



VoD问题排查 与实战手册

云运维工程师从入门到精通



精解基本概念快速入门阿里云视频点播

40+技术问答详解常用功能排查及解决方案

针对多种业务场景的最佳实战分享



阿里云开发者电子书系列



云技术服务大学
云产品干货高频分享



云技术服务课堂
和大牛零距离沟通



阿里云开发者“藏经阁”
海量电子书免费下载

目录

点播原理介绍和入门	4
视频点播功能方案介绍	5
视频点播入门介绍1	11
视频点播入门介绍2	13
视频点播快速上手	14
点播常用功能排查	21
播放失败常见问题汇总1	22
播放失败常见问题汇总2	28
上传视频异常排查实战	35
点播最佳实践	38
直播转点播实践	39
播放地址获取实践	46
视频点播添加字幕实战	48
点播试看最佳实践	52
多场景化转码实践	55
OSS SDK上传到点播方案实践	58
点播视频安全宝典系列1	65
点播视频安全宝典系列2	66
点播解决方案	69
播放常见所需参数介绍	70
播放器SDK播放问题实战排查	75
短视频解决方案介绍	79
端SDK相关汇总	82

点播原理介绍和入门

视频点播功能方案介绍

简介： 阿里云视频点播（VoD）是集音视频采集、编辑、上传、自动化转码处理、媒体资源管理、分发加速、视频播放于一体的一站式音视频点播解决方案。今天我们将介绍点播的优势、开通和计费方式。

产品简介

阿里云视频点播（VoD）是集音视频采集、编辑、上传、自动化转码处理、媒体资源管理、分发加速、视频播放于一体的一站式音视频点播解决方案。其构建在阿里云强大的基础设施服务之上，借助灵活和可伸缩的存储、高质量的视频转码处理技术，以及稳定快速的内容分发服务，帮助企业和开发者快速搭建安全、弹性、高可定制的点播平台和应用。

视频点播服务提供Web管理控制台和软件开发工具包（API+SDK）。您可以通过它们使用、管理视频点播服务，也可以与您自己的应用和服务集成。所有服务按使用付费，服务能力自动伸缩，告别复杂的架构设计和编程开发，维护成本几近于零，使您可以专注于业务逻辑实现及最终用户体验的提升。

产品优势

一、灵活简单易用

点播场景完整支持，简单易用的控制台，并提供丰富的SDK及开放API，支持定制的开发需求；按使用付费，服务能力自动伸缩，维护成本几近于零，您可专注于业务实现及最终的用户体验。

二、领先转码技术

窄带高清和H.265技术，提供更佳画质、更低码率的自适应转码输出；高速稳定的并行转码系统，转码任务规模无缝扩展，保障视频转码质量和效率。

三、极致播放体验

多码率、多清晰度、多格式的差异化视频流，满足全终端的播放需求；跨运营商、跨地域全网覆盖的视频分发网络，提供更稳定、更流畅的视听体验。

四、酷炫视频玩法

快速集成短视频录制编辑功能，可按模块组合使用，多分辨率支持，UI自定义；实时滤镜、人脸贴图等前卫玩法应有尽有。

五、多重服务保障

高可靠的云存储服务保障海量音视频永久安全存储；灵活定制的防盗链和播放鉴权功能保障媒体资源安全；完善的监控体系&服务体系，7*24小时为点播服务保驾护航。

购买指导

一. 使用注册成功的阿里云账号登录 www.aliyun.com，点左上角 产品，在视频服务中找到 视频点播 进入产品主页。



二. 单击 立即开通 进入产品开通页面。



三. 如果未实名认证, 需单击 立即认证 进行实名认证, 完成认证后返回开通页面并刷新; 如果已经实名认证, 请仔细阅读 视频点播服务服务条款, 然后单击 立即开通 开通服务。



四. 开通服务后, 进入视频点播控制台, 完成初始化配置即可使用, 具体的使用方法请查看官网文档快速入门。

计费方式

视频点播产品价格分为两个部分: “基础服务” + “增值服务”。

• 基础服务

oÂÂ 存储空间: 上传的视频源文件和转码后的视频文件占用的存储空间, 按占用容量大小计费。

oÂÂ 视频加速: 视频在内容分发网络 (CDN) 中加速产生的费用, 默认按流量计费。

• 增值服务

oÂ Â 视频转码：除转码设置中“流畅”和“标清”两种格式的首次转码为免费之外，其他格式的转码均按转码时长进行计费。

一、按[存储空间]计费说明

- 计费项：容量大小
- 计费价格：

存储阶梯			价格（元/GB/月）
0-50GB（含）			0
大于50G			0.25

• 计费规则：

1. 付费方式：后付费
2. 计费规则：根据实际文件占用存储空间大小，按小时超额累计
3. 计费周期：按小时计费，实时扣费

• 计费公式：累计的存储容量（GB） * 对应存储阶梯的小时单价

• 举例说明：

用户当月累计上传视频 100G，转码后视频等文件为900G。则当月存储费用 = 0.25 元/GB/月*（100 + 900 - 50）GB= 237.5 元

二、按[流量]计费说明

- 计费项：下行流量
- 计费价格：0.5元/GB

- 计费规则：

1. 付费方式：后付费
2. 计费规则：根据播放视频时产生的下行流量消耗计费
3. 计费周期：按天计费，实时扣费

- 计费公式：每天下行流量消耗（GB）* 下行流量单价（元/GB）
- 收费示例

3月31日 0:00 至 3月31日 24:00 累计消耗的流量为 102 GB，则当天产生账单金额为102GB × 0.5元/GB = 51元

三、按[转码]计费说明

- 计费项：转码时长
- 计费价格

计费类型	价格（元/分钟）
H.264 流畅、标清（960*540及以下）	免费

H.264 高清 (1280*720)	0.0465
H.264 超清 (1920*1080)	0.093

注：H.265编码格式单独议价。如有需要，请工单咨询。

- 计费规则：

1. 付费方式：后付费
2. 计费规则：在免费范围外，不同格式的转码均按转码时长进行计费
3. 计费周期：按小时计费，实时扣费

- 计费公式：转码视频的时长（分钟）* 不同视频格式的转码单价（元/GB）

- 举例说明：

5月1日进行了两次付费转码操作，其中视频A 时长为120分钟、转码格式分别为高清（H.264 SD）和超清（H.264 HD），视频B 时长为100分钟、转码格式分别为超清（H.264 HD）和2K（H.264 2K）。则

1. 视频A转码费用 = $120 * (0.0465 + 0.093) = 16.74$ 元
2. 视频B转码费用 = $100 * (0.093 + 0.2) = 29.3$ 元

当天产生转码费用金额为 $16.74 + 29.3 = 46.04$ 元

以上就是阿里云点播解决方案的产品简介、优势、开通和计费，如有疑问请[参考官网](#)，如有意见欢迎留言哦~

视频点播入门介绍1

简介：熟悉视频点播服务的基本概念和开发流程，带领大家操作视频点播的主要场景。

目标

- 熟悉视频点播服务的基本概念和开发流程
- 带领大家操作视频点播的主要场景

准备工作

- 视频点播服务主页和控制台

[视频点播服务主页](#)

[视频点播服务控制台](#)

[视频点播产品功能介绍](#)

[视频点播开发手册](#)

[阿里云Code](#)

- 注意事项

视频点播服务是一个按量计费的服务，根据视频时长、接口访问次数来计费。具体的参考[产品价格](#)。

视频点播服务目前每月赠送免费额度，包含最多 50 分钟的H.264内容，25 分钟的H.265内容、10 分钟的转封装内容及 30 分钟的音频内容。

考虑到视频点播服务的免费额度，在搭建视频网站过程中，除OSS存储和CDN网络外，将不会产生其他相关费用，因此课程中产生的总费用极低。

步骤

1、配置媒体工作流

- 媒体库设置
- 创建媒体工作流

2、配置CDN

3、上传视频

- 下载演示视频xxx
- 打开首页，进入我的视频，选择上传视频
- 填写基本信息，选择本地视频文件
- 开始上传，并查看上传进度，直到完成

4、查看视频状态，等待转码完成

- 进入我的视频
- 查询视频状态，直到完成

5、播放

- 点击视频进行播放
- PC浏览器直接观看
- 手机扫描二维码观看

视频点播入门介绍2

简介：展示点播服务的基本概念

1、点播服务是什么？

阿里云视频点播（VoD）是集音视频采集、编辑、上传、自动化转码处理、媒体资源管理、分发加速、视频播放于一体的一站式音视频点播SAAS解决方案。定义成的是一个完全SAAS的方案，目标是需要让用户能够更加简单的使用所有关于音视频相关的阿里云服务.而不用关注更深层次的服务。如CDN，MTS,OSS等。

2、为什么需要点播服务？

在之前阿里视频云一直是一个PAAS方案，对于一个开发者来说，比如要使用一个视频上传，转码，存储需要对接阿里云的存储，转码，CDN等多个产品。但是现在有了点播服务，对于开发者来说只要开通一个点播服务，就可以直接做到一站式解决用户的上传---转码---播放整个链路.并且点播服务还提供各种开发者需要的端SDK，如上传的SDK，短视频录制的SDK，端播放器的SDK，web播放器SDK、服务端的SDK来帮助开发者来快速上线一个视频网站或者是视频app。

3、点播服务由哪几部分组成？

Web管理控制台 + openAPI + 短视频SDK + 上传SDK + 播放器。

web管理控制台：视频管理、视频审核、视频编辑、视频转码、添加水印、视频上传、下载配置等。

openApi: 视频上传相关接口、视频播放相关接口、视频管理接口、视频分类相关接口、事件通知、视频安全等。

短视频客户端SDK：完成视频的采集，本地编辑，添加特效之后生成一个mp4文件。

上传客户端SDK：实现文件的上传，比如短视频SDK录制完成的视频上传。

播放器客户端SDK和web播放器：播放点在播的视频。

本文为【问答合集】阿里视频云常见问题汇总——内含技术文档40+ 持续更新中 系列文章之一，如果本文还不能解决你的问题可以在汇总合集中查看，有任何问题也欢迎回复，谢谢。

视频点播快速上手

简介： 本文档介绍如何快速开始使用视频点播服务

写在前面

阿里云视频点播（VoD）是集音视频采集、编辑、上传、自动化转码处理、媒体资源管理、分发加速、视频播放于一体的一站式音视频点播解决方案。其构建在阿里云强大的基础设施服务之上，借助灵活和可伸缩的存储、高质量的视频转码处理技术，以及稳定快速的内容分发服务，帮助企业 and 开发者快速搭建安全、弹性、高可定制的点播平台和应用。

视频点播服务提供Web管理控制台和软件开发工具包（API+SDK）。您可以通过它们使用、管理视频点播服务，也可以与您自己的应用和服务集成。所有服务按使用付费，服务能力自动伸缩，告别复杂的架构设计和编程开发，维护成本几近于零，使您可以专注于业务逻辑实现及最终用户体验的提升。

