阿里云 开发者社区 | 云服务技术课堂 Cloud Technology Class

CDN排坑指南

云运维工程师从入门到精通

作者：胡夫

阿里云工程师多年从业经验
真实案例详解问题排查要点

藏经阁

阿里云开发者电子书系列

阿里云 开发者社区

云服务技术大学
云产品干货高频分享

云服务技术课堂
和大牛零距离沟通

阿里云开发者“藏经阁”
海量免费电子书下载

目录

CDN 原理及快速入门

CDN 加速的核心原理是什么

简介：了解和学习阿里云 CDN 的工作原理非常重要，这对于网站优化、解决用户问题都有非常大的帮助。本文主要介绍了阿里云 CDN 的加速原理和缓存策略，举了一些实际的例子方便读者能清晰地理解阿里云 CDN。

什么是 CDN

CDN 的全称是 Content Delivery Network，即内容分发网络。CDN 是构建在现有网络基础之上的智能虚拟网络，依靠部署在各地的边缘服务器，通过中心平台的负载均衡、内容分发、调度等功能模块，使用户就近获取所需内容，降低网络拥塞，提高用户访问响应速度和命中率。CDN 的关键技术主要包括了节点调度、节点负载均衡和内容存储、分发、管理技术。

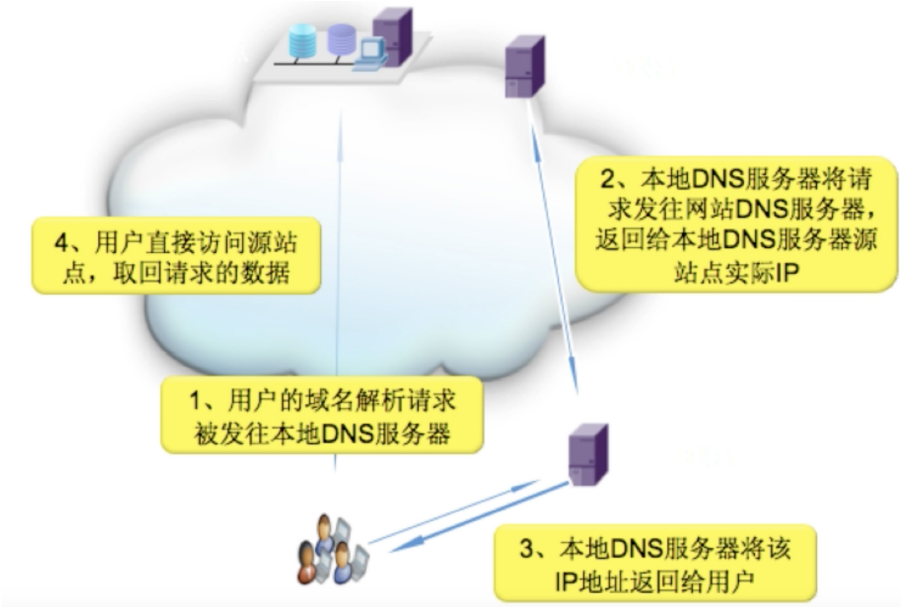
阿里云 CDN

阿里云在全球拥有 2800+ 节点。中国内地（大陆）拥有 2300+ 节点，覆盖 31 个省级区域，大量节点位于省会等一线城市。海外、中国香港、中国澳门和中国台湾拥有 500+ 节点，覆盖 70 多个国家和地区。同时，阿里云所有节点均接入万兆网卡，单节点存储容量达 40TB~1.5PB，带宽负载达到 40Gbps~200Gbps，具备 130Tbps 带宽储备能力。点击文档三分钟了解阿里云 CDN。

CDN 加速前

使用 CDN 加速前，用户侧发起的请求通过用户侧 DNS 递归到网站 DNS 解析以后，最终用户侧直接请求网站服务器。这里可能会造成以下几种情况：

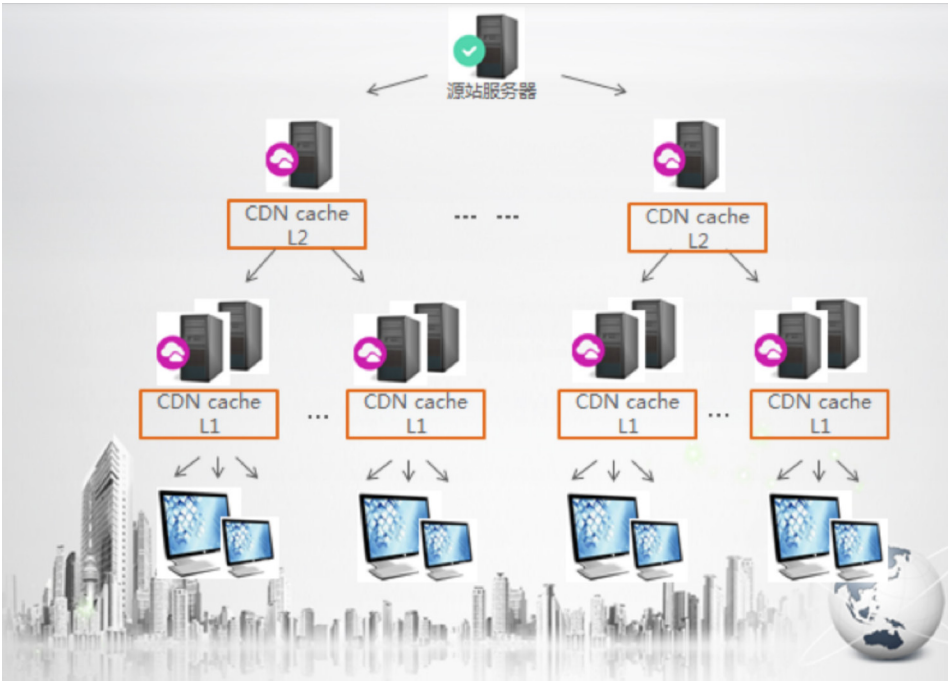
1. 中心服务器负载过高，因为所有客户端发起的请求都会请求到服务器上
2. 终端用户内容获取延时高，比如服务器在北京，而用户在广州
3. 服务稳定性差



CDN 加速后

CDN 通过在现有网络中增加一层新的缓存节点，将源站的资源发布到最接近用户的网络节点，使得客户端在请求时直接访问到就近的 CDN 节点并命中该资源，减少回源情况，提高网站访问速度。

阿里云 CDN 缓存节点可分为 L1 节点（一级节点）和 L2 节点（二级节点），请求的流程是：客户端 -->CDN_L1-->CDN_L2--> 源站。CDN 的 L1 节点分布在全国各省市，L2 节点分布在几个大区下，可以把 L2 节点理解为汇聚式节点，简单架构如下图所示。



CDN 节点缓存策略如下：

1. 客户端向 CDN 节点发起连接请求，当 L1 节点有缓存资源时，会命中该资源，直接将数据返回给客户端。当 L1 节点无缓存资源时，会向 L2 节点请求对应资源，如果 L2 节点有缓存资源，则将资源同步到 L1 节点，并返回给用户；如果 L2 节点无缓存资源，则直接回客户源站获取资源，并按照配置的缓存策略进行缓存。
2. 为了方便理解，再举一个简单例子，假设有杭州移动节点 L1-hz 和宁波移动节点 L1-nb 两个 L1 节点，这两个 L1 节点都回源到同一个 L2 这个节点，源站在北京。这几个 CDN 节点初始的时候都没有用户的缓存资源。当 ABC 三个用户依次请求同一个图片的时候，过程如下：

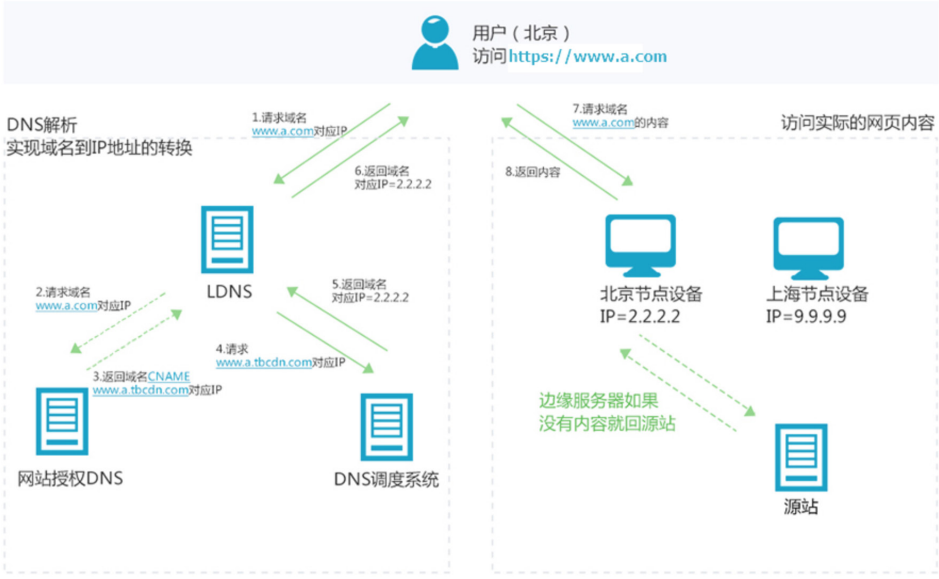
- 杭州移动用户 A 被 CDN 调度到杭州移动 L1-hz 节点，L1-hz 由于没有缓存，则回源到 L2，L2 由于也没有缓存，则回源到北京源站，请求到数据以后再返回给 L1-hz，L1 再返回给用户 A。

- 用户 A 请求完以后，L1-hz 和 L2 节点都有了缓存资源。此时杭州移动用户 B 也开始访问这个图片，用户 B 也被分配到了 L1-hz 节点，由于 L1-hz 已经有这个图片的缓存了，因此不需要再去回源了，而是直接返回缓存给用户 B。
- 宁波移动用户 C 此时也访问了同一个图片，用户 C 被分配到了宁波移动节点 L1-nb，由于 L1-nb 还没有缓存，就会回源到 L2，而 L2 已经有缓存，因此 L2 会直接返回缓存数据给 L1-nb，然后 L1-nb 再返回给用户 B。此过程存在 L1-nb 向 L2 回源的过程，而 L2 不需要再去回源到源站了。
- 通过 CDN 加速，杭州用户 A 和 B 可以直接从杭州节点读取缓存数据，宁波用户 C 可以直接从宁波节点读取数据，不需要每一次都去请求北京服务器了，提高了用户侧的访问速度，降低了服务器压力。

CDN 工作原理

通过以下案例，可以进一步了解 CDN 的工作原理。

假设加速域名为 www.a.com，接入 CDN 网络，开始使用加速服务后，当终端用户（北京）发起 HTTP 请求时，处理流程如下图所示。



1. 当终端用户（北京）向 www.a.com 下的某资源发起请求时，首先向 LDNS（本地 DNS）发起域名解析请求。
2. LDNS 检查缓存中是否有 www.a.com 的 IP 地址记录。如果有，则直接返回给终端用户；如果没有，则向授权 DNS 查询。
3. 当授权 DNS 解析 www.a.com 时，返回域名 CNAME www.a.tbcdn.com 对应 IP 地址。
4. 域名解析请求发送至阿里云 DNS 调度系统，并为请求分配最佳节点 IP 地址。
5. LDNS 获取 DNS 返回的解析 IP 地址。
6. 用户获取解析 IP 地址。
7. 用户向获取的 IP 地址发起对该资源的访问请求。

- 如果该 IP 地址对应的节点已缓存该资源，则会将数据直接返回给用户，例如，图中步骤 7 和 8，请求结束。
- 如果该 IP 地址对应的节点未缓存该资源，则节点向源站发起对该资源的请求。获取资源后，结合用户自定义配置的缓存策略，将资源缓存至节点，例如，图中的北京节点，并返回给用户，请求结束。配置缓存策略的操作方法，请参见缓存配置。

特别注意

CDN 调度系统分配节点的时候，是根据客户端的 LocalDNS 来分配节点的，而不是根据客户端 IP 来分配节点的。因此，如果客户端 LocalDNS 设置不正确的话会导致无法分配调度到最优的节点。

什么资源可以被加速

在 HTTP 请求的资源，请求可以分为静态请求和动态请求。

静态请求

静态请求是指在不同请求中访问到的数据都相同的静态文件。例如：图片、视

频、网站中的文件（html、css、js）、软件安装包、apk 文件、压缩包文件等。

CDN 加速的本质是缓存加速，将您服务器上存储的静态内容缓存在阿里云 CDN 节点上，当您访问这些静态内容时，无需访问服务器源站，就近访问阿里云 CDN 节点即可获取相同内容，从而达到加速的效果，同时减轻服务器源站的压力。

动态请求

动态请求是指在不同请求中访问到的数据不相同的动态内容。例如：网站中的文件（asp、jsp、php、perl、cgi）、API 接口、数据库交互请求等。

当客户端访问这些动态内容时，每次都需要访问用户的服务器，由服务器动态生成实时的数据并返回给客户端。因此 CDN 的缓存加速不适用于加速动态内容，CDN 无法缓存实时变化的动态内容。对于动态内容请求，CDN 节点只能转发回源站服务器，没有加速效果。

如果用户的网站或 App 应用有较多动态内容，例如需要对各种 API 接口进行加速，则需要使用阿里云全站加速产品。全站加速能同时加速动态和静态内容，加速方式如下：

- 静态内容使用 CDN 加速。
- 动态内容通过阿里云的路由优化、传输优化等动态加速技术以最快的速度访问您的服务器源站获取数据。从而达到全站加速的效果。

CDN 的缓存策略

静态内容可以在 CDN 上缓存多久，这个是根据 CDN 的缓存策略的。如果用户没有主动到 CDN 上配置过期时间，则会遵循 CDN 的默认缓存策略。CDN 的默认缓存时间比较短，最大不超过 3600 秒，因此很容易缓存过期。因为网站开发及其相关技术人员更清楚自身网站的业务逻辑、静态和动态因素，所以建议用户通过控制台按照文件类型和目录设置缓存时间。

用户可以登录阿里云 CDN 控制台配置缓存过期时间，针对静态资源配置指定目录和文件后缀名的缓存过期时间和优先级，资源过期后，自动从 CDN 节点删除。

全站加速和 CDN

全站加速（Dynamic Route for Content Delivery Network）是阿里云融合了动态加速和静态加速技术的 CDN 产品。该产品一站式优化了页面动静态资源混杂、跨运营商、网络不稳定、单线源站、突发流量、网络拥塞等诸多因素导致的响应慢、丢包、服务不稳定的问题，提升全站性能和用户体验。全站加速和 CDN 的对比如下

对比项	CDN	全站加速
支持资源类型	仅支持静态内容加速。	同时支持静态内容和动态内容加速。
加速方式	将您服务器上的静态内容缓存在阿里云CDN节点上供您就近访问。	静态内容使用CDN加速。 动态内容通过智能路由、协议优化等动态加速技术快速访问您的服务器源站获取。
源站适配	建议您对服务器源站的动静态内容进行分离，静态内容使用CDN加速，动态内容不使用CDN加速。	您无需对服务器源站上的资源进行改造，全站加速会智能区分动静态内容并分别加速。

全站加速的静态加速和 CDN 的加速原理一致，是通过将静态资源缓存到边缘节点的方式，提供用户就近访问去做加速。全站加速的动态加速是对于动态请求回源的时候，通过智能路由优化、协议优化等动态加速技术快速回源获取。

> 注：全站加速默认走了动态加速，动态加速是每次回源的。如果需要走缓存的话，需要配置静态加速。目前配置静态加速支持按照文件类型、URI 以及路径方式配置。

更多

CDN 应用场景

CDN 初次使用快速导航

简介：本文介绍了用户使用阿里云 CDN 加速时的快速入门手册，包括如何添加域名、如何配置 CNAME 解析、如何验证 CDN 是否生效，并介绍了一些常见的问题。

概述

当您初次使用 CDN 时，可以快速了解其操作流程和操作场景。本文档指导您快速开通 CDN，并加速您的域名，操作流程如入门概述所示。主要包括：【开通 CDN 服务】-->【添加 CDN 加速域名】-->【配置 CNAME 解析】-->【验证 CDN 是否生效】

添加 CDN 加速域名

登录阿里云 CDN 控制台，按照添加加速域名帮助文档去完成域名的添加。该帮助文档对每一个选项做了解释，如不清楚基本概念可先阅读帮助文档，以下是一些添加域名的时候遇到的常见问题。

如何选择加速域名

- Q. 加速域名是添加主域名还是子域名

假如您的网站域名是 www.test.com，您想加速该网站，那么这里的 CDN 加速域名应该填写 www.test.com ，而不是 test.com。也就是说，您需要加速哪个域名，就添加对应的这个域名。

- Q. 是否支持泛域名

支持添加泛域名。泛域名是指使用通配符做加速域名以实现所有的次级域名加速效果。例如，您添加了 .test.com 作为加速域名，将 .test.com 解析至 CDN 生成的 CNAME 域名后，则所有 .test.com 的次级域名 a.test.com 均支持 CDN 加速。泛

域名 .test.com 的三级域名 b.a.test.com 不提供加速服务。

- Q. 泛域名限制

最多支持三级泛域名，3 个点，例如：*.b.c.com

如何选择业务类型

- Q. 业务类型的种类和概念是什么

目前 CDN 支持 " 图片小文件 "、" 大文件下载 "、" 视音频点播 "、" 直播流媒体 "、" 全站加速 "、" 安全加速 " 业务类型，可以根据实际的业务类型，选择适合自己业务的类型。

- Q. 如何加速直播业务

若您需要加速直播流媒体，请直接登录直播控制台，添加域名并进行相关配置。如果业务是推到自己的流媒体服务器，而不是直接推到阿里云，只是希望通过阿里云 CDN 加速直播的话，可以考虑使用直播服务的拉流直播功能。直播流方向：主播 --> 服务器 --> 阿里云直播中心 --> 观众。

- Q. 如果选择音视频点播业务，是否无法加速图片小文件

CDN 针对每一个业务类型都有特定的优化。音视频点播业务类型只是对音视频的加速效果更佳，比如该场景支持对视频的拖拽等。但同时该业务类型也是支持加速图片小文件的，并不是说音视频业务只能加速音视频。同理，其他业务场景也是一样的道理，但是具体的选择，还是需要根据您主要加速的资源和业务场景，去选择合理的业务类型。

如何填写源站

- Q. 什么是源站

源站就是指您实际业务的服务器，当 CDN 节点未缓存请求资源或缓存资源已到期时，CDN 会回源到源站获取资源，返回给客户端。源站类型可以选择 OSS 域名、

IP、源站或函数计算域名。假设在使用 CDN 前，网站域名是 www.test.com，该域名解析到服务器 1.1.1.1 , 那么这里的源站 IP 就填写 1.1.1.1 。特别注意，源站必须公网可达。

- Q. 什么是源站域名

源站域名决定了回源时，CDN 请求到哪个 IP。假设源站域名是 www.a.com，那么 CDN 回源的时候，在 CDN 服务器上会先 DNS 解析 www.a.com 得到源站 IP 地址，然后再请求到该 IP。请注意：源站域名不能和 CDN 加速域名相同。

- Q. 源站是否支持第三方的服务器或 OSS

支持，只要保证源站公网可达，可正常提供服务即可。

- Q. 源站是阿里云 OSS 或 ECS，回源是否可以走内网

不支持走内网，因为 CDN 节点都是架设在公网上的。

- Q. 是否支持多个源站 IP

支持多个服务器外网 IP。CDN 主要支持主备方式切换源站场景。当多个源站回源时，优先回源优先级为主的源站。如果主站连续 3 次健康检查均失败，则回源优先级为备的源站。如果该源站的主站健康检查成功，则该源站将重新标记为可用，恢复其优先级。当所有源站的回源优先级相同时，CDN 将自动轮询回源。

如何填写端口

- Q. 端口是什么意思

这个端口是指 CDN 的回源端口，它决定了 CDN 回源的时候，请求到源站的哪个端口。如选择 80 端口，则 CDN 以 HTTP 协议访问源站资源。如选择 443 端口，则 CDN 以 HTTPS 协议访问资源。

如何选择加速区域

- Q. 加速区域概念

CDN 提供三个加速区域："仅中国大陆"、"全球"和"全球（不包含中国大陆）"。"全球（不包含中国大陆）"这个加速区域，只有海外的 CDN 节点，没有中国大陆的 CDN 节点，因此使用这个加速区域的情况下，中国大陆用户访问的时候都会访问到海外的 CDN 节点，因此中国大陆用户没有加速效果。同理，"仅中国大陆"这个加速区域只有中国大陆的 CDN 节点，使用这个加速区域的情况下，海外用户访问的时候都会访问到中国大陆的 CDN 节点，因此海外用户没有加速效果。

- Q. 域名没有备案怎么办

"全球（不包含中国大陆）"这个区域只有海外的 CDN 节点，所以不用备案。"仅中国大陆"和"全球"这两个区域包含了国内的 CDN 节点，所以域名必须备案的。如希望加速中国大陆用户，则推荐您进入阿里云 ICP 代备案管理系统进行备案。

添加域名失败常见问题

提示“域名已添加”

- 阿里云的 CDN、全站加速 (DCDN)、安全加速 (SCDN)、视频直播 (Live)、视频点播 (VOD) 产品，底层都是基于 CDN 网络的，同一个域名不能同时添加到以上两个产品上。因此如果您的域名在以上其中一个产品下已经添加了，则到其他一个产品里去添加同一个域名，会提示域名已存在。
- 检查域名是否在自己的其他阿里云账号里添加了。如无法找到原因，提交阿里云工单协助处理。

