示此IP无法访问当前加速域名

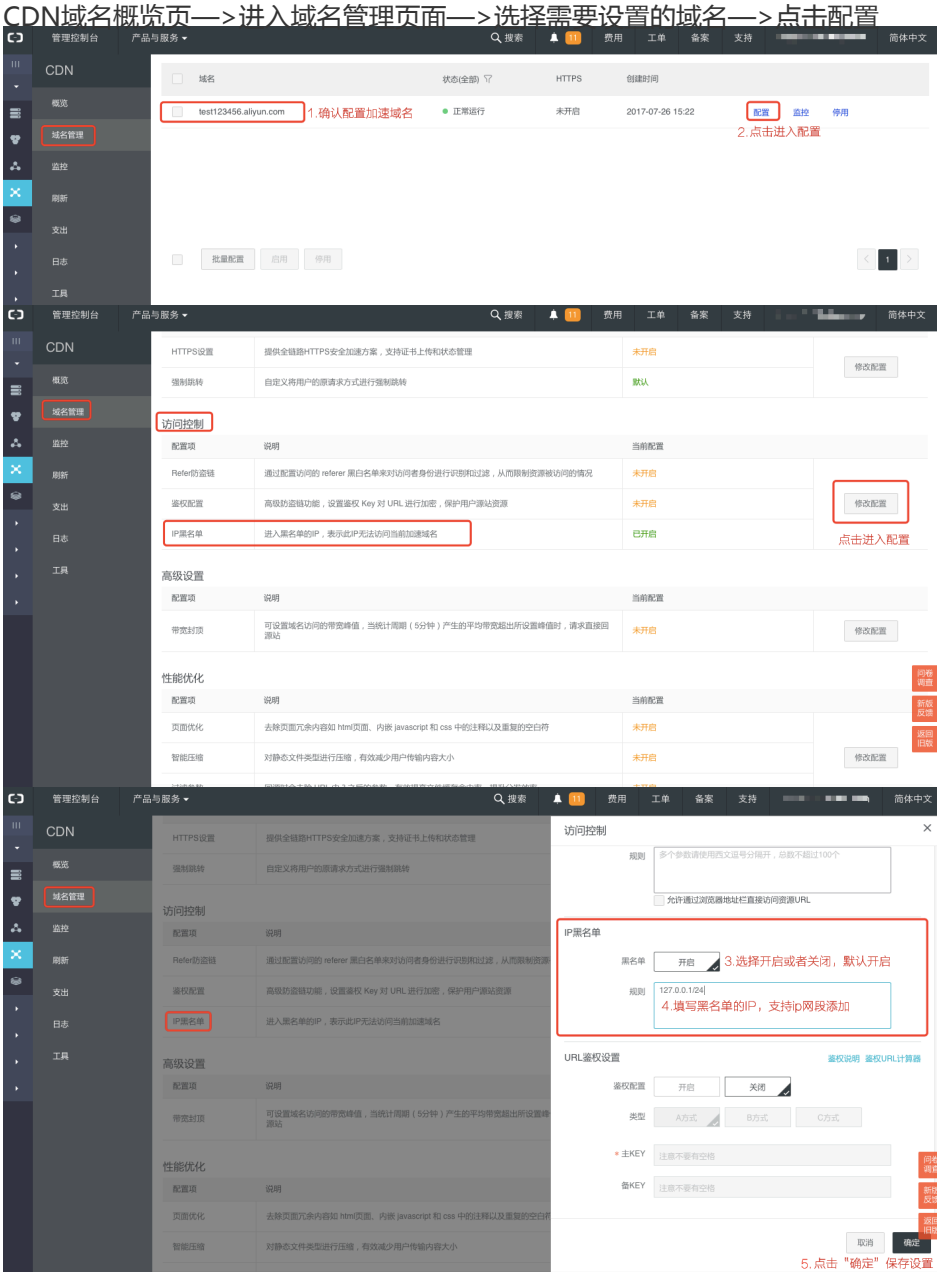
注意事项

- IP黑名单当前支持ip网段添加，例如127.0.0.1/24

  例如：127.0.0.1/24 24表示采用子网掩码中的前24位为有效位，即用32-24=8bit来表示主机号，该子网可以容纳2^8 - 2 = 254 台主机。故127.0.0.1/24 表示IP网段范围是：127.0.0.1~127.0.0.255

配置引导

CDN域名概览页—>进入域名管理页面—>选择需要设置的域名—>点击配置



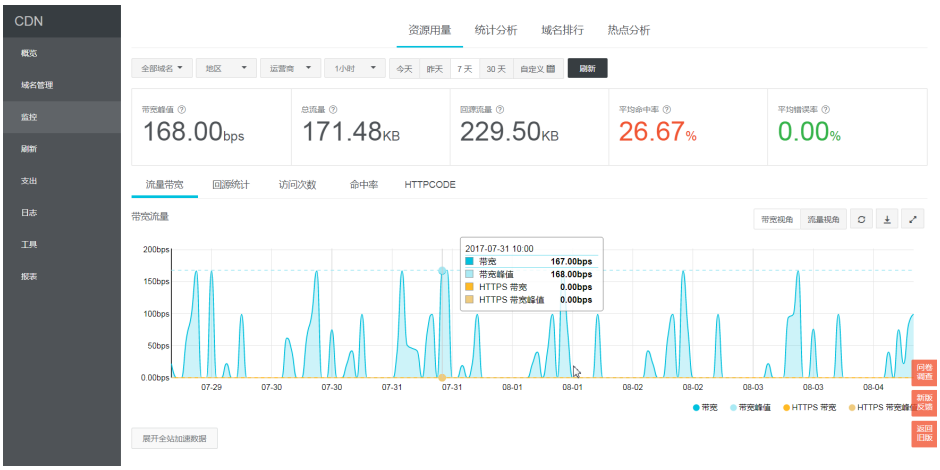
资源监控

监控页面功能说明：

- 资源监控包含四部分，资源用量、统计分析、域名排名、热点分析。

- 支持原始数据导出，如网络带宽、流量，域名按流量占比排名以及访客区域、运营商分布等详细数据。
- 资源监控部分的曲线图数据和计费数据有一定差别，如30天统计曲线取点粒度为14400s，计费数据粒度为300s，故曲线图会忽略掉其中的一些计量点作图，主要用作带宽趋势描述，带宽使用以精确粒度的计费数据为准。

注：原始数据采集粒度随时间段变化，日维度导出数据，粒度为300s；周维度导出数据，粒度为3600s；月维度导出数据，粒度为14400s。



项目	监控指标	可选时间
资源用量	网络流量、域名排行、回源流量、按日流量统计	今天、昨天、7 天内、30 天、自定义90天内
统计分析	PV、UV、用户区域分布、运营商占比	今天、昨天、7 天内、30 天、自定义90天内
域名排名	各个加速域名的访问排名	今天、昨天、7 天内、30 天、自定义90天内
热点分析	文件响应占比、URL 访问次数统计、页面引用 URL 占比	支持查看单日数据，自定义90天内