本文档介绍如何快速开始使用视频点播服务，主要步骤有：

1. 登录官网
2. 开通服务
3. 初始化域名
4. 全局设置
5. 上传文件
6. 查看视频

STEP 1、登录阿里云官网

访问[阿里云官网](#)，点左上角登录。

STEP 2、开通视频点播服务

在阿里云官网视频点播产品详情页快速了解产品。

单击立即开通，在订购页面选择计费方式，确认订单，视频点播服务即开通。



STEP 3、初始化域名

进入视频点播控制台，首次使用点播服务需要初始化域名。

填写用于视频分发与加速的自有域名。

****>注：**请确保该域名已经备案，并拥有使用权。



输入手机号验证码进行验证。

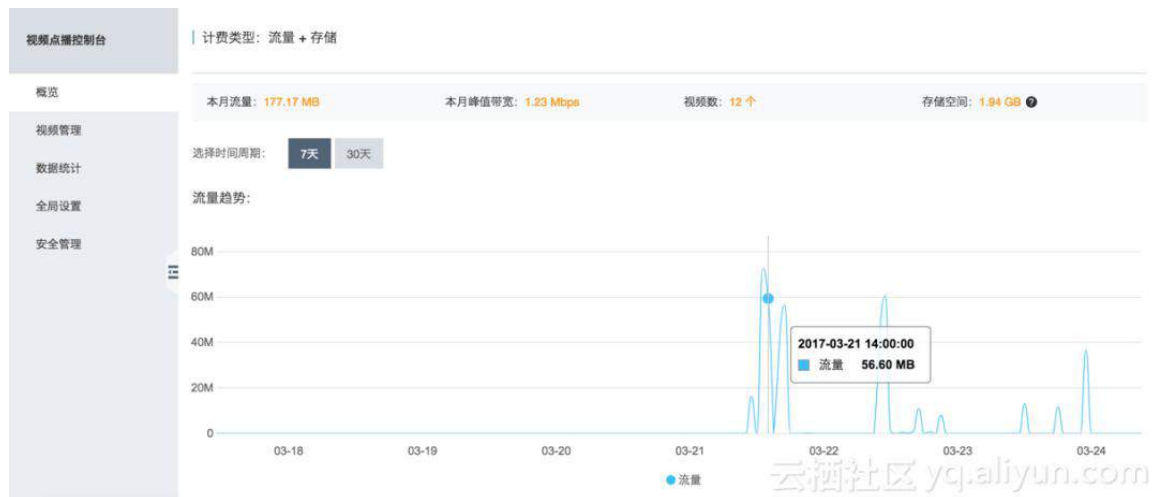


如果域名和手机验证均通过，则将获得阿里云的加速域名CNAME地址。

注：请根据提示信息，到您的域名解析商处完成CNAME绑定。

如果您跳过绑定，则需要确保在视频播放前完成绑定，否则会影响视频的正常播放。

点击立即体验，则进入到视频点播控制台的概览页面。



STEP 4、全局设置

设置转码格式

在上传视频之前，请进入全局设置-转码设置 TAB页，您可以根据自己的需求在列表中选择启用格式和清晰度。

如果您的视频需要更高的清晰度（如2K、4K）或码率设置，请通过工单提交申请。



启用视频水印

如需要使用视频水印，则请切换到全局设置-水印管理TAB页，选择启用水印，然后上传一张PNG图片并设置为默认水印即可。



STEP 5、上传文件

进入 视频管理 页面，点击页面右上角的上传视频按钮，在弹出的上传视频窗口中添加本地视频文件。

在添加视频页，您可以根据需要连续添加多个视频或选择移除某个视频。



如果在全局设置中已经启用了视频水印并配置了默认水印图片，则上传的视频默认都将压制水印。

若需要对某个视频取消使用水印，则可以在当前窗口对该视频取消选中使用水印即可。

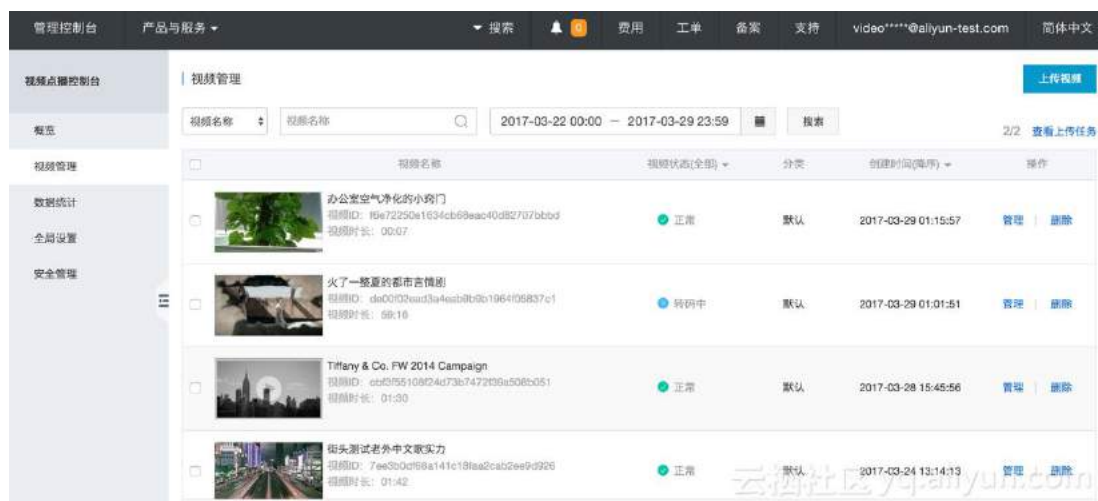
点击开始上传，已添加的本地视频文件则进入上传阶段，可以取消或重新上传视频。



在上传列表页，如果需要进行继续添加视频，则可以点击继续上传按钮返回到添加视频页。

STEP 6、查看视频

在 视频管理 页面中查看视频的处理状态。



点击某个视频所在行的管理按钮，可查看编辑视频的详细信息，包括标题、描述、分类、标签等。

如果需要添加视频分类，请点击管理分类进入全局设置-分类管理 TAB页进行管理操作。



转码结束后，点击视频列表中的视频名称列的某个视频，可以预览该视频，并支持切换清晰度。



关于视频点播的快速接入操作方式，已经介绍完毕了。

在官网上，我们还准备了操作演示视频：

点击[首次视频转码配置](#)、[微视频添加水印](#)、[进行视频上传和管理](#)



点播常用功能排查

播放失败常见问题汇总1

简介：本文由阿里云视频云高级开发工程师筱配撰写，主要介绍视频点播播放的前置条件以及用户在使用过程中经常遇到的问题，便于用户快速的定位并解决播放相关的问题。

1. 概述

视频上传到视频点播(VOD)，并且转码成功后即可进行播放，可以在视频点播控制台的视频管理页面对视频进行预览播放，也可集成VOD播放器SDK，或者将获取到的视频播放地址传递给自己的播放器（系统原生或开源或自研）来播放视频。本文由阿里云视频云高级开发工程师筱配撰写，主要介绍视频点播播放的前置条件以及用户在使用过程中经常遇到的问题，便于用户快速的定位并解决播放相关的问题。正式使用点播播放服务前需确认以下流程的配置及处理正确无误：

- 服务开通
- 视频处理
- 安全配置
- 网络环境

2. 服务开通

2.1 点播服务开通

- 在[阿里云视频点播产品页](#)点击【立即开通】。
- 若无阿里云账号请先[注册](#)；若未完成实名认证，请参考[实名认证](#)进行操作。
- 点播帐号未欠费。关于欠费请参考[欠费说明](#)。

2.2 域名配置

在[视频点播控制台](#)添加备案过的域名，过程参考[初始化设置](#)。

- 域名获取：域名注册、转入或购买请移步[万网](#)，可在[万网域名管理页面](#)选择需要使用的域名。

- 域名配置：获得域名后需要进行CNAME绑定才能完成接入，请参考[万网域名接入](#)，同时也支持[新网](#)和[DNSPod](#)的域名接入。
- 域名检测：在Linux命令行、Mac的Terminal或Windows的CMD命令行输入：nslookup -type=cname [用户点播域名]。

Linux（如CentOS）下若提示“command not found”则需要先安装：yum install bind-utils -y

如果显示的canonical name值与点播控制台显示的CNAME值一致则表示配置成功。

结果参考：https://help.aliyun.com/knowledge_detail/40173.html

https://help.aliyun.com/knowledge_detail/40144.html

CDN的IP检测页面：<https://cdn.console.aliyun.com/home#/Tool/ip>

3. 视频处理

视频文件上传至点播帐号下，经过视频处理流程后才可用于播放。根据业务需要，用户可以选择普通转码和上传后直接分发加速两种处理方式。用户上传的源文件必须是多媒体文件，需要满足特定文件格式和编码的要求才可以转码成功，最终用于播放。

3.1 文件有效

源文件和输出文件格式和编码限制参考文档：

https://help.aliyun.com/knowledge_detail/42250.html

如果用户的视频选择的是上传后直接分发加速的处理方式，点播需要对该视频进行截图处理和CDN分发加速，那么为了保障源文件分发后是默认可播放的，则视频源文件仅限于上传mp4、flv格式的文件。

当用户上传的源文件不符合视频处理要求时，会导致文件转码失败，目前转码失败的原因大部分是由于上传的源文件视频流缺失或Meta存在问题导致的，因此用户在上传文件前需要确认源文件能否在本地正常播放。

3.2 转码配置

在[转码设置](#)中选择转码模板组，点播服务开通时默认会添加不转码即分发和系统模板组，配置参考[转码配置](#)。进入转码流程处理时会根据转码配置转出相应格式和清晰度的文件用于播放。当视频选择不转码即分发模板组时，源文件上传完成后即可用于播放。

3.3 视频状态

视频处理成功后且只有[视频状态](#)为Normal的视频才能用于播放。为了快速获取可播放的视频，用户可以通过状态轮询和事件通知两种方式确定当前视频是否可以播放。状态轮询的方式时间上不可预期，建议用户根据视频处理的方式设置不同的事件通知：

- 普通转码，只要有一个清晰度转码完成后即可播放，可以配置：

[单个清晰度转码完成事件](#)

[全部转码完成事件](#)

- 不转码即分发，视频上传完成后即可播放，可以配置：

[视频上传完成事件](#)

在点播控制台的[回调设置](#)中配置相应的事件通知，配置参考[事件通知](#)。

4. 安全配置正确

阿里云视频点播提供了完善的内容安全保护机制，可以满足不同业务场景的安全需求，但如果安全配置不当，也会导致用户视频无法播放的问题。

4.1 审核配置

在[审核设置](#)中可以配置审核流程，只有审核通过的视频才可用于播放，配置参考[审核设置](#)。

- 在"审核设置"开启了"先审后发"才会出现状态为Checking的视频
- 在"视频点播控制台-视频审核-审核详情"中点击"屏蔽"或者调用审核接口对视频进行屏蔽操作后才会出现状态为Blocked的视频

状态为Checking和Blocked状态的视频只能在点播控制台内或者来自审核IP白名单内的请求才可播放。

4.2 访问限制

在[防盗链](#)中配置访问的 Referer黑、白名单（二者互斥）来限制视频资源被访问的情况。符合规则可顺利请求到视频数据，若不符合，请求会被拒绝，并返回403响应码。配置参考[防盗链设置](#)。

4.3 URL鉴权

播放地址若固定不变会带来持久的非法扩散传播，且无法有效遏制。视频点播提供的[URL鉴权](#)可通过生成动态的加密URL（包含权限验证、过期时效等信息）来区分合法请求，以达到保护视频资源的目的。在 URL鉴权 可配置URL鉴权信息，配置参考[URL鉴权配置](#)。

开启URL鉴权后，点播的播放器SDK、获取播放地址的API/SDK都会自动生成带时效的播放URL。

开启URL鉴权后，所有媒体资源，包括视频、音频、封面、截图等地址都会进行鉴权。

当开启URL鉴权后，只有在有效期内的URL才可以播放，URL鉴权过期后请求会被拒绝，并返回403响应码。

4.4 视频加密

阿里云视频加密是对视频数据加密，即使下载到本地，视频本身也是被加密的，无法恶意二次分发。如果用户在配置转码模板时选择了HLS加密，相应的转码输出文件只能通过集成阿里云播放器SDK播放。目前播放器内核SDK，涵盖iOS/Android/Flash多平台，自

动对加密内容进行解密播放；H5播放器不支持加密视频的播放。

更多信息参考[阿里云-视频加密](#)

5. 网络环境

播放服务强依赖于网络环境，如果没有稳定畅通的网络环境，会严重影响播放服务质量甚至无法播放视频。当网络出现异常时，可以参考以下两种网络排查方法：

5.1 域名劫持检查

访问 <https://www.ipip.net/ip.html> 默认显示的是本地IP。

用户本地PING一下要访问的域名，看是否能解析出对应的IP。

查看解析出的IP是否是阿里云CDN的节点，在 <https://www.ipip.net/ip.html> 中输入IP会显示相应地区。

如果不是阿里云CDN节点的IP，那么说明被劫持的，建议向客户端访问所在网络运营商进行投诉。

5.2 源站检查

如果通过域名解析出的IP是阿里云CDN的节点IP，且能PING通域名，那么本地域名绑定host测试，看下是CDN问题还是源站问题。具体操作步骤请参考 [CDN访问异常](#)。

6. 常见问题

- 上传和播放服务均无法使用时，请确认点播帐号是否欠费。
- 控制台内可播放，调用API/SDK获取播放地址时返回错误码InvalidDomain.NotFound，请确认域名配置。
- 调用API/SDK获取播放地址时返回错误Forbidden.IllegalStatus，请确认视频状态。
- 调用API/SDK获取播放地址时返回错误码InvalidVideo.NotFound，请确认视频是否被删除。
- 当请求CDN加速地址返回403时，请确认安全配置。
- 当某个地区用户无法播放视频时，请检查用户网络环境。
- 使用点播Web端播放器诊断页面的输出信息定位播放问题。目前支持原生H5、阿里云H5/Flash播放。诊断页面地址：
<http://player.alicdn.com/detection.html?source=http://common.qupai.me/player/qupai.mp4>

播放失败常见问题汇总2

阿里云视频点播是集音视频采集、编辑、上传、自动化转码处理、媒体资源管理、分发加速、视频播放于一体的一站式音视频点播解决方案。但是对于使用者来说经常遇到的问题即是视频点播中的视频如何对外提供服务，并且当播放出现异常时如何进行排查呢？本文主要从该方面为大家介绍视频点播的播放方式和异常排查。

播放方式

视频点播提供了三种方式进行播放分别适用于不同的场景。用户可以根据自己的使用场景选择合适的方法进行播流，下面依次介绍该三种播放方式：

1. 使用播放URL进行播放

视频点播的使用是存储在产品统一的 Bucket 中，因此与 OSS 中的视频都可以直接简单获取 URL 进行访问方式不同。产品统一的并不属于用户自己，因此用户是无法通过自己账号的 AccessKeyId 和 AccessKeySecret 获取播放的权限。

用户获取视频播放 URL 是需要通过视频点播提供的 GetPlayInfo 接口获取播放信息，具体接口信息请参考[获取视频播放地址](#)，该接口会根据用户在视频点播配置的转码规则分别提供所有的转码后的资源的 URL 以及对应的视频的基本信息。

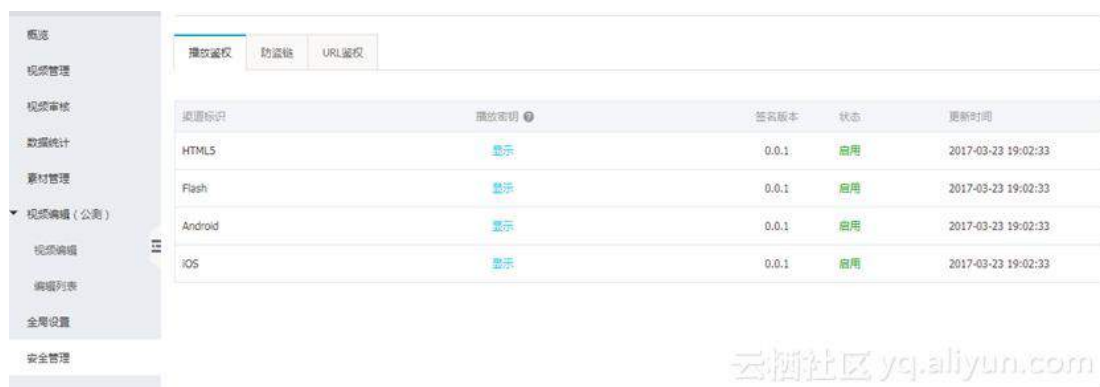
注意：

- 对于所有状态不是“正常”状态的视频资源将无法调用该接口（例如：转码中、审核中等执行状态），否则将直接抛出错误：Forbidden.IllegalStatus : Status of the video is illegal。
- 对于内容加密的视频通过该接口获取得到的播放 URL 是不需要解密播放并且可以直接下载得到该视频，因此对于内容加密的视频并不适合与使用 URL 的方式进行播放。
- 由于后续的方案都仅能够支持视频点播官方提供的播放器，如果需要使用第三方的播放器插件的话只能够使用本方法播放。

- 播放 URL 在没有开启 URL 鉴权的情况下获取的 URL 是永久可以访问的，但是在开启鉴权后获取的 URL 是带有 auth_key 参数的，对应的 URL 地址也是存在有效时间的，超过有效时间将出现 403 的错误。

2. 使用播放鉴权播放

播放鉴权是视频点播在阿里云 AK 安全认证基础上的二次鉴权机制。播放鉴权的机制是根据使用不同的终端设备播放器的 SDK 提供了不同的鉴权 Key，然后用户可以使用账号的 AccessKeyId 和 AccessKeySecret 结合对应客户端播放鉴权进行播放即可。我们分别提供了html5、flash、android和IOS不同平台的鉴权，查看方法如图1所示。



鉴权标识	播放密钥	签名版本	状态	更新时间
HTML5	显示	0.0.1	启用	2017-03-23 19:02:33
Flash	显示	0.0.1	启用	2017-03-23 19:02:33
Android	显示	0.0.1	启用	2017-03-23 19:02:33
iOS	显示	0.0.1	启用	2017-03-23 19:02:33