提示根域名被占用

目前 CDN 添加域名的时候，有如下限制。请检查自己是否有多个阿里云账号，如无法定位请提交阿里云工单域名迁移。

- 添加精确域名时候：如果已经在 CDN 上有添加泛域名，则必须跟泛域名在同一个账号，否则报错。

- 添加泛域名时候：如果已经在 CDN 上添加了精确域名，那么添加的泛域名必须在同一个账号，否则报错。

域名达到数量上限

每个阿里云账户下，最多支持加速 50 个域名。如果您的域名的总带宽日均峰值大于 50MB，且业务无风险，则可提交工单申请增加域名个数。如果带宽不满足 50MB，则暂时无法添加，建议使用泛域名加速。具体请参照 CDN 的使用限制说明。

提示非法错误

通常情况是因为之前使用 CDN 加速服务时有一些违规业务导致域名被 CDN 加入黑名单，请参照 CDN 的域名准入标准核实业务。

域名一直审核中或审核失败

- 目前 CDN 域名有专门的审核人员做审核，需要确保 CDN 加速的内容是符合接入标准的，因此如果没有及时审核通过，则需要耐心等待。
- 如果最终审核失败，则需要确认域名加速的内容是否符合准入标准，可以参照域名审核失败帮助文档排查确认。

配置 CNAME 域名解析

域名添加成功后，阿里云 CDN 会分配对应的 CNAME 地址。如果您想启用 CDN 加速服务，则需要将加速域名指向 CNAME 地址，访问加速域名的请求才能转发到 CDN 节点上，达到加速效果。

什么是域名解析

如果您是初次接触域名解析，您可以会有一堆疑问。" 什么是域名解析 "、" 为什么要解析域名 "、" 如何进行域名解析 "、" 什么是 A 记录 "、" 什么是 CNAME 记录 "、"CNAME 记录与 A 记录的差别 "，这些问题在这篇域名解析帮助文档里有很好的

解释，建议您参考下。

如何配置 CDN 的 CNAME 解析

如果您的域名是在阿里云 / 万网，请参考以下域名解析参考文档：阿里云 / 万网配置流程

如果您的域名是在腾讯云（原 DNSPod），请参考以下域名解析参考文档：DNSPod 配置流程

如果您的域名是在新网，请参考以下域名解析参考文档：新网配置流程

域名解析冲突

很多用户在配置 CNAME 记录的时候，出现了域名解析冲突的情况。在同一个域名解析服务商下，域名解析是存在冲突规则的，比如 A 记录和 CNAME 记录冲突，MX 记录和 CNAME 记录冲突等，具体可以看这个域名解析冲突规则。

	NS	CNAME	A	URL	MX	TXT	AAAA	SRV	CAA
NS	可重复	冲突	冲突	冲突	冲突	冲突	冲突	冲突	冲突
CNAME	冲突	可重复	冲突	冲突	冲突	冲突	冲突	冲突	冲突
A	冲突	冲突	可重复	冲突	不冲突	不冲突	不冲突	不冲突	不冲突
URL	冲突	冲突	冲突	冲突	不冲突	不冲突	冲突	不冲突	不冲突
MX	冲突	冲突	不冲突	不冲突	可重复	不冲突	不冲突	不冲突	不冲突
TXT	冲突	冲突	不冲突	不冲突	不冲突	可重复	不冲突	不冲突	不冲突
AAAA	冲突	冲突	不冲突	冲突	不冲突	不冲突	可重复	不冲突	不冲突
SRV	冲突	冲突	不冲突	不冲突	不冲突	不冲突	不冲突	可重复	不冲突
CAA	冲突	冲突	不冲突	不冲突	不冲突	不冲突	不冲突	不冲突	可重复

（1）CNAME 记录和 A 记录冲突

- Q. 如何处理

需要删除 A 记录，然后再去配置 CNAME 记录。

- Q. 删除 A 记录是否无法访问网站

只要配置了 CNAME 记录以后，客户端的请求会请求到 CDN 上，然后 CDN 再去访问源站服务器，因此就没必要再配置 A 记录了。CNAME 在 CDN 加速中的原理，请参见工作原理。

（2）CNAME 记录和 MX 记录冲突

请参照 CNAME 和 MX 冲突的解决方法处理。

示例

预期要为域名dnswork.top同时添加主机记录为@的MX和CNAME记录

记录类型	主机记录	记录值
MX	@	mx1.hichina.com
CNAME	@	example.com

结论：因为MX和CNAME冲突的规则，无法正常完成添加。

建议方案

您可以通过使用URL转发记录来解析CNAME记录和MX记录冲突问题，需要注意的是URL转发前和转发后的域名都需要接入备案，URL转发配置参考

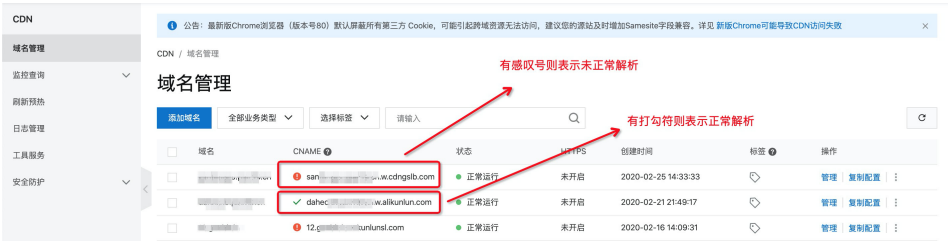
解析记录配置如下：

记录类型	主机记录	解析线路	记录值	TTL
URL	@	默认	example.com	10分钟
MX	@	默认	mx1.hichina.com	10分钟

验证 CDN 是否生效

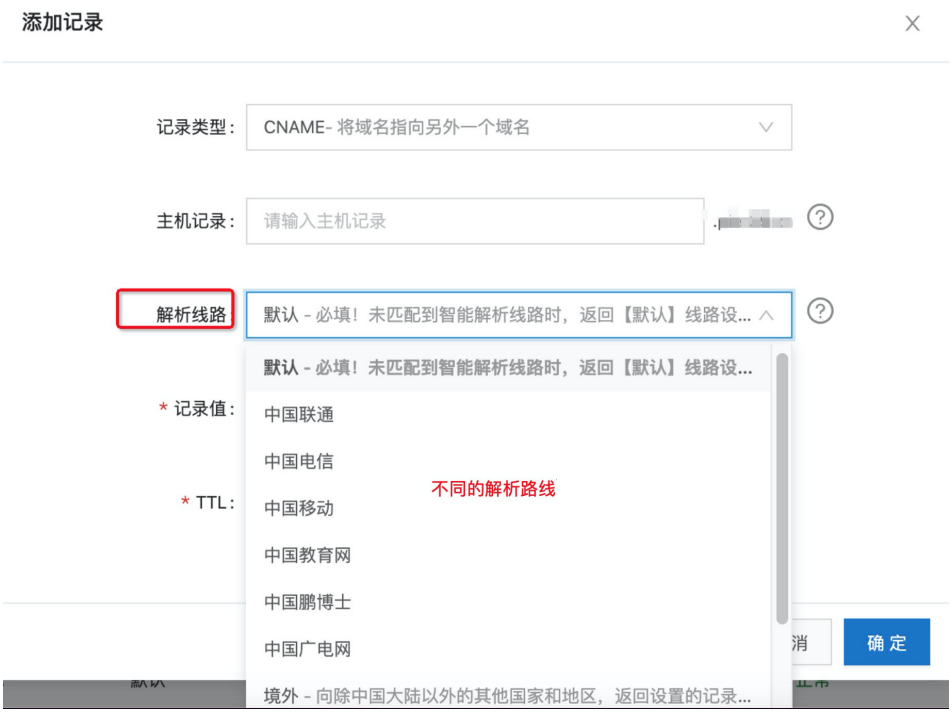
按照前面的步骤操作 CNAME 解析，如果 CNAME 解析正确，则 CDN 控制台

会显示正常的 " √ " 符号。也可以参考如何验证 CDN 节点是否生效文档来确认是否正常解析到 CDN。如果已经正常解析解析到 CDN，可以通过如何通过浏览器的审查元素判断 CDN 缓存是否成功文档判断是否可以命中 CDN 缓存。



如果控制台显示不正常的解析，则可能有以下几种原因：

- 确认配置的 CNAME 解析的记录值是否和 CDN 控制台获取的记录值一致，如不一致则解析失败
- 配置完域名解析以后，运营商 DNS 的 TTL 还未更新，则需要耐心等待下，一般情况下 TTL 时间为 10 分钟，具体以解析配置的时候选择的 TTL 为准
- CDN 服务会去全网检查加速域名域名是否解析到 CDN，如果大部分区域已经解析，但是还是有个别地区没有解析的话，也会显示感叹号，需要全网解析生效以后才会显示正常
- 有一种特殊情况是，用户配置域名解析的时候设置了解析路线，需求部分地区不走 CDN 加速。比如国内的解析路线是解析到 CDN，海外的解析路线是 A 解析到服务器，这种情况下，因为海外没有解析到 CDN，因此控制台没显示正常，但在这种需求场景下，不影响用户实际使用，如下图


添加记录
记录类型：
CNAME- 将域名指向另外一个域名
主机记录：
请输入主机记录
解析线路：
默认 - 必填！未匹配到智能解析线路时，返回【默认】线路设...
* 记录值：
* TTL：
默认 - 必填！未匹配到智能解析线路时，返回【默认】线路设...
中国联通
中国电信
中国移动
中国教育网
中国鹏博士
中国广电网
境外 - 向除中国大陆以外的其他国家和地区，返回设置的记录...
不同的解析路线
消
确 定

CDN 进阶功能排查

7 个问题搞懂 HTTPS 证书配置

简介：本文主要介绍如何在阿里云 CDN 上配置 HTTPS 证书以及一些常见问题的解答。

什么是 HTTPS？

HTTP 协议以明文方式发送内容，不提供任何方式的数据加密。HTTPS 协议是以安全为目标的 HTTP 通道，简单来说，HTTPS 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 协议。HTTPS 提供了身份验证与加密通讯方法，被广泛用于万维网上安全敏感的通讯，例如交易支付。

根据 2017 年 EFF（Electronic Frontier Foundation）发布的报告，目前全球已有超过一半的网页端流量采用了加密的 HTTPS 进行传输。更多 HTTPS 的信息请参考阿里云 CDN 官方帮助文档什么是 HTTPS 加速。

CDN 如何 HTTPS 加速

使用了 CDN 以后，域名解析到了 CDN，因此必须要在 CDN 侧配置 HTTPS 证书。如果 CDN 上没有配置 HTTPS 证书，则 CDN 只支持 HTTP 访问；如果 CDN 上配置了 HTTPS 证书，则 CDN 支持 HTTP 和 HTTPS 访问。具体配置请参考帮助文档“配置 HTTPS 证书”。

源站已经配置了 HTTPS，CDN 上是否还需要配置

HTTPS 是客户端和服务端的交互，没有用 CDN 前，是客户端直接和源站交互，因此源站需要配置 HTTPS。使用 CDN 以后，是客户端和 CDN 交互，因此如果需要 HTTPS 访问 CDN，则 CDN 上必须要配置 HTTPS 证书。源站配置了 HTTPS 证书只是支持 CDN 以 HTTPS 回源到源站。

为什么配置了 HTTPS，客户端还是 HTTP 访问的

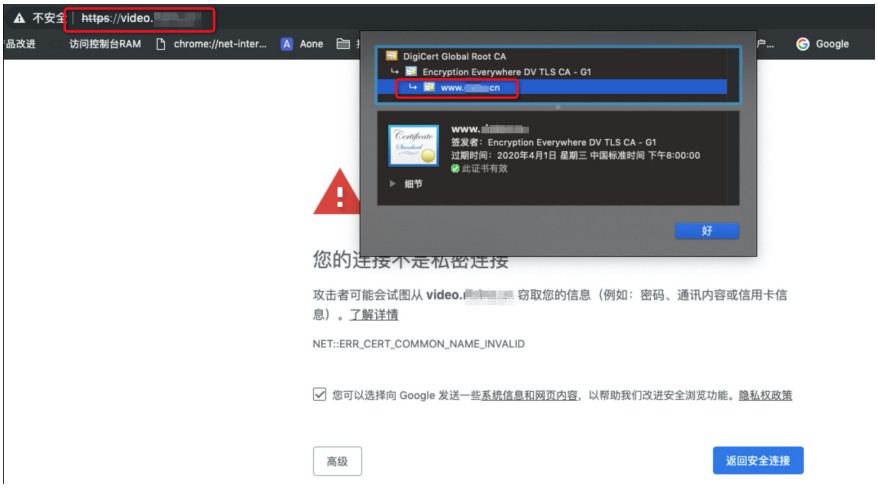
客户端是 HTTP 访问还是 HTTPS 访问完全是客户端的行为，如果希望客户端强制用 HTTPS 访问，可以在 CDN 上开启强制 HTTPS 跳转。

申请 CDN 免费 HTTPS 证书失败

在阿里云 CDN 控制台中申请免费 HTTPS 证书时，存在一些限制。您可以参考“在 CDN 的 HTTPS 设置中申请免费证书失败”的文档去排查和解决。

CDN 配置 HTTPS 以后还是无法访问

（1）如果是购买证书以后自定义上传的情况，需要特别注意 SSL 证书根据其适用范围可以分为：通配符域名、单个域名和多个域名。根据其名称即可查看购买的证书分别适用于主域名下某个级别的全部子域名、单个域名或者多个域名。用户是需要保证购买的证书必须适用于加速域名后续才可以添加在 CDN 中生效。如下图所示的即是添加的 SSL 证书（适用于 www 域名）与 CDN 加速域名（video 的子域名）是不相匹配的，因此会抛出 NET::ERR_CERT_COMMON_NAME_INVALID 的错误。



（2）CA 机构提供的证书为了兼容性可能会提供多种形式的证书，CDN 支持的证书仅有 PEM 格式，并且私钥需要 RSA 格式。如果客户获取得到的是其他格式的证书是需要转换后然后提交到 CDN 服务中的，常见格式切换格式请参考：CDN 证书格式说明，而其中的私钥文件如果是 -----BEGIN PRIVATE KEY-----，-----END PRIVATE KEY----- 样式的话是需要通过如下命令转换成 RSA 格式：

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

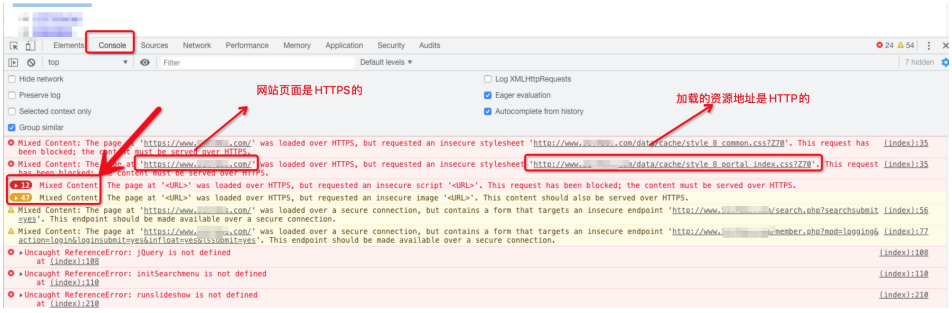
（3）CDN 是不支持设置密码的私钥。如图 3 所示即是经过加密的私钥，这类私钥文件是需要经过解密后才可以正常使用，因此 CDN 是无法正常使用的。

（4）证书链需要补全中间证书。对于中级 CA 机构提供的证书，那么拿到的证书将包括多份证书，而 CDN 需要添加的是包括中间证书的完整证书链，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。另外有一些中间证书 CA 机构提供了不同的服务器使用的证书，由于 CDN 是基于 Tengine 提供服务的，因此用户是需要使用 Nginx 对应的证书到视频中心的。

（5）CDN 的 HTTPS 技术是基于 SNI 技术实现的。SNI 技术主要是用来在同一台服务器上配置多个证书的需求，而 SNI 是需要客户端发送请求的时候带有 SNI 的信息以标识是哪个域名的 SSL 请求，因此 SNI 技术对客户端有一定的要求，部分低版本系统中的低版本浏览器不满足该要求。SNI 技术对于客户端的限制详细请参考：SNI 对客户端浏览器限制。

为什么网站开启 HTTPS 以后显示不全

打开浏览器开发者模式，切换到 console 页面，如果看到 Mixed Content 错误，则说明是浏览器安全限制导致的。浏览器要求 Https 的页面里只能加载 Https 的地址，不能加载 Http 的资源。如果您的 Https 的页面里加载了很多 http 的资源，这些资源加载不出来的，因此会引起网站显示异常，这种情况需要网站技术人员把 htm 代码里加载的资源地址都改成 https 的。



CDN 刷新和预热常见问题

简介：本文详细介绍了刷新和预热功能并列举了常见问题以及解决方案。

刷新和预热的概念

刷新功能是指提交 URL 刷新或目录刷新请求后，该加速域名下的所有 CDN 节点里的缓存内容将会被强制过期，当用户再次向 CDN 节点请求资源时，CDN 会直接回源站获取对应的资源返回给用户，并将其缓存。

（1）因为刷新会强制清除缓存，因此刷新功能会降低缓存命中率。

（2）刷新支持 URL 刷新和目录刷新。

（3）URL 刷新使用限制为 2000 条 / 日 / 每账户，目录刷新使用限制为 100 个 / 日 / 每账户。

预热功能是指提交 URL 预热请求后，源站将会主动将对应的资源缓存到 CDN 节点，当用户首次请求时，就能直接从 CDN 节点缓存中获取到最新的请求资源，无需再回源站获取。预热功能会提高缓存命中率。

（1）预热只支持 URL 预热，不支持目录预热。

（2）同一个 ID 每天最多预热 500 个 URL，每次最多只能提交 100 条。

点击【操作类型】可以选择 " 刷新 " 或者 " 预热 "

源站更新以后 CDN 多久时间更新

CDN 是否更新完全取决于 CDN 节点上的缓存是否过期了，如果缓存没有过期，那么 CDN 依然会返回缓存数据，因为 CDN 并不知道源站更新数据了。如果缓存过期了，那么客户端请求到 CDN 的时候，CDN 因为没有缓存，就会回源向源站去获取数据，这个时候就获取到新的数据，进而依据缓存规则来把资源缓存下来。

如果用户没有在 CDN 上配置缓存规则，那么 CDN 是依赖于默认的缓存规则来缓存的，最大缓存时间不超过 3600 秒。如果用户有在 CDN 配置缓存规则，那么缓存过期时间就是依赖于用户配置的缓存规则的。

因此如果源站更新了资源，需要手动到 CDN 控制台去刷新缓存，或调用 CDN 的 API/SDK 接口去刷新缓存，一般情况下刷新是 5 分钟内生效。如果是调用 API/SDK 接口，可以考虑写一个自动化脚本。阿里云 CDN 为用户提供了 Python 示例脚本，帮助用户对文件或目录快速进行刷新和预热，具体可以参见刷新预热自动化脚本。

如何查看预热任务是否执行完成

在 CDN 控制台刷新预热功能界面单击【操作记录】，即可查看预热任务的执行

状态。预热任务的状态为成功，表示预热任务提交成功，并不代表文件已经预热结束。执行如下命令，查看预热任务的执行状态

```
curl -I 'http://cdnoss.xxxxxxxxxx.com/test.json'
```

系统显示类似如下

（1）CDN 缓存节点可分为 L1 节点（一级节点）和 L2 节点（二级节点），请求的流程是：客户端 -->CDN_L1-->CDN_L2--> 源站。CDN 的 L1 节点是边缘节点，分布在靠近用户侧，L2 节点分布在几个大区下，可以把 L2 节点理解为汇聚式节点，一个 L2 对应多个 L1 节点。预热功能是指提交 URL 预热请求后，CDN 的 L2 节点作为一个客户端，主动向源站发起请求，将源站的资源缓存到 CDN 的 L2 节点上。

（2）CDN 的 HTTP 响应头里有 Via 字段（如上图），Via 的前半部分代表二级节点状态，其中的“H”表示命中，说明文件已经预热到二级节点，即预热成功了，不需要再回源站。

（3）Via 的后半部分代表一级节点的状态，“M”表示一级节点上没有缓存，需要向二级节点回源。

影响预热完成时间的因素

预热完成的时间跟**预热的资源大小、预热的 URL 数量、源站的性能、回源网络**

等因素有关。

（1）如前文所说，CDN 是 L1+L2 的架构，预热是指 L2 节点回源。假设用户的 CDN 加速域名的调度域里有 10 个 L2 节点，那么预热一个 URL，那么这 10 个 L2 节点就会同时向源站去请求这个资源。依次类推，如果同时预热 100 个 URL，那就同时会有 1000 个请求去请求源站了。

（2）基于以上说明，因为预热会有并发的请求去请求源站，因此需要保证源站的性能、公网带宽等能满足，否则很可能会导致预热失败等情况。例如用户的源站带宽是 5M，但是实际同时预热了大量的 URL，造成大量来自 CDN 的预热请求，源站的带宽被打满，那么会导致 CDN 无法正常请求到数据，最终导致预热失败。

（3）另外请求资源的大小和数量也影响预热时间，比如预热 1 个 5M 的文件跟预热一个 500M 的文件，那时间肯定不一样。比如预热 1 个 5M 的文件，跟同时预热 100 个 5M 的文件，那时间也不一样。