CDN子账户使用指南

本文档提供给CDN域名资源组管理需求的客户，通过子账户+资源组授权实现不同部门之间资源的隔离操作，接入流程如下

接入流程

步骤1 登录企业控制台

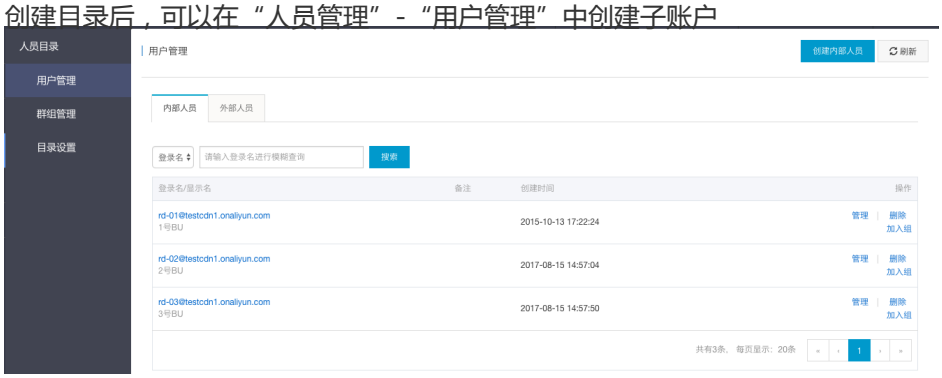
- 说明：资源组设置和子账号管理需要在企业控制台完成，设置好相应的资源组和权限后，子账号登录CDN控制台就会按照已定的规则进行有限的资源查看和操作，保证子账户间的操作和资源展示完全隔离。

使用主账号登录企业控制台：https://enterprise.console.aliyun.com/（附：企业控制台使用手册）

步骤2 创建子账户

- 进入“人员管理”模块，初次进入需要创建目录,一个用户必须且只能归属于某一个目录下。

创建目录后，可以在“人员管理”-“用户管理”中创建子账户



注，根据业务需求还可以创建群组，统一管理

步骤3 创建资源组 + 授权

- 进入“资源管理”模块，创建资源组，如下创建“1号BU资源组”

- 选择需要管理的资源组，完成该组内的“资源”、“成员”和基础信息“设置”

- 进入“资源管理”-“资源”实现加速域名分组设置，在筛选区选择产品CDN，勾选需要加入该资源组的加速域名，点击“转入”，完成资源组内加速域名设置

- 进入“资源管理”-“成员”完成子账户的授权，点击“新增成员”，可以选择需要管理本资源组的子账户，并完成策略授权，附：授权模板说明

步骤4 使用子账号登录CDN控制台

- 登录地址: http://signin.aliyun.com/<自定义域>.onaliyun.com/login.htm

子账户登录后，可以选择展示当前子账户拥有权限的资源组，根据资源组罗列加速域名

子账户支持 域名管理、监控、刷新和日志下载，其他操作同主账号完全一致，请参考 快速入门

附录

当前RAM模板策略

1.CDN管理授权：支持增删查改

```
{
"Version": "1",
"Statement": [
{
"Action": "cdn:*",
"Resource": "*",
"Effect": "Allow"
}
]
}
```

2.CDN只读权限

```
{
"Version": "1",
"Statement": [
{
"Action": "cdn:Describe*",
"Resource": "*",
"Effect": "Allow"
}
]
}
```

全站加速设置

动态协议跟随回源

功能介绍

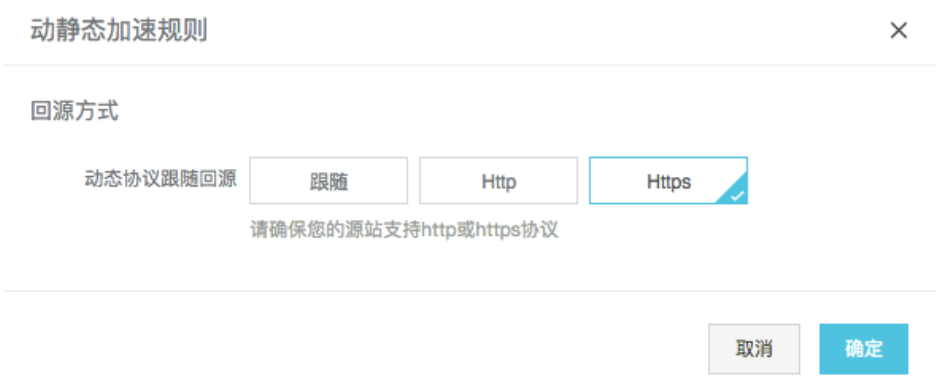
动态资源回源时使用协议和客户端访问资源的协议保持一致。例如客户端以HTTP协议请求动态资源，CDN节点也会以HTTP协议回源获取资源，同理如果客户端以HTTPS协议请求动态资源，CDN节点会以HTTPS协议回

源获取资源。

配置引导

【域名管理】—>选择域名进入域名【配置】页面—>【动静态加速规则设置】：

选择【动态协议跟随回源】进行选择：



注意：“动态协议跟随回源”是针对动态资源的请求的配置，而源站设置里的“协议跟随回源”则是针对静态资源的请求的配置，二者有区别。

特殊header头设置

功能介绍

根据Header中的Cache-Control字段，选择是否动态加速。Header相应头的Cache-Control内容符合配置中任一规则就**强制开启动态加速**，不再检查其余配置。

配置引导

【域名管理】—>选择域名进入域名【配置】页面—>【动静态加速规则设置】：

选择【特殊Header头设置】，设置要强制开启动态加速的Cache-Control规则：

例如：设置了规则为 no-cache，则所有响应头中的Cache-Control中带no-cache的资源都强制开启动态加速，不缓存在边缘节点上。

静态文件类型

功能介绍

全站加速默认为纯动态加速，即所有资源请求都使用动态加速，通过最优路由回源获取资源。因此静态资源也不会被边缘节点缓存。可通过配置指定静态文件的类型，以智能区分动静态资源，达到静态资源使用边缘缓存，动态资源用动态加速的最优方案。

配置引导

【域名管理】—>选择域名进入域名【配置】页面—>【动静态加速规则设置】：

选择【静态文件类型】进行配置：

勾选静态资源的文件类型，选中的资源类型将使用边缘缓存，而不用每次请求都回源获取资源。

静态路径设置

功能介绍

全站加速默认为纯动态加速，即所有资源请求都使用动态加速，通过最优路由回源获取资源。因此静态资源也不会被边缘节点缓存。可通过配置指定静态文件的路径，以区分动静态资源，达到静态资源使用边缘缓存，动态资源用动态加速的最优方案。

配置引导

【域名管理】—>选择域名进入域名【配置】页面—>【动静态加速规则设置】：

选择【静态路径设置】，指定静态资源的路径：

静态路径的资源将使用边缘节点缓存，供用户就近获取，达到更好的加速效果。

静态URI设置

功能介绍

全站加速默认为纯动态加速，即所有资源请求都使用动态加速，通过最优路由回源获取资源。因此静态资源也不会被边缘节点缓存。可通过配置指定静态文件的URI，以区分动静态资源，达到静态资源使用边缘缓存，动态资源用动态加速的最优方案。

配置引导

【域名管理】—>选择域名进入域名【配置】页面—>【动静态加速规则设置】：

选择【静态URI设置】，指定静态URI：

静态URI的资源将使用静态资源加速，缓存在边缘节点上，供用户就近获取。

视频相关配置

Notify_URL设置

功能介绍

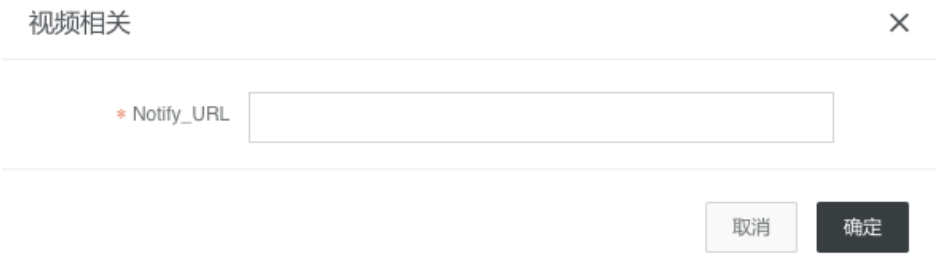
- 流状态实时信息回调，可以及时通知用户推流或断流操作结果。

注意事项

- 原理：通过 HTTP 接口向用户服务器发送GET请求，将视频流推送成功，断流成功的状态实时反馈给用户，用户服务器通过 200 响应返回接口返回结果。
- URL无需标识，只需可正常访问，URL 的应答有要求如下：
- 如果访问超时，会重试这个 URL，目前超时时间是 5s，重试次数是 5 次，重试间隔为 1s。