图 1. 点播播放鉴权示意图

对应的web端实例代码请参考如下demo：

```
1.      <!DOCTYPE HTML> <html> <head>
2.          <meta charset="UTF-8"><meta name="viewport" content="width=device-width,
height=device-height, initial-scale=1, maximum-scale=1, minimum-scale=1, user-
scalable=no"/> <link rel="stylesheet"
href="http://g.alicdn.com/de/prismplayer/1.6.3/skins/default/index.css" />
<script type="text/javascript"
src="http://g.alicdn.com/de/prismplayer/1.6.3/prism.js"></script> </head>
<body> <div class="prism-player" id="J_prismPlayer"></div> <script> var player
= new prismplayer( {
3.      id: "J_prismPlayer",
4.      autoplay: true,
5.      width: "1920px",
6.      height: "1280px",
7.      // 播放方式二
```

```
8.      prismType:2,
9.      //固定参数
10.     vid : '视频ID',
11.     accId : '用户AK id, 推荐子账号',
12.     accSecret : '用户AK secret, 推荐子账号',
13.     apiKey : 'H5播放器对应的播放密钥',
14.     flashApiKey : 'Flash播放器对应的播放密钥',
15.     }); </script> </body> </html>
```

注意:

- 上述的 AccessKeyId 和 AccessKeySecret 推荐使用 RAM 子账号进行设置，因为该页面是展示在前端的应用或者页面中，需要进行更细粒度的安全控制，仅需要给该子账号赋权 AliyunMTSPlayerAuth 即可。
- 上述方式由于多个参数均是固定参数并且没有时间限制导致可能出现安全风险，因此建议尽量避免使用本方法。
- 上述的方法可以同时配置 apiKey 和 flashApiKey 的话将可以使用兼容模式同时在 html5 和 flash 模式下播放。

3. 使用播放凭证播放

播放凭证方式播放是通过 vid 和临时播放凭证（playAuth）进行播放的，由于 playAuth 是服务器端根据 AccessKeyId 和 AccessKeySecret 获取的，并且是有 100 秒的有效时间的，因此较为有效的保护资源的安全性的。具体的 web 端播放的 demo 如下：

```
1.      <!DOCTYPE html><html><head><meta charset="UTF-8"><meta name="viewport"
2.      content="width=device-width, height=device-height, initial-scale=1, maximum-
3.      scale=1, minimum-scale=1, user-scalable=no"/><title>用户测试用例</title><link
4.      rel="stylesheet"
5.      href="//g.alicdn.com/de/prismplayer/1.9.9/skins/default/index.css" /><script
6.      type="text/javascript" src="//g.alicdn.com/de/prismplayer/1.9.9/prism-
7.      min.js"></script></head><body><div class="prism-player" id="J_prismPlayer"
8.      style="position: absolute"></div><script>var player = new prismplayer({
9.
10.     id: 'J_prismPlayer',
11.
12.     width: '100%',
13.
14.     autoplay: false,
```

```
5.     vid : '',
6.     playauth : '',
7.     });</script></body></html>
```

注意:

- playAuth 我们提供的 SDK 均是服务器端的 SDK，包括有Java、.net、php和 python 的版本，其主要原因是因为 playAuth 生成由于涉及到 AccessKeyId 和 AccessKeySecret 因此不建议在客户端直接生成，而建议客户端向服务器端发请求获取 playAuth 后直接播放，如果用户一定需要在客户端获取的话是需要使用我们提供的 api 接口实现的，其文档请参考：[获取视频播放凭证](#)；
- playAuth 可以通过有效时间保护视频资源，但是其并不能完全避免盗链的情况发生，建议如果是 flash 播放器 hls 资源可以考虑使用视频点播提供的内容加密功能，详细请参考：[内容加密](#)

播放异常排查

使用上述播放方式播放点播中的资源时经常会出现无法正常播放的情况，建议我们可以考虑如下的步骤逐个排查其无法播放的详细原因并修改。

1. 域名解析导致播放异常

视频点播在开通服务时即需要用户绑定自己的域名到视频点播服务中。而视频点播中所有的视频都是需要通过该域名提供链接进行播放的，因此该域名的解析配置正确将是能够正确播放的第一步。

视频点播的域名管理截图如图 2 所示。视频点播是可以添加多个域名在视频点播服务中的，但是同一时间启用的仅能够有一个域名。在配置完域名后需要用户到 DNS 提供商出配置该域名解析到视频点播提供的 CNAME 地址上才可以正常访问的。

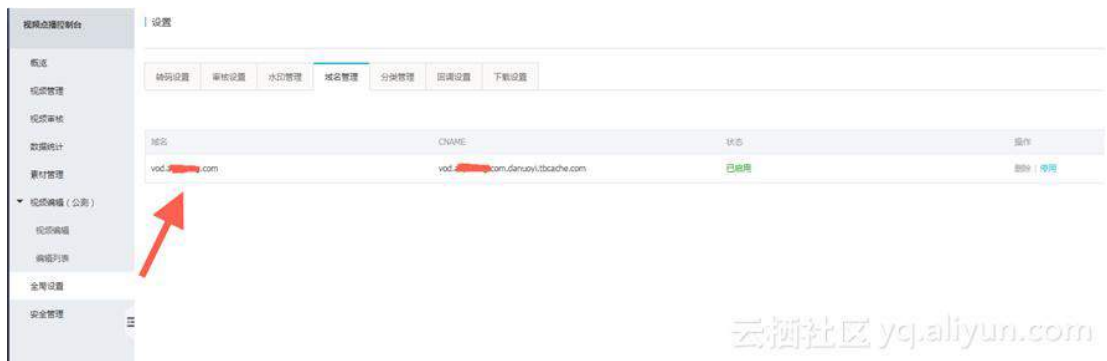


图 2. 点播域名管理示意图

注意：

- CNAME 记录和其他的 A 记录是相冲突的，因此某个域名解析到视频点播的话是无法再解析到其他的服务器的，因此这里建议使用还没有使用的子域名后添加并设置解析，而不要使用绑定在站点上的主域名。
- DNS 解析配置截图请参考图 3，其中记录类型是需要选择“CNAME”，主机记录填写子域名的记录值，例如图 2 中的子域名为 vod，那么这里我们应该填写的即是 vod。记录值填写视频点播提供的完整的 CNAME 域名，如图 2 中的 vod.xxxx.com.danuoyi.tbcache.com。



2. 防盗链设置导致播放异常

在设置完成上述解析后会可以通过视频点播控制台的播放功能测试是否正常的。当出现控制台播放正常，但是通过上述方法播放测试仍然有问题时刻查看安全防护设置是否禁止了访问。

防盗链是通过 HTTP 请求头中的 Referer 字段来控制访问来源是否允许，具体的设置请参考图 4，可以分别选择 Referer 白名单或者 Referer 黑名单，两者仅能够同一时间生效一种。当设置了之后就会仅允许或者限制某些 Referer 头访问。



图 4. 点播防盗链示意图

图5的示例即是被Referer防盗链限制的结果。



图 5. 防盗链拦截示意图

3. 播放凭证选择错误导致播放异常

当使用上述第二种播放方式时需要根据对应的播放器终端设备选择对应的播放凭证，如果设置错误的话将会导致无法播放，用户需要根据客户端选择正确的播放凭证。另外针对于 web 播放器的兼容模式可以同时设置 apiKey 和 flashApiKey 参数保证同时在 flash 和 html5 播放正常。

4. playAuth 超时导致播放异常

当采用上述第三种播放方式播放时需主要获取 playAuth 的业务逻辑，由于 playAuth 仅有 100 秒的有效时间，超过该有效时间后将导致新发起的播放请求无法播放（但是不影响正在播放的视频），因此业务端需要在少于 100 秒更新一次 playAuth。

上传视频异常排查实战

作者：张医博

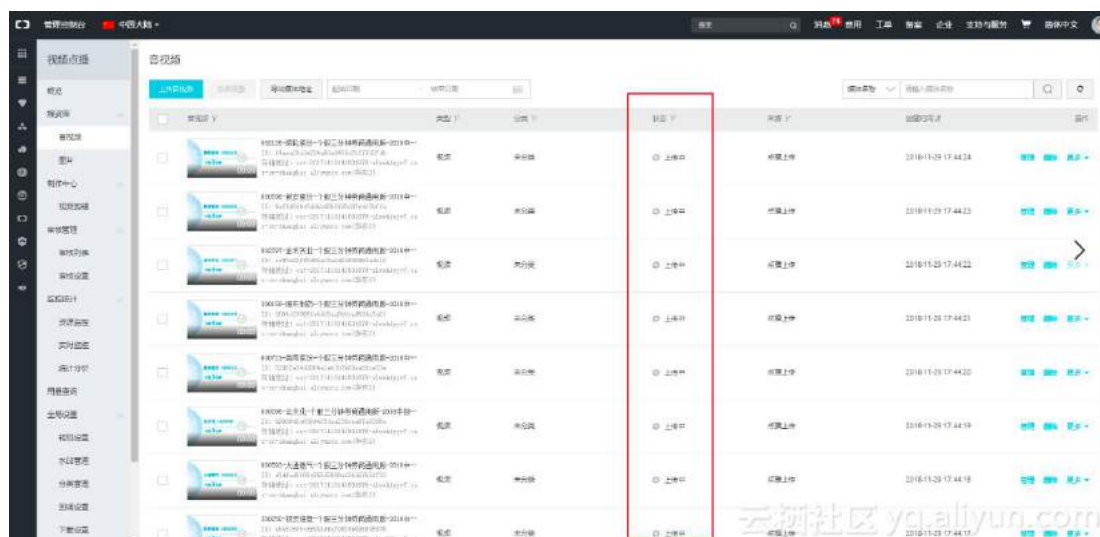
浅谈

本章节主要讲解上传视频到点播 VOD 存储时，控制台上显示的状态异常问题排查。

案例分析

案例

用户调用 createuploadvideoauth 获取上传凭证成功后，开始上传点播存储，发现大量的视频都是上传中的状态。



排查

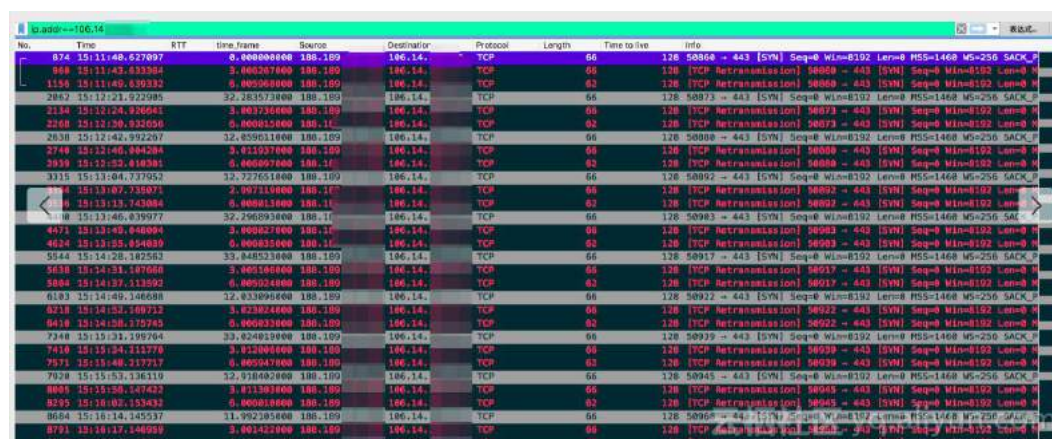
1) 先了解下用户点播上传凭证上传的流程

- 先通过 CreateUploadVideo 接口获取一个有效的上传地址。
- 然后再通过 OSS 上传文件到点播的存储 bucket，在这个过程中当获取的点播上传凭证后，控制台就会把视频的状态改为上传中。当用户使用上传凭证开始上传到点播存储 bucket 成功后，视频的状态才会变为已经上传。如果只是获取的上传凭证而没有上传视频的话，状态就是一直上传中，造成的误判就是点播上传任务堆积。

2) 了解了上面的流程后我们展开分析

- 既然已经出现了上传中的状态说明点播凭证肯定是获取到了。那我关键就是判断下是否发起了上传的操作。
- 经过客户端沟通反馈是已经发起了上传操作，但是如何证明呢？让客户端在本地使用 wireshark 进行了抓包，wireshark 的使用方式可以 Google 看下。

3) 分析抓包结果



No.	Time	RTT	Time frame	Source	Destination	Protocol	Length	Time to live	Info
874	15:11:40.627007		0.000000000	100.14	100.14	TCP	60	128	58888 → 443 [SYN] Seq=0 Win=0 Len=0
909	15:11:42.033304		3.000000000	100.14	100.14	TCP	60	128	TCP Retransmission 58888 → 443 [SYN] Seq=0 Win=0 Len=0
1156	15:11:49.639332		6.000000000	100.14	100.14	TCP	60	128	TCP Retransmission 58888 → 443 [SYN] Seq=0 Win=0 Len=0
2062	15:12:21.922985		32.283573000	100.14	100.14	TCP	60	128	58873 → 443 [SYN] Seq=0 Win=0 Len=0
2134	15:12:24.926041		3.000000000	100.14	100.14	TCP	60	128	TCP Retransmission 58873 → 443 [SYN] Seq=0 Win=0 Len=0
2260	15:12:26.922050		0.000000000	100.14	100.14	TCP	60	128	TCP Retransmission 58873 → 443 [SYN] Seq=0 Win=0 Len=0
2630	15:12:42.922207		32.030011000	100.14	100.14	TCP	60	128	58809 → 443 [SYN] Seq=0 Win=0 Len=0
2749	15:12:46.944284		3.911927000	100.14	100.14	TCP	60	128	TCP Retransmission 58809 → 443 [SYN] Seq=0 Win=0 Len=0
2939	15:12:52.410381		0.000000000	100.14	100.14	TCP	60	128	TCP Retransmission 58809 → 443 [SYN] Seq=0 Win=0 Len=0
3315	15:13:04.737952		32.727651000	100.14	100.14	TCP	60	128	58892 → 443 [SYN] Seq=0 Win=0 Len=0
3584	15:13:07.735071		2.907119000	100.14	100.14	TCP	60	128	TCP Retransmission 58892 → 443 [SYN] Seq=0 Win=0 Len=0
3710	15:13:15.743004		0.000000000	100.14	100.14	TCP	60	128	TCP Retransmission 58892 → 443 [SYN] Seq=0 Win=0 Len=0
3710	15:13:15.743004		0.000000000	100.14	100.14	TCP	60	128	TCP Retransmission 58892 → 443 [SYN] Seq=0 Win=0 Len=0
4471	15:13:19.940894		3.900827000	100.14	100.14	TCP	60	128	TCP Retransmission 58893 → 443 [SYN] Seq=0 Win=0 Len=0
4624	15:13:55.954889		0.000000000	100.14	100.14	TCP	60	128	TCP Retransmission 58893 → 443 [SYN] Seq=0 Win=0 Len=0
5544	15:14:28.187563		33.848523000	100.14	100.14	TCP	60	128	58917 → 443 [SYN] Seq=0 Win=0 Len=0
5630	15:14:31.187068		3.000000000	100.14	100.14	TCP	60	128	TCP Retransmission 58917 → 443 [SYN] Seq=0 Win=0 Len=0
5884	15:14:37.112992		0.000000000	100.14	100.14	TCP	60	128	TCP Retransmission 58917 → 443 [SYN] Seq=0 Win=0 Len=0
6183	15:14:45.146688		12.433090000	100.14	100.14	TCP	60	128	58922 → 443 [SYN] Seq=0 Win=0 Len=0
6218	15:14:52.140712		3.823824000	100.14	100.14	TCP	60	128	TCP Retransmission 58922 → 443 [SYN] Seq=0 Win=0 Len=0
6416	15:14:58.170745		0.000000000	100.14	100.14	TCP	60	128	TCP Retransmission 58922 → 443 [SYN] Seq=0 Win=0 Len=0
7348	15:15:31.199764		23.024019000	100.14	100.14	TCP	60	128	58939 → 443 [SYN] Seq=0 Win=0 Len=0
7610	15:15:34.111776		3.812006000	100.14	100.14	TCP	60	128	TCP Retransmission 58939 → 443 [SYN] Seq=0 Win=0 Len=0
7971	15:15:40.112717		0.000000000	100.14	100.14	TCP	60	128	TCP Retransmission 58939 → 443 [SYN] Seq=0 Win=0 Len=0
7972	15:15:40.112717		0.000000000	100.14	100.14	TCP	60	128	TCP Retransmission 58939 → 443 [SYN] Seq=0 Win=0 Len=0
8005	15:15:58.147422		3.811389000	100.14	100.14	TCP	60	128	TCP Retransmission 58945 → 443 [SYN] Seq=0 Win=0 Len=0
8295	15:16:02.153432		0.000000000	100.14	100.14	TCP	60	128	TCP Retransmission 58945 → 443 [SYN] Seq=0 Win=0 Len=0
8604	15:16:14.145537		11.992185000	100.14	100.14	TCP	60	128	58969 → 443 [SYN] Seq=0 Win=0 Len=0
8791	15:16:17.146999		3.001422000	100.14	100.14	TCP	60	128	TCP Retransmission 58969 → 443 [SYN] Seq=0 Win=0 Len=0
8809	15:16:23.160997		0.001122000	100.14	100.14	TCP	60	128	TCP Retransmission 58969 → 443 [SYN] Seq=0 Win=0 Len=0