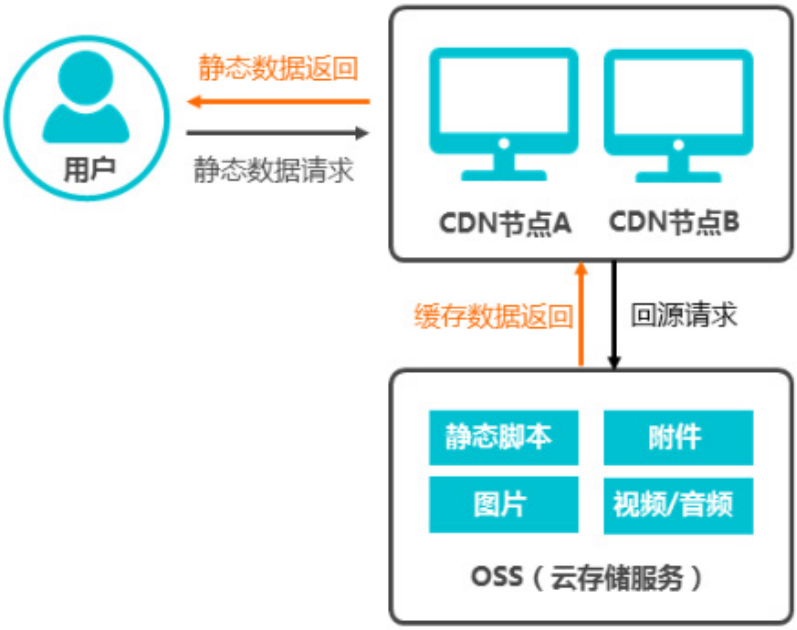
基于以上说明，根据源站的性能情况，结合实际的业务情况，合理的预热可以到达更优的访问效果。

高效低价！ CDN 加速 OSS 架构优势

简介：本文主要介绍 CDN 加速 OSS 的原理、入门配置以及常见问题。

背景信息

OSS 源站上存储的静态资源包括静态脚本、图片、附件和音频 / 视频。当终端用户请求访问或下载静态资源时，CDN 对 OSS 源站上的静态资源进行加速，源站上的资源缓存到 CDN 的加速节点，系统自动调用离终端用户最近的 CDN 节点上已缓存的资源。加速 OSS 架构如下图所示。具体操作可以参考 CDN 加速 OSS 最佳实践。



架构优势

CDN 加速 OSS 的优势如下

（1）用户访问网站资源，全部通过 CDN，降低源站压力。

（2）使用 CDN 流量，单价低于 OSS 直接访问外网流量。

（3）资源从距离客户端最近的 CDN 节点获取，减少网络传输距离，保证静态资源质量。

常见问题

一、如何设置缓存自动刷新

CDN 加速 OSS 的场景下，在 OSS 的 Bucket 中修改了 Object，使用 CDN 加速域名访问得到的 Object 仍是更新前的旧版本，必须在 CDN 中手动刷新后才能访问到新版本的 Object。

这是因为 Bucket 中的 Object 更新了，但是 CDN 中该 Object 的缓存未到期，所以访问的时候还是老的资源。用户可以开启 CDN 缓存自动刷新功能。开启此功能后，若 Object 有更新，OSS 会自动刷新 CDN 上的缓存，从而实现文件更新时缓存自动刷新。具体可以登录 OSS 控制台，在对应 Bucket 下选择【传输管理】>【域名管理】界面，设置开启 CDN 缓存自动刷新。

二、如何解决静态文件强制下载

出于安全考虑，从 2019 年 9 月 23 日起，针对之后新建的 Bucket，直接使用 OSS 提供的默认域名，从互联网访问 OSS 上该 Bucket 的图片类型文件，即 mimetype 为以下值：

image/jpeg、image/gif、image/tiff、image/png、image/webp、image/svg+xml、image/bmp、image/x-ms-bmp、image/x-cmu-raster、image/exr、image/x-icon、image/heic，扩展名包括：jpg、jpeg、jpe、png、tif、tiff、gif、svg、bmp、ico、ras、dib、svgz、webp、bm、jfif、x-png、exr 和 heic 时，Response Header 中会自动加上 Content-Disposition:'attachment=filename;'。即从浏览器访问图片类型文件时，会以附件形式进行下载。用户使用自有域名访问 OSS 的请求，Response Header 中不会加上此信息。解决方案如下：

（1）检查并设置 Bucket 是否绑定 CDN 加速域名，详情请参见绑定 CDN 加速域名。

（2）检查并配置 CDN 配置的回源 Host 为用户的加速域名，而不是源站域名。如果设置源站域名（也就是 OSS 的域名），则 CDN 回源的时候所带的 Host 就是 OSS 域名，这样会导致 OSS 返回强制下载的 HTTP 头最终导致资源直接下载。

注：如果访问 URL 仍然为强制下载，则可能是 CDN 缓存了强制下载的 HTTP 头，需要在 CDN 控制台刷新 URL。另外需要检查 OSS 源文件的 HTTP 头内 Content-Type 的值是否正确，详情请参见 OSS 如何设置 Content-Type。

三 . CDN 加速导致 OSS 配置的 CORS 跨域失效

使用 CDN 加速 OSS 跨域访问失败，原因是可能存在这样的场景：

第一个用户访问 CDN 时，没有发起跨域访问，然后 CDN 回源到 OSS 的时候 OSS 返回了不带跨域头的 Respons Headers 信息，并且被 CDN 缓存下来了。 当第二个用户访问时，发起了跨域请求，但是由于 CDN 有缓存，直接把之前缓存下来的不带跨域头的 Respons Headers 信息返回了，导致本次跨域请求失败。因此建议使用 CDN 加速 OSS 时，直接在 CDN 上去配置跨域规则，具体请参考 CDN 如何配置跨域资源共享（CORS）。

四 . CDN 加速 OSS 资源返回 403 状态码

为了防止 OSS 被盗链，保护 OSS 的资源安全，用户把 OSS 的 Bucket 权限设置为私有权限，这样就需要带了签名参数的 URL 去访问。如果 CDN 加速 OSS 的访问 URL，不带 OSS 签名参数的话，就会导致 403。这种情况下可以开启阿里云 OSS 私有 Bucket 回源授权。当开启私有 OSS Bucket 回源授权后，即表示开启 CDN 对所有 Bucket 的只读权限，CDN 在回源的时候会计算 OSS 的签名参数，从而可以从 OSS 上正常获取资源。

五 . CDN 加速 OSS 访问静态托管页面返回 403

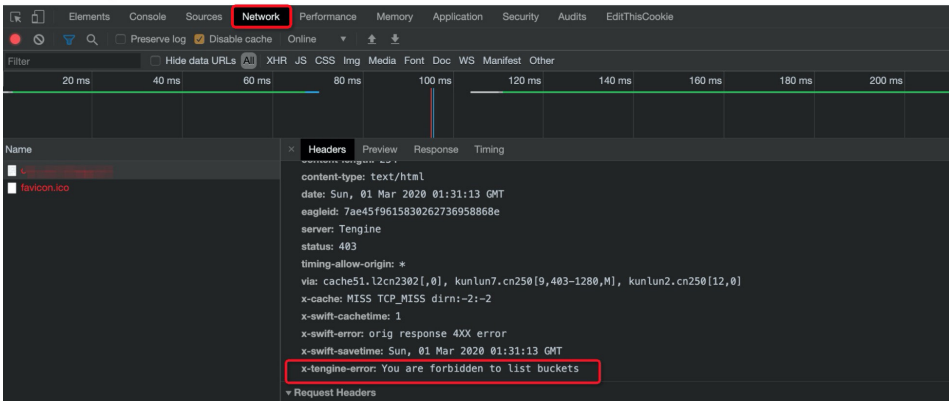
可以通过 Network 下获取 403 请求的 Response Headers 信息去查看对应的错误信息，如果出现如下错误，说明是开启私有 Bucket 回源授权的情况下访问了 OSS 的静态首页。需要注意，目前 CDN 的私有 Bucket 回源功能和 OSS 的静态网站托管功能冲突，无法一起使用。

```
You are forbidden to list buckets
```

403 Forbidden

You don't have permission to access the URL on this server.

Powered by Tengine



必备 API 接口和 SDK 工具包

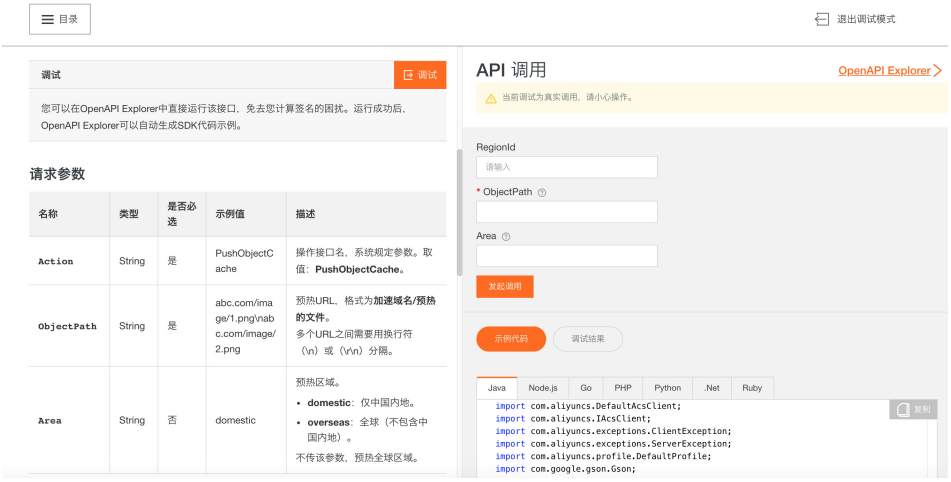
简介：阿里云 CDN 提供了丰富的 API 接口，除了控制台，CDN 还开发了多样化的接口。目前我们推荐用户使用新版 API，详情请参见新版 API 参考。同时 CDN 提供了多语言 SDK 工具包，并且准备了 SDK 使用说明，以便用户了解如何获取、安装和调用阿里云 SDK。用户可以单击 CDN SDK 下载，并参考文档去集成。目前 CDN 提供的 SDK 如下：Java、Python、PHP、.NET、C 或 C++、Go。

API 调用

CDN 的 API 是 RPC 风格，用户可以通过发送 HTTP GET 请求调用 API，并按照接口说明在请求中加入相应请求参数，调用后系统会返回处理结果。CDN 支持通过 HTTP 或 HTTPS 通道进行请求通信，为了获得更高的安全性，推荐使用 HTTPS 通道发送请求。我们提供了 API 的调用规则，并且提供了使用 Java、Python 封装 API 的示例，具体可以参见 API 调用方式。

SDK 调用

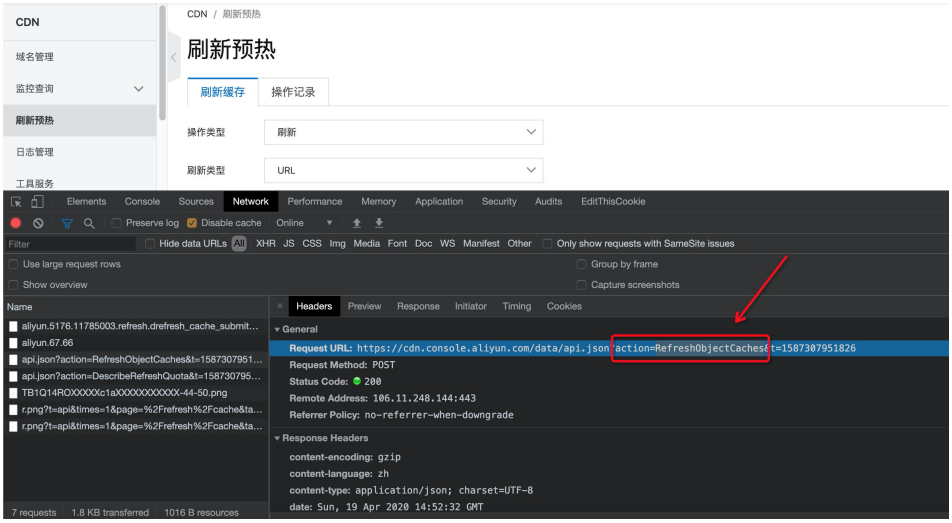
由于 API 调用需要按照调用规则去计算签名，经常会遇到一些开发者计算签名错误导致无法正常调用 API 的情况，建议用户直接使用 SDK 来调用接口，SDK 里封装了接口，会自动计算签名，免去自己计算签名的困扰。用户可以直接在 OpenAPI Explorer 中运行接口，填写接口私有参数运行成功后，OpenAPI Explorer 可以自动生成 SDK 代码示例，集成 SDK 以后调用示例代码即可。如下图，在 API 接口文档里单击 " 调试 " 按钮即可进入调试模式，设置参数以后调试成功可以查看示例代码。

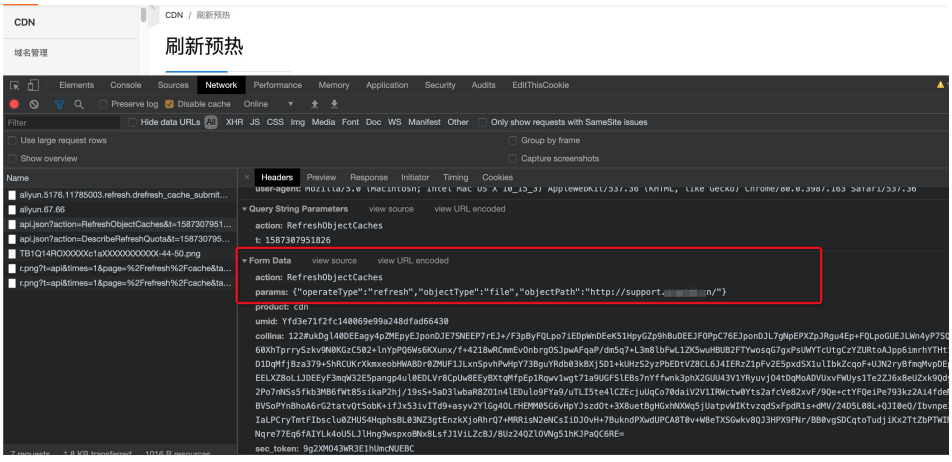


注：接口文档提供了调用接口的错误码，如果出现错误时，可以根据具体的错误码以及错误解释查找原因，确认是否是哪个参数传的不正确。也可以参见这个 CDN 错误代码汇总匹配错误信息。

控制台查看接口参数

由于 CDN 控制台也是通过调用 CDN 的 API 接口来实现对 CDN 的相关操作，因此如果对于调试时调用接口的参数有疑问的时候，可以通过控制台先去设置相关的功能，然后通过打开浏览器 Network 去看控制台具体调用了什么接口，传了什么参数，这样方便理解以便进一步完成自己的开发工作。以刷新功能为例，在控制台刷新功能页面下打开 Network，然后执行刷新工作，可以看到控制台调用了 RefreshObjectCaches 接口，并且在 Form Data 下面可以看到控制台调用该接口发的参数。具体可以看下面的图一和图二。


CDN
域名管理
监控查询
刷新预热
日志管理
工具服务
CDN / 刷新预热
刷新预热
刷新缓存
操作记录
操作类型
刷新
刷新类型
URL
Elements Console Sources Network Performance Memory Application Security Audits EditThisCookie
Preserve log Disable cache Online
Filter Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other Only show requests with SameSite issues
Use large request rows Group by frame
Show overview Capture screenshots
Name
aliyun.5176.11785003.refresh.drefresh_cache_submit...
aliyun.67.66
api.json?action=RefreshObjectCaches&t=1587307951...
api.json?action=DescribeRefreshQuota&t=15873079...
TB1Q14ROXXXXXc1aXXXXXXXXXX-44-50.png
r.png?t=api&time=1&page=%2Frefresh%2Fcache&ta...
r.png?t=api&time=1&page=%2Frefresh%2Fcache&ta...
Headers Preview Response Initiator Timing Cookies
General
Request URL: https://cdn.console.aliyun.com/data/api.json?action=RefreshObjectCaches&t=1587307951826
Request Method: POST
Status Code: 200
Remote Address: 106.11.248.144:443
Referrer Policy: no-referrer-when-downgrade
Response Headers
content-encoding: gzip
content-language: zh
content-type: application/json; charset=UTF-8
date: Sun, 19 Apr 2020 14:52:32 GMT
7 requests 1.8 KB transferred 1016 B resources


CDN
域名管理
CDN / 刷新预热
刷新预热
Elements Console Sources Network Performance Memory Application Security Audits EditThisCookie
Preserve log Disable cache Online
Filter Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other Only show requests with SameSite issues
Use large request rows Group by frame
Show overview Capture screenshots
Name
aliyun.5176.11785003.refresh.drefresh_cache_submit...
aliyun.67.66
api.json?action=RefreshObjectCaches&t=1587307951...
api.json?action=DescribeRefreshQuota&t=15873079...
TB1Q14ROXXXXXc1aXXXXXXXXXX-44-50.png
r.png?t=api&time=1&page=%2Frefresh%2Fcache&ta...
r.png?t=api&time=1&page=%2Frefresh%2Fcache&ta...
Headers Preview Response Initiator Timing Cookies
Query String Parameters view source view URL encoded
action: RefreshObjectCaches
t: 1587307951826
Form Data view source view URL encoded
action: RefreshObjectCaches
params: {"operateType":"refresh","objectType":"file","objectPath":"http://support
product: cdn
umid: Yfd3e71f2fc140069e99a248dfad66430
collina: 122#ukDgl40DEEagy4pZMEpyEJponOJE7SNEEP7rEJ+/F3pByFQLpo7iEDpWnDEeK51HpyGZp9hBuDEEJFOPpC76EJponDJL7gNpEPXZpJRgu4Ep+FQLpoGUEJLWn4yP7SQ
60XhTprrySzkv9N0KGzC502+lnYpPQ6Ws6KXunx/f+4218wRCmmEvOnbrgOSJpwAFqaP/dm5q7+L3m8lbFwL1ZK5wuHBUB2FTYwosqG7gxPsUWYTcUtgCzYZURtoAJpp6imrhYTHt
D1DqMfjBza379+ShRCUKrXkmxeobHWABDr0ZMUF1JLxnSpvhPwHpY73BguYRdb03kBXj5D1+kUHzS2yzPbEDtVZ8CL6J4IERzZ1pFv2E5pxdSX1ulIbkZcqoF+UJN2ryBfmqMvpDE
EELXZ8oLiJDEEyF3mqW32E5pangp4ul0EDLVr8CpUw8EEyBXtqMfpEp1Rqwv1wgt71a9UGF5lEBs7nYffwnk3phX2GUU43V1YRyuvjO4tDqMoADVUxvFWUys1Te2ZJ6x8eUZxk9QdF
2Po7nNSs5fkb3MB6fWt85sikaP2hj/19sS+5aD3lwbaR8ZO1n4lEDul9FYa9/uTLI5te4lCZEcjuUqCo70daiV2V1IRWctw0YtsZafcVe82xvF/9Qe+ctYFQeiPe793kz2Ai4fde
BVSoPYn8hoA6rG2tatvQtSobK+ifJx53ivITd9+asyv2YlGg4OLrHEMM05G6vHpYJszdOt+3X8uetBgHGxhNXWq5jUatpvWIKtvzqdSxFpdR1s+dMV/24D5L08L+QJI0eQ/IbvnpeI
IaLPCryTmtFIbsclu0ZHUS4HqphsBL03NZ3gtEnzkXjoRhrQ7+MRRisN2eNCsIiDJOvH+7BuknqPXwdUPCA8T0v+W8eTXSGwkv8QJ3HPX9FNr/BB0vgSDCqtoTudjiKx2TtZbPTWIf
Nqre77Eq6fAIYLk4oU5LJlHng9wspxoBNx8LsfJ1ViLZcBJ/8Uz24QZlOVNg51hKJPaQC6RE=
sec_token: 9g2XM043WR3E1hUmcNUEBC

CDN 性能优化与安全防护

核心课题——学会从缓存命中率解决看问题

简介：本文详细介绍旅 CDN 缓存命中率的概念，分析了无法命中缓存的问题原因以及命中率降低影响因素，并针对命中率降低及如何优化做了分析和介绍。

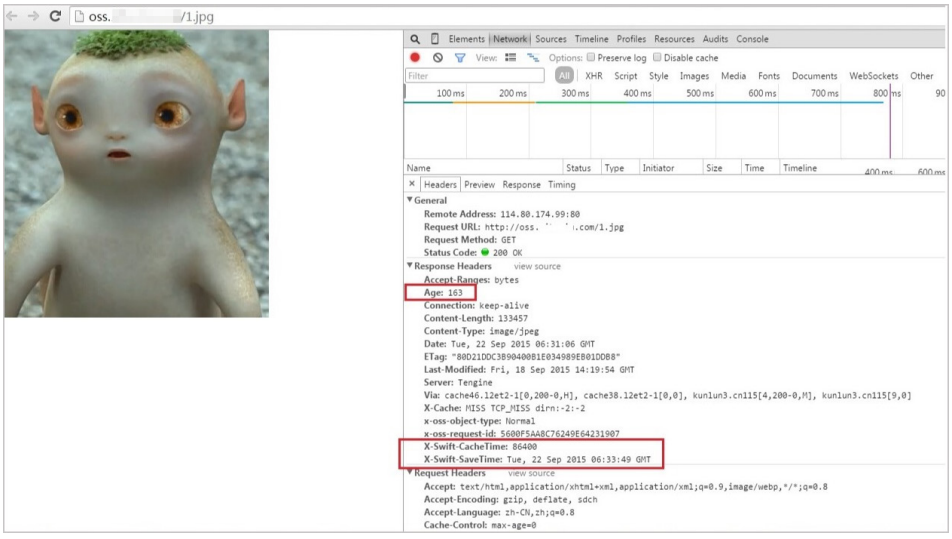
提升缓存命中率的意义

CDN 在静态资源加速场景的应用，是将静态资源缓存在距离客户端最近的 CDN 节点上。用户访问该资源时，直接从缓存中获取资源，避免通过较长的链路回源。如果 CDN 缓存命中率低，则会导致源站压力大，静态资源访问效率低。因此，CDN 缓存命中率的高低直接影响用户体验，而保证较高的缓存命中率也成为了 CDN 的核心课题。可以针对导致 CDN 缓存命中率低的具体原因，选择对应的优化策略，来优化 CDN 的缓存命中率。CDN 缓存命中率包括字节缓存命中率和请求缓存命中率。