配置引导

- 支持在控制台配置，为可选配置



- 举例如下

```
http://1.1.1.1/pub?action=publish&app=xc.cdnpe.com&appname=hello&id=world&ip=42.120.74.183&node=cdnvideocenter010207116011.cm3
```

参数	取值说明
time	unix 时间戳
usrargs	用户推流的参数
action	publish表示推流，publish_done表示断流
app	默认为自定义的推流域名，如果未绑定推流域名即为播放域名
appname	应用名称
id	流名称
node	cdn接受流的节点或者机器名
ip	推流的客户端ip

拖拽播放

功能介绍

- 拖拽播放是指在视频点播场景中，发生拖拽播放进度时，客户端会向server端发送类似 http://www.aliyun.com/test.flv?start=10 ，这样的URL请求，然后server端会向客户端响应从第10字节的前一个关键帧（如果start=10不是关键帧所在位置）的数据内容。
- 开启该功能，CDN节点则可以支持此项配置，可以在响应请求的时候直接向client响应从第10字节的

前一个关键帧（如果start=10不是关键帧所在位置）（FLV格式）或第10s（MP4格式）开始的内容。

注意事项

- 需要源站支持range请求,即对于http请求头中包含 Range 字段,源站能够响应正确的206文件分片
- 目前支持文件格式有：MP4和FLV
- 目前对于flv只支持音频aac并且视频是avc编码格式，其余编码格式不支持拖拽。

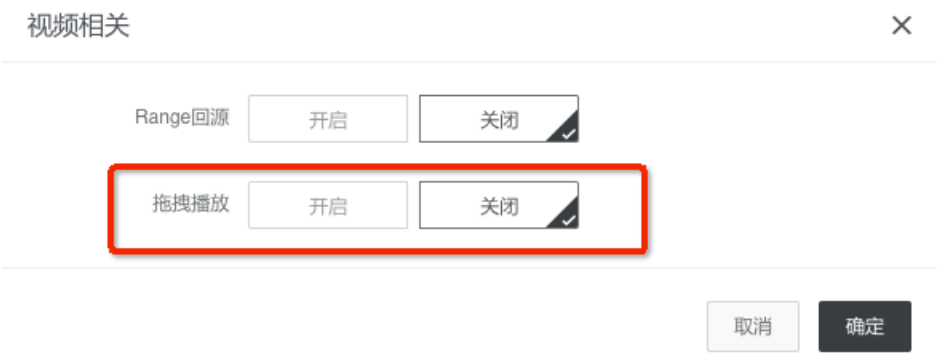
文件类型	meta信息	start参数	举例
MP4	源站视频的meta信息必须在文件头部，不支持meta信息在尾部的视频	start参数表示的是时间，单位是s，支持小数以表示ms（如start=1.01，表示开始时间是1.01s），CDN会定位到start所表示时间的前一个关键帧（如果当前start不是关键帧）	请求http://domain/video.mp4?start=10就是从第10秒开始播放视频
FLV	源站视频必须带有meta信息	start参数表示字节，CDN会自动定位到start参数所表示的字节的前一个关键帧（如果start当前不是关键帧）	对于http://domain/video.flv,请求http://domain/video.flv?start=10就是从第10字节的前一个关键帧（如果start=10不是关键帧所在位置）开始播放视频

配置引导

可选配置项，默认不开启

变更配置

CDN域名管理页面—>点击配置 —>视频相关 开启/关闭【拖拽播放】功能



range回源

功能介绍

- Range回源是指客户端通知源站服务器只返回部分内容，以及部分内容的范围。这对于较大文件的分发加速有很大帮助，开启Range回源功能，可以减少回源流量消耗，并且提升资源响应时间。
- 需要源站支持range请求,即对于http请求头中包含 Range 字段,源站能够响应正确的206文件分片

开启【Range回源】，则该参数可以请求回源站。此时源站需要依据 Range 的参数，响应文件的字节范围。同时CDN节点也会向客户端响应相应字节范围的内容。

> 例如：客户端向CDN请求中含有range:0-100，则源站端收到的请求中也会含有range：0-100这个参数。并且源站响应给CDN节点，然后CDN节点响应给客户端的就是范围是0-100的一共101个字节内容

关闭【Range回源】，CDN上层节点会向源站请求全部的文件，并且由于客户端会在收到Range定义的字节后自动断开http链接，请求的文件没有缓存到CDN节点上。最终导致缓存的命中率较低，并且回源流量较大。

> 例如：客户端向CDN请求中含有range:0-100，则server端收到的请求中没有range这个参数。源站响应给CDN节点完整文件，但是CDN节点响应给客户端的就是101个字节，但是由于连接断开了，会导致该文件没有缓存到CDN节点上。

注意事项

- 需要源站支持range请求,即对于http请求头中包含 Range 字段,源站能够响应正确的206文件分片

配置引导

- 可选配置项，默认不开启

- 变更配置

CDN域名管理页面—>点击配置—>选择 开启/关闭【Range回源】功能

日志管理

日志下载

- 日志文件延迟4小时，可以在日志管理模块查询到4小时之前的日志文件
- 日志文件按小时粒度分割
- 支持 **1月** 的日志数据下载
- 日志命名规则：加速域名_年_月_日_时间开始_时间结束
- 日志字段格式说明

日志内容：

```
[9/Jun/2015:01:58:09 +0800] 188.165.15.75 - 1542 "-" "GET http://www.aliyun.com/index.html" 200 191 2830 MISS "Mozilla/5.0 (compatible; AhrefsBot/5.0; +http://ahrefs.com/robot/)" "text/html"
```

字段含义：

字段	参数
时间	[9/Jun/2015:01:58:09 +0800]
访问ip	188.165.15.75
代理ip	-
responsetime(单位 ms)	1542
referer	-
method	GET
访问url	http://www.aliyun.com/index.html
httpcode	200
requestsize(单位 byte)	191
responsesize(单位 byte)	2830
cache命中状态	MISS
UA头	Mozilla/5.0 (compatible; AhrefsBot/5.0; +http://ahrefs.com/robot/)
文件类型	text/html

控制台位置： 

增值服务

HTTPS安全加速

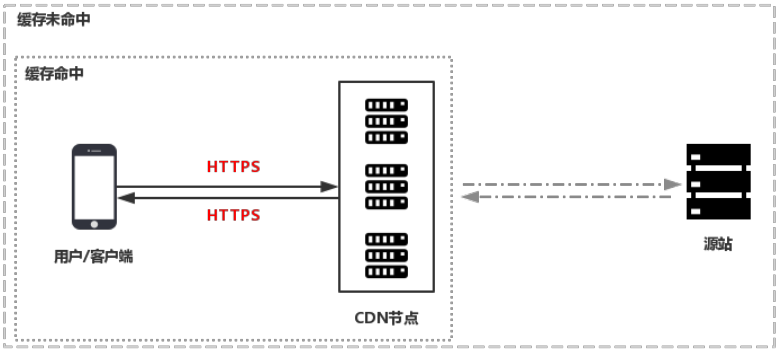
HTTPS安全加速设置

功能介绍

- HTTPS是以安全为目标的HTTP通道，简单讲是HTTP的安全版。即将HTTP用SSL/TLS协议进行封装，HTTPS的安全基础是SSL/TLS。
- HTTPS加速优势：
    - 传输过程中对用户的关键信息进行加密，防止类似Session ID或者Cookie内容被攻击者捕获造成的敏感信息泄露等安全隐患；
    - 传输过程中对数据进行完整性校验，防止DNS或内容遭第三方劫持、篡改等中间人攻击（MITM）隐患，了解更多。使用HTTPS防止流量劫持
- 阿里云CDN 提供HTTPS安全加速方案，仅需开启HTTPS后 **上传证书 和 私钥**，并支持对证书进行查看、停用、启用、编辑操作。用户自定义上传的证书仅支持PEM格式的证书，具体请看 证书格式说明及转化方法。
- 您可以在 阿里云云盾 快速申请**免费的证书** 或 购买高级证书。
- 证书配置正确及开启状态，同时支持HTTP访问和HTTPS访问；证书不匹配或者停用证书，仅支持HTTP访问。
- 注意：目前 **不支持sni 回源。**