通过抓包可以看到客户端的上行 SYN 包一直发不出来，而且并不想网络丢包，从重传的规律看像是被安全策略或者路由不通导致的重传。

于是让客户端做了基本的 ping 测试，发现连 ping 都失败。



4) 原因确定

经过和用户的沟通发现，客户端是有网络出口的安全策略，必须加白才能正常访问。客户是获取上传凭证，但是发起上传时一直没有成功，但是用户还没有基本的网络知识导致了问题的拖延。

5) 视频一直上传中的状态如何清理

- 先调用这个接口。https://help.aliyun.com/document_detail/52838.html?spm=a2c4g.11186623.6.679.7fbd10d5VVkEQg
设置 status 参数为上传中，会获取到所有上传中视频的 videoid。
- 然后调用这个接口去删除上传中的 videoid
https://help.aliyun.com/document_detail/52837.html?spm=a2c4g.11186623.6.678.1a552b29OTV5Ya

总结

遇到类似问题，一定要先排除客户端能否正常的获取上传凭证，获取上传凭证后能否正常上传到点播存储，一步步进行排除。

另外排查过程中一定要借助网络知识快速的定位问题。比如，Windows 端的 Wireshark Linux 端的 tcpdump。

点播最佳实践

直播转点播实践

场景简介

直播转点播（直转点）是将直播流同步录制为点播视频，并支持媒资管理、媒体处理（转码及内容审核/智能首图等AI处理）、内容制作（云剪辑）、CDN分发加速等一系列操作，可配置工作流自动处理，也可通过API/SDK灵活触发。

准备工作

1. 开通视频点播服务，[开通指引](#)
2. 开通视频直播服务，[开通指引](#)
3. 添加直转点录制配置，[帮助文档](#)
4. 上述准备工作完成后，即可开始进行接入(注：下述文档中的仅存储、仅合成模板组需联系点播进行激活)

名词解释

直转点，结合视频点播的转码、云剪辑、AI处理、事件通知等功能，可适应多种业务场景。

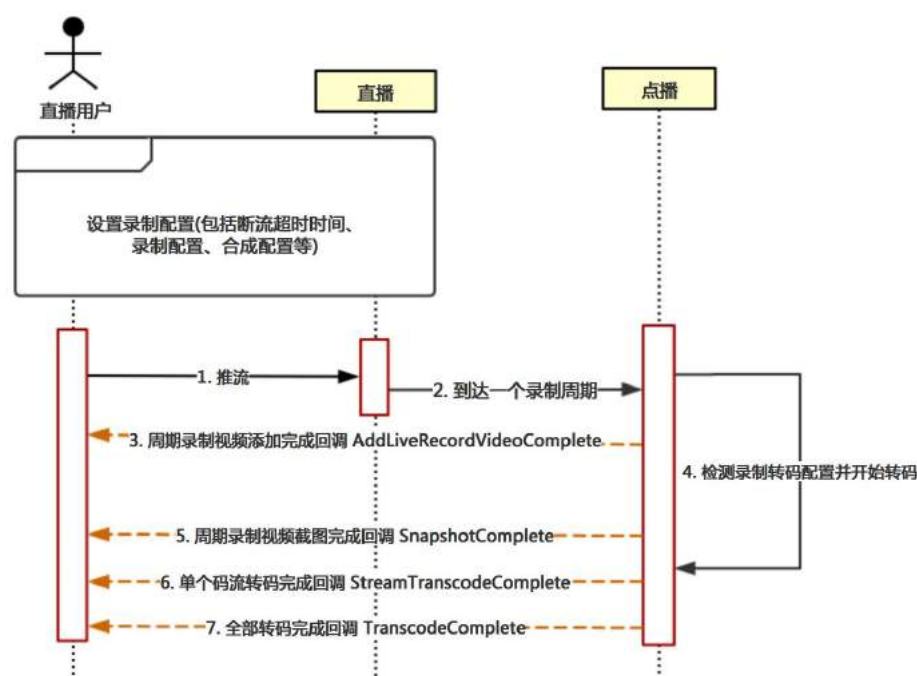
名词解释：

- 录制转码模板组：直播录制到点播同时，点播会使用该模板组对视频进行转码操作。
- 合成转码模板组：多个录制视频进行自动合成时，点播会使用该模板组对视频进行合成+转码操作。
- 仅存储：对直播内容进行录制后，不进行任何后续操作。
- 仅合成：对直播内容进行合成后，不进行任何后续操作。
- 直播录制周期：直播录制到点播的周期，如一场直播3个小时，如果需要在直播过程中就可提供已录制内容的点播服务，则可设置录制周期为1小时，即直播1小时过后，点播可提供前一小时内容的点播观看服务。

实践1：直播录制 + 自动转码 + CDN加速

直播录制后快速将录制文件进行转码和CDN加速，供用户进行点播播放，适用于大部分直播场景(不需要对内容进行二次加工)。

流程如下：



1. 客户进行直播推流。
2. 推流达到一个录制周期，则会自动将录制文件添加到点播系统。
3. 点播记录完成后，会生成点播系统的唯一视频ID，并将该视频信息回调给用户，即AddLiveRecordVideoComplete通知，并附带直播相关的DomainName、AppName、StreamName信息。客户收到回调后，需记录该视频信息，并以该VideoId作为索引进行后续视频状态更新。
4. 点播系统检测用户录制配置中的录制转码组ID(该转码组中含有具体码流转码任务)，进行对应的转码操作。
5. 截图完成、单个码流转码完成、全部码流转码完成时，会给用户进行回调(回调顺序无时序)，用户需根据回调信息中的VideoId进行视频状态更新。转码完成后，即可进行后续的播放操作(转码回调信息中含播放地址或后续通过GetPlayInfo接口根据VideoId进行播放地址获取，该播放地址已经经过CDN加速)。

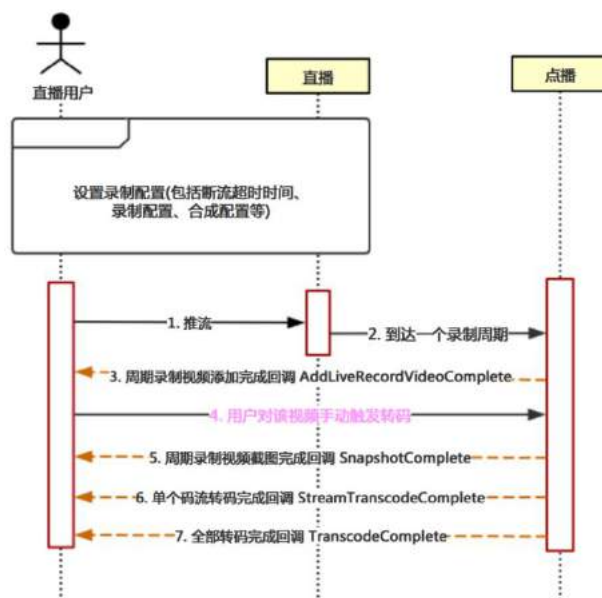
实践2：直播录制 + 仅存储到点播 + 手动发起转码 + CDN加速

部分用户希望将直播录制的视频仅先存储到点播，且先不进行后续的转码操作，则可在新建录制配置时，将录制转码组选择为仅存储模板组(注:仅存储模板组需联系点播进行激活)如果后续希望对视频进行转码，则可进行手动触发转码操作，同时，可配合点播云剪辑功能进行使用，效果更佳。

适用场景：

直播完成后，需要对内容进行二次加工，如体育赛事、游戏直播剪辑等，后续由用户主动发起转码和CDN加速全流程(转码完成后点播会自动对输出文件进行CDN加速)。

流程如下：



1. 客户进行直播推流。
2. 推流达到一个录制周期，则会自动将录制文件添加到点播系统。
3. 点播记录完成后，会生成点播系统的唯一视频ID，并将该视频信息回调给用户，即AddLiveRecordVideoComplete通知，并附带直播相关的DomainName、AppName、StreamName信息。客户收到回调后，需记录该视频信息，并以该VideoId作为索引进行后续视频状态更新。
4. 点播系统检测用户录制配置中的录制转码组ID(此时为仅存储模板组)，则点播系统不进行后续转码操作。

5. 用户对该视频手动触发转码操作(调用转码任务API)，在此之前可进行云剪辑等操作。
6. 截图完成、单个码流转码完成、全部码流转码完成时，会给用户进行回调(回调顺序无时序)，用户需根据回调信息中的Videoid进行视频状态更新。转码完成后，即可进行后续的播放操作(转码回调信息中含播放地址或后续通过GetPlayInfo接口根据Videoid进行播放地址获取，该播放地址已经经过CDN加速)。

实践3：直播录制 + 多周期视频自动合并

部分用户希望将自己录制周期生成的多个文件(如录制周期为20分钟，一次直播1个小时，则会生成三个视频)进行视频合成，再将合成后的视频进行处理，点播提供了自动合成的功能可在新建直播录制配置时，将自动合成开关打开，并且配置进行合成时所使用的合成转码模板组(也可配置为仅合成，后续由用户触发转码，与前两节所述区别相同)。

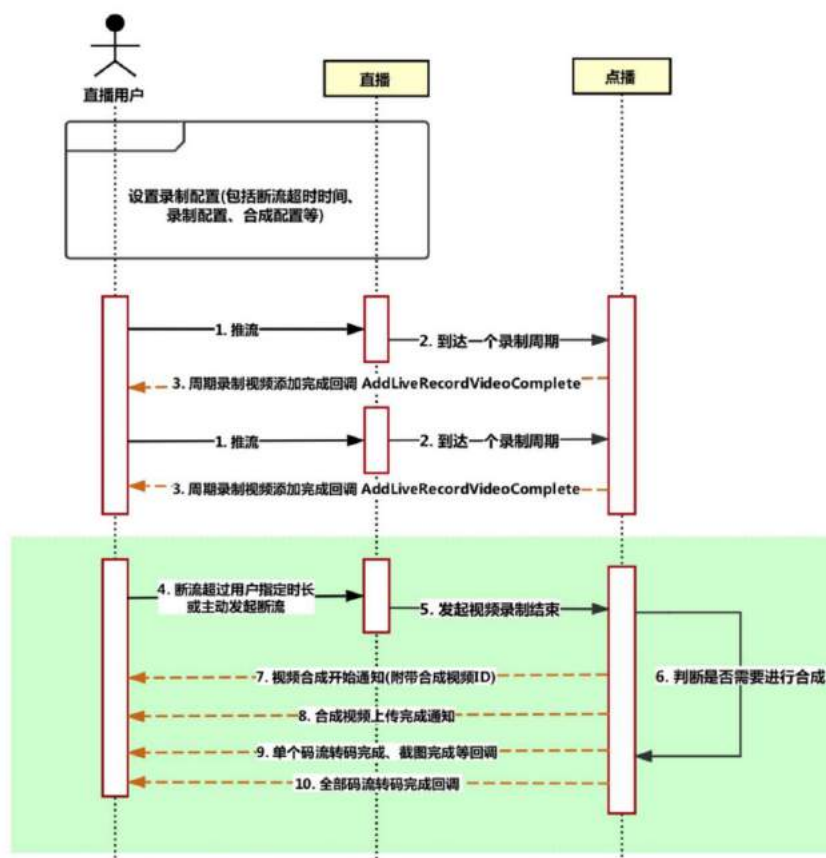
点播会在用户断流超过指定时间(可由直播进行配置断流超时时间)后，进行本次直播的视频自动合成并根据转码配置进行后续操作。

合成 + 自动转码

适用场景：

一场直播完成后，需要对所有录制周期内的分段进行自动合并，同时发起转码等全流程。如体育赛事、教育多节授课合并等。

流程如下：



1. 客户进行直播推流。
2. 推流达到一个录制周期，则会自动将录制文件添加到点播系统。
3. 点播记录完成后，会生成点播系统的唯一视频ID，并将该视频信息回调给用户，即AddLiveRecordVideoComplete通知，并附带直播相关的DomainName、AppName、StreamName信息。客户收到回调后，需记录该视频信息，并以该VideoId作为索引进行后续视频状态更新。
4. 客户断流超时或主动触发断流。
5. 点播收到直播发送的本次直播结束消息。
6. 点播系统检测用户录制配置中的合成配置，判断是否需要发起自动合成。如果需要，则按照录制配置中的合成转码组进行合成和转码。
7. 视频开始合成，点播会生成一个合成后视频的唯一视频ID，并将该视频信息回调给用户，即LiveRecordVideoComposeStart通知，并附带直播相关的DomainName、AppName、StreamName信息。客户收到回调后，需记录该视频信息，并以该VideoId作为索引进行后续视频状态更新。
8. 视频源文件合成完成后，点播会将该状态回调给用户，即FileUploadComplete通知。

9. 截图完成、单个码流转码完成、全部码流转码完成时，会给用户进行回调(回调顺序无时序)，用户需根据回调信息中的Videoid进行视频状态更新。转码完成后，即可进行后续的播放操作(转码回调信息中含播放地址或后续通过GetPlayInfo接口根据Videoid进行播放地址获取，该播放地址已经经过CDN加速)。