- 字节缓存命中率 = CDN 缓存命中响应的字节数 / CDN 所有请求响应的字节数
- 请求缓存命中率 = CDN 缓存命中的请求数 / CDN 所有的请求数

如何判断缓存是否成功

我们可以通过打开浏览器审查元素来分析 CDN 返回的 Response Header，其中 X-Cache 字段来判断是否命中缓存，具体可以参见如何通过浏览器的审查元素判断 CDN 缓存是否成功。



在 Response Headers 字段内，可以查看详细的请求和返回的报文信息。

- Age：为 CDN 返回的头部字段，表示该文件在 CDN 节 – 点上缓存的时间，单位为秒。只有文件存在于节点上 Age 字段才会出现，当文件被刷新后或者文件被清除的首次访问，在此前文件并未缓存，无 Age 头部字段，需要注意当 Age 为 0 时，表示节点已有文件的缓存，但由于缓存已过期，本次无法直接使用该缓存，需回源校验。
- X-Swift-SaveTime：CDN 节点上的缓存 RS（swift）的时间，即该文件是在什么时间缓存到 CDN 节点上。
- X-Swift-CacheTime：CDN 节点上的允许缓存时间，即该文件可以在 CDN 节点上缓存多久，是指文件在 CDN 节点缓存的总时间。计算还有多久需要回源刷新 = 'X-Swift-CacheTime' – 'Age'。
- X-Cache："HIT" 表示已缓存，"MISS" 表示节点上无该文件的缓存，回源请求。

为什么无法命中缓存

（1）客户端请求是动态请求

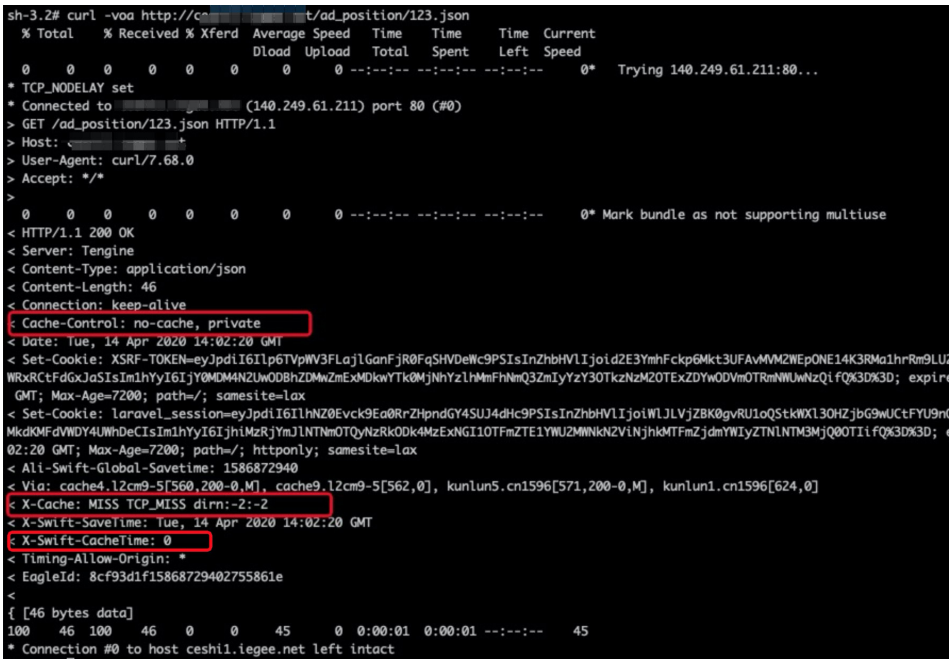
如果请求是动态请求，则无法命中 CDN 缓存。当客户端访问这些动态内容时，每次都需要访问用户的服务器，由服务器动态生成实时的数据并返回给客户端。

（2）源站返回强制不缓存的 HTTP 头

当源站配置了以下响应头时，即使配置了缓存规则，CDN 也不会对该资源进行缓存，因为这些响应头在 CDN 缓存规则中的优先级较高。

- 1： 有 s-maxage=0、max-age=0、no-cache、no-store、private 中 的任一种。
- 2：有 s-maxage 或 s-maxage=0。
- 3：有 Pragma: no-cache。

如下图，源站响应了 no-cache 和 private 导致 CDN 无法缓存，X-Cache 字段为 MISS，X-Swift-CacheTime 为 0。这种情况需要源站去掉这个 HTTP 响应头，具体可以参见 Nginx 缓存策略设置、Apache 缓存策略的设置、IIS 缓存策略的设置方法。



（3）未返回响应头 Etag 和 Last-modified

当 CDN 未配置缓存规则时，如果静态文件未返回响应头 Etag 和 Last-modified，则该静态文件不能缓存在 CDN 节点上。解决方案就是源站配置返回 Etag 和 Last-modified 或者直接在 CDN 上配置缓存规则。

（4）全站加速未配置静态加速

全站加速默认走了动态加速，动态加速是每次回源的。如果需要走缓存的话，需要配置静态加速。目前配置静态加速支持按照文件类型、URI 以及路径方式配置。如果全站加速没有配置静态加速的情况，则都是走动态加速的，全站加速节点响应的

HTTP 头没有 X-Cache、X-Swift-CacheTime 等字段的，类似如下图。

```
sh-3.2# curl -I http://[redacted]n/1.html
HTTP/1.1 200 OK
Server: Tengine
Content-Type: text/html
Content-Length: 1144
Connection: keep-alive
Vary: Accept-Encoding
Date: Sun, 19 Apr 2020 17:41:36 GMT
Last-Modified: Fri, 09 Jun 2017 10:26:53 GMT
Vary: Accept-Encoding
ETag: "593a77ed-478"
Expires: Tue, 19 May 2020 17:41:36 GMT
Cache-Control: no-cache
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: X-Requested-With
Access-Control-Allow-Methods: GET,POST,OPTIONS
Accept-Ranges: bytes
Via: cache27.l2et2[26,0], cache9.cn590[35,0]
Timing-Allow-Origin: *
EagleId: 3ad8761d15873180964265824e
```

影响 CDN 缓存命中率下降的因素

影响 CDN 缓存命中率下降的因素：

（1）刷新缓存，可能导致短时间内命中率下降。

（2）带宽突增，会导致 CDN 节点回源较多，命中率会表现有下降趋势。

（3）CDN 节点访问新内容，导致 CDN 节点回源较多，命中率会表现有下降趋势。

（4）缓存规则调整，可能会影响命中率。

缓存命中率低分析及优化

CDN 控制台统计的缓存命中率仅仅是 CDN L1 层的命中率，实际情况 L2 层的

缓存数据也是从 CDN 节点获取，并不会从源站获取数据，所以真实的 CDN 命中率是略高于 CDN 控制台显示的命中率。

另外查看 CDN 加速域名流量情况，在加速域名流量不高的情况下，即便 MISS 状态的 URL 不多，但是对命中率的统计计算影响很大。例如，某 CDN 加速域名一共对外提供了 10 个可以访问的 URL，其中有一个 URL 源站上设置了 no-cache，导致不缓存，在其他 URL 访问都命中的情况下，命中率也仅有 90%。

在之前检查正常的情况下，有如下几种可能导致命中率低的情况，请逐一进行排查：

（1）源站上缓存 Header 设置不当，或者缺少必要的 Header，如果 CDN 的缓存规则是不缓存，那么每次访问都是 MISS 状态，影响命中率，具体请参考前文“为什么无法命中缓存”的描述。

（2）CDN 控制台设置了不缓存的规则，即某目录或者某种后缀的文件设置的缓存时间为 0 秒，相关信息可以在 CDN 控制台查看。

（3）源站动态内容较多，目前 CDN 主要是加速静态资源，例如 CSS、JS、HTML、图片、txt、视频等资源，针对动态资源 PHP、JSP、包含内部逻辑处理甚至 Cookie 等资源都会回源数据。

（4）CDN 的加速 URL 中带有可变参数。例如 URL 地址为 http://XXX.XXX.cn/1.txt?timestamp=14378923，其中 timestamp 值为时间戳，每次访问此值均不同。CDN 针对第一次访问的 URL，即之前未预热的 URL，无论该 URL 是否符合 CDN 的缓存规则，由于节点上还没有这个文件，第一次访问肯定都是 MISS 状态。但是 timestamp 参数会变化，所以每次访问都是一个全新的 URL，则每次都返回 MISS 状态，从而影响命中率。

（5）检查是否存在频繁刷新缓存的操作。

（6）文件热度不够。不经常被用户访问到的 URL，即使符合所有缓存规则，但是

经常有被节点去除缓存的风险。CDN 节点上缓存的文件，可以理解为按照热度属性采取末尾淘汰制，热度就是该文件在该节点上被访问的频率，文件热度不够，其实一定程度上跟这个域名本身的流量不高有关系。

针对以上情况，可以考虑通过 " 预热 URL"、" 配置资源缓存规则 "、" 过滤 URL 中可变参数 " 来优化缓存命中率，具体操作请参见优化 CDN 缓存命中率。

流量突增?！CDN 帮你分析

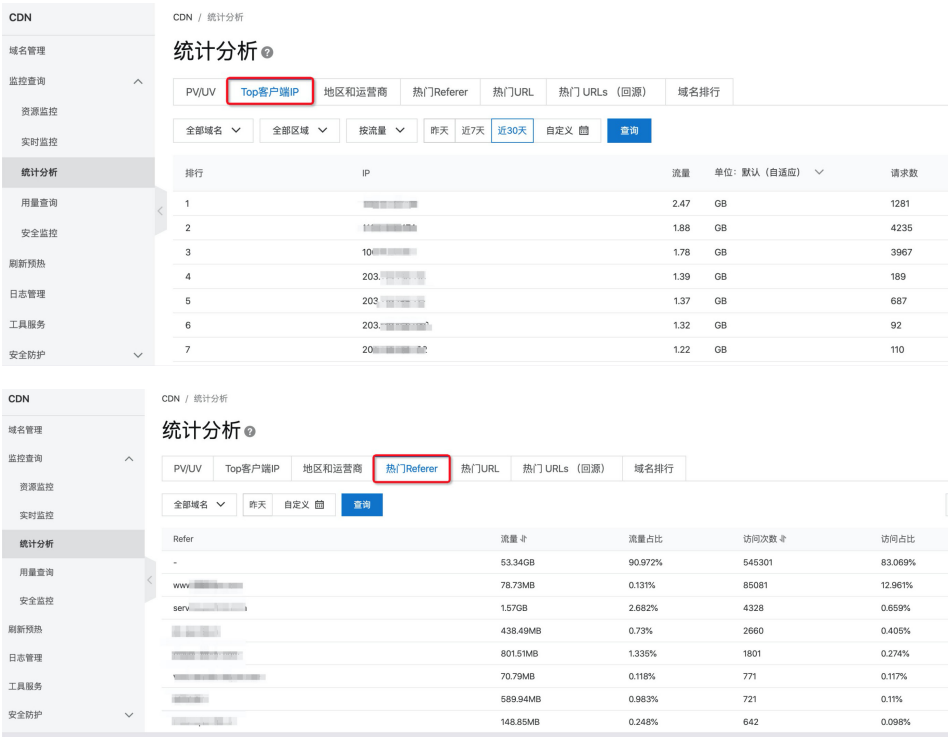
简介：当 CDN 出现流量突增、明显大于业务正常范围流量时，则很可能是被恶意攻击或者刷流量导致，CDN 本身是提供了一些安全防护策略，本文主要介绍如何去定位异常流量来源以及如何去做 CDN 的安全防护和监控。

问题分析

CDN 流量出现突增时，先要检查是否是有一些业务上的推广导致的流量增加。如果不是业务推广，却突然产生了大量的异常流量，则很可能是被恶意攻击或者刷流量导致，需要定位客户端来源信息来加以防护。通常是借助 CDN 提供的监控统计以及日志来加以分析。

一、监控统计

CDN 控制台提供了监控统计分析，包含七个部分：PV/UV、Top 客户端 IP、地区和运营商、域名排名、热门 Referer、热门 URL、热门 URLs（回源）、域名排行。用户可以导出原始详细数据，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。用户可以通过这些监控统计分析客户端的来源信息、请求 URL、Top 客户端等信息。其中中等是分析 Top 客户端 IP 以及 Top Referer 信息，具体可以看如下图



注意：Referer 为 "-" 则表示是空 Referer 的请求。

二、日志分析

CDN 控制台的显示统计分析报表数据会有延迟，对于正在发生的恶意刷流量或攻击行为，往往无法获取实时的信息，这种情况下需要借助 CDN 提供的日志来分析定位客户端的来源信息。目前 CDN 提供的日志分为两种：离线日志和实时日志。

（1）离线日志：默认开启，日志文件延迟一般情况下延迟在 24 小时之内，但是也有可能超过 24 小时。登录 CDN 控制台，下载日志文件，日志相关字段说明请参日志下载。可以借助 Linux 命令来分析日志，例如执行如下命令，查询访问量前十的 IP：

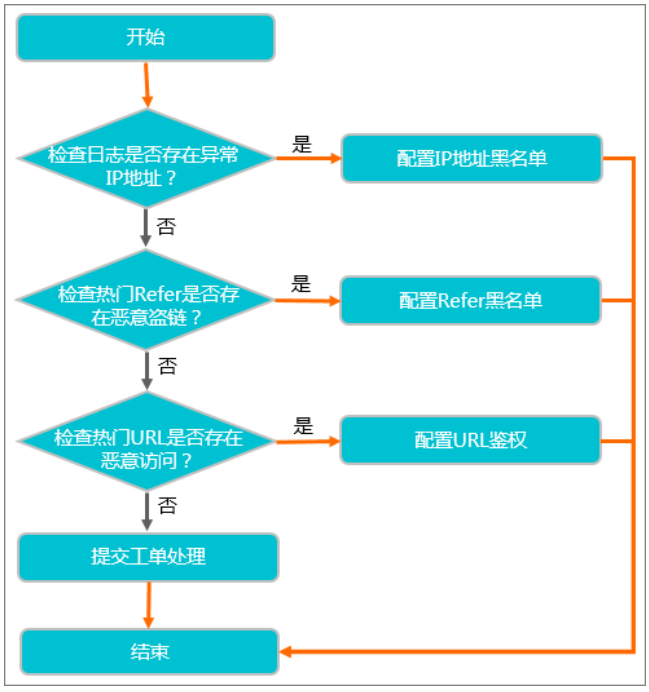
```
cat [$Log_Txt] | awk '{print $3}' |sort|uniq -c|sort -nr |head -10
```

用同样的方法可以分析访问量前 10 的 Top Referer、URL 信息，更多请参考日志分析。

（2）实时日志：属于计费服务，默认不开启，需要手动开启，日志数据延迟不超过 3 分钟。阿里云 CDN 通过与日志服务融合，将采集到的实时日志实时推送至日志服务，并进行日志分析。通过日志的实时分析，用户可以快速发现和定位问题。除此之外，通过对日志数据的挖掘，提高数据的决策能力，将您的业务推向一个新的高度，更多请参考实时日志帮助文档。

安全防护

通过分析出客户端来源信息以后，可以按照以下处理流程图来处理



（1）配置 IP 黑白名单：通过监控统计分析和日志查看是否存在异常 IP 地址访问资源。如果有短时间大量访问的恶意 IP 地址，请将 IP 地址配置为黑名单。

（2）配置 Referer 防盗链：该功能是根据 HTTP 请求的 Referer 字段来对请求来源的域名进行筛选和链接。CDN 支持三种防盗链设置：白名单、黑名单以及是否允许空 refer。防盗链功能主要通过 URL 过滤的方法对来源 Host 的地址进行过滤，其中黑名单和白名单只能有一种生效，可以将恶意 Referer 加入黑名单或将正常业务 Referer 加入白名单，通过该功能可以对请求来源进行限制。

（3）配置频次控制：如果恶意 IP 量比较大且不固定，不容易配置 IP 黑名单，可以考虑配置频次控制。可以根据实际场景自定义配置频次控制功能，设定单位时间内单 IP 访问频次超过设定的阈值则触发阻断，通过频次控制功能，可以秒级阻断访问该网站的请求，提升网站的安全性。

（4）配置 URL 鉴权：URL 鉴权功能主要用于保护用户站点的资源不被非法站点下载盗用。通过防盗链方法添加 Referer 黑名单和白名单的方式可以解决一部分盗链问题，由于 Referer 内容可以伪造，所以 Referer 防盗链方式无法彻底保护站点资源。因此，可以采用 URL 鉴权方式保护源站资源更为安全有效。

（5）配置 UA 黑白名单：可以通过配置 User-Agent 黑名单和白名单来实现对访客身份的识别和过滤，从而限制访问 CDN 资源的用户，提升 CDN 的安全性。

安全加速

阿里云 CDN 是公共的加速服务，承载着成千上万的域名加速，默认不提供抗攻击能力。所以当用户域名遭受大量攻击时，CDN 系统会自动将对应域名切入沙箱，防止影响其他正常用户的加速服务，域名进入沙箱后，服务质量不再保证且无法恢复，因此做好防护工作十分重要。

（1）对于 CC 攻击，可以考虑配置 WAF 防护功能，使用 CDN WAF 功能不能解决恶意刷流量问题，但是可以防数据泄密，避免因黑客的注入入侵攻击，导致网站核心数据被拖库泄露；阻止木马上传网页篡改，保障网站的公信力；提供虚拟补丁，针对网站被曝光的最新漏洞，最大可能地提供快速修复规则。

（2）如果域名经常遭受攻击，可以根据自身业务需求考虑使用 SCDN 来做安全加速。SCDN（Secure Content Delivery Network），即拥有安全防护能力的 CDN 服务，提供稳定加速的同时，深度集成抗 DDoS、CC 攻击的防护功能。基于阿里云飞天平台的计算能力，使用深度学习的算法，智能预判攻击行为，通过智能的调度系统将 DDoS 恶意请求平滑切换至高防 IP 完成清洗，保护源站。

系统监控

CDN 被恶意刷流量导致流量异常时，会产生一定的经济损失，因此建议提前做好监控和安全防护工作。通常我们可以通过 CDN 的带宽封顶功能以及云监控的阈值报警功能来做监控。

一、带宽封顶

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设置的带宽最大值时，为了保护 CDN 域名安全，此时域名会自动下线，所有的请求会回到源站，CDN 将停止加速服务，避免异常流量给用户带来的异常消费。域名下线后，可以在控制台重新启用该域名，具体请参考带宽封顶。

> 注：因为触发带宽封顶以后域名会停止 CDN 加速，域名会解析到源站，因此相当于会把源站地址暴露出去，这里也带宽安全隐患，因此建议根据实际情况考虑使用启用该功能。

二、云监控报警规则

结合阿里云的云监控服务，通过设置云监控的报警功能，可以设置带宽峰值和下行流量的报警规则。当流量达到阈值时，系统自动通过电话、短信、邮件等方式通知用户，请及时采取措施。登录云监控控制台，依次选择【报警服务】>【报警规则】>【阈值报警】>【创建报警规则】，选择产品 CDN 以后去设定规则。创建针对 CDN 的报警规则，详情请参见创建阈值报警规则。

最佳实践——运维仔教你优化加速

简介：使用 CDN 加速以后还是存在访问慢的情况，如何去分析定位问题、优化网站速度、解决用户问题是一个十分重要的课题。本文介绍了 CDN 加速访问慢的分析思路，通过归纳的一些原因结合搜集的信息去进一步判断定位问题，帮助用户在遇到问题时有一个更清晰的思考方法论。同时介绍了一些典型的问题场景，结合这些问题场景可以更快速的去发现问题并优化。

问题背景

运维技术人员使用 CDN 加速以后发现还是有用户反馈访问慢的情况，而实际造成访问慢的影响因素很多，如何去分析定位问题、优化网站速度、解决用户问题是一个十分重要的课题。

分析思路

正所谓“工欲善其事，必先利其器”，在排查分析问题前，了解 CDN 的加速原理十分重要，它将有助于帮助你如何去思考和分析问题存在的可能原因。简单来说，CDN 主要是通过在现有网络中增加一层新的缓存节点，将网站服务器的资源发布到最接近用户的网络节点，使得用户侧客户端在请求时直接访问到就近的 CDN 节点并命中该资源，减少回源情况，提高网站访问速度。因此，造成访问慢的可能原因可以简单归纳为以下几个方向：

1. 客户端本地网络因素，比如客户端下行带宽不足、DNS 配置错误等
2. 客户端到 CDN 节点之间的网络不佳，网络延迟高
3. CDN 节点异常，响应速度慢
4. 资源内容比较大，导致下载比较耗时
5. CDN 回源到源站时，回源网络不佳

6. 源站本身响应速度慢

通过搜集一些问题现象和信息，我们可以进一步再继续往下分析，确定一下初步的排查方向，这也是一个非常重要的环节。

（1）可以先确认下是全网都存在访问慢的问题，还是只是个别用户访问慢，亦或是某一个地区、某一个运营商的用户访问慢。可以借助一些基调探测平台去探测，免费平台推荐 17 测。付费平台可以考虑“听云”、“博睿”等探测平台去探测，这些平台可以设定某一地区、某一运营商网络的探测机器去探测，精准性更高。

- 如果只是极个别用户访问不佳，那么可能跟用户侧的网络有强相关性，很可能就是用户侧的网络问题
- 异常用户是否有集中性，比如某市大量移动用户访问异常，而该市联通和电信用户访问正常。这种情况就有可能跟该地区的运营商网络有一定关联，可以使用一些基调工具，用该地区的一些探测机器去探测一下
- 如果全网用户都存在访问慢的问题，那就很有可能是源站响应问题或者是一些配置方面的问题了，因为几乎不可能同时所有的 CDN 节点或者所有地区的网络都处问题了。比如是不是加速区域选择的不对，是不是动态请求或者无法缓存的请求，源站响应慢，需要重点往这方面考虑了