功能示意图

在阿里云CDN控制台开启的HTTPS，将实现 **用户 和 阿里云CDN节点** 之间请求的HTTPS加密。而CDN节点返回源站获取资源的请求仍按您源站配置的方式进行，建议您源站也配置并开启HTTPS，实现全链路的HTTPS加密：



注意事项

配置相关

1. 支持开启“HTTPS安全加速”功能的业务类型为：
    - 图片小文件加速
    - 大文件下载加速
    - 视音频点播加速
    - 直播流媒体加速
    - 暂不支持移动加速业务类型
2. 支持泛域名HTTPS服务。
3. 支持该功能的“停用”和“启用”：
    - 启用：支持修改证书，默认兼容用户的HTTP和HTTPS请求，支持“强制跳转”设置。
    - 停用：不支持HTTPS请求且将不再保留证书/私钥信息，再次开启证书，需要重新上传证书/私钥。
4. 允许用户查看证书，但是只支持查看证书，由于私钥信息敏感不支持私钥查看，请妥善保管证书相关信息。
5. 支持修改编辑证书，但注意生效时间大约为10分钟，请慎重操作。

计费相关

- HTTPS安全加速属于增值服务，开启后将产生HTTPS请求数计费，当前计费标准详见 HTTPS计费详情。注意：HTTPS根据请求数单独计费，费用不包含在CDN流量包内，请确保账户余额充足再开通HTTPS服务，以免HTTPS服务导致欠费影响CDN服务。

附：如何查看HTTPS请求数使用情况。

证书相关

1. 开启“HTTPS安全加速”功能的加速域名，须要上传证书，包含证书/私钥，均为 PEM 格式(注：CDN采用的Tengine服务是基于Nginx的，因此只支持Nginx能读取的证书，即PEM格式)。具体请看 证书格式说明及转化方法。
2. 只支持带SNI信息的SSL/TLS握手。
3. 用户上传的证书和私钥要匹配，否则会校验出错。
4. 更新证书的生效时间约为10分钟。
5. 不支持带密码的私钥。

配置引导

步骤1 购买证书

开启HTTPS安全加速，需要您具备匹配加速域名的证书，可以在 阿里云云盾 快速申请**免费的证书** 或 购买高级证书。

步骤2 加速域名配置

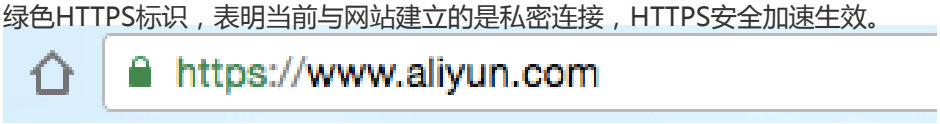
CDN域名列表页—>选择域名进入配置页面—>HTTPS设置—>修改配置。

点击修改配置，可以进行相应设置：

1. 确认当前域名“HTTPS设置”是否开启，点击“修改配置”按钮进入设置界面并“开启”，注：HTTPS安全加速属于增值服务，开启后将产生HTTPS请求数计费，了解**计费详情**。
2. 选择证书：
   - 可在“**阿里云盾证书服务**”**快速申请免费证书** 或 购买高级证书， 云盾的证书，可以通过证书名称直接选择适配该加速域名；
   - 若证书列表中无当前适配的证书可以选择自定义上传，需要设置证书名称后上传证书内容和私钥，该证书将会在“云盾证书服务”中保存，可以在“我的证书”部分查看。
3. 仅支持 PEM 的证书格式。具体请看 **证书格式说明及转化方法**。
4. 支持设置“强制跳转”：自定义将用户的原请求方式进行强制跳转：
   - 例如开启“强制HTTPS跳转”后，用户发起了一个HTTP请求，服务端返回302重定向响应，原来的HTTP请求强制重定向为HTTPS请求。
   - 默认：兼容用户的HTTP和HTTPS请求。
   - 强制HTTPS跳转：用户的请求将强制重定向为HTTPS请求。
   - 强制HTTP跳转：用户的请求将强制重定向为HTTP请求。

步骤3 验证证书是否生效

设置完成待证书生效后（设置HTTPS证书后约1小时后生效），使用HTTPS方式访问资源，如果浏览器中出现绿色HTTPS标识，表明当前与网站建立的是私密连接，HTTPS安全加速生效。

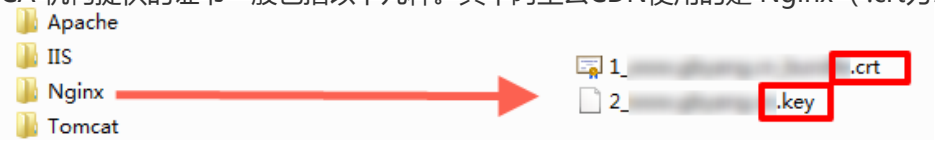


证书格式说明

在您 开启HTTPS 服务之前，需要配置证书。您可以直接选择在 阿里云盾 托管或购买的证书，或自行上传自定义证书。自定义上传只支持PEM格式证书、证书及私钥格式及其他格式转PEM格式方法。