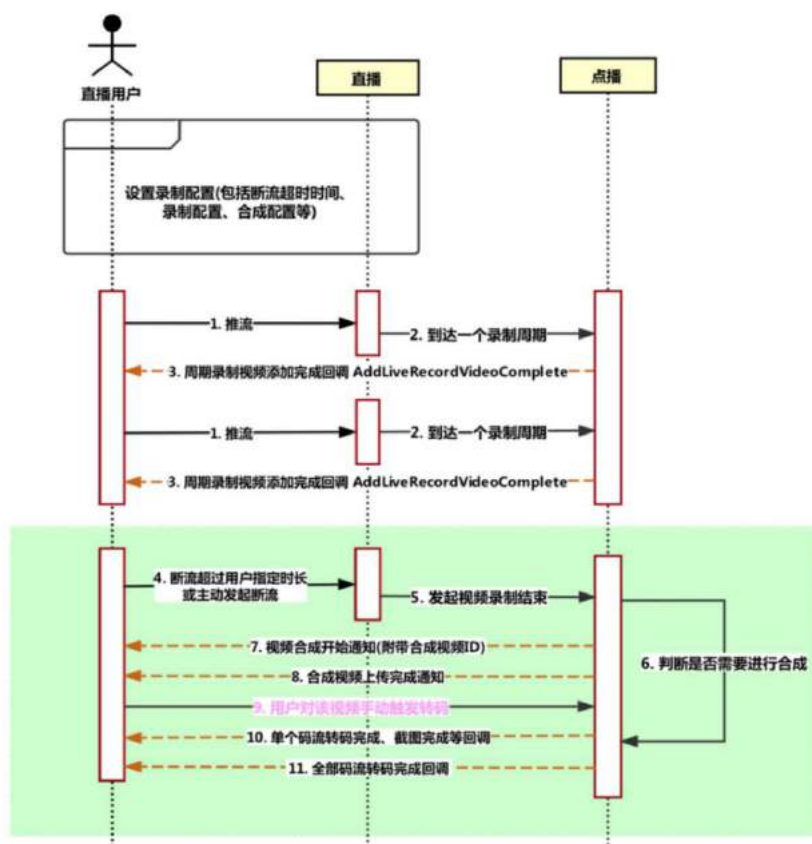
合成 + 手动发起转码

可在新建直播录制配置时，将自动合成开关打开，并且配置进行合成时所使用的合成转码模板组(本情况配置成仅合成，后续由用户触发转码)。

适用场景：

一场直播完成后，需要对所有录制周期内的分段进行自动合并，合并后文件不做任何处理，由用户进行二次加工，如云剪辑等，后续再进行手动转码，如视频内嵌广告、体育赛事空挡部分内容剪切等。

流程如下：



1. 客户进行直播推流。
2. 推流达到一个录制周期，则会自动将录制文件添加到点播系统。
3. 点播记录完成后，会生成点播系统的唯一视频ID，并将该视频信息回调给用户，即AddLiveRecordVideoComplete通知，并附带直播相关的DomainName、AppName、StreamName信息。客户收到回调后，需记录该视频信息，并以该VideoId作为索引进行后续视频状态更新。
4. 客户断流超时或主动触发断流。
5. 点播收到直播发送的本次直播结束消息。
6. 点播系统检测用户录制配置中的合成配置，判断是否需要发起自动合成。如果需要，则按照录制配置中的合成转码组进行合成和转码，由于本场景配置的仅合成，则点播系统不会自动发起转码。
7. 视频开始合成，点播会生成一个合成后视频的唯一视频ID，并将该视频信息回调给用户，即LiveRecordVideoComposeStart通知，并附带直播相关的DomainName、AppName、StreamName信息。客户收到回调后，需记录该视频信息，并以该VideoId作为索引进行后续视频状态更新。
8. 视频源文件合成完成后，点播会将该状态回调给用户，即FileUploadComplete通知。此时，代表本次录制合成的源文件已经处于正常就绪状态，用户可对该视频进行后续的转码触发等操作。
9. 用户对该视频手动触发转码操作(调用转码任务API)，在此之前可进行云剪辑等操作。
10. 截图完成、单个码流转码完成、全部码流转码完成时，会给用户进行回调(回调顺序无时序)，用户需根据回调信息中的VideoId进行视频状态更新。转码完成后，即可进行后续的播放操作(转码回调信息中含播放地址或后续通过GetPlayInfo接口根据VideoId进行播放地址获取，该播放地址已经经过CDN加速)。

播放地址获取实践

在使用视频点播服务时最常见的情况就是使用一个播放器去播放一个视频URL,这个播放器可能是h5播放器、安卓播放器、iOS播放器。那么如何获取视频播放的URL呢?

1、通过服务端接口获取

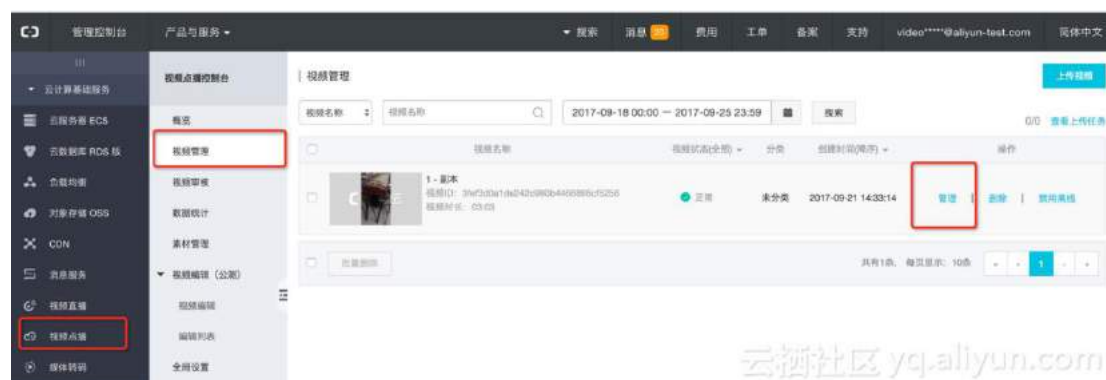
视频点播提供了openApi来让开发者通过videoid获取播放地址: ----->[openApi说明](#)

OpenApi需要用户传递视频id等信息。开发者从返回的参数PlayInfo中获取PlayURL即为视频流的播放地址。

2、通过控制台获取

用户在开通点播服务后在控制台可以直接查看到已经上传的视频url信息,具体查看路径:

阿里云控制台----->选择视频点播----->视频管理----->选择视频“管理”----->选中视频地址。





如此就可以获取到视频点播的播放地址了。

本文为【[问答合集](#)】[阿里视频云常见技术问题汇总——内含技术文档40+ 持续更新中](#) 系列文章之一，如果本文还不能解决你的问题可以在[汇总合集中](#)查看，有任何问题也欢迎回复，谢谢。

视频点播添加字幕实战

业务场景

点播服务上传的原始视频没有字幕，需要对点播视频加上视频字幕。

业务流程

1. 上传字幕文件
2. 添加带字幕的转码模版
3. 使用带字幕的转码模版转码

使用步骤

1、上传字幕文件

字幕文件无法通过控制台上传，需要通过上传SDK的上传接口上传，并且指定subtitle参数：

```
1.     from voduploadsdk.AliyunVodUtils import *from
    voduploadsdk.AliyunVodUploader import AliyunVodUploaderfrom
    voduploadsdk.UploadAttachedMediaRequest import UploadAttachedMediaRequest

2.     def testUploadLocalAttachedMedia(accessKeyId, accessKeySecret,
    filePath):

3.         try:

4.             uploader = AliyunVodUploader(accessKeyId, accessKeySecret)

5.             uploadAttachedRequest = UploadAttachedMediaRequest(filePath,
    'subtitle')

6.             uploadAttachedRequest.setTitle('test upload local watermark file')

7.             media = uploader.uploadAttachedMedia(uploadAttachedRequest, True)

8.             print(media)

9.         except AliyunVodException as e:

10.            print(e)
```



```
11.     localFilePath = '/Users/mac/Downloads/VodUploadSDK-  
    Python_1.3.1/dlfiles/subtitle-02.srt'  
12.     testUploadLocalAttachedMedia(accessKeyId, accessKeySecret,  
    localFilePath)
```

上传完成后会返回字幕文件的url信息:

```
{'FileURL': u'https://outin-a30b2feff15a11e8ad9400163e1a625e.oss-cn-shanghai.aliyun.com/subtitle/5D55590D555C40BA8DC9BF307D3AC28C-3-3.srt', 'MediaId': u'68928b9ab5744b65b7b04f0cca97f54d', 'MediaURL': u'http://vod.pier39.cn/subtitle/5D55590D555C40BA8DC9BF307D3AC28C-3-3.srt'}
```

2、设置转码模版

点播控制台的转码模版目前还不支持选择字幕, 所以也是通过接口来完成对带字幕转码模版的设置的:

```
1.     from aliynsdskcore import clientfrom aliynsdskvod.request.v20170321  
    import AddTranscodeTemplateGroupRequest  
2.     clt = client.AcsClient(access_key_id, access_key_secret, region)  
3.     def AddTranscodeTemplateGroup(clt):  
4.         request =  
    AddTranscodeTemplateGroupRequest.AddTranscodeTemplateGroupRequest()  
5.  
6.         transcodeTemplateList = []  
7.         transcodeTemplate = {}  
8.         # 设置模板名称  
9.         transcodeTemplate["TemplateName"] = "Subtitle_Transcode"  
10.        # 清晰度  
11.        transcodeTemplate["Definition"] = "LD"  
12.        # 视频流转码配置  
13.        videoConfig = {"Width": 640, "Bitrate": 400, "Fps": 25, "Remove":  
    False, "Codec": "H.264", "Gop": "250"}  
14.        transcodeTemplate["Video"] = videoConfig  
15.        # 音频流转码配置
```

```
16.     audioConfig = {"Codec": "AAC", "Bitrate": "64", "Channels": "2",
17.         "Samplerate": "32000"}
18.
19.     transcodeTemplate["Audio"] = audioConfig
20.
21.     # 封装容器
22.     container = {"Format": "m3u8"}
23.     transcodeTemplate["Container"] = container
24.
25.     # 添加字幕参数
26.     subtitleList = {"SubtitleUrl": "https://outin-
27.         a30b2feff15a11e8ad9400163e1a625e.oss-cn-
28.         shanghai.aliyuncs.com/subtitle/D973C24DA668448A80E235F9D020BEFE-3-3.srt",
29.         "CharEncode": "UTF-8"}
30.     transcodeTemplate["SubtitleList"] = subtitleList
31.     request.set_TranscodeTemplateList(json.dumps(transcodeTemplateList))
32.     request.set_Name('subtitle')
33.
34.     response = clt.do_action_with_exception(request)
35.     return response
36.     print AddTranscodeTemplateGroup(clt)
```

代码里subtitleList参数是设置字幕的，里面的SubtitleUrl参数填写第一步中返回的FileURL地址。

3、使用带字幕的转码模版转码

这里可以通过控制台的媒体处理选项选择上面创建的转码模版，或者通过转码接口来实现转码：

```
1.     from aliyunsdkcore import client
2.     from aliyunsdkvod.request.v20170321
3.     import SubmitTranscodeJobsRequest
4.
5.     clt = client.AcsClient(access_key_id, access_key_secret, region)
6.
7.     def SubmitTranscodeJobs(clt):
8.
9.         request = SubmitTranscodeJobsRequest.SubmitTranscodeJobsRequest()
10.        request.set_VideoId('4bee0ee927ad4bed8ca865471c6680a7')
```

```
6.     request.set_TemplateGroupId('89a90b2cb8fd9765747dd39fae31e')
7.     response = clt.do_action_with_exception(request)
8.     return response
9.     print SubmitTranscodeJobs(clt)
```

4、播放通过上面转码模版转码后的视频就会有字幕的效果了

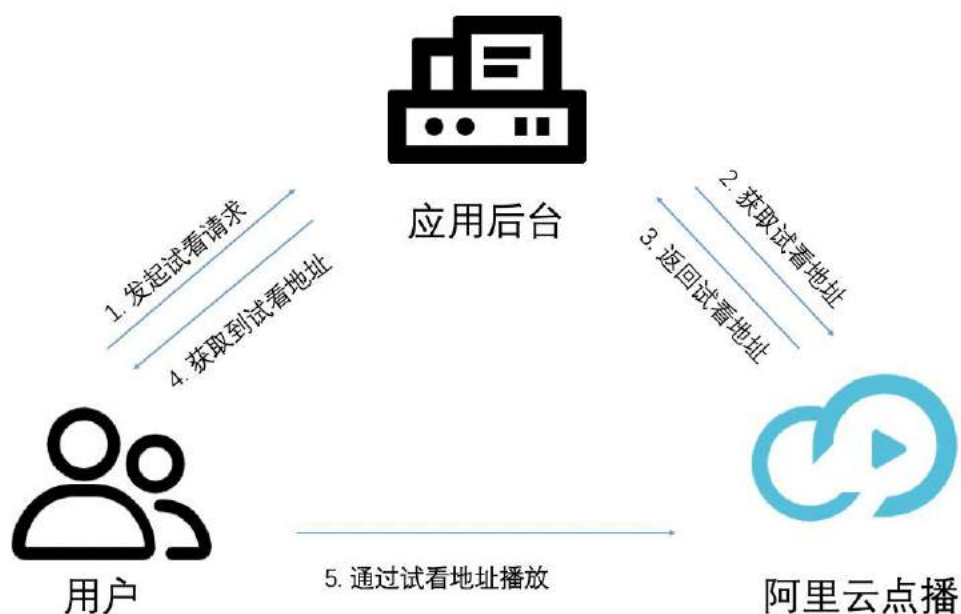


点播试看最佳实践

业务场景

在线教育场景下，提供视频课程给用户试看一段时间（如前5分钟），用户试看完后如果要继续观看，需要付费购买。

业务流程

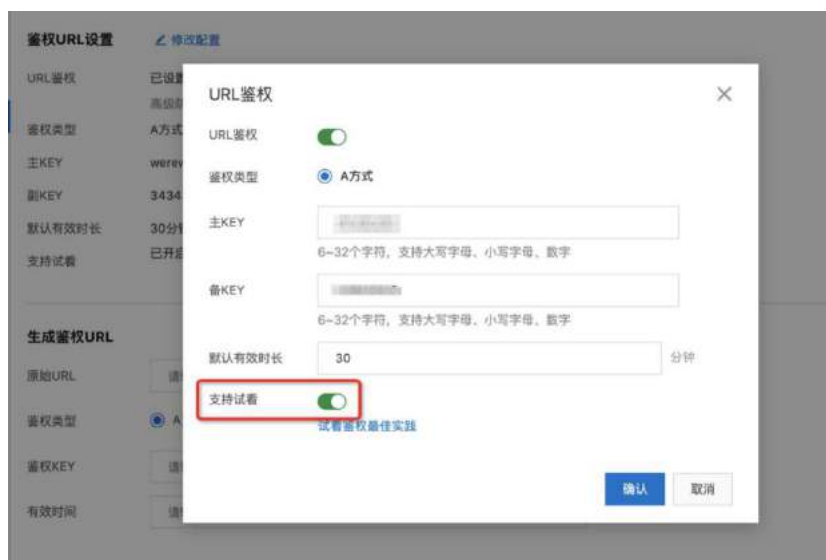


1. 客户端到应用后台请求试看地址
2. 应用后台请求阿里云点播服务，获取视频试看地址
3. 阿里云点播服务返回试看地址
4. 应用后台返回试看地址给客户端播放
5. 客户端通过试看地址播放