（2）确认下访问慢或者异常的请求是否被 CDN 缓存了

- 如果是命中 CDN 缓存的请求，那么就不存在 CDN 回源了，因此 CDN 会直接把节点上的缓存数据返回给客户端，这种情况就和源站没什么关系了
- 如果是没有命中缓存，那么需要重点看是客户端到 CDN 慢了，还是源站响应慢了

衡量指标

使用 CDN 加速，除了通用的数据观测指标外，不同的场景下也有更具体的指标。观测这些指标，不仅可以帮助用户体验 CDN 加速的效果，也能观测自身业务使

用 CDN 的情况，帮助您更好地做出调整和决策。阿里云 CDN 官方帮助文档中心提供了 CDN 的衡量指标的介绍文档。

信息搜集

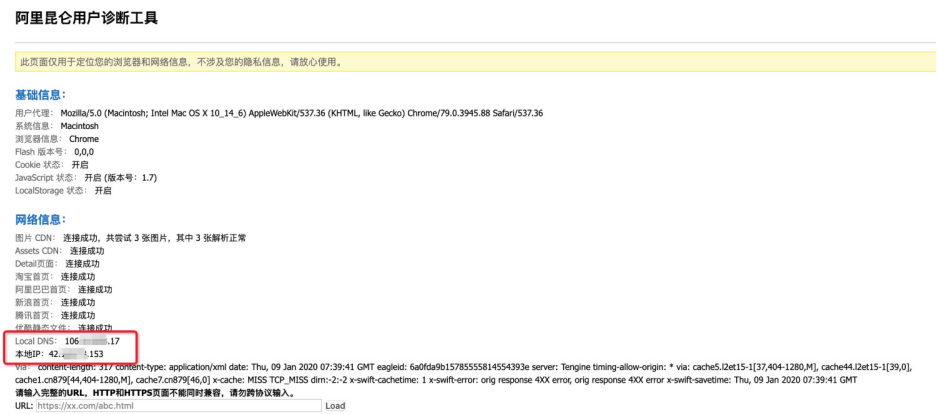
我们知道一次完整的 HTTP 请求需要经过 DNS 解析 -->TCP 建连 -->SSL 握手（HTTPS 需要 SSL 握手）--> 客户端发送请求 --> 服务端响应请求 的过程，了解 HTTP 请求的过程将有助于我们更深层次的去分析问题，因此在客户端侧搜集一些信息很有必要，通常可以搜集以下的几点信息

(1) 搜集客户端网络情况和 CDN 节点 IP

在客户端侧 ping 加速域名，确认是否正确解析到 CDN，以及客户端到 CDN 节点之间网络是否是通的，网络延迟如何。如果无法 ping 通，则还需要做一些链路诊断，具体可以参考这里的链路诊断方法。如果是手机侧，则需要借助一些第三方的应用来协助诊断，例如 Android 手机可以用“网络万用表”，iOS 可以用 iNetTools。

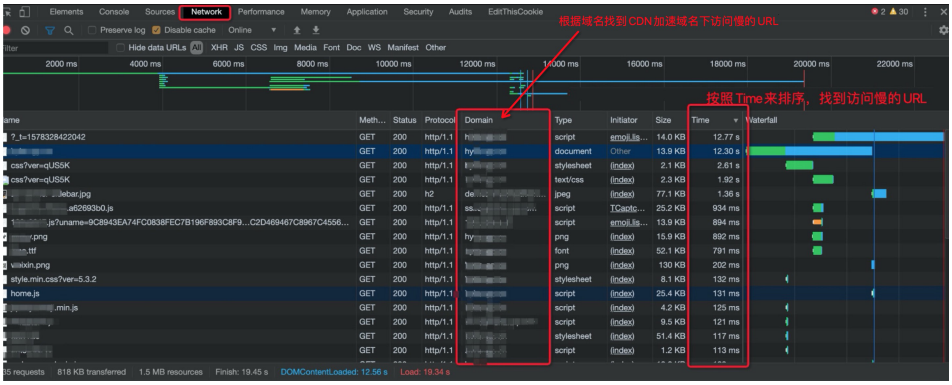
(2) 搜集客户端 IP 和 LocalDNS

CDN 的节点调度策略是根据客户端的 LocalDNS 来分配调度的，因此确认客户端的 LocalDNS 是否设置正确非常重要。可以通过客户端访问这个地址来获取客户端 IP 以及客户端 DNS：https://cdn.dns-detect.alicdn.com/https/doc.html



(3) 找到访问慢的 URL

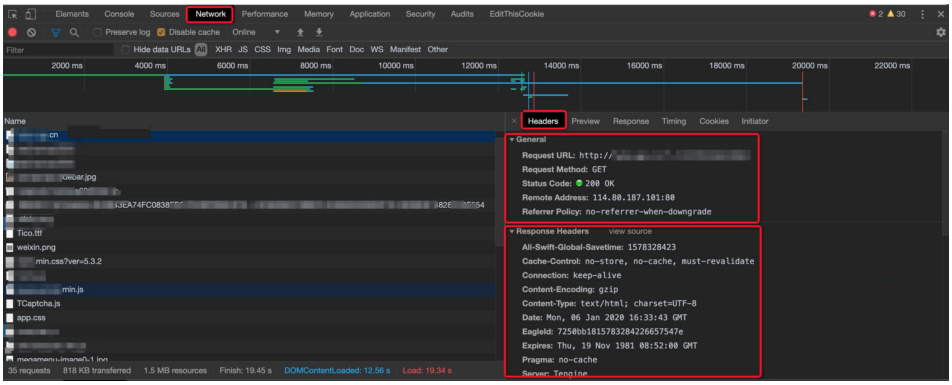
可以打开浏览器开发者模式，切换到 Network 标签页，输入 URL 以后可以在 Network 标签页下看到浏览器发出的所有的 HTTP 请求。点击“Time”选项按照时间来排序，看具体是哪些请求慢了，找到这个访问慢的 URL



特别注意：通常情况下一个网站加载的资源比较多，当然这里可能还有一些非 CDN 加速的一些 URL，有时候可能存在一些非 CDN 的资源访问慢，而 CDN 加速的资源都访问快，但是就是这些非 CDN 加速的资源加载慢导致整个网站响应速度变慢。因此根据 Time 排序来确认到底是哪些 URL 访问慢了很重要。

(4) 搜集 HTTP 请求的请求头和响应头

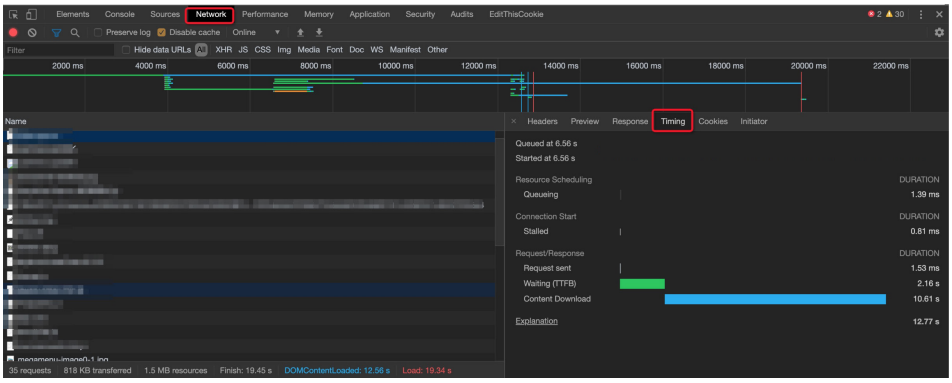
单击访问慢的 HTTP 请求 Name 值，在 Headers 标签下可以看到这次请求的 General、Response Headers 和 Request Headers 信息。通过请求头和响应头，我们可以了解这次请求是否是一个静态请求，是否命中了缓存等信息。



如果是手机 4G 慢，则需要在手机侧抓包来获取信息了，这个一般用户可能会有一些困难。可以考虑手机开热点，PC 连接热点，这样就可以在 PC 上搜集信息了。

(5) 搜集 HTTP 请求的 Timing 信息

Timing 标签中可以显示资源在整个请求生命周期过程中各部分时间花费信息。对于 Timing 里的信息介绍可以参考下这个介绍。



常见案例

在了解 CDN 的加速原理、HTTP 请求过程的基础下，结合问题现象做一个初步分析，然后根据搜集到的客户端侧的信息一起判断，基本已经可以大致的发现或定位

一些问题了。下面我们来介绍一些典型的问题案例。

案例一　客户端到 CDN 节点网络质量不佳

客户端 ping 加速域名网络延迟大，甚至丢包，这种情况需要搜集客户端的 IP、客户端的 DNS 以及 ping 截图、mtr 截图信息。因为 CDN 调度节点是通过客户端的 DNS 来分配调度的，根据客户端 IP、DNS 以及 CDN 节点可以判断调度是否异常，通过 ping 以及 mtr 截图可以看到网络延迟以及具体延迟在哪个网络链路节点。通常这类情况可能有以下几种情况：

(1) 加速区域设置错误

比如中国大陆的用户被解析到了海外的节点，或者海外用户被解析到国内。这种情况建议将加速区域设置为“全球加速”。

- 如果 CDN 的加速区域选择的是“仅中国大陆”，那么该域名的调度域就只有中国大陆的 CDN 节点，海外用户访问的时候也会调度到中国大陆的 CDN 节点
- 如果加速区域选择的是“全球（不包含中国大陆）”，那么该域名的调度域里就只有海外的 CDN 节点，中国大陆用户也会请求到海外的 CDN 节点

(2) 客户端 DNS 设置错误

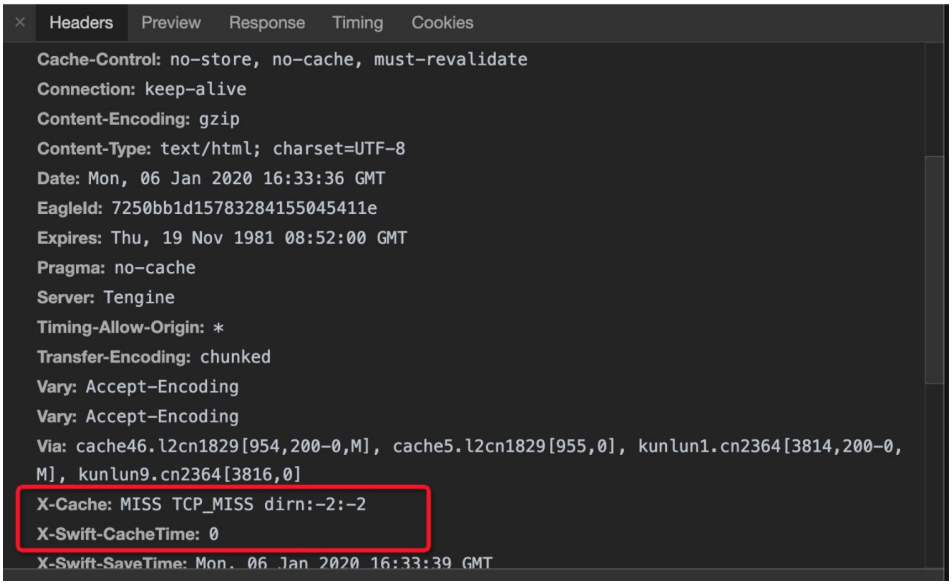
- 例如一个广东移动的用户，用了联通的 DNS，则会导致该用户被调度到联通 CDN 节点上，存在跨运营商的情况
- 例如一个广东移动的用户，用了哈尔滨移动的 DNS，则会导致该用户被调度到哈尔滨移动的 CDN 节点上，远距离调度拉长了网络链路。

这种场景需要用户侧修改使用对应所在地对应运营商的 DNS。

说明：如果加速区域和 DNS 设置正确，在 CDN 正确分配调度的情况下，网络质量还是差，那就需要搜集 traceroute 和 mtr 信息来进一步诊断了

案例二　缓存命中率低，频繁回源

CDN 在静态资源加速场景的应用，是将静态资源缓存在距离客户端最近的 CDN 节点上。用户访问该资源时，直接从缓存中获取资源，避免通过较长的链路回源。如果 CDN 缓存命中率低，则会导致源站压力大，静态资源访问效率低。因此，CDN 缓存命中率的高低直接影响用户体验，而保证较高的缓存命中率也成为了 CDN 的核心课题。可以针对导致 CDN 缓存命中率低的具体原因，选择对应的优化策略，来优化 CDN 的缓存命中率。我们可以通过 CDN 返回的 Response Header 里的 X-Cache 字段来判断是否命中缓存



X-Cache 字段：MISS 表示未命中缓存，是回源处理的；HIT 表示命中了 CDN 的缓存，直接读取的缓存数据。

X-Swift-CacheTime 字段：表示 CDN 节点上的允许缓存时间，即该文件可以在 CDN 节点上缓存多久，如果是 0 表示该请求无法缓存。

通常的一些现象和优化方案如下

(1) 首次访问资源慢，第二次访问正常

首次访问会比直接访问源站相对还慢些，因为第一次 CDN 节点没有缓存，要回源取数据。此情况推荐使用【预热】功能，将源站的内容主动预热到 CDN 节点上，用户首次访问可直接命中缓存，提高加载速度。

(2) 资源访问量较低，文件热度不够，CDN 收到请求较少无法有效命中缓存

CDN 节点作为所有使用 CDN 的用户公用的节点资源，因此 CDN 配置的缓存规则表示了该资源在 CDN 上的缓存最长时间，如果您的 CDN 加速域名流量较低，则可能提前从 CDN 节点的缓存中清除。即缓存按照热度属性采取末尾淘汰制。热度是指文件在节点上被访问的频率，文件热度不够，被提前剔除。

(3) 缓存配置不合理，缓存时间过短，CDN 节点频繁回源。

- 当 CDN 未配置缓存规则时，如果静态文件未返回响应头 Etag 和 Last-modified，则该静态文件不能缓存在 CDN 节点上。优化方案是需要源站配置这两个响应头，或者考虑在 CDN 侧配置缓存规则。
- 当 CDN 未配置缓存规则时，CDN 用的是默认缓存策略，缓存时间很短，最长不超过 3600 秒，因此容易造成频繁过期回源的情况，建议可以根据业务情况到 CDN 侧设置合理的缓存时间。
- 当源站配置了一些强制不缓存的 Cache-Control 的响应头时，即使您配置了缓存规则，CDN 也不会对该资源进行缓存，因为这些响应头在 CDN 缓存规则中的优先级较高。以下有 "s-maxage=0"、"max-age=0"、"no-cache"、"no-store"、"private"、"Pragma: no-cache" 中的任一种，都会导致 CDN 无法缓存，需要源站侧去修改这些响应头，比如修改成 Public 等可以被缓存的响应头。参考文档：设置 Nginx 缓存策略

(4) URL 带可变参数

访问资源的 URL 带参，并且参数不断变化，当用不同的 URL 去访问 CDN 的

时候，CDN 会认为这是一个新请求（即便这两个不同的 URL 其实是访问到了同一个文件，并且该文件已经缓存在节点上），还是会回源去拉取所请求的内容，建议开启【过滤参数】功能。

（5）大文件 Range 回源

对于一些大文件，建议开启 Range 回源来优化回源

案例三　动态请求访问慢

如果访问慢的请求是一个动态请求，当客户端访问这些动态内容时，每次都需要访问用户的服务器，由服务器动态生成实时的数据并返回给客户端。这种场景下，CDN 无法缓存实时变化的动态内容，因此 CDN 的缓存加速不适用于加速动态内容。对于动态内容请求，CDN 节点只能转发回源站服务器，没有加速效果。如果用户的网站或 App 应用有较多动态内容，例如需要对各种 API 接口进行加速，可以考虑如下方案

(1) 做动静分离，静态资源用 CDN 域名来加速，动态请求用另一个直接解析到源站的域名来访问

(2) 考虑使用全站加速来加速动态请求。不过要注意的是，全站加速对于动态请求的加速是通过阿里云的路由优化、传输优化等动态加速技术以最快的速度访问您的服务器源站获取数据，是一个四层链路的优化，如果源站服务器本身响应速度就很慢，那这种情况还是需要优化源站。

案例四　源站响应慢

访问慢的请求是一个不缓存的请求，或者是一个动态请求，CDN 都是回源处理的，如果源站的响应速度非常慢，则会导致最终响应的速度慢。这种情况可以直接本地绑定 Host 到源站去测试源站的响应速度。这种情况一般可能有以下情况：

(1) 源站性能限制，本身处理速度比较慢，比如源站的带宽、CPU 等达到瓶颈，或

者源站程序处理速度慢等，需要考虑优化源站。如果是性能不足则需要对源站扩容。

(2) 源站侧网络比较差，或者源站涉及到跨境链路，比如中国大陆用户请求 CDN，而源站在境外。由于 CDN 回源到源站也是走的公网，如果涉及到跨境链路的话确实可能会受到一些影响，因为跨境链路涉及到不同的运营商、境外运营商，而且需要走国际互联网出口，这些 CDN 侧和源站侧都不可控，CDN 单方面优化的空间很小，建议部署双源站（境外 + 境内）调整架构来优化。

案例五　网站首页加载慢

我们知道打开一个网站，实际的过程是浏览器发起一个请求以后服务端返回一个 html（也就是首页的请求），浏览器拿到首页请求以后解析 html 以后才会继续去请求网页里的图片、css、Js 等资源。如果首页是一个动态请求或者是不缓存的请求，会导致每次请求首页的时候，CDN 都是回源处理的。如果源站响应慢就会导致最终首页加载慢，该请求在 Network 下 Pending 状态持续时间比较久。具体是否命中缓存可以参考本文案例二的介绍。

这种首页不缓存的请求访问慢的场景，造成的现象就是首页请求一直 Pending，等到首页请求到了以后后续的静态资源很快都加载出来了。

> 如果是首页慢的情况，这种情况用“站长工具”、“17 测”等平台去验证 CDN 的加速效果结果可能不准确。因为探测地址如果填写是 http://{ 网站域名 }，那么探测平台实际探测的就是首页的地址，并没有去探测网站里的一些静态文件的资源。如果探测 URL 输入的是一个具体的静态资源的 URL，那才可以验证加速效果。

案例六　网站加载的内容比较大

如果网站加载的资源比较大，可以通过设置加速域名的性能优化功能，缩小访问文件的体积，提升加速效率和页面可读性。目前智能压缩支持的内容格式：text/html、text/xml、text/plain、text/css、application/javascript、application/x-ja-

vascript、application/rss+xml、text/javascript、image/tiff、image/svg+xml、application/json、application/xmltext

案例七　某地区某运营商用户访问慢

有一些问题场景，客户端有一些共性。比如某一个时间段，某市移动用户有大量用户反馈访问慢或者异常，而联通电信用户都正常。这类问题就很有可能跟当地运营商网络或者该地区请求到的 CDN 节点有关联，通常的排查方法就是在用户侧去搜集 ping 信息，先确认客户端和 CDN 节点之间的网络延迟情况，另外根据用户请求到的 CDN 节点 IP 可以绑定到该 CDN 节点去测试，测试方法跟绑定到源站去测试类似，把 IP 地址换成 CDN 节点的 IP 即可。绑定节点测试可以先验证下这个节点本身是否确实存在响应慢的情况。如果响应慢，再查看该请求是否命中缓存、加载的资源是否过大等，结合前面的案例进一步分析。如果无法定位也可以通过提交工单等方式联系阿里云。

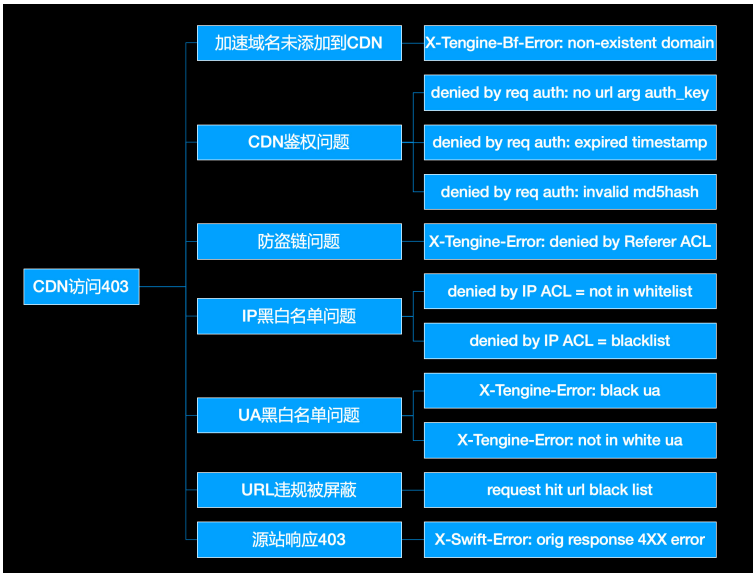
CDN 访问异常排查

403 错误怎么办？七种原因帮你精准定位

简介：使用阿里云 CDN 加速站点访问后出现 403 错误，通常情况下可能是由域名配置、CDN 安全策略以及源站响应 403 导致。CDN 的 Response Headers 返回的错误字段明确标识了该 403 是什么原因引起的，本文详细介绍了 CDN 常见的引发 403 错误的问题场景。

概述

CDN 访问出现 403 通常情况下可能是由以下几种情况导致的，我们可以打开浏览器开发者模式，切换到 Netwrok 标签页以后重新请求异常的 URL，复现 403 的问题，然后在 Headers 下查看 CDN 返回的 Response Header，通过这个信息我们可以判断这个 403 错误是什么原因引起的。本文会对这些情况做具体讲解，以下是本文概图。