证书格式要求

CA 机构提供的证书一般包括以下几种。其中阿里云CDN使用的是 Nginx （ .crt为证书，.key为私钥 ）：



如果证书是通过 root CA机构颁发，则您的证书为唯一的一份。

如果证书是通过中级CA机构颁发的证书，则您的证书文件包含多份证书，需要手工将服务器证书与中间证书拼接后，一起上传。

> **拼接规则为：**服务器证书放第一份，中间证书放第二份，中间不要有空行。一般情况下，机构在颁发证书的时候会有对应说明， 请注意规则说明。

示例

请确认格式正确后上传。

Root CA机构颁发的证书

证书格式为linux环境下 PEM 格式为：

```
-----BEGIN CERTIFICATE-----
MIIE+TCCA+GgAwIBAgIQU306HIX4KsioTW1s2A2krTANBgkqhkiG9w0BAQUFADCB
tTELMAkGA1UEBhMCVVMxFzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMR8wHQYDVQQL
ExZWZXJpU2lnbiBUcnVzdCBOZXR3b3JrMTswOQYDVQQLEzJUZXJtcyBvZiB1c2Ug
YXQgaHR0cHM6Ly93d3cudmVyaXNpZ24uY29tL3JwYSoAYykwOTEvMC0GA1UEAxMm
VmVyaVNpZ24gQ2xhc3MgMyBTZWN1cmUgU2VydmVyIENBIC0gRzIwHhcNMTAxMDA4
MDAwMDAwWhcNMTMxMDA3MjM1OTU5WjBqMQswCQYDVQQGEwJVUzETMBEGA1UECBMK
V2FzaGluZ3RvbjEQMA4GA1UEBxQHU2VhdHRsZTEYMBYGA1UEChQPQW1hem9uLmNv
bSBJbmMuMRowGAYDVQQDFBFpYW0uYW1hem9uYXdzLmNvbTCBnzANBgkqhkiG9w0B
AQEFAAOBjQAwgYkCgYEA3Xb0EGea2dB8QGEUwLcEpwvGawEkUdLZmGL1rQJZdeeN
3vaF+ZTm8Qw5Adk2Gr/RwYXtpx04xvQXmNm+9YmksHmCZdruCrW1eN/P9wBfqMMZ
X964CjVov3NrF5AuxU8jgtw0yu//C3hWnOuIVGdg76626ggOoJSaj48R2n0MnVcC
AwEAAaOCAdEwggHNMAkGA1UdEwQCMAAwCwYDVR0PBAQDAgWgMEUGA1UdHwQ+MDww
OqA4oDaGNGh0dHA6Ly9TVlJTZWN1cmUtRzItY3JsLnZlcmlzaWduLmNvbS9TVlJT
ZWN1cmVHMi5jcmwwRAYDVR0gBD0wOzA5BgtghkgBhvhFAQcXAzAqMCgGCCsGAQUF
BwIBFhxodHRwczovL3d3dy52ZXJpc2lnbi5jb20vcnBhMB0GA1UdJQQWMBQGCCsG
AQUFBwMBBggrBgEFBQcDAjAfBgNVHSMEGDAWgBSl7wsRzsBBA6NKZZBIshzgVy19
RzB2BggrBgEFBQcBAQRqMGgwJAYIKwYBBQUHMAGGGGh0dHA6Ly9vY3NwLnZlcmlz
aWduLmNvbTBABggrBgEFBQcwAoY0aHR0cDovL1NWUlNlY3VyZS1HMi1haWEudmVy
aXNpZ24uY29tL1NWUlNlY3VyZUcyLmNlcjBuBggrBgEFBQcBDARiMGChXqBcMFow
WDBWFglpbWFnZS9naWYwITAfMAcGBSsOAwIaBBRLa7kolgYMu9BSOJsprEsHiyEF
GDAmFiRodHRwOi8vbG9nby52ZXJpc2lnbi5jb20vdnNsb2dvMS5naWYwDQYJKoZI
hvcNAQEFBQADggEBALpFBXeG782QsTtGwEE9zBcVCuKjrsl3dWK1dFiq3OP4y/Bi
ZBYEywBt8zNuYFUE25Ub/zmvmpe7p0G76tmQ8bRp/4qkJoiSesHJvFgJ1mksr3IQ
3gaE1aN2BSUIHxGLn9N4F09hYwwbeEZaCxfgBiLdEIodNwzcvGJ+2LlDWGJOGrNI
NM856xjqhJCPxYzk9buuCl1B4Kzu0CTbexz/iEgYV+DiuTxcfA4uhwMDSe0nynbn
1qiwRk450mCOnqH4ly4P4lXo02t4A/DI1I8ZNct/Qfl69a2Lf6vc9rF7BELT0e5Y
R7CKx7fc5xRaeQdyGj/dJevm9BF/mSdnclS5vas=
-----END CERTIFICATE-----
```

证书规则为：

- 请将开头-----BEGIN CERTIFICATE-----和结尾 -----END CERTIFICATE-----一并上传；
- 每行64字符，最后一行不超过64字符。

中级机构颁发的证书链：

-----BEGIN CERTIFICATE-----

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

-----END CERTIFICATE-----

证书链规则：

- 证书之间不能有空行；
- 每一份证书遵守第一点关于证书的格式说明；

RSA私钥格式要求

rsa私钥规则：

本地生成私钥：openssl genrsa -out privateKey.pem 2048 其中privateKey.pem为您的私钥文件。

-----BEGIN RSA PRIVATE KEY-----开头, -----END RSA PRIVATE KEY----- 结尾；请将这些内容一并上传。

每行64字符，最后一行长度可以不足64字符。

如果您并未按照上述方案生成私钥，得到如-----BEGIN PRIVATE KEY-----、 -----END PRIVATE KEY----- 这种样式的私钥，您可以按照如下方式转换：

```
openssl rsa -in old_server_key.pem -out new_server_key.pem
```

然后将new_server_key.pem的内容与证书一起上传。

证书格式转换方式

CDN HTTPS安全加速只支持 PEM 格式的证书，其他格式的证书需要转换成 PEM 格式，建议通过openssl 工具进行转换。下面是几种比较流行的证书格式转换为 PEM 格式的方法。

DER 转换为 PEM

DER格式一般出现在java平台中

证书转化：

```
openssl x509 -inform der -in certificate.cer -out certificate.pem
```

私钥转化：

```
openssl rsa -inform DER -outform pem -in privatekey.der -out privatekey.pem
```

P7B 转换为 PEM

P7B格式一般出现在windows server和tomcat中

- 证书转化：

```
openssl pkcs7 -print_certs -in incertificat.p7b -out outcertificate.cer
```

获取outcertificat.cer里面-----BEGIN CERTIFICATE-----， -----END CERTIFICATE-----的内容作为证书上传。

- 私钥转化：P7B证书无私钥，因此 只需在CDN控制台只需填写证书部分，私钥无需填写。

PFX 转换为 PEM

PFX格式一般出现在windows server中。

证书转化：

```
openssl pkcs12 -in certname.pfx -nokeys -out cert.pem
```

私钥转化：

```
openssl pkcs12 -in certname.pfx -nocerts -out key.pem -nodes
```

其它证书相关

- 您可以停用、启用和修改证书。停用证书后，系统将不再保留证书信息。再次开启证书时，需要重新上传证书或私钥。请参考HTTPS安全加速设置教程。
- 只支持带SNI信息的SSL/TLS“握手”。

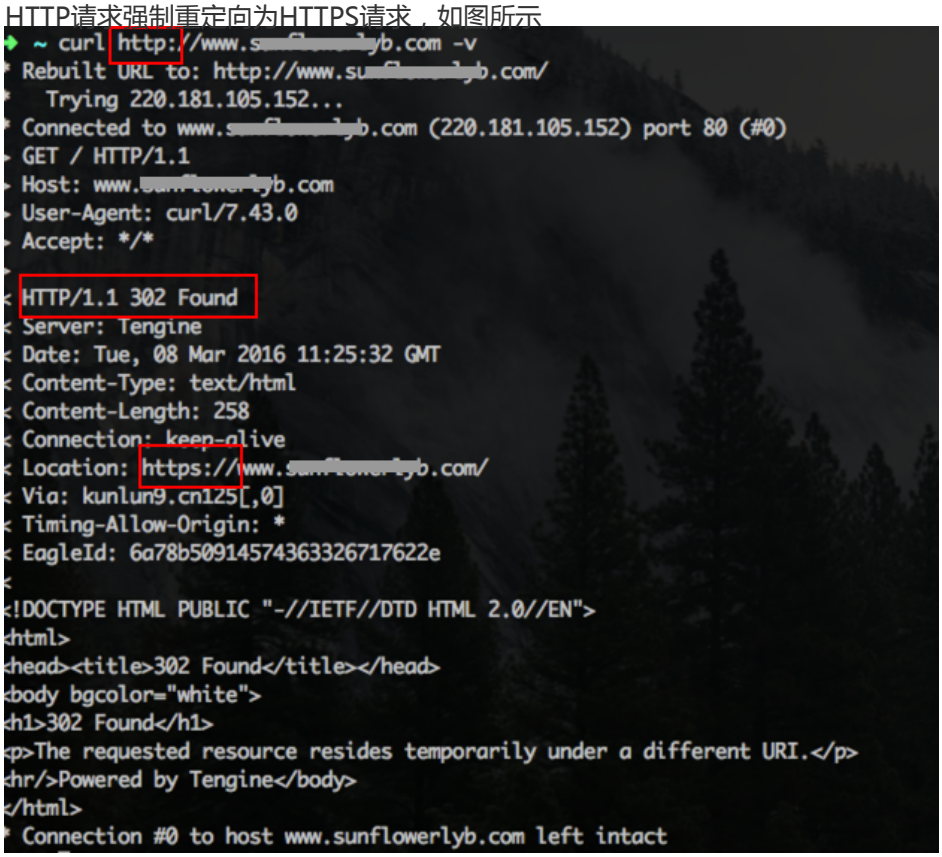
- 请确保上传的证书和私钥匹配。
- 更新证书的生效时间为10分钟。
- 不支持带密码的私钥。