使用步骤

- 点播域名开启试看功能

登陆点播控制台，配置管理-分发加速配置-域名管理-配置(对应域名处)-访问控制-URL鉴权中，开启A方式鉴权，同时，勾选”支持试看”选项。



- 试看地址获取

调用 [获取播放地址接口](#)，通过参数 PlayConfig 结构中的 PreviewTime 来获取指定时长的播放试看地址。示例代码：

```
1. from aliynsdkvod.request.v20170321 import GetPlayInfoRequest
   from aliynsdkcore import client

2. access_key_id = "your access_key_id"

3. access_key_secret = "your access_key_secret"

4. region = "cn-shanghai"

5. videoId = "your videoId"

6. def GetPreviewURL(clt):

7.     request = GetPlayInfoRequest.GetPlayInfoRequest()

8.     request.set_VideoId(videoId)

9.     #设置set_PlayConfig参数指定试看时长为15秒

10.    request.set_PlayConfig({'PreviewTime':"15"})

11.    request.set_accept_format('JSON')
```

```
12.     response = clt.do_action(request)
13.     return response
14.     clt = client.AcsClient(access_key_id, access_key_secret,
    region)print GetPreviewURL(clt)
```

- 使用获取到的试看地址播放

获取到的播放地址格式（最后end=15参数说明是15s试看）：https://vod.aliyunvod.cn/8f026a6b21004454bc1524f3bfa71015/6c2bed8cd51e4e02b261d00a19ce76e4-3c166f6cad28fc09ff0ddada54f4ab08-fd.mp4?auth_key=1575626158-f644bdd753434db08813410fa04e4492-0-d7976721890fc12348ea4cabfb70b815&end=15

- 播放效果

实际播放可以看到视频只有15秒



注意事项

试看的基本原理是，播放的CDN加速地址带有试看的指定时长信息，云端会对该信息进行鉴权，鉴权通过会返回指定的文件内容，否则拒绝访问、返回403。

1. 点播试看功能基于阿里云CDN加速实现，且必须在视频点播(VOD)控制台配置CDN加速域名。
2. 此方案必须开启A鉴权，同时，为了防止试看参数被篡改，试看参数也作为auth_key计算的一部分。
3. 域名必须开启 range回源 和 拖拽播放。具体可在域名管理-配置(对应域名处)-视频相关处开启。

多场景化转码实践

背景概述

在面对不同行业用户丰富的转码场景需求时，如何将用户偏向自身业务特性的定制化需求通用化、产品化并赋予给其他用户使用？这是点播转码一直在思考并努力去解决的，本文由阿里云视频点播技术团队张立磊撰写，讲述视频点播针对多种业务场景提供的适应多场景化的转码处理方案，如何实现将用户定制化的场景需求进行抽象提取，最终以适用于其他点播用户的同样或类似的业务场景需求。

用户准备工作

- 开通点播服务，[开通服务](#)
- 添加转码模板组，[转码设置](#) > 上传仅存储、分发并转码模板组请联系点播后台提供响应的模板组

名词解释

- 不同规格视频：主要是指视频的分辨率、码率等编码参数不同的转码输出视频，这不同规格视频可以适应不同的网络带宽环境。
- 转码后分发：视频上传完成后，源片经转码处理后，输出不同规格视频再经CDN加速分发，供终端播放。
- 分发并转码：视频上传完成后，源片可经CDN加速供终端播放，同时会触发转码处理流程，输出不同规格视频。
- 不转码即分发：视频上传完成后，源片可经CDN加速播放，但不会触发任何转码处理流程。
- 上传仅存储：视频上传完成后，源片仅仅存储在OSS当中，不可经CDN加速分发供终端播放，也不触发任何转码处理流程。

通用视频转码场景

通常视频上传到点播会先经过转码处理后，生成不同规格的视频以适应不同的网络带宽环

境及多终端处理，然后经过CDN加速分发，最终供终端播放。

这类场景用户只需要在[点播控制台-->转码设置](#)，创建转码模板组并设置为默认模板组，然后上传视频即可，后续流程自动完成。

该场景的视频的处理流程如下图所示：



短视频不转码场景

由短视频SDK录制并上传的视频，这类视频的编码信息实际已经基本符合在网络带宽传输环境且能被各个终端兼容处理，因此用户可选择对这类视频无需再经由转码处理,而是直接通过CDN加速分发，既可快速响应播放请求，也可节约用户的转码成本。

这类场景用户只需在[点播控制台-->转码设置](#)，将不转码即分发模板组激活并设置为默认模板组，然后上传视频即可。

该场景的视频处理流程如下图所示：

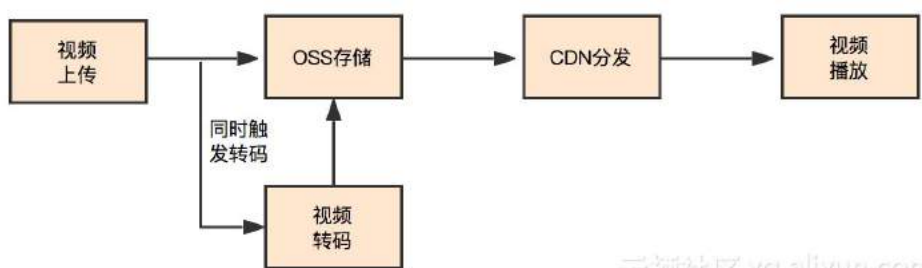


快速响应播放请求 + 多规格视频输出场景

视频上传至点播，即可用源片进行CDN加速分发供终端播放，这样可快速响应用户播放请求，但为了输出不同规格视频以适应不同带宽环境及终端兼容处理，同时可以触发相应转码处理流程，而当不同规格的视频生成后，则可以给用户提供不同规格选择并替换源片播放，以此可提升用户的播放体验。

这类场景用户可以联系点播后台提供分发并转码模板组并设置为默认模板组，然后上传视频即可，后续流程自动完成。

该场景的视频处理流程如下图所示：



云栖社区 yq.aliyun.com

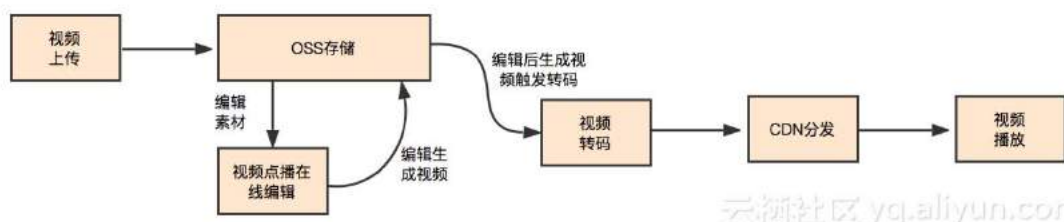
直播录制转点播场景

针对部分直播录制用户，他们的视频录制周期相对较长、文件较大，实际上这类用户真正关心的仅仅只是录制文件中的某些精彩片段，比如：游戏录制视频中的某些精彩操作、体育直播录制的精彩片段等等。如果将整个视频文件全部转码，将会给用户带来高昂的转码成本，一般对于用户来说都是难以接受的。

因此，如果将这类视频可以上传点播仅作为素材存储在OSS当中，而后可以用通过视频点播在线编辑能力，将精彩片段抽后取并生成新的视频，再去触发相应的转码流程生成不同规格视频，最终经过CDN加速分发供终端播放，这样不仅降低了用户的转码成本，也达到了用户去摘取精彩视频片段的目的。

这类场景用户可以联系点播后台提供上传仅存储模板组并设置为默认模板组，然后上传视频即可，后续可以通过视频点播的在线编辑功能去提取精彩内容。

该场景的视频处理流程如下图所示：



云栖社区 yq.aliyun.com

OSS SDK上传到点播方案实践

场景

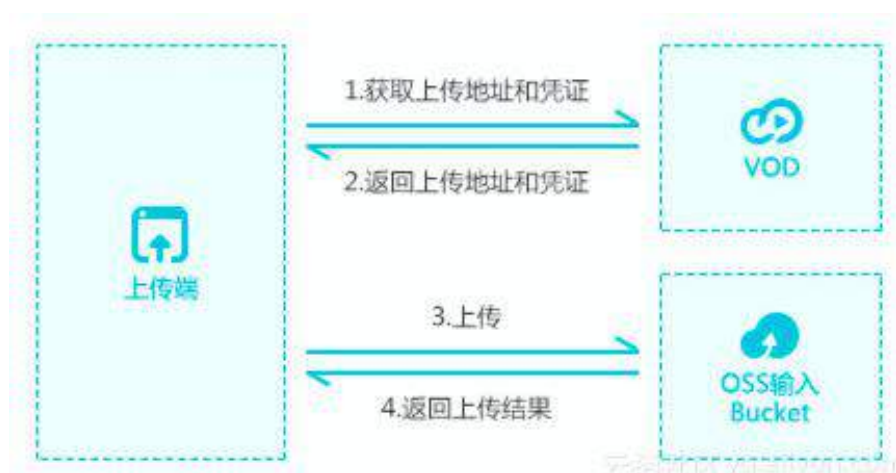
点播上传SDK缺乏需要的语言版本（如C/C++、Go等）或相应的功能（如网络流上传、追加上传），可以直接使用OSS的SDK进行上传。

准备工作

确认已开通点播服务并完成了相关配置。

确认已准备了阿里云账号AK，并授予了上传权限。

上传步骤



一、访问点播服务获取上传地址和上传凭证。

这一步还会创建视频媒资记录，返回视频ID，请妥善保存，后续可根据视频ID进行视频播放、管理和AI处理等。

二、对上传地址（UploadAddress）和上传凭证（UploadAuth）分别进行Base64解析，得到OSS的上传地址和授权信息。

- UploadAddress字段Base64解码后，得到JSON格式字符串，包含字段如下：

字段	含义
Bucket	OSS Bucket名称
Endpoint	OSS区域地址
FileName	分配的视频文件名，对应OSS的Object参数

- UploadAuth字段Base64解码后，得到JSON格式字符串，包含字段如下：

字段	含义
AccessKeyId	上传授权密钥ID
AccessKeySecret	上传授权密钥
SecurityToken	上传授权安全令牌
Expiration	上传授权过期时间，视频为3000秒，过期需要刷新上传凭证

三、调用OSS SDK将视频文件上传至指定的bucket中，注意使用STS Auth方式，且使用UploadAddress和UploadAuth进行初始化，不要使用自己的AK等信息。

代码实现

核心代码实现共分4步：

- 使用AK初始化VOD客户端
- 获取视频上传地址和凭证
- 使用上传凭证和地址初始化OSS客户端
- 上传本地文件

可以使用点播服务的接口SDK（[Java](#)、[PHP](#)、[Python](#)、[.NET](#)），或API（[接口说明](#)、[调用示例](#)）获取上传地址和凭证。

- 使用OSS SDK在服务端上传，目前支持如下版本：

[OSS-Java-SDK](#)、[OSS-PHP-SDK](#)、[OSS-Python-SDK](#)、[OSS-C-SDK](#)、[OSS-Go-SDK](#)、[OSS-Ruby-SDK](#)、[OSS-.NET-SDK](#)

客户端上传建议使用VOD的上传SDK：VOD-Android-SDK、VOD-iOS-SDK、[VOD-JavaScript-SDK](#)；客户端上传时需要在服务端获取上传地址和凭证下发给客户端，以确保数据安全。

PHP上传示例

环境准备

- PHP 5.3+，可通过php -v命令查看当前的PHP版本。
- cURL 扩展，可通过php -m命令查看curl扩展是否已经安装好。

安装

1. 在您的PHP项目中添加空文件夹 aliyun-php-sdk。
2. 从 [aliyun-openapi-php-sdk](#) 下载整个源码，解压后拷贝 aliyun-php-sdk-core 和 aliyun-php-sdk-vod 两个文件夹到 aliyun-php-sdk 目录下。
3. 从 [aliyun-oss-php-sdk](#) 下载最新的OSS PHP SDK的源码，解压ZIP文件后添加文件夹到 aliyun-php-sdk 目录下。以下载 [v.2.2.4 Source code \(zip\)](#) 为例，解压后文件夹为 aliyun-oss-php-sdk-2.2.4。
4. 打开 aliyun-php-sdk/aliyun-php-sdk-core/Config.php 文件，找到 “//config sdk auto load path.”，在这行下面添加：

```
Autoloader::addAutoloadPath("aliyun-php-sdk-vod");
```

5. 在代码中引用VOD和OSS的文件：

```
1.     require_once './aliyun-php-sdk/aliyun-php-sdk-core/Config.php';  
    // 假定您的源码文件和aliyun-php-sdk处于同一目录。  
2.     require_once './aliyun-php-sdk/aliyun-oss-php-sdk-  
    2.2.4/autoload.php';  
3.     use vod\Request\V20170321 as vod;  
4.     use OSS\OssClient;  
5.     use OSS\Core\OssException;
```

更多信息可参考[VOD PHP SDK安装](#) 和 [OSS PHP SDK安装](#)。

参考代码

定义核心步骤的函数：

1. 使用AK初始化VOD客户端

```
1.     function init_vod_client($accessKeyId, $accessKeySecret) {  
2.         $regionId = 'cn-shanghai';    // 点播服务所在的Region，国内请填写cn-shanghai，不要填写别的区域  
3.         $profile = DefaultProfile::getProfile($regionId, $accessKeyId, $accessKeySecret);  
4.         return new DefaultAcsClient($profile);  
5.     }
```

2. 获取视频上传地址和凭证

```
1.     function create_upload_video($vodClient) {  
2.         $request = new vod\CreateUploadVideoRequest();  
3.         $request->setTitle("视频标题");    // 视频标题(必填参数)  
4.         $request->setFileName("文件名称.mov"); // 视频源文件名称，必须包含扩展名(必填参数)  
5.         $request->setDescription("视频描述"); // 视频源文件描述(可选)  
6.         $request->setCoverURL("http://img.alicdn.com/tps/TB1qnJ1PVXXXXCXXXXXXXXXXXX-700-700.png"); // 自定义视频封面(可选)  
7.         $request->setTags("标签1,标签2"); // 视频标签，多个用逗号分隔(可选)  
8.         return $vodClient->getAcsResponse($request);  
9.     }
```