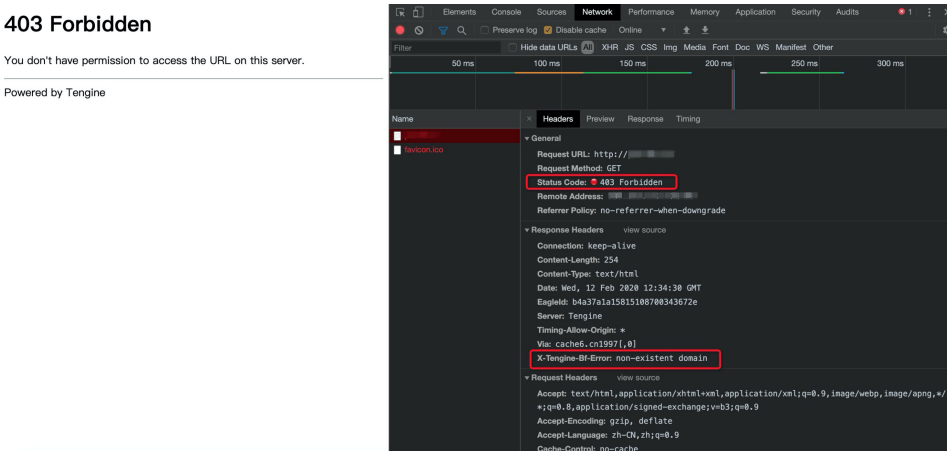


一、加速域名未添加到 CDN

用户在 CDN 上添加了主域名 test.com，对应的 CNAME 是 test.com.alikunlun.com，然后用户的一些其他的二级域名比如 a.test.com、b.test.com 等域名并没有添加到 CDN 上，但是却直接将这些二级域名解析到 test.com.alikunlun.com，这种情况会导致 CDN 响应 403，具体报错如下。

> 注意：主域名的 CNAME 不能被二级域名使用，如果需要加速这些二级域名，需要把二级域名单独都添加到 CDN 上，并解析到对应的 CNAME 地址上。或者考虑使用泛域名的方式，泛域名的 CNAME 是可以被二级域名使用的。

```
X-Tengine-Bf-Error: non-existent domain
```

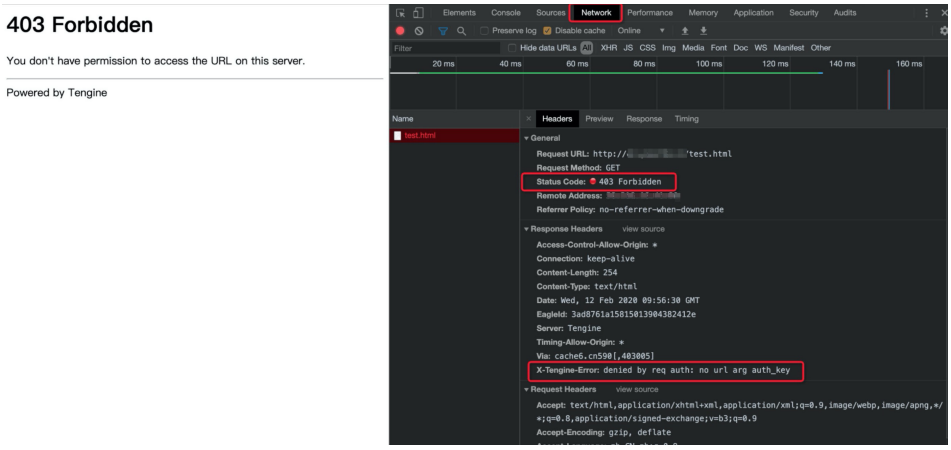


二、CDN 鉴权问题

CDN 鉴权问题通常表现在没有带鉴权参数、鉴权过期、鉴权计算错误，需要根据 URL 鉴权的文档了解鉴权的原理然后去进一步排查和解决。

1. **CDN 开启了鉴权，但是实际的访问 URL 里没有带鉴权参数，导致报错如下**

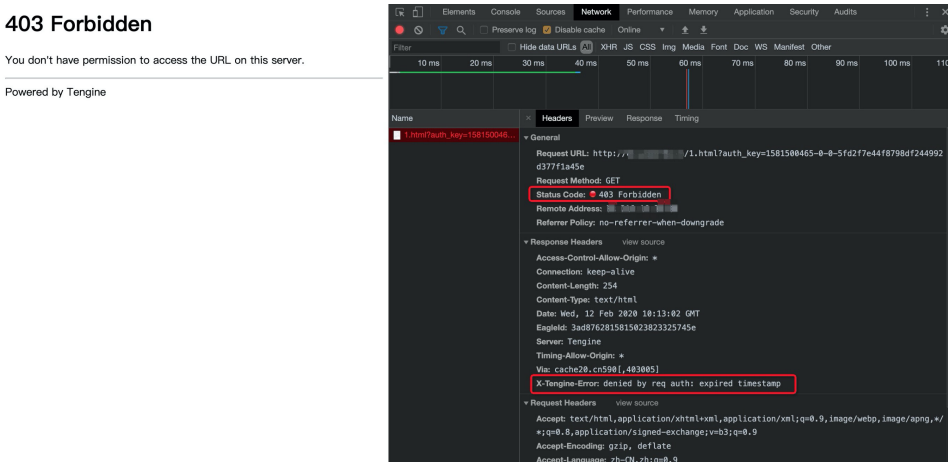
```
X-Tengine-Error:denied by req auth: no url arg auth_key
```



2. 鉴权参数过期

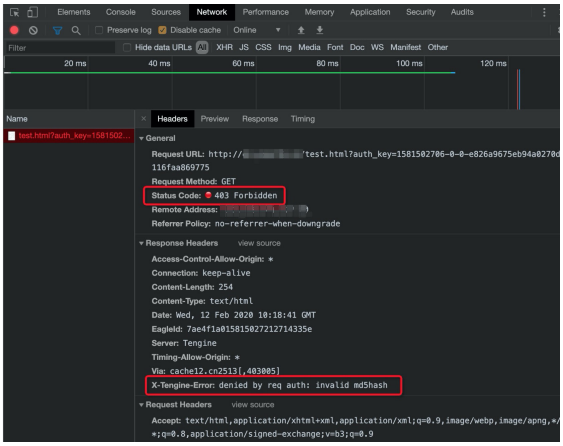
CDN 开了鉴权，并且 URL 带了鉴权参数，但是鉴权参数过期

```
X-Tengine-Error: denied by req auth: expired timestamp
```



3. 鉴权参数的 md5 计算不正确

```
X-Tengine-Error: denied by req auth: invalid md5hash
```



解决方案：

1. 如果不需要 CDN 的鉴权功能，可以在 CDN 控制台关闭鉴权。
2. 如果鉴权过期，请重新生成鉴权 URL。
3. 如果鉴权的 md5 计算不正确，建议先用 CDN 控制台的地址生成器生成 URL 来对比自己的鉴权代码，也可以参考官方帮助文档提供的鉴权示例代码。

三、防盗链问题

开启了防盗链功能，但是实际 Request Headers 请求头里的 Referer 头不符合防盗链规则导致失败，因防盗链问题导致的 403，在 CDN 的 Response headers 里的 X-Tengine-Error 会返回 denied by Referer ACL。防盗链配置请参考配置文档。

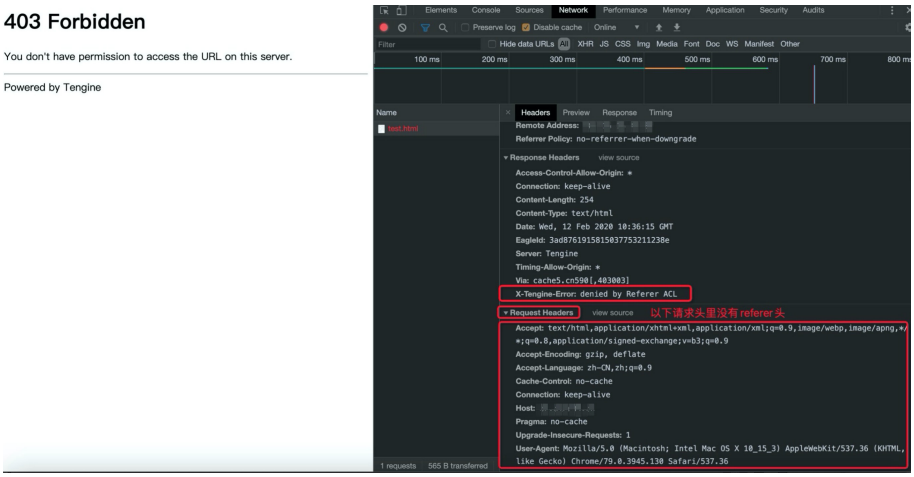
```
X-Tengine-Error: denied by Referer ACL
```

Referer 防盗链类型如下：

- 黑名单：黑名单内的域名均无法访问当前的资源。
- 白名单：只有白名单内的域名能访问当前资源，白名单以外的域名均无法访问当前的资源。

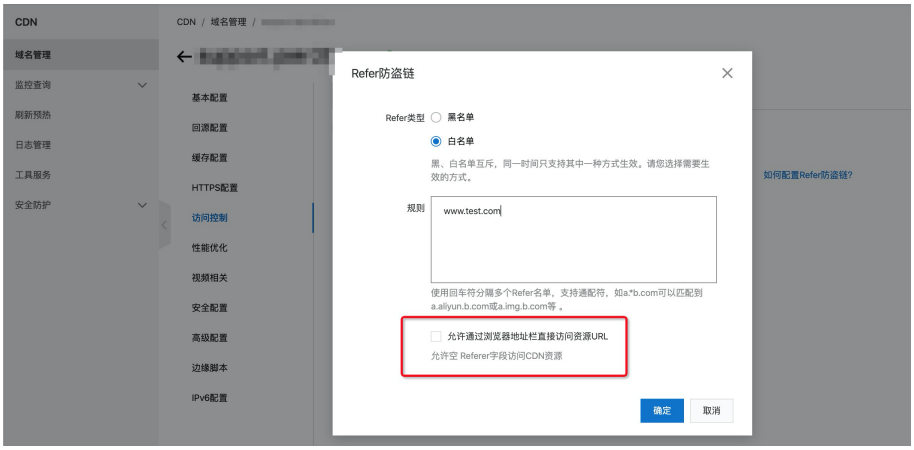
黑名单和白名单互斥，同一时间只支持其中一种方式生效。

1. **请求头里没有带 referer 头，也就是说该 HTTP 请求是一次空 referer 的请求的。而 CDN 控制台又设置了不允许空 referer，因此该请求会被 403，参考如下案例。**

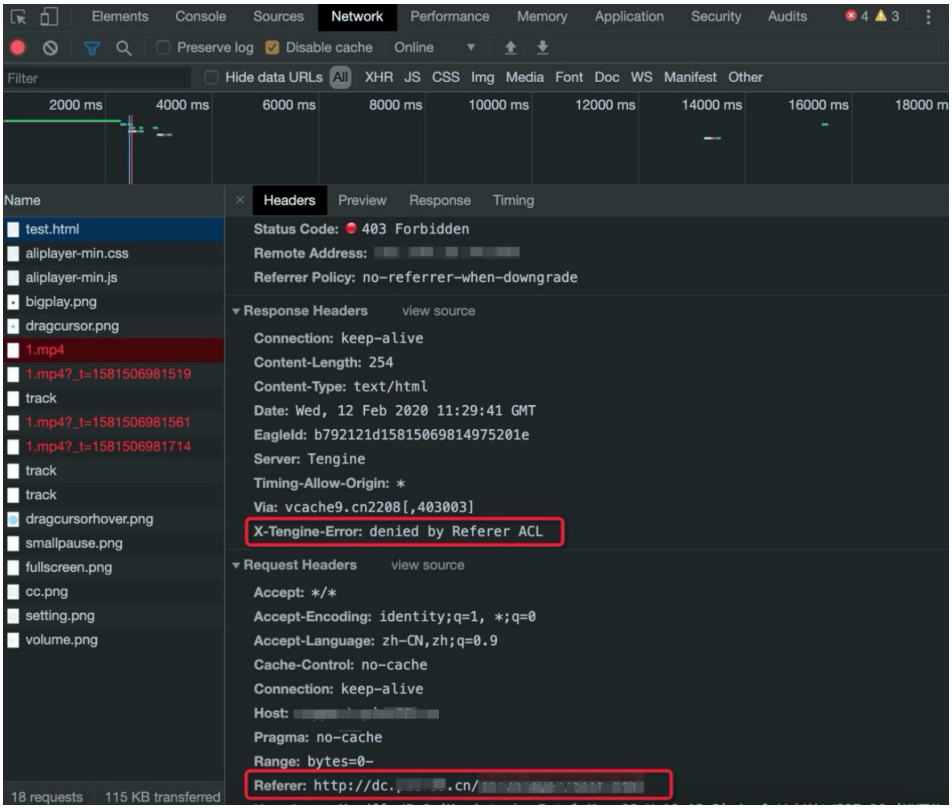


解决方案：如果希望允许空 referer 的请求，可以登录 CDN 控制台，单击对应的加速域名 管理，选择 访问控制 > Refer 防盗链 > 修改配置，，勾选“允许通过浏览器地址栏直接访问资源 URL”。

注：如将防盗链设置不允许为空 Referer 访问，这样操作，有被盗链的风险。



2. **设置了防盗链白名单，但是实际请求时，请求头里的 referer 头不在白名单里。例如如下案例，设置的白名单是 www.test.com，但是实际访问的时候，请求头里的 referer 头是 dc.xxx.cn，未在白名单里，因此 403。**

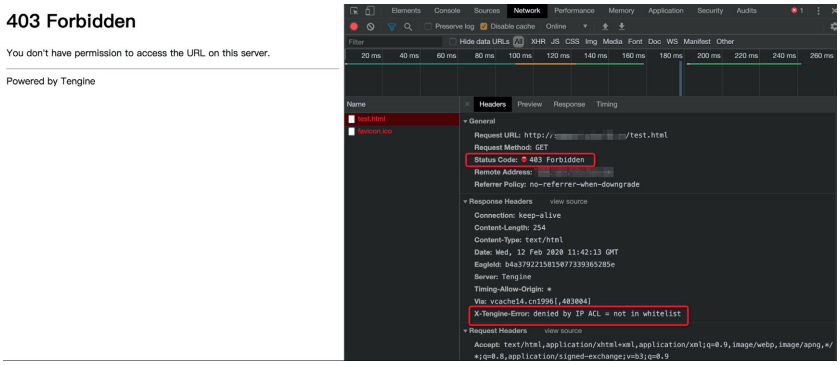


四、IP 黑白名单问题

在 CDN 控制台配置了 IP 黑白名单，实际访问的 IP 不符合配置规则，导致出现 403。

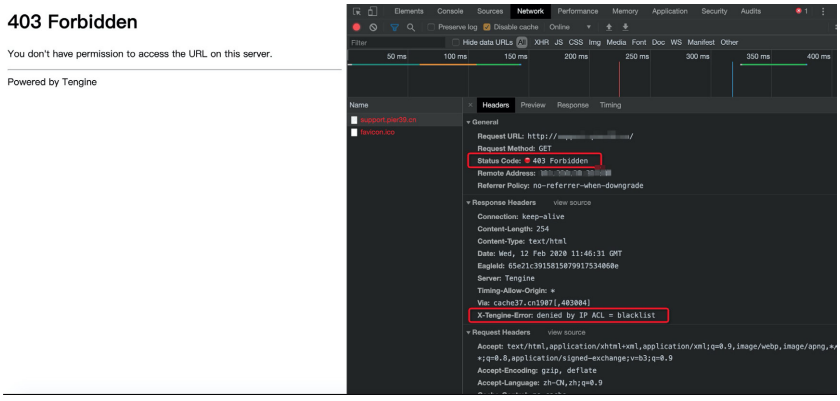
1. **配置了 IP 白名单，实际访问的客户端 IP 不在 IP 白名单里，导致 403，具体报错如下**

```
X-Tengine-Error: denied by IP ACL = not in whitelist
```



2. **配置了 IP 黑名单，实际访问的客户端 IP 在 IP 黑名单里，导致 403，具体报错如下**

```
X-Tengine-Error: denied by IP ACL = blacklist
```



常见问题

- 问：为什么配置了 IP 黑名单，还是可以正常访问，响应 200，而不是 403？

答：这种情况一般都是客户端真实出口 IP 跟 IP 黑名单里配置的 IP 不一致导致的。建议获取客户端真实出口 IP，可以通过 IP 工具查询；也可以通过下载 CDN 的日志，从 CDN 的日志去查找这条请求，CDN 的日志里记录了客户端 IP。

- 问：发现恶意请求的情况，把恶意请求的客户端 IP 配置到黑名单了，为什么还是不断有请求 CDN？

答：CDN 作为一个服务端，无法控制客户端不请求 CDN，CDN 能做的是当恶意请求到 CDN 的时候，CDN 根据配置的安全规则拒绝不合法的请求，以 403 的形式拒绝访问。

五、UA 黑白名单问题

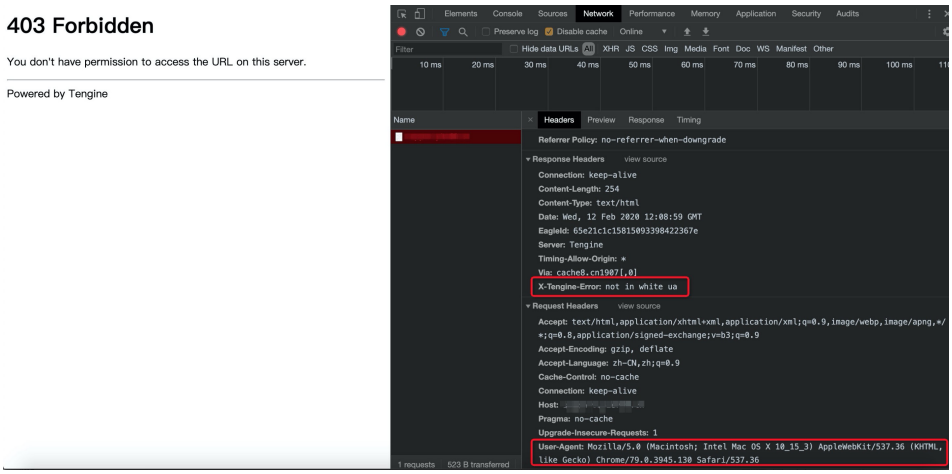
配置了 UA 黑白名单，User-Agent 名单类型如下：

- 黑名单：黑名单内的 User-Agent 字段均无法访问当前资源。
- 白名单：只有白名单内的 User-Agent 字段能访问当前资源，白名单以外的 User-Agent 字段均无法访问当前资源。

黑名单和白名单互斥，同一时间只支持其中一种方式生效。

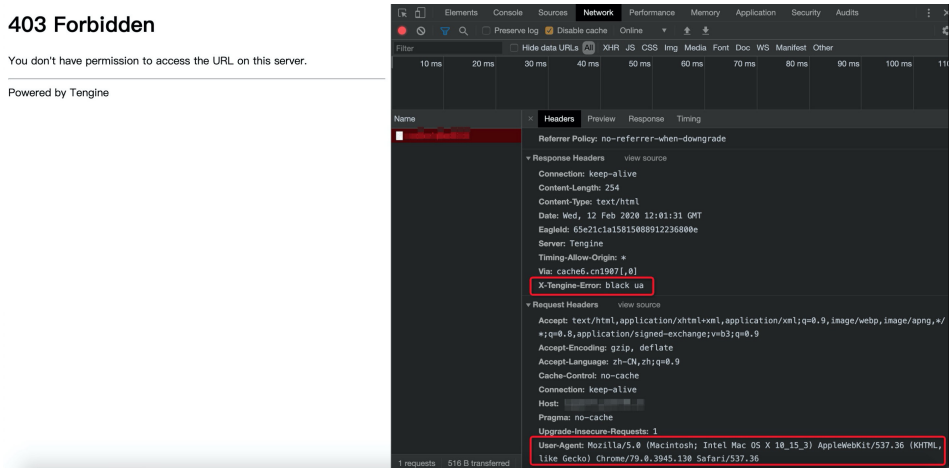
1. **配置了 UA 黑名单，客户端 UA 命中了黑名单规则，报错如下**

```
X-Tengine-Error: black ua
```



2. **配置了 UA 白名单，客户端 UA 不在 UA 白名单列表里，报错如下**

```
X-Tengine-Error: not in white ua
```