其他证书相关的常见问题，请见更多证书问题：

强制跳转

功能介绍

- 加速域名开启“HTTPS安全加速”的前提下，支持自定义设置，将用户的原请求方式进行强制跳转
- 例如开启“强制HTTPS跳转”后，用户发起了一个HTTP请求，服务端返回302重定向响应，原来的HTTP请求强制重定向为HTTPS请求，如图所示



注意事项

- 仅支持启用“HTTPS安全加速”功能后设置，同时支持 HTTP 和 HTTPS 方式的请求。
- 默认为不强制跳转。

配置引导

- 【强制跳转】为可选配置项，默认设置同时支持 HTTP 和 HTTPS 方式的请求

可选项分别是：默认、强制HTTPS跳转、强制HTTP跳转

- 强制HTTPS跳转：用户的请求将强制重定向为HTTPS请求
- 强制HTTP跳转：用户的请求将强制重定向为HTTP请求

变更配置：

CDN域名管理页—>选择域名进入配置页面—>HTTPS设置—>强制跳转—>修改配置， 在设置页底部可设置强制跳转类型

HTTP/2

什么是HTTP/2？

HTTP/2是最新的HTTP协议，已于2015年5月份正式发布， Chrome、 IE11、Safari以及Firefox 等主流浏览器已经支持 HTTP/2协议

HTTP/2优化了性能而且兼容了HTTP/1.1的语义，其几大特性与SPDY差不多，与HTTP/1.1有巨大区别，比如它不是文本协议而是二进制协议，而且HTTP头部采用HPACK进行压缩，支持多路复用、服务器推送等等。

HTTP/2的优势

- 采用二进制协议

头部压缩：HTTP/2消息头采用HPACK格式进行压缩传输，并对消息头建立索引表，相同的消息头只发送索引号，从而提高效率和速度

多路复用：在HTTP/2中，不用按照次序一一对应，而且并发的多个请求或者响应中任何一个请求阻塞了不会影响其他的请求或者响应，这样就避免了“队头堵塞”

服务器推送：在HTTP/2中服务器未经请求可以主动给客户端推送资源，大大提高了网页加载的速度

安全：HTTPS将是未来的趋势，HTTP/2基于HTTPS也是未来的趋势，安全也是HTTP/2的重要特性之一

如何开启HTTP/2？

开启HTTP/2前，请确保HTTPS的证书已经配置成功；若您是第一次配置HTTPS证书，需要等到证书配置完成并且证书生效后，才能打开HTTP/2。

若您已经开启了HTTP/2，但是又关闭了https证书功能，HTTP/2会自动失效。

如何设置：

进入域名配置—HTTPS设置—找到HTTP/2后点击“修改配置”：

点击打开后，保存即可

全站加速

应用场景介绍

全站加速即 **动态加速**，适用于各行业动静态内容混合、含较多**动态资源请求**（如asp、jsp、php等格式的文件）的站点，阿里云CDN全站加速提供：

- 动静分离加速，动态内容采用智能路由、传输协议优化和链路复用技术，静态内容采用边缘缓存，提升整站资源加载速度。
- 实时探测及平滑跨网技术稳定高效处理高流量负载，提供全天候全网可用性。
- 回源负载均衡、多源主备、连接复用和有序回源技术降低源站压力和故障风险。
- 全链路HTTPS安全加速、防盗链、IP限流等保证源站安全。
- 自定义设置动静规则、缓存规则并配备全景信息监控和告警功能。

注意：**全站加速默认纯动态加速**，即所有动静态请求都通过最优路由回源获取资源，可通过配置指定静态文件类型或路径，实现智能区分动静态资源，静态资源缓存在边缘节点上，动态资源使用动态加速，达到

最快的加速效果。

计费规则

- 全站加速为增值服务，计费项为“**基础费用**”+“**请求数费用**”。其中“**基础费用**”是根据CDN服务所选择的“按峰值带宽”或“按流量”计费的基础费用。“**请求数费用**”包含动态**HTTP请求数**、**动态HTTPS请求数**和**静态HTTPS请求数**，分别按照单价按日计费。全站加速详情请参考 **全站加速介绍**。

图片鉴黄

产品介绍

- CDN图片鉴黄是CDN加速的一项增值服务，开通此功能后，用户在使用CDN服务过程中，系统会自动检测通过CDN加速的图片是否涉黄，违规图片的URL将会被记录下来供用户导出和删除。
- CDN图片鉴黄按照扫描张数计费，以回源的图片作为检测基数，同一条图片URL只会被检测一次，不会重复计费，同时用户还可以设置每日检测张数的上限，控制消费额度。
- CDN的图片鉴黄基于云计算平台，能对海量数据进行快速检测，可以帮助用户节省90%以上的人力成本。

使用方法

图片鉴黄功能可在CDN控制台的【增值服务】中使用：

1、设置待检测域名和限额：首次进入未设置检测域名，CDN中的域名需要在设置中添加到检测列表才会开始检测：

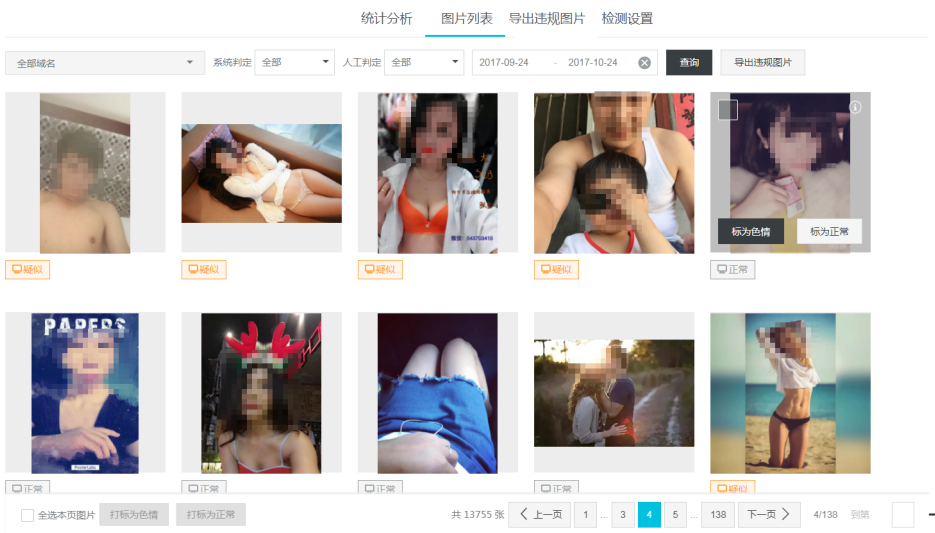
点击【马上设置】后进入设置菜单：

- 检测域名的设置：在左侧选择要检测的域名，添加到右侧检测中的域名栏中。
- 最后点击确定保存后，云端即开始检测通过CDN加速的新增图片**(注意：首次开通服务是第二天00:00开始检测)**。该功能对已有的**存量图片不会检测**。如需检测存量图片，可以通过手动刷新缓存的方式实现，刷新缓存后，待下次用户通过CDN访问该图片后即会自动检测，整个检测结果会延迟3-4小时。