3. 使用上传凭证和地址初始化OSS客户端（注意需要先Base64解码并Json Decode再传入）

```
1.     function init_oss_client($uploadAuth, $uploadAddress) {
2.         $ossClient = new OssClient($uploadAuth['AccessKeyId'],
3.             $uploadAuth['AccessKeySecret'], $uploadAddress['Endpoint'],
4.             false, $uploadAuth['SecurityToken']);
5.         $ossClient->setTimeout(86400*7);    // 设置请求超时时间，单位秒，默认
6.             是5184000秒，建议不要设置太小，如果上传文件很大，消耗的时间会比较长
7.         $ossClient->setConnectTimeout(10); // 设置连接超时时间，单位秒，默认
8.             是10秒
9.         return $ossClient;
10.    }
```

4. 上传本地文件

```
1.     function upload_local_file($ossClient, $uploadAddress, $localFile)
2.     {
3.         return $ossClient->uploadFile($uploadAddress['Bucket'],
4.             $uploadAddress['FileName'], $localFile);
5.     }
```

5. 刷新上传凭证

```
1.     function refresh_upload_video($vodClient, $videoId) {
2.         $request = new vod\RefreshUploadVideoRequest();
3.         $request->setVideoId($videoId);
4.         return $vodClient->getAcsResponse($request);
5.     }
6.     执行完整流程（注意捕获异常）：
7.     $accessKeyId = '<AccessKeyId>';           // 您的
8.     $accessKeySecret = '<AccessKeySecret>';     // 您的
9.     $localFile = '/Users/yours/Video/testVideo.flv'; // 需要上传到VOD
10.    的本地视频文件的完整路径try {
```

```
10.      // 初始化VOD客户端并获取上传地址和凭证
11.      $vodClient = init_vod_client($accessKeyId, $accessKeySecret);
12.      $createRes = create_upload_video($vodClient);
13.      // 执行成功会返回VideoId、UploadAddress和UploadAuth
14.      $videoId = $createRes->VideoId;
15.      $uploadAddress = json_decode(base64_decode($createRes->UploadAddress), true);
16.      $uploadAuth = json_decode(base64_decode($createRes->UploadAuth), true);
17.      // 使用UploadAuth和UploadAddress初始化OSS客户端
18.      $ossClient = init_oss_client($uploadAuth, $uploadAddress);
19.      // 上传文件，注意是同步上传会阻塞等待，耗时与文件大小和网络上行带宽有关
20.      // $result = upload_local_file($ossClient, $uploadAddress, $localFile);
21.      $result = multipart_upload_file($ossClient, $uploadAddress, $localFile);
22.      printf("Succeed, VideoId: %s", $videoId);
23.      } catch (Exception $e) {
24.      // var_dump($e);
25.      printf("Failed, ErrorMessage: %s", $e->getMessage());
26.      }
```

Demo下载

可下载完整的 [PHP版上传Demo源码](#)，更多使用信息请参考[OSS-PHP-SDK上传文件](#)。

为了扶持更多创业者，降低入行门槛，视频点播已在原有的视频点播服务五个套餐包基础上，推出了超值体验套餐包。

只需9.9元即可获得10GB流量、50GB存储、100分钟转码，可供个人、小微企业门户网站体验使用。



点击进入活动页面，立即购买，完成支付



进入[视频点播产品页面](#)，点击立即开通，开通视频点播服务，进行初始化配置，用户可以在控制台进行视频上传，转码设置、视频管理和视频预览等操作。



另外，体验完成后，阿里云也为提供了五个点播套餐包，供各种需求用户选择。



点播视频安全宝典系列1

概述

如何保障视频内容的安全，不被盗链、非法下载和传播，是众多企业困扰已久的问题，特别是在在线教育、财经金融、行业培训等在线版权视频领域尤为迫切，处理不好会造成极为严重的经济损失，甚至法律风险。

阿里云视频点播提供了完善的内容安全保护机制，本系列文章将全面介绍这些安全保护机制，以满足不同业务场景的安全需求。

安全方案对比

视频点播提供了多个纬度的安全方案，下面的表格对比了不同安全方案的特性，可以根据实际的业务情况选择。

安全机制	安全方案	安全特点	安全等级	技术门槛
访问限制	IP黑白名单	基于客户端IP限制，不适合面向大量To C用户的场景	低	低，仅需控制台配置
	Refer防盗链	基于Http Header判断来源，但容易被伪造	低	低，仅需控制台配置
带鉴权的URL	URL鉴权	动态生成请求URL，自定义过期时间	中	较低，需要服务端API请求或者根据鉴权Key生成
视频加密	阿里云加密	使用阿里云私有加密，但需要使用阿里云播放器且不支持H5	高	中，控制台配置+服务端调用API接口+集成阿里云播放器
	HLS标准加密	使用HLS通用加密方案，适配所有播放器，但也会存在被下载的可能	较高	高，需自行搭建转码、令牌分发、解密服务
安全下载	对下载到本地的适配加密	视频被加密后存储，只能使用唯一的应用解密播放	高	较低，简单集成+集成阿里云播放器

下面的章节将会对上面列出的安全方案逐一介绍，详细描述方案特点、实现方法和使用限制。

点播视频安全宝典系列2

简介

本章节主要介绍通过访问限制配置来实现对点播视频的保护。访问限制类的视频保护使用门槛低、快速生效等优点。

IP黑白名单

视频点播支持配置IP黑名单或白名单，拒绝或只允许特定IP的访问。

操作步骤

1. 登录VOD控制台，选择**域名管理**。
2. 单击目标域名后的**配置**，选择**访问控制**。
3. 选择**IP黑/白名单**，单击**修改配置**进行设置。

比如下图是针对120.55.XXX.34这个IP设置黑名单：

IP黑/白名单

名单类型

☒ IP黑名单

☐ IP白名单

黑、白名单互斥同一时间只支持一种方式（当时所选方式）

规则

120.55.XXX.34

最多100个，使用回车符分隔，不可重复，支持网段添加 如：
127.0.0.1/124

确定

取消

测试可以发现通过这个IP请求点播视频的时候会返回403报错，报错原因是IP黑名单报错：denied by IP ACL = blacklist

```
[root@iZbp1cd217ux0klkx1mlbZ ~]# curl -I https://vod.pier39.cn/customerTrans/ad8c4b5b23d35bef4292c87e29210d1d/912c8ee-170140abf6f-0006-1da4-0bc-8bc7f.mp4
HTTP/1.1 403 Forbidden
Server: Tengine
Date: Wed, 05 Feb 2020 07:00:35 GMT
Content-Type: text/html
Content-Length: 254
Connection: keep-alive
X-Tengine-Error: denied by IP ACL = blacklist
Access-Control-Allow-Origin: *
Via: cache4.cn1576[,403004]
Timing-Allow-Origin: *
EagleId: 6525b79815808860352914242e
```

Referer防盗链

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，可配置访问的Referer黑、白名单来限制视频资源被访问的情况。

操作步骤

1. 登录VOD控制台，选择域名管理。
2. 单击目标域名后的配置，选择访问控制。
3. 选择Refer防盗链，单击修改配置进行设置。

比如下图是针对aliyun.com这个域名设置了白名单，且不允许空Referer访问：

Refer防盗链

Refer类型

☐ 黑名单

☒ 白名单

黑、白名单互斥，同一时间只支持一种方式（当时所选方式）

规则

aliyun.com

使用回车符分隔多个Refer名单，支持通配符，如a.*b.com可以匹配到a.aliyun.b.com或a.img.b.com等

☐ 允许通过浏览器地址栏直接访问资源URL

允许空 Referer 字段访问CDN资源

确定

取消

测试可以发现如果请求时候Referer参数不是aliyun.com这个域名就会有报错：denied by Referer ACL

```
[root@izbp1cd217ux0klkx1mlbZ ~]# curl -I https://vod.pier39.cn/customerTrans/ad8c4b5b23d35bef4292c87e29210d1d/912c8ee-170140abf6f-0006-1da4-0bc-8bc7f.mp4
HTTP/1.1 403 Forbidden
Server: Tengine
Date: Wed, 05 Feb 2020 07:00:35 GMT
Content-Type: text/html
Content-Length: 254
Connection: keep-alive
X-Tengine-Error: denied by IP ACL = blacklist
Access-Control-Allow-Origin: *
Via: cache4.cn1576[,403004]
Timing-Allow-Origin: *
EagleId: 6525b79815808860352914242e
```

注意事项

- 可以通过点播的日志查看用户的IP和Referer：
https://help.aliyun.com/document_detail/86099.html
- IP黑/白名单支持添加IP网段，例如127.0.0.1/24。
网段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 $32-24=8$ bit来表示主机号，该子网可以容纳 $2^8 - 2 = 254$ 台主机，故可表示IP网段范围：127.0.0.1~127.0.0.255。
- Referer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。
- 由于移动端一般拿不到Referer，当前默认支持空Referer访问，可选择关闭。

点播解决方案

播放常见所需参数介绍

阿里云视频点播总是会出现Videoid、AccessKeyId、AccessKeySecret、playKey、playauth这几个值，这些值到底是什么？为什么会需要这些值？这些值到底从哪里拿到？有什么区别呢？本文将试图将这件事情讲清楚。

1. 获取videoid

1.1 videoid是什么？为什么需要videoid？

上传点播之后的视频ID.主要还是安全性考虑，用户拿到的都是一个一个的视频ID，而不是视频URL，避免被爬数据。（当然也可以拿到视频URL----->传送门）

1.2 这个值从哪里来？

使用点播上传完成之后都会得到一个videoid.在阿里云点播控制台里面的视频列表里面会有一个“视频ID”即为videoid.可以拿到控制台的视频ID来做下载和播放的测试。

如何实现点播上传见：

https://help.aliyun.com/document_detail/52200.html?spm=a2c6h.12873639.0.0.3188258a2DrbXy

2. 获取AccessKeyId和AccessKeySecret

2.1 AccessKeyId和AccessKeySecret是什么？

阿里云 access key ID 和 access key secret 是您访问阿里云API的唯一凭证。Access key ID 是类似身份的标识，而 access key secret 的作用是签名您的访问参数，以防被篡改。Access key secret 类似您的登录密码，不要向任何人泄漏。

2.2 这个值从哪里来？

- 1.登录阿里云官方网站。
- 2.单击页面上方菜单控制台。
- 3.鼠标放在右上方的用户名区域，在弹出的快捷菜单中单击AccessKeys。
- 4.系统弹出安全提示对话框，单击 继续使用AccessKey。页面显示 Access Key ID 和 Access Key Secret 。

3. 获取playKey(同apikey)

3.1 playKey是什么?

playkey(apikey):播放密钥,用于播放器SDK获取视频播放地址时验证身份,播放鉴权是视频点播在阿里云AK安全认证基础上的二次鉴权机制。

播放密钥用于播放器SDK获取视频播放地址时验证身份,可有效防止盗链。

根据用户播放时可能使用的平台,默认提供Flash、H5、iOS、Android四个平台的播放密钥。

为保证密钥安全,查看播放密钥时需要输入手机号验证码确认身份。

3.2 这个值从哪里来?

- 1.登录阿里云官方网站。
- 2.单击页面上方菜单控制台。
- 3.选择云计算基础服务。
- 4.选择视频点播。
- 5.选择安全设置。
- 6.选择播放鉴权。
- 7.选择对应的平台点击显示即可。

4. 获取playauth

4.1 playauth是什么?

目前播放器播放视频分为三种模式,三种模式有着不同的使用场景.playauth就是最安全的方法,也就是setAuthInfo的方式来实现。

播放模式	适用场景	优劣	是否建议使用
setDataSource	适用于测试时为了方便测试使用	危险，需要将自己的ak写死在客户端，客户端如果被破解将有泄露风险	不再建议使用
setAuthInfo	适用于正式商用	安全，所有的视频地址和链接都不暴露	建议商用使用
播放本地和网络URL	可以播放本地视频且能够播放视频URL	简单，可以播放其他平台的视频	在有播放本地视频需求和播放网络视频需求时使用.

4.2 这个值从哪里来？

playauth可以理解为点播服务将所有的信息(Videoid、AccessKeyId、AccessKeySecret)都做了一个混合然后做了一个简单的加密。这样用户拿到的就是一串包含多个信息的数据.这样播放器就可以播放了。

如何获取见以下流程：

流程：服务端获取播放凭证 -> 将播放凭证下发给客户端 -> 完成视频播放。

- 1. 获取播放凭证

客户在服务器侧通过调用播放鉴权SDK(server端的SDK)，向视频点播服务获取播放凭证。

注意：播放凭证时效为100秒，只能用于获取指定视频的播放地址，不能混用或重复使用；如果凭证过期则无法获取播放地址，请重新获取凭证。

- 2. 完成视频播放

播放器SDK根据视频ID和播放凭证向视频点播服务获取视频的播放地址，从而加载视频流并解码完成播放。

注意：

(1)播放器SDK根据播放凭证会自动获取播放地址进行解码播放，播放地址的时效为30分钟，若失效请重新获取播放凭证回传给播放器SDK用于刷新播放地址。

(2)为保障主账号安全，建议使用子账号的Access Key，尤其是Web播放场景下。

4.3 通过python脚本获取播放凭证进行调试?