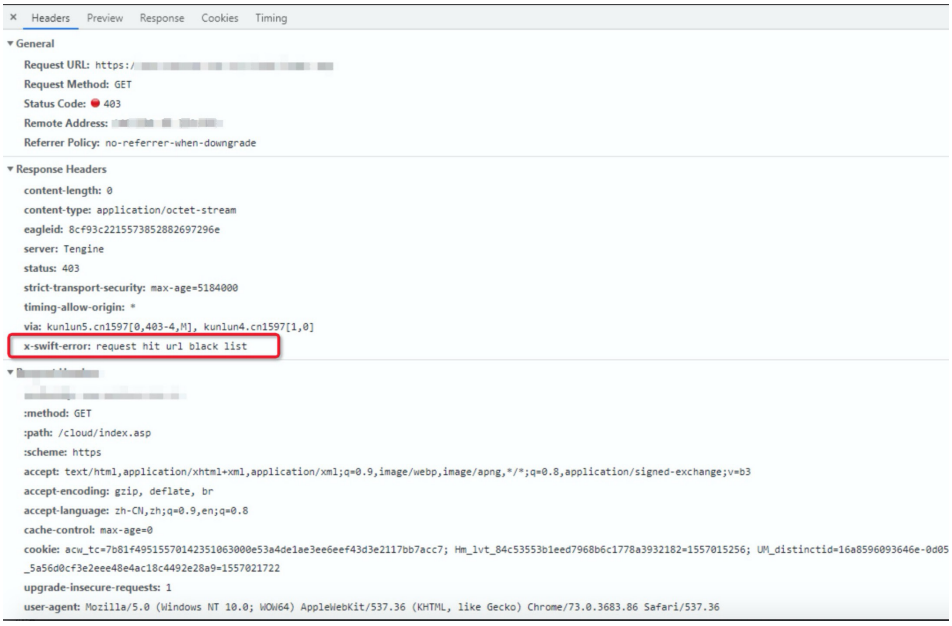


六、URL 违规被屏蔽

403 的 URL 涉及违法不良信息，违反了相关服务协议和《互联网信息服务管理办法》第十五条规定，这种情况下违法 URL 会被 CDN 做屏蔽访问处理。通常这种

情况会收到邮件或短信通知，请注意确保 CDN 加速的内容是合法的内容。报错如下

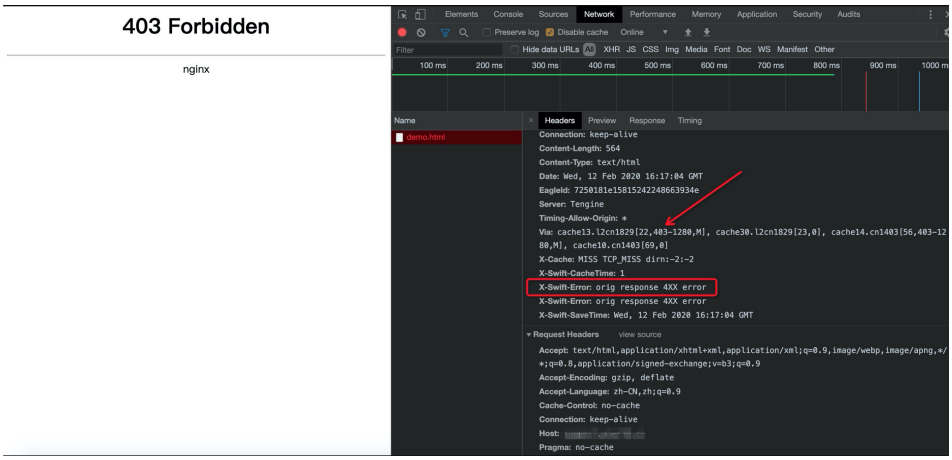
```
x-swift-error:request hit url black list
```



七、源站响应 403

源站响应了 403 给 CDN，CDN 再把 403 响应给客户端。源站响应的 403 会报错如下

```
X-Swift-Error: orig response 4XX error
```



- **源站是用户服务器**

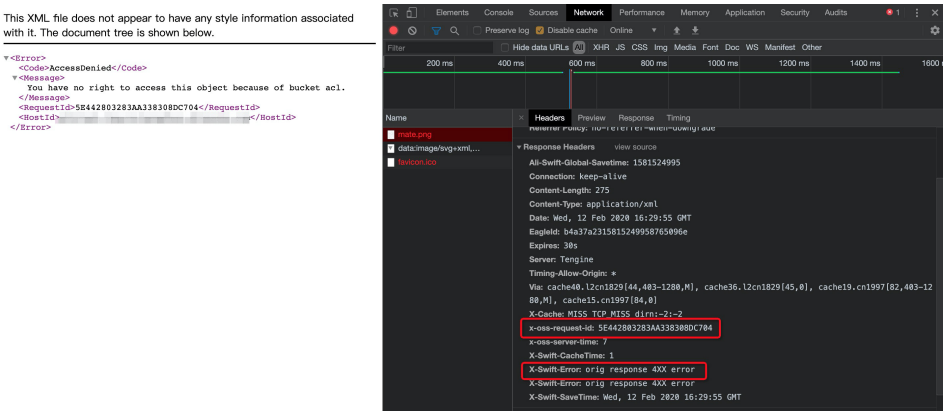
可以绑定 Host 到源站访问测试是否一样存在 403 的情况，如果源站就有 403 的情况，需要先解决源站的 403 问题。另外还有一点需要注意，CDN 的回源 Host 配置错误也可能导致 403 错误。回源 HOST 跟源站的区别就是，源站决定了回源时请求到的具体 IP 地址，而回源 HOST 决定了回源请求访问到该 IP 地址上的具体站点。

- **源站是阿里云 OSS**

如果源 Bucket 的访问权限是私有权限，但是访问 URL 里没有带上 OSS 的私有签名参数（Signature、Expires、OSSAccessKeyId），就会导致 CDN 回源请求 OSS 的时候通不过 OSS 的鉴权导致 403，报错如下

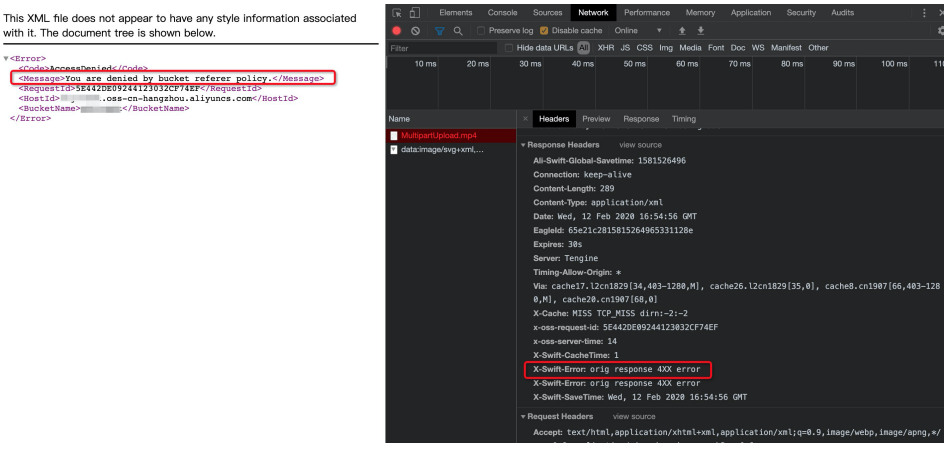
```
You have no right to access this object because of bucket acl 。
```

这种情况建议开启 CDN 的阿里云 OSS 私有 Bucket 回源授权功能。



如果出现如下错误，说明是 OSS 防盗链鉴权返回的 403，则需要检查 OSS 的防盗链设置。

```
You are denied by bucket referer policy
```



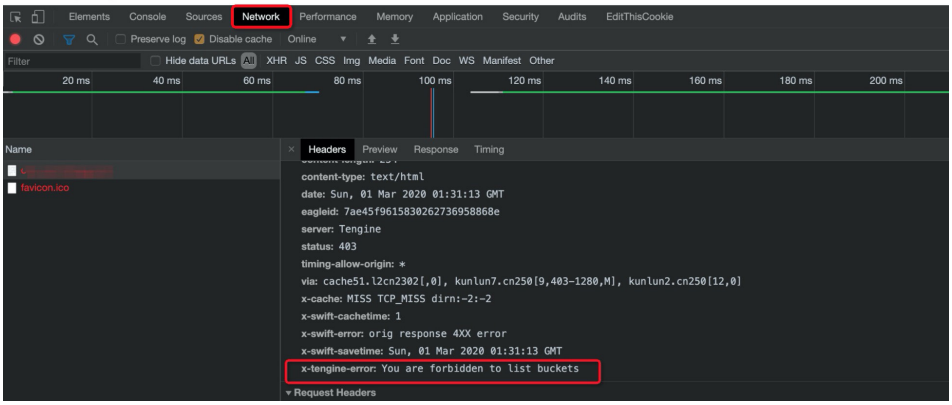
如果出现如下错误，说明是开启私有 Bucket 回源授权的情况下访问了 OSS 的静态首页，目前 CDN 的私有 Bucket 回源功能和 OSS 的静态网站托管功能冲突，无法一起使用。

```
You are forbidden to list buckets
```

403 Forbidden

You don't have permission to access the URL on this server.

Powered by Tengine

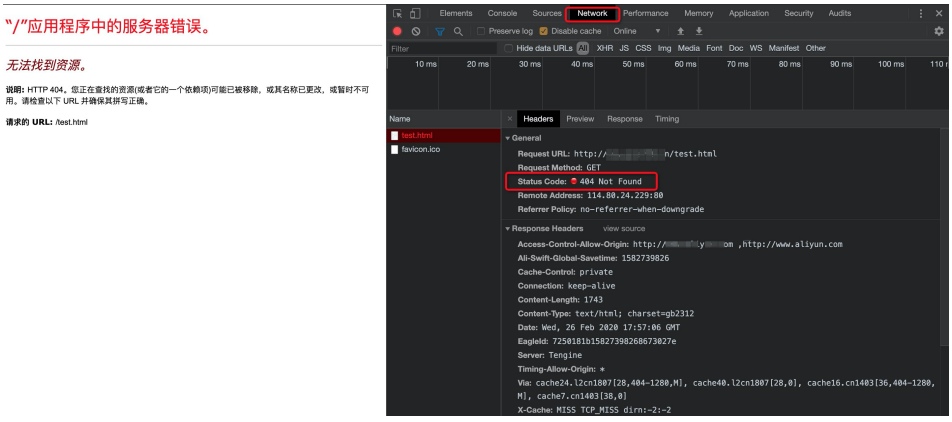


三招快速定位 404 错误

简介：本文详细介绍了使用 CDN 以后出现 404 错误的常见问题。

问题现象

通过 CDN 加速以后，出现访问 404 not found 的情况。



问题原因

如果是通过监控发现 404 的错误，则可以通过 CDN 的日志去确认出现 404 的 URL。如果已知 404 的 URL，则可以绑定源站去测试确认资源是否存在，通常可能有以下几种原因。

源站资源不存在

请参考 CDN 加速域名绑定 Host 到源站测试方法 绑定到源站去测试访问 404 的 URL，确认源站是否返回 404，如果源站返回 404，请确认源站的资源是否存在。

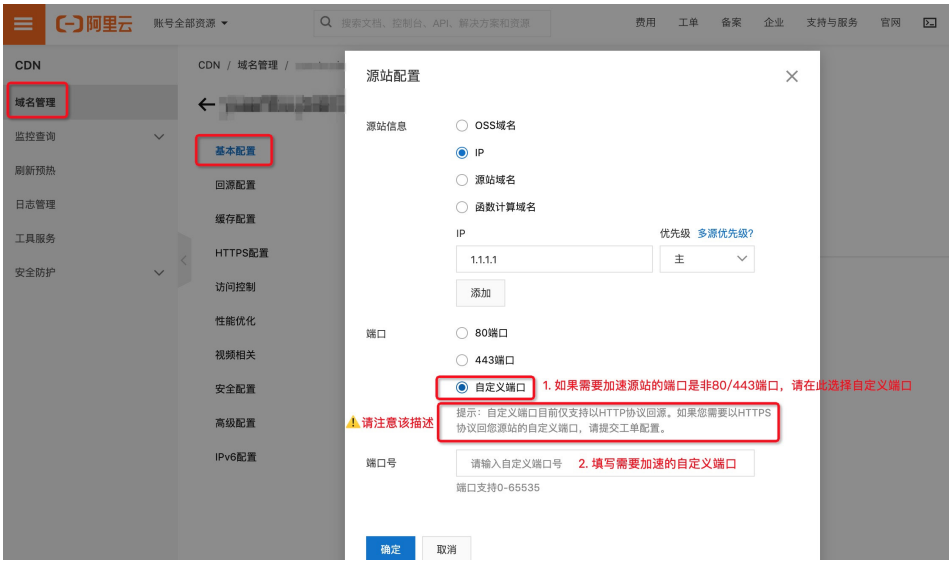
回源 Host 配置错误

回源 HOST 是指 CDN 节点在回源过程中，在源站访问的站点域名。如果您需要自定义 CDN 节点回源时需要访问的具体服务器域名，则需要配置回源 HOST 的域名类型。回源 HOST 可选域名类型包括：加速域名、源站域名和自定义域名。如果回源 Host 配置不对，源站无法识别该回源 Host，源站也会响应 404。特别注意，源站和回源 HO− ST 的区别如下：

- 源站：源站决定了回源时请求到的具体 IP 地址。
- 回源 HOST：回源 HOST 决定了回源请求访问到该 IP 地址上的具体站点。

回源端口配置错误

该问题通常发生在源站的端口是非 80/443 端口，例如源站 http 服务的端口是 8080，则在 CDN 上配置回源端口时，需要配置自定义回源端口为 8080，且需要关闭协议跟随回源功能，否则自定义端口无法生效。



502/503/504 错误排查攻略

简介：当客户使用阿里云 CDN 加速站点访问后，客户端的请求将首先发送到 CDN 的 L1 节点（一级节点），再回源到 L2 节点（二级节点），然后再回源到源站。因此如果访问过程中出现问题就可能涉及到多级网络链路的问题。当 CDN 回源源站异常失败时就会出现 5xx 的错误，主要包括 502 Bad Gateway、503 Service Temporarily Unavailable、504 Gateway Time-out。很多情况下是因为一些细节被忽略了导致了错误的发生，本文介绍了一些常见的引发 5xx 错误的问题场景。

问题分析

通过阿里云 CDN 访问出现 5xx 错误，在响应的 Response Header 里的 X-Swift-Error 字段会有相关的错误，例如 forward retry timeout 或者 orig response 5xx error，同时用 curl 测试或者浏览器 Network 开发者模式下可以看这个请求消耗的时间。

遇到这类问题可以初步分析，如果是全局都是 5xx 的错误，通常就是 CDN 的配置错误或者源站错误导致的，比如“**源站不通或源站域名无法解析**”、“**CDN 配置了 HTTPS 回源，但源站不支持 HTTPS**”、“**源站开启了 SNI 校验，但是 CDN 没有开启回源 SNI**”等，这些配置会导致 CDN 必然会回源失败，响应 5xx 错误。如果只是部分区域有问题，或者问题是偶发出现的，那么有可能跟部分地区回源网络或其他因素有关。比如源站的安全策略把部分 CDN 节点 IP 屏蔽了，就会导致对应区域的用户访问 CDN 异常；比如源站侧网络不稳定或者回源跨境链路不稳定或者源站动态接口响应速度不稳定，就会有偶发出现 5xx 的情况。


```
[root@iZbp1cd2l7ux0klkmxlmlbZ ~]# curl -voa http://s[redacted]cn
* About to connect() to [redacted] port 80 (#0)
*   Trying 61.164.147.118... connected
* Connected to [redacted] (61.164.147.118) port 80 (#0)
> GET / HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.44 zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: [redacted]
> Accept: */*
>
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:--  0:00:36 --:--:--     0< HTTP/1.1 503 Service Temporarily Unavailable
< Server: Tengine
< Content-Type: application/octet-stream
< Content-Length: 0
< Connection: keep-alive
< Age: 0
< X-Cache: MISS TCP_MISS dirn:-2:-2
< X-Swift-SaveTime: Thu, 01 Jan 1970 00:00:00 GMT
< X-Swift-CacheTime: 1580994646
< Via: vcache10.cn1898[0,503-274,M], vcache11.cn1898[36000,608,504001]
< X-Swift-Error: forward retry timeout
< Timing-Allow-Origin: *
< EagleId: 3da4931f15809946098452981e
<
  0     0    0     0    0     0      0      0 --:--:--  0:00:36 --:--:--     0* Connection #0 to host support.pier39.cn left intact

* Closing connection #0
```

nginx 对于 502/503/504 的错误解释如下

```
nginx中502/503/504通用意义
502
#define NGX_HTTP_BAD_GATEWAY 502 # connection reset, bad gateway (HTTP/1.1 502 Bad Gateway)
> (kevent() reported that connect() failed (61: Connection refused))
> (connect() failed (111: Connection refused) while connecting to upstream)
503
#define NGX_HTTP_SERVICE_UNAVAILABLE 503 # limit qps/limit conn/ sysguard
504
#define NGX_HTTP_GATEWAY_TIME_OUT 504 # connection/read timeout (HTTP/1.1 504 Gateway Time-out)
```

详细信息

一、源站不通或源站域名无法解析

CDN 都是公网上的节点，CDN 配置的源站必须要公网可达。如果配置的源站 IP 公网不可达、端口不通或者源站域名没有解析，则会导致 CDN 回源请求源站失败，报错 5xx。

常见的几种异常情况如下：

(1) 源站网络不通，测试无法 ping 通源站 IP。ping 测试命令：ping 源站 IP

(2) 源站端口不通或源站直接响应 5xx 错误。例如以下案例，telnet 端口报错

Connection timed out

i) 如果源站端口配置的是 80，则测试 80 端口是否通：telnet 源站 IP 80

ii) 如果源站端口配置的是 443，则测试 443 端口是否通。如果源站端口配置的是自定义端口，则测试自定义端口是否通。

iii) 可以在 CDN 控制台获取配置的源站地址和端口，然后本地 host 绑定到源站，固定源站做七层测试，查看是否是源站直接无响应或源站直接响应 5xx，具体可以参考这里。

(3) 源站配置的是域名，但是源站域名未配置解析，会导致 CDN 请求源站失败。可以用 ping 和 nslookup 命令检查源站域名的解析是否正常。例如以下案例，配置源站域名 www.a.com，ping www.a.com 报错 unknown host，nslookup unknown host 报错 server can't find www.a.com: NXDOMAIN，表示域名未解析。


```
[root@iZbp1cd2l7ux0klkmxlmlbZ ~]# ping www.a.com
ping: unknown host www.a.com
[root@iZbp1cd2l7ux0klkmxlmlbZ ~]#
[root@iZbp1cd2l7ux0klkmxlmlbZ ~]# nslookup www.a.com
Server:         223.5.5.5
Address:        223.5.5.5#53

** server can't find www.a.com: NXDOMAIN
```

二、CDN 配置了 HTTPS 回源，但源站不支持 HTTPS

（1）源站端口配置成 443，但源站不支持 HTTPS

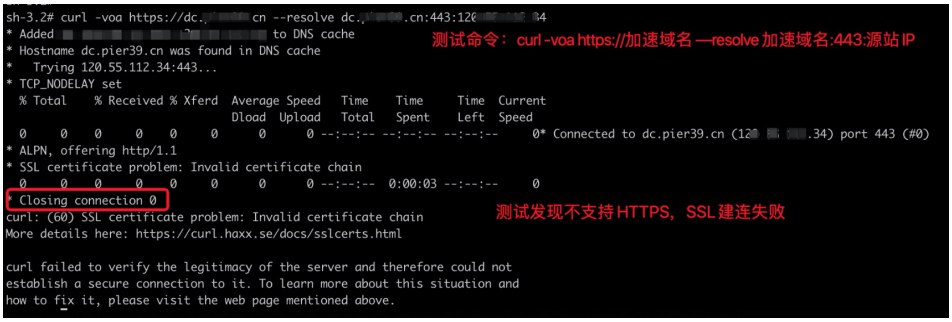
在 CDN 控制台的源站配置界面，如果源站端口配置成 443，则 CDN 回源的时候是 HTTPS 回源到源站的 443 端口。源站需要开放 443 端口，且配置 HTTPS 证书。如果源站不支持 HTTPS 访问，则 CDN 回源失败，报错 5xx。对于这种情况，可以把回源端口改成 80；如果业务需要 443 回源的话，那么需要在源站配置 HTTPS 证书。

（2）CDN 配置了协议跟随回源，但是源站不支持 HTTPS 访问。

协议跟随回源如果设置成“HTTPS”，则 CDN 是以 HTTPS 回源；协议跟随回

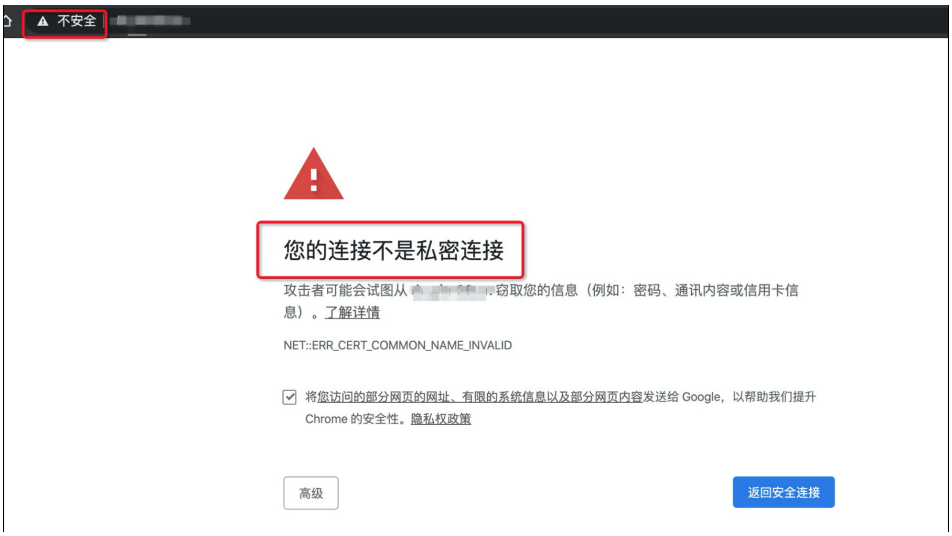
源如果设置成“跟随”，则当客户端是 HTTPS 访问的时候，CDN 是 HTTPS 回源。源站不支持 HTTPS 的情况下，会出现访问失败。对于这种情况，需要关闭协议跟随回源功能，或设置为 HTTP 回源。

可以通过 curl 命令直接绑定到源站去测试，测试命令：curl –voa http://dc.xxx.cn ––resolve dc.xxx.cn:443:a.a.a.a (dc.xxx.cn 是 CDN 加速域名 ,a.a.a.a 是源站 IP)