2、查看统计数据及操作：配置完成后等待云端开始检测，3-4个小时后会有第一批结果出来。

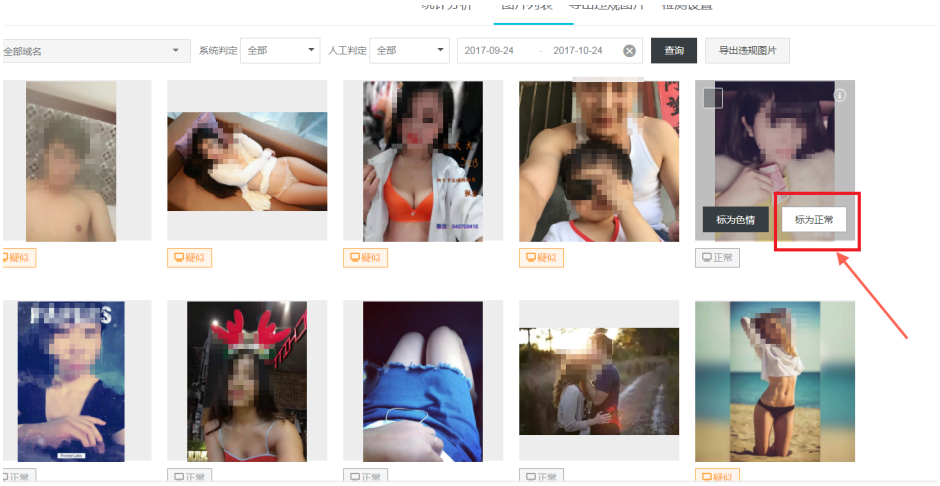
- 打开图片鉴黄菜单可以看到检测的统计数据信息，包括今日已检测的总量，疑似色情的图片量以及判定为色情的图片量：

- 点击【图片列表】标签可查看图片列表，选择查询条件可以筛选图片：



下方可以翻页，图片列表中可以查看所有检测过的图片（如果图片在源站被删除则可能会导致控制无法显示此图片）。

- 色情图片手工打标。 由于检测系统判定无法做到100%准确率，会有少量图片会识别成疑似色情或识别结果不对，此时可以通过手工打标的方式将图片打标为色情或正常。还可以同时选中多张图片批量打标：



3、导出涉黄图片列表：系统会将检测结果和手工打标的结果综合起来判定图片是否违规，通过【导出违规图片】按钮将所有违规图片导出:

用户可以根据导出的列表到自己的系统中进行删除，然后刷新对应的CDN缓存。

产品定价

- CDN图片鉴黄计费规则：

1. 计费周期为1天1次；
2. 按照当日扫描量收费，每日扫描量越大，单价越低；
3. 算法确定部分和待用户确认部分按照不同的单价计费。

- 后付费模式的详细计费标准如下：

阶梯（张/日）	确定部分 单价（元/千张）	待用户确认部分 单价（元/千张）
>0	¥1.80	¥0.45
>5000	¥1.62	¥0.41
>50000	¥1.53	¥0.38
>130000	¥1.44	¥0.36
>260000	¥1.35	¥0.34
>850000	¥1.26	¥0.32

- 鉴黄资源包的价格如下：

鉴黄包规格	价格	对应折扣
50万张	810元	9折
300万张	4590元	8.5折
500万张	7200元	8折
1000万张	13500元	7.5折
1亿张	126000元	7折
5亿张	540000元	6折

- 鉴黄资源包抵扣规则：

算法确定部分按照1：1抵扣，待用户确认部分按照1：0.25抵扣。

性能优化设置

智能压缩

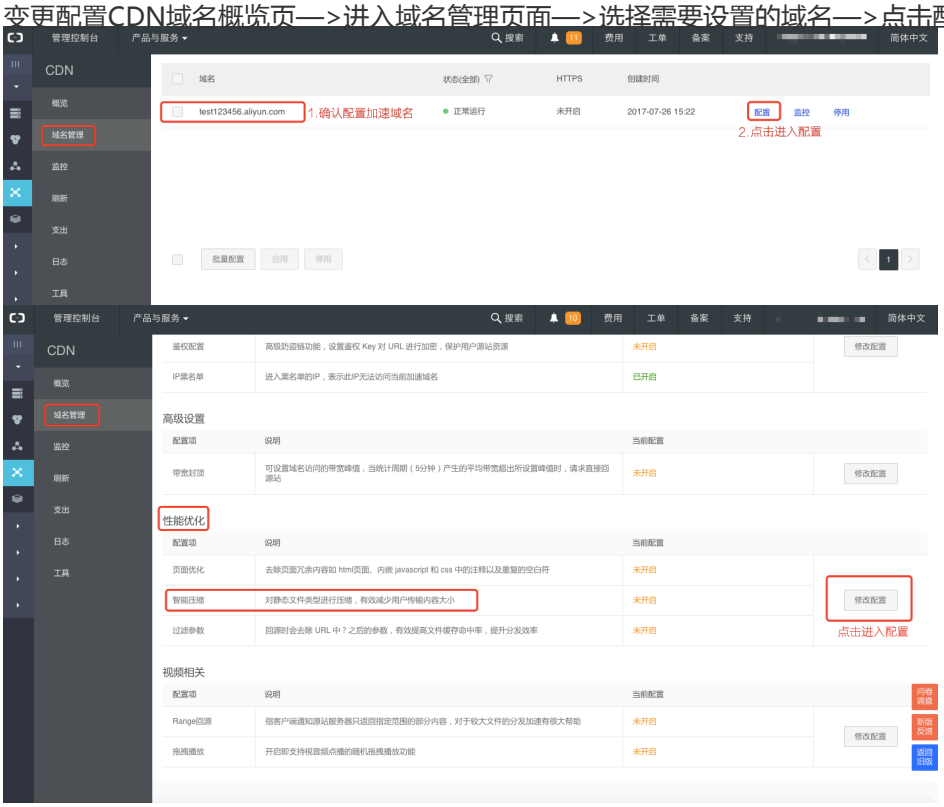
功能介绍

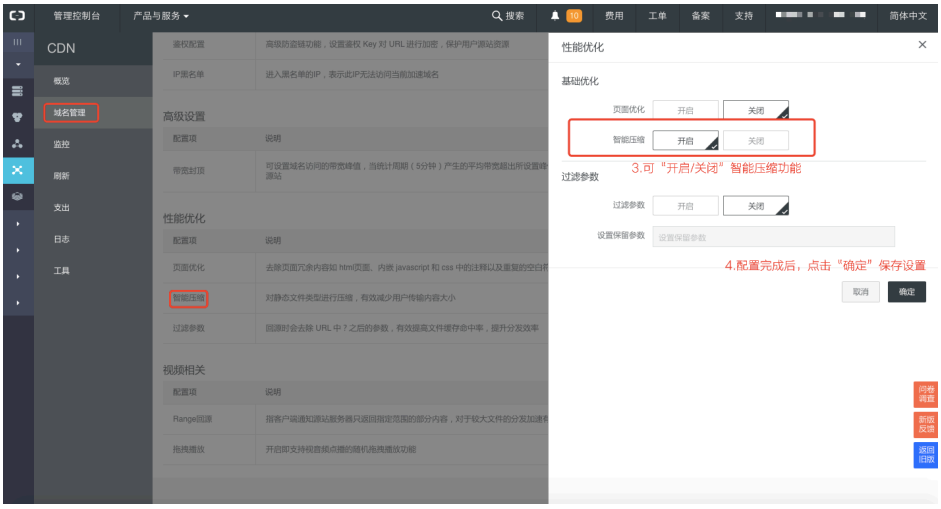
- 开启智能压缩功能，可以对大多数静态文件类型进行压缩，有效减少用户传输内容大小，加速分发效果
- 当前支持的压缩内容格式有：“content-type：text/xml ，text/plain ，text/css ，application/javascript ，application/x-javascript ，application/rss+xml， text/javascript ，image/tiff， image/svg+xml，application/json”

配置引导

适用业务类型：所有

变更配置CDN域名概览页—>进入域名管理页面—>选择需要设置的域名—>点击配置



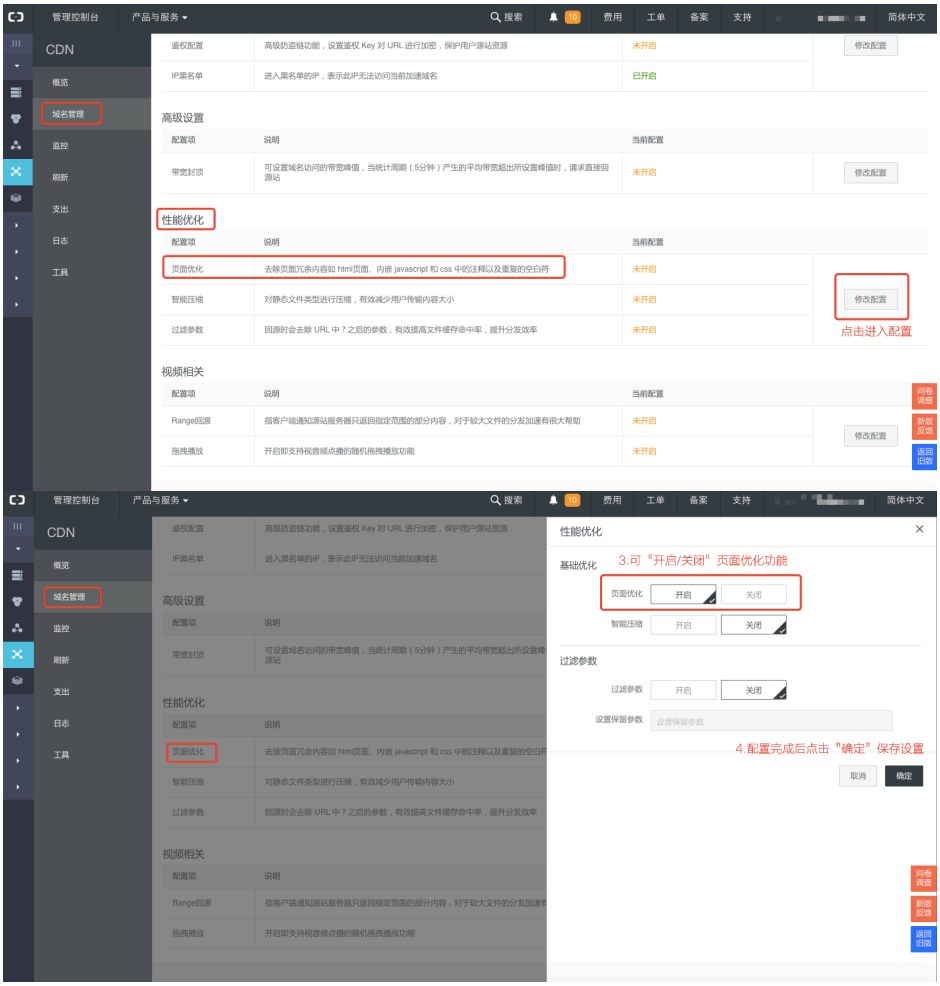


页面优化

功能介绍

- 开启页面优化功能，可以删除 html中的注释以及重复的空白符；这样可以有效地去除页面的冗余内容，减小文件体积，提高加速分发效率

配置引导



过滤参数

功能介绍

- 过滤参数是指当URL请求中带？并携带参数请求到CDN节点的时候，CDN节点在收到该请求后是否将该带参数的请求URL请求回源站。如果开启过滤参数的话，该请求到CDN节点后会截取到没有参数的URL向源站请求。并且CDN节点仅保留一份副本。如果关闭该功能，则每个不同的URL都缓存不同的副本在CDN的节点上
- http 请求中多包含参数，但是参数内容优先级不高，可以忽略参数浏览文件，适合开启该功能；开启后可以有效提高文件缓存命中率，提升分发效率
- 若参数有重要含义，例如包含文件版本信息等，推荐设置“保留参数”，支持设置多个保留参数，如请求中包含任一“保留参数”，会带保留参数回源，保留参数不忽略

使用示例

- 例如：http://www.abc.com/a.jpg?x=1 请求URL到CDN节点；
- 开启“过滤参数”功能后CDN节点向源站发起请求 http://www.abc.com/a.jpg （忽略参数x=1）待源站响应该请求内容后，响应到达CDN节点后，CDN节点会保留一份副本；然后继续向终端响应 http://www.abc.com/a.jpg 的内容。所有类似的请求 http://www.abc.com/a.jpg?参数 均响应CDN副本 http://www.abc.com/a.jpg 的内容。
- 关闭“过滤参数”功能则每个不同的URL都缓存不同的副本在CDN的节点上。例如：http://www.abc.com/a.jpg?x=1 和 http://www.abc.com/a.jpg?x=2 会响应不同参数源站的响应内容。

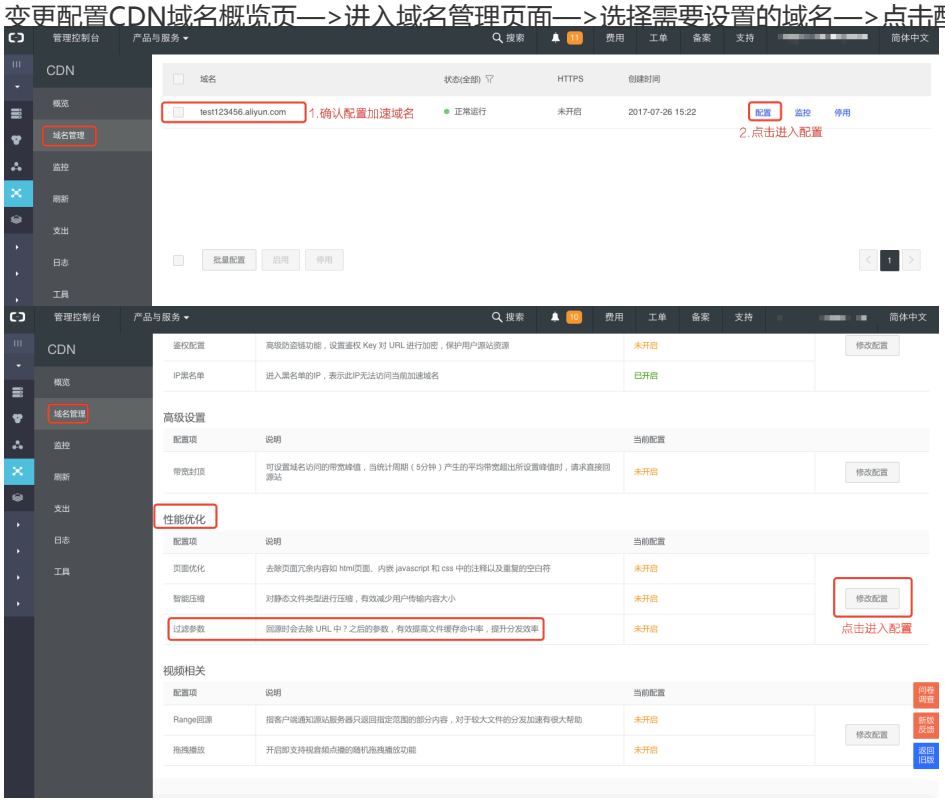
注意事项

- URL鉴权功能的优先级高于过滤参数，由于A类型鉴权信息包含在http请求的参数部分，系统会先进行鉴权判断，鉴权通过后在CDN节点缓存一份副本

配置引导

适用业务类型：所有

变更配置CDN域名概览页—>进入域名管理页面—>选择需要设置的域名—>点击配置



诊断工具

控制台的【工具】页面提供IP地址检测工具，可以验证输入的IP是否为阿里云CDN节点的IP。