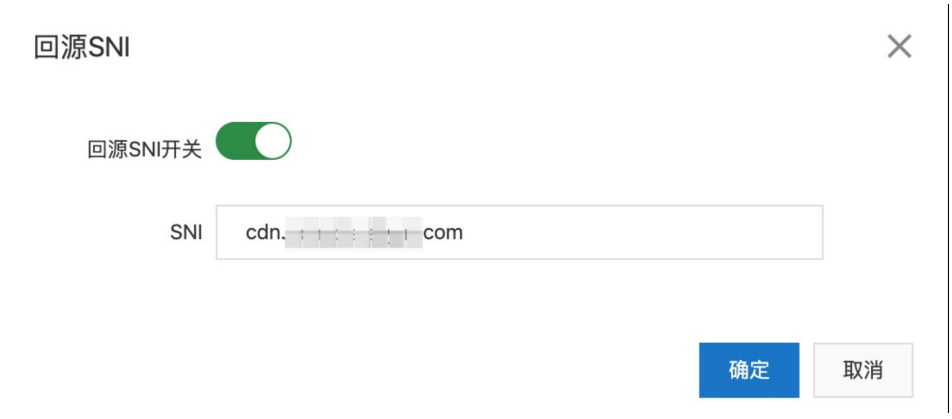

```
sh-3.2# curl -voa https://dc.  cn --resolve dc.  .cn:443:12  4
* Added  to DNS cache
* Hostname dc.pier39.cn was found in DNS cache
*   Trying 120.55.112.34:443...
* TCP_NODELAY set
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0* Connected to dc.pier39.cn (12  .34) port 443 (#0)
* ALPN, offering http/1.1
* SSL certificate problem: Invalid certificate chain
  0     0    0     0    0     0      0      0 --:--:--  0:00:03 --:--:--     0
* Closing connection 0
curl: (60) SSL certificate problem: Invalid certificate chain
More details here: https://curl.haxx.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.
```

也可以修改本地 etc/hosts 文件绑定到源站，用浏览器发起 HTTPS 访问。以下案例报错“您的连接不是私密连接”，则表示不支持 HTTPS 访问。



三、源站开启了 SNI 校验，但是 CDN 没有开启“回源 SNI”

CDN 回源默认是不带 SNI 信息的，如果您的源站 IP 绑定了多个域名，当 CDN 节点以 HTTPS 协议访问您的源站时，由于没有带 SNI 信息，会导致源站无法正确响应 HTTPS 证书，导致回源失败。因为这个问题导致的错误，一般是 503 Service Temporarily Unavailable 错误，而且很快就会返回这个错误。您可以在 CDN 控制台设置开启回源 SNI，指明具体访问域名。具体 SNI 的介绍以及配置方法参考这里。



四、源站存在安全防护规则

源站的相关安全防护规则导致的 CDN 回源异常，通常会在 10 秒以内就返回 5xx 错误，大部分情况会返回 503 Service Temporarily Unavailable 的错误，具体的排查方法和解决方案可以参考文档源站安全策略导致 5xx。

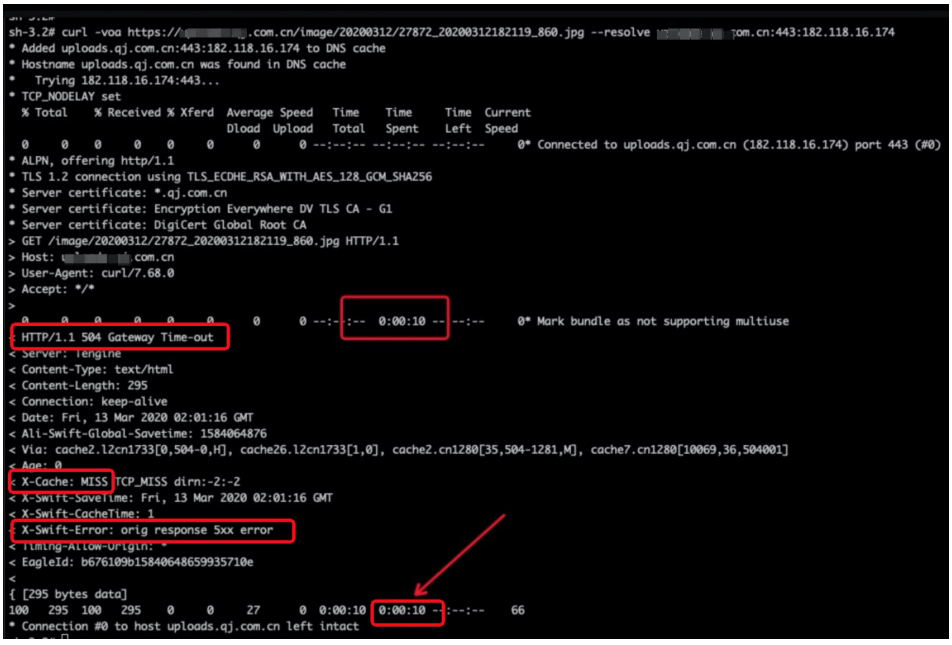
（1）源站服务器开启了安全组限制，限制了 CDN 节点的访问

（2）源站服务器配置了单 IP 访问次数限制，把 CDN 的回源 IP 当成了异常 IP

（3）源站存在云锁、安全狗、防火墙等安全策略，拦截了 CDN 的回源 IP

（4）源站 Web 服务异常或服务器超载

例如下图就是一种比较典型的 TCP 超时，每次访问到 10 秒的时候就超时断开了，响应 504。



五、源站超时无响应导致 CDN 回源超时

CDN 回源有严格的超时时间，四层 TCP 是 10 秒超时，七层 HTTP / HTTPS 是 30 秒超时，当超过该时间时即使后续源站响应正常也是会返回 5xx 错误，通常因 CDN 回源超时导致的问题，会响应 504Gateway Time-out 错误。可以绑定源站去测试源站的响应速度，如果超过 30 秒，需要检查源站服务，优化源站的响应速度，确保源站返回请求时间控制在一个较短的时间内，另外也可以申请延长 CDN 域名的默认超时时长，详细请参考配置回源请求超时时间。

> 请注意这个回源超时时间的配置是设置 HTTP 层面的超时时间，如果 TCP 层面就已经超时，那么这个设置是不生效的。

通常这类回源超时的问题发生在一些动态请求上，比如请求源站的程序、数据库、接口等，源站处理需要一些时间。这类情况建议源站使用 CDN 的站点都做动静分离改造，静态资源用 CDN 加速域名，动态资源直接用源站域名，因为如果源站响应慢，部分动态资源可能出现 30 秒仍然无法响应的情况。

六、跨境回源或源站侧网络异常

回源存在跨境链路导致的 CDN 回源超时，响应 5xx 错误。例如源站在境外，中国大陆的用户访问的时候，是先访问到中国大陆的 CDN 节点，然后中国大陆的 CDN 节点走跨境链路，回源到境外的源站；亦或者源站在中国大陆，境外用户访问的时候先请求到境外的 CDN 节点，境外的 CDN 节点走跨境链路，回源到中国大陆的源站。由于 CDN 回源走的都是公网，这种情况涉及到跨境链路，需要走国际互联网出口以及境外运营商的链路，本身就存在一定的不稳定因素。还有一种情况是源站侧机房的网络差，或源站侧网络不稳定。

通常这两类问题 CDN 层面的优化难度比较大，因为 CDN 只是提供了节点，做缓存服务，很难去控制公网的网络以及源站侧的网络。对于源站侧网络的问题，建议优化源站；对于跨境回源的问题，建议优化 CDN 的缓存命中率，尽量减少回源，降低 5xx 比率。或者考虑使用海外源站 + 国内源站的双源站架构。

在一些静态加速没有命中缓存，回源又一直超时的情况，可以考虑使用全站加速的动态加速，动态加速通过智能路由技术为动态内容，进行路由决策，选择最佳回源路径，会有一定的优化效果，但是不一定能完全解决此类问题。

相关文档

域名绑定 Host 操作

源站存在安全防护等原因导致访问 CDN 域名报 503 错误

回源请求超时时间

使用 CDN 后访问域名提示“504”Gateway Time-out" 错误

适用于

CDN

DCDN

服务器陷入死循环？ 508 错误的解法

简介：508 Loop Detected 是回环错误，CDN 的源站域名不能走 CDN 加速，否则可能引起 508 回环错误。

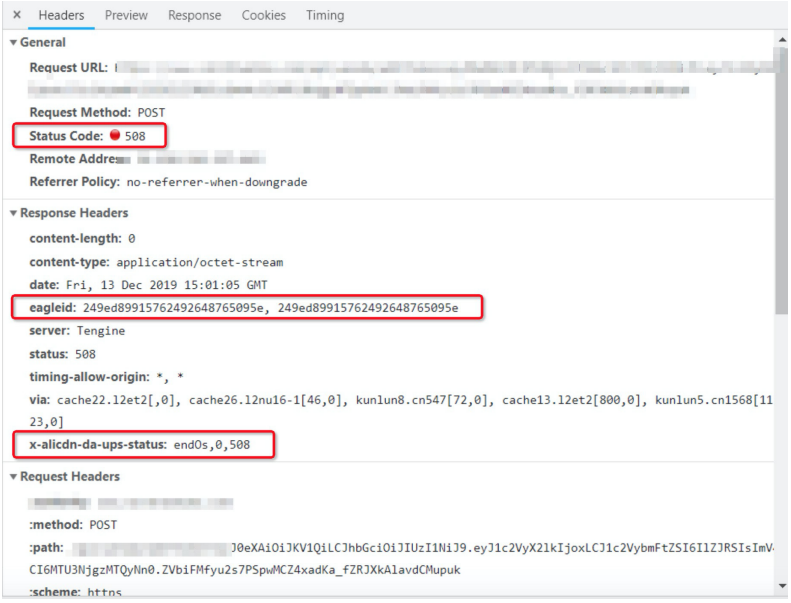
问题场景

用户使用阿里云 CDN 加速服务以后出现 508 错误。我们先来看一下，Http_code 508 的定义：

The server terminated an operation because it encountered an infinite loop while processing a request with "Depth: infinity". This status indicates that the entire operation failed。

Source: RFC5842 Section 7.2

也就是说，服务器在处理请求时陷入死循环，此状态表示整个操作失败。



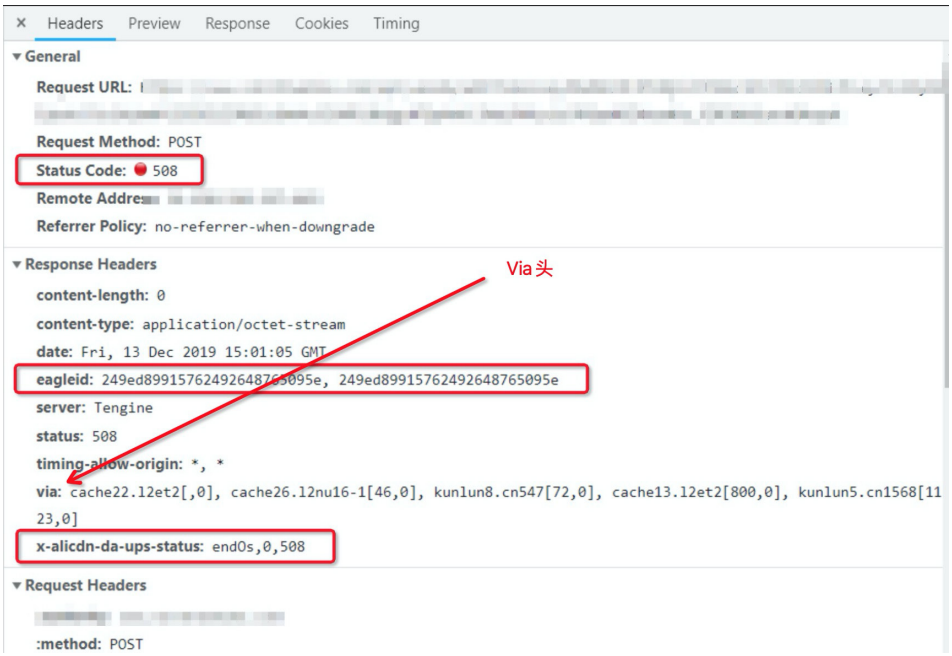
问题阐述

从上面的 CDN 的 508 错误里，可以看到 CDN 返回的 Response Headers 里，x-alicdn-da-ups-status 响应了 508，eagleid 响应了两个值。通常情况下，一次通过 CDN 加速的请求，eagleid 只会记录一个值，用于唯一标识这次请求，像这种情况，eagleid 记录了多个值，说明这个请求在 CDN 里循环了。造成这个现象的可能原因一般有如下两种情况：

- CDN 的源站也是在阿里 CDN 上加速的，比如 CDN 的源站域名也是用了一个阿里 CDN 域名。
- CDN 的源站有一些代理的逻辑，会将 CDN 加速域名的请求代理到另外一个走阿里 CDN 加速的域名上。

这里请注意，因为数据的走向是：客户端 -->CDN--> 源站。如果源站也是走了 CDN 的话，就会变成 : 客户端 -->CDN--> 源站 -->CDN--> 源站 ... 以此循环，这种情况就可能导致回环的情况，出现 508。CDN 产品本身有限制，要求 CDN 的源站不能是走阿里 CDN 加速，这是一种非标准化的操作。

另外还有一种定位方法，可以直接绑定 Host 到源站去测试，也就是不通过 CDN 访问，然后看返回的 Response headers 响应头，是否带有 CDN 的特殊 header 头，一般就是看 Via 头就可以了（阿里云 CDN 返回的 header 里会带有 Via 头），可以参考下图。如果直接访问源站的时候，源站返回的 header 头却有 CDN 特有的 header 头，那就说明源站有请求 CDN 的逻辑。



特别注意

CDN 添加域名的时候，CDN 控制台会做判断，如果用户填写的源站域名是阿里 CDN 加速域名的话，会直接报错。但是有两种情况目前 CDN 层面还无法避免，需要用户侧去修改：

- 用户配置加速域名 A 的源站域名是域名 B，此时域名 B 未走 CDN 加速，因此域名 A 的源站可以配置成功。但是配置成功以后，用户将域名 B 添加到 CDN 上做加速。
- 用户配置的源站不走 CDN，但是源站会做一些代理，例如方向代理，把 CDN 加速域名的请求代理到另外一个 CDN 加速域名，这种情况 CDN 也是无法提前监测到的。


CDN
域名管理
监控查询
刷新预热
日志管理
工具服务
安全防护
CDN / 域名管理
正常运行
基本配置
回源配置
缓存配置
HTTPS配置
访问控制
性能优化
视频相关
安全配置
高级配置
边缘脚本
IPv6配置
基础信息
CNAME
创建时间
加速区域
源站信息
类型
地址
错误
源站已经是cdn的域名，请修改成其他源站。
RequestId: B659B983-F64A-48A4-8527-9CFC12B74C7A
确定

重定向次数过多？三个方法搞定

简介：用户配置了阿里云 CDN 或者全站加速后，使用浏览器进行访问，出现重定向的次数过多的错误。

问题描述

配置了阿里云 CDN 或者全站加速后，使用浏览器进行访问，出现如下错误提示。

XXX.XXX.XXX 将您重定向的次数过多。

尝试清除 Cookie.

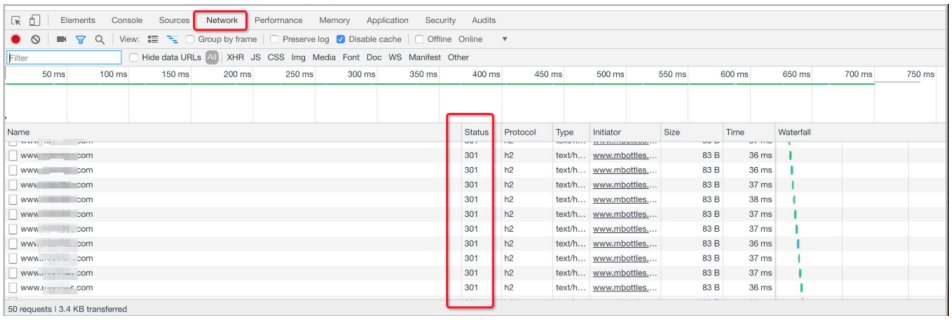
ERR_TOO_MANY_REDIRECTS

问题原因

源站开启了 HTTP 重定向至 HTTPS 的功能，并且 CDN 控制台上配置的回源端口为 80。在这种情况下，由于 CDN 回源端口为 80，客户端无论是通过 HTTP 还是 HTTPS 访问 CDN 加速域名时，CDN 在回源的时候都是使用 HTTP 请求源站，此时会触发源站的 HTTPS 强制跳转逻辑，然后源站会要求 CDN 重新发送一个 HTTPS 的请求，但是 CDN 回源的时候仍然会发送 HTTP 回源请求，然后再进行跳转，以此类推，就会出现反复重定向问题，最终导致出现报错。

分析过程

1. 打开浏览器的开发者模式，切换至 Network 标签页，然后重新访问源站，发现出现无限 301 重定向的情况。



2. 使用 HTTP 协议访问源站域名进行测试，确认源站开启了 HTTP 重定向至 HTTPS 的功能，并且确认 CDN 控制台上配置的回源端口为 80。

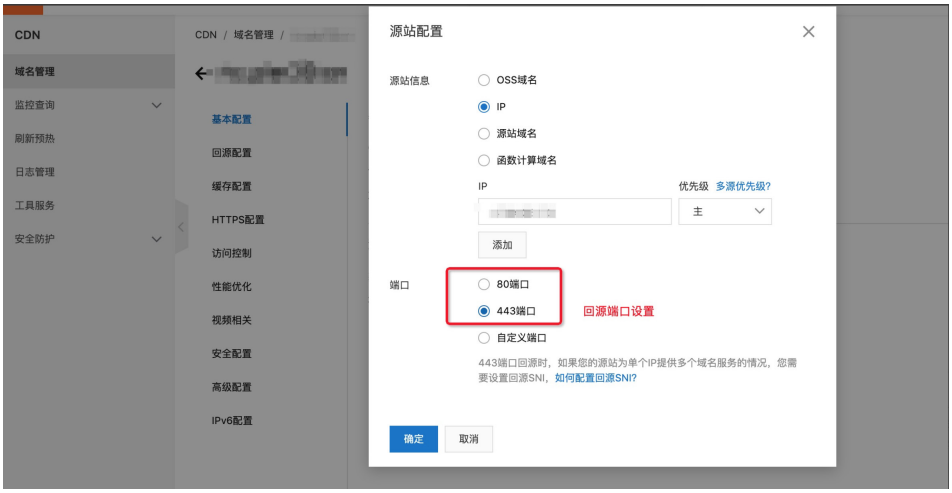
解决方法

本文介绍如下三种解决方法，请根据实际情况选择具体的方法。

方法一

登录 CDN 控制台，在域名管理页面单击目标域名对应的管理，然后单击 修改配

置，将 CDN 的回源端口设置为 443，并单击 确定。设置为 443 端口以后，CDN 回源时会以 HTTPS 协议请求源站，就不会触发源站的强制跳转逻辑。



方法二

将协议跟随回源设置为“跟随”。设置为跟随以后，源站发起 HTTPS 重定向以后，CDN 回源协议跟随为 HTTPS 回源。

方法三

如果不希望将 CDN 的回源端口改成 443，仍希望 CDN 以 HTTP 协议回源，这种情况下可以考虑关闭源站的 HTTP 重定向至 HTTPS 的强制跳转功能。

特别注意

如果按照上述建议修改配置以后问题还未解决，则可能是 301 被 CDN 节点缓存了，需要刷新下 CDN 的缓存，具体请参考刷新缓存操作。下面是一个异常现象的案例，访问 HTTPS 的 URL 以后，301 重定向 Location 到同样的 HTTPS 地址，从 Response Headers 里可以看到 301 被 CDN HIT 缓存住了。


```
sh-3.2# curl -voa https://www.      cn/customer.html --resolve www.      .cn:443:111.48.155.242
* Added      :443:111.48.155.242 to DNS cache
* Hostname www.      was found in DNS cache
*   Trying 111.48.155.242:443...
* TCP_NODELAY set
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0* Connected to www.      (111.48.155.242) port 443 (#0)
* ALPN, offering http/1.1
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: *.shixunbao.cn
* Server certificate: GlobalSign Organization Validation CA - SHA256 - G2
* Server certificate: GlobalSign Root CA
> GET /customer.html HTTP/1.1
> Host:
> User-Agent: curl/7.68.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 301 Moved Permanently
< Server: Tengine
< Content-Type: text/html
< Content-Length: 185
< Connection: keep-alive
< Date: Mon, 03 Feb 2020 00:17:53 GMT
< Location: https://www      /customer.html
< Strict-Transport-Security: max-age=31536000
< Ali-Swift-Global-Savetime: 1580689073
< Via: cache6.l2et15-7[0,301-0,H], cache70.l2et15-7[0,0], vcache33.cn1890[0,301-0,H], vcache8.cn1890[2,0]
< Age: 400694
< X-Cache: HIT TCP_MEM_HIT dirn:10:334798902
< X-Swift-SaveTime: Tue, 04 Feb 2020 09:57:35 GMT
< X-Swift-CacheTime: 5184000
< Timing-Allow-Origin: *
< EagleId: 6f309b9315810897676711178e
<
{ [185 bytes data]
100   185  100   185    0     0    692      0 --:--:-- --:--:-- --:--:--   695
* Connection #0 to host www.shixunbao.cn left intact
```

更多信息

由于使用了 CDN，且客户端的请求都是先请求到 CDN 节点。如果希望 CDN 节点获取客户端的请求是 HTTPS 协议时，可以在 CDN 控制台上设置 HTTP 强制跳转，具体信息请参考配置强制跳转。

适用于

CDN

全站加速

阿里云 开发者社区

云服务技术大学
云产品干货高频分享

云服务技术课堂
和大牛零距离沟通

阿里云开发者“藏经阁”
海量免费电子书下载