通过PlayAuth方式去播放云视频，具体参数获取方法查看文档 [获取播放凭证](#)。建议使用此方式播放视频。

客户端开发也可以通过python脚本获取播放凭证进行调试，具体流程如下：

- 安装python2.7+pip环境（Mac环境下自带，Windows环境自行安装）
- 使用终端安装SDK，运行以下命令：

```
pip install aliyun-python-sdk-core
```

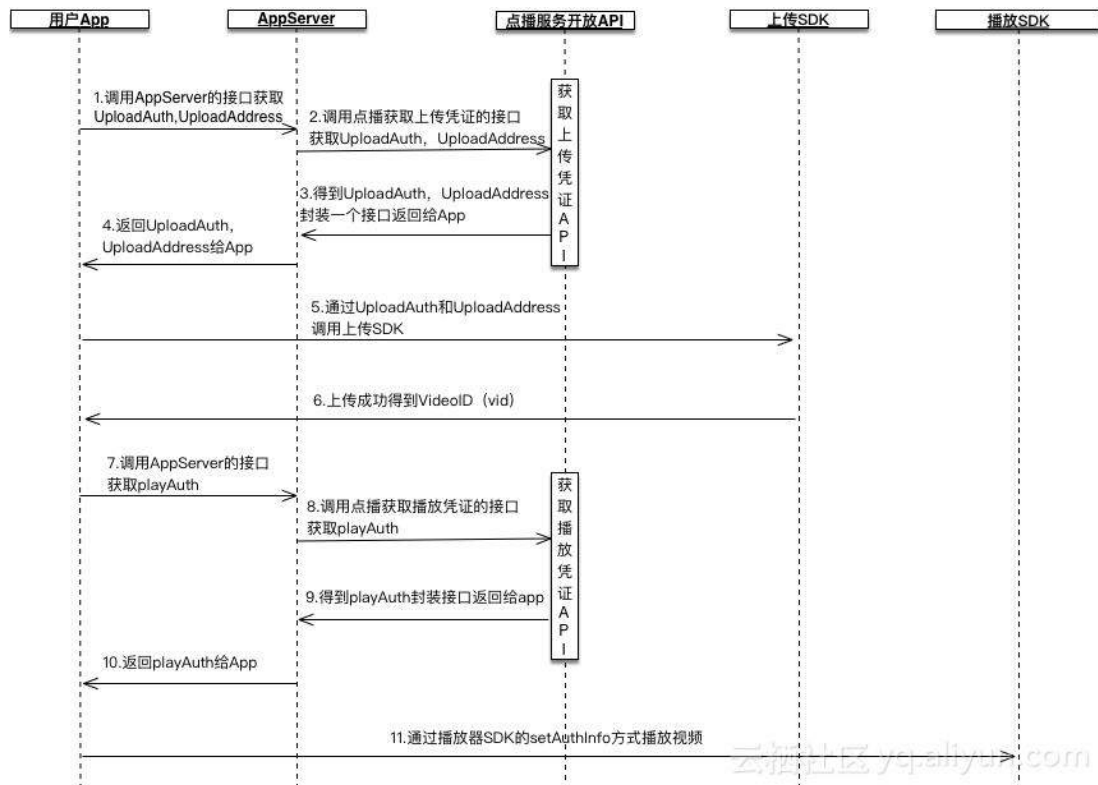
```
pip install aliyun-python-sdk-vod
```

- 下载Python脚本，从阿里云控制台获取accessKeyId和accessKeySecret并替换脚本里面的字段内容，从点播控制台获取已经上传视频的videoId并替换脚本里面的字段内容。
- 在python脚本所在目录下通过终端执行以下命令：

```
python playAuth.py
```

- 在终端中查看获取的PlayAuth和VideoId。
- 在播放器SDK中使用获取的PlayAuth和VideoId进行播放。

4.4 上传、播放最安全且最推荐的做法流程（重要）



本文为【[问答合集](#)】阿里视频云常见技术问题汇总——内含技术文档40+ 持续更新中系列文章之一，如果本文还不能解决你的问题可以在汇总合集中查看，有任何问题也欢迎回复，谢谢。

播放器SDK播放问题实战排查

使用阿里云播放器

目前随便手机型号的增加，和大量短视频的接入，VOD 点播播放器会遇到各种奇葩问题，今天从几个非常多的案例入手讲下坑点。

案例分析：ios 3.4.7 播放器 crash

现象：

客户 app 播放教学视频时初始化播放器后播放时直接 crash 。

分析：

出现问题，先看下视频编码，发现客户源文件是 vcodec 是 h264，acodec 是 amr，可以使用 ffprobe 命令分析视频的编码。而点播播放器目前仅支持 MP3、AAC 两种 acodec。

```
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'https://oss-cn-beijing.aliyuncs.com/lesson/ppt/106180/M7k5ESwyY7.mp4':
  Metadata:
    major_brand      : 3gp4
    minor_version    : 512
    compatible_brands: isomiso23gp4
  Duration: 00:04:10.27, start: 0.000000, bitrate: 216 kb/s
  Stream #0:0(eng): Video: h263 (s263 / 0x33363273), yuv420p(progressive), 176x144 [SAR 12:11 DAR 4:3], 189 kb/s, SAR 228:157 (default)
  Metadata:
    handler_name     : VideoHandler
  Stream #0:1(eng): Audio: amr_wb (sawb / 0x62776173), 16000 Hz, mono, flt, 24 kb/s (default)
  Metadata:
    handler_name     : SoundHandler
```

结论：

已知的一个问题 3.4.7 的播放器在播放的音频不支持的情况下会出现 crash，正常的情况应该是不支持播放会卡住而不是 crash，此问题会随着新版本发布解决掉。

案例分析：ios 3.4.7 切后台会从头重新播放

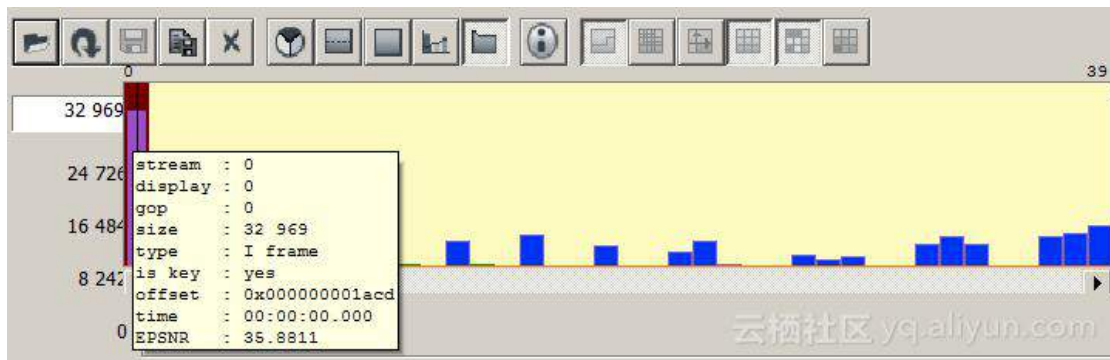
客户播放架构：

客户集成 3.4.7 的 demo 后，本地调苹果手机的剪辑库后，将文件转码上传都点播平台，app 收到点播 URL 后，播放短视频的 URL 。

现象：

应用切到后台，再切回应用中时，视频从 starttime=0 的位置重新播放，而不是从当前视频开始播放，而 android 没有这种问题。

分析：



- 问题分析还是老套路，先看下原视频分析下基础的内容。以下是视频的封装帧的内容。比较特殊，全场 27s 秒的小视频只有一个 I 帧。
- 3.4.7 的 ios 点播播放是通过硬解的方式 decode，播放时是寻找 I 播放，而软件码是可以从当前的视频开始播放。

结论：

- 3.4.7 的 ios SDK 利用硬解如果找不到 I 帧就会重头开始播放，而 android 是软解，直接从当前视频帧开始播。经过 MTS 产品对视频补帧后，直接传输到点播后，再次播放就不会从头播放了，但是会在下一个 I 帧找到之前会有卡顿。

案例分析：web 播放器 android 播放异常

```
{type: "error", target: div#player-live.prism-player, paramData: {...}, relatedTarget: und
efined, preventDefault: f, ...}
  defaultPrevented: false
  ▶ isDefaultPrevented: f i()
  ▶ isImmediatePropagationStopped: f i()
  ▶ isPropagationStopped: f i()
  ▼ paramData:
    cdnIp: "219.141.176.98"
    display_msg: "获取m3u8文件失败(manifestLoadError)"
    error_code: 4006
    error_msg: "manifestLoadError"
    localIp: "1.202.220.123"
    mediaId: ""
    requestId: "521D1B86-E100-4A58-BF52-7A0F92296C96"
    uuid: "F341EA60-137C-43A9-81C1-838C1A4258B5"
    vt: 0
    ▶ __proto__: Object
  ▶ preventDefault: f ()
  ▶ relatedTarget: undefined
  ▶ stopImmediatePropagation: f ()
  ▶ stopPropagation: f ()
  ▶ target: div#player-live.prism-player
    type: "error"
    which: undefined
  ▶ __proto__: Object
```



背景:

android 的手机全都不能播放, ios 的手机可以播放。

分析:

android 出现这种异常是对 跨域协议的只有有问题, 而 m3u8 是 HLS 协议苹果支持的, Safari 对跨域协议有良好的支持所以不会铲平报错。

案例分析: ios 3.4.7 播放器播放出现慢镜头



分析:

先分析原视频是否存在异常, 可以用 ffmpeg 播放看下, 看是否出现异常, 发现原视频在第 50s 左右没有视频包只有音频包, 在 1分25秒 左右恢复了视频和声音。

结论:

- 针对第 50s 没有视频的画面 ios SDK 的行为是直接 seek 过异常的 duration, 跳到 1分 25秒。
- 在画面恢复后, 由于音视频不同步, 所以 ios 会在短暂的几帧内进行音视频同步所以会比较慢。

案例分析：



背景：

web 点播 SDK同层播放时，点播列表向上滑动时，view 层不是固定的，会随着屏幕向上滑动将视频窗口覆盖。

结论：

出现这种问题不是播放器的 bug，是 cs 代码没有调整好，客户端需要自己调整下 cs 的代码，提供一个参考建议。

```
1.     extraInfo:{
2.       "x5-playsinline":""
3.     },
4.     x5_type:""
```


短视频解决方案介绍

相信很多人都已经被从去年年底接开始踵而来的短视频行业融资、战略事件、新品发布刷屏，短视频已经走上了风口浪尖，成为大众用户和行业人士关注的对象。

实际上，互联网的内容行业正在进阶，从文字、图片到视频、直播，再到能填补用户碎片时间的短视频，这是一个趋于互动性、实时性的迭代。而且，4G普及、资费下调、编解码技术进步和移动硬件的品质提升，也为短视频的发展提供了良好客观因素。从产品形态层面看，短视频本身也拥有创作门槛低、内容精炼、易于发酵等属性，更易于传播。综合这几项，短视频创业才如此蓬勃。那作为国内领先的云服务厂商，释放更多技术能力，给开发者带来便利是阿里云义不容辞的责任。

期待已久的阿里云短视频解决方案终于上线了。集短视频拍摄SDK、上传、视频点播等功能的一站式短视频解决方案。短视频SDK提供短视频录制和编辑基本功能，支持多种分辨率选择、实时美颜、实时滤镜、摄像头切换、闪光灯切换、视频画面和时长裁剪，同时支持视频文件上传到视频点播平台，提供丰富的回调。

五大核心优势

1. 提供产品级SDK，最快2小时接入，节省自行开发耗费的人力物力，助你快速实现APP短视频功能。
2. 接口简单易用，开放性强，标准版（UI开源）可以根据业务自由定制UI。
3. 录制功能自带断点录制、实时滤镜、高效美颜功能，支持本地视频导入压缩裁剪。
4. 迭代打磨，稳定可靠，视频技术经钉钉、美柚、梨视频、迅雷、贝贝网、宝宝树、蚂蜂窝等1000+应用商用验证，稳定可靠。
5. 除短视频端功能外，可结合阿里云全链路的视频云服务和云计算产品，需求一站式解决。

功能说明

1. 录制：支持断点录制、回删、点击拍摄、长按拍摄、美颜、实时滤镜、闪光灯、摄像头切换、分辨率设定等功能。
2. 导入：支持从相册选择视频、按视频时长和画面进行裁剪。



视频录制界面



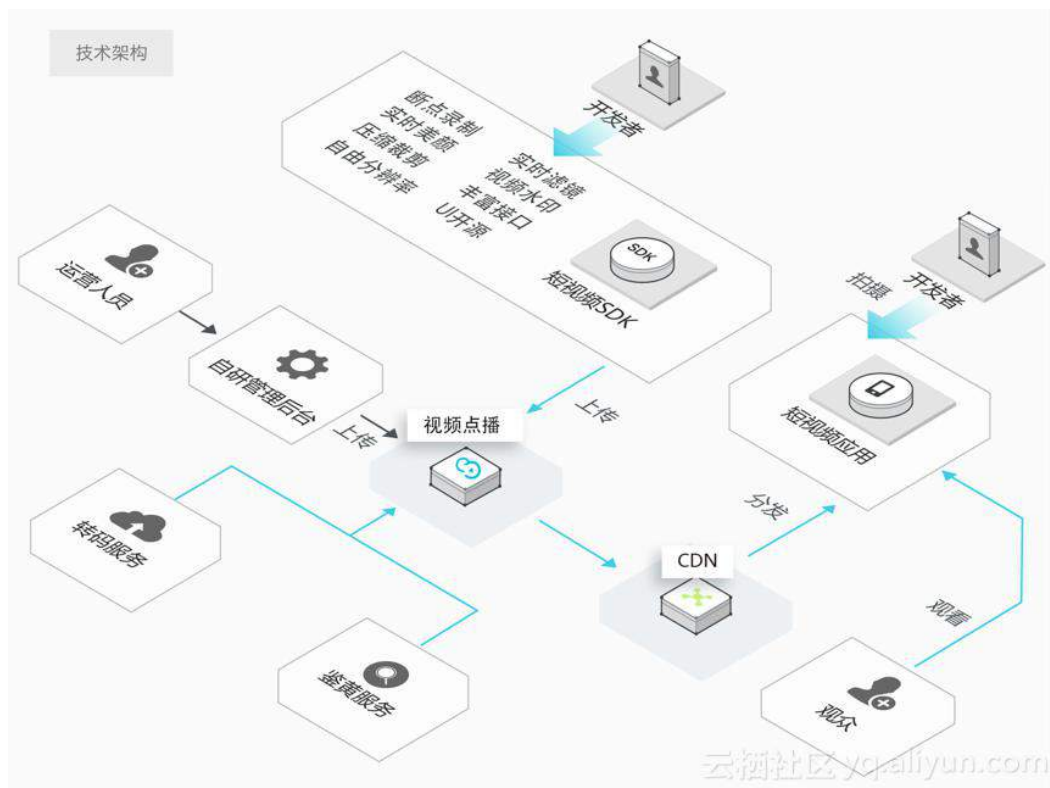
视频导入界面

3. 上传：上传本地文件到阿里云视频点播平台。

产品功能对比

功能点	功能说明	基础版	标准版
自定义UI	SDK包含一套默认的UI，布局、交互、界面可二次开发，基础版支持图标和背景颜色替换、标准版UI完全自定义	×	√
UI开源	提供完整的UI交互源码，用户可定制UI界面	×	√
多段录制	支持断点拍摄和连续拍摄	√	√
自定义时长	自定义最长和最短拍摄时长	√	√
摄像头切换	选择使用前/后摄像头进行录制	√	√
闪光灯	支持打开、关闭、自动闪光灯模式	√	√
焦距调节	录制中可调节画面焦距进行放大缩小	√	√
自定义分辨率及质量	可设定拍摄的画面尺寸、比例和质量，基础版仅支持9:16、3:4、1:1三种比例，标准版任意分辨率	√	√
美颜	录制实时美颜，平滑无极调整强度	√	√
滤镜	拍摄预览界面实时切换滤镜	√	√
相册选择	支持从相册过滤视频，也支持视频时长过滤	√	√
视频裁剪	支持视频画面大小和时长裁剪，同时支持画面填充和画面裁剪	√	√
原比例裁剪	支持保持原始视频比例裁剪视频时长	×	√
单视频导入	支持单视频导入，跳转进入用户定义的页面	√	√

产品架构



license申请说明（请仔细查看）

1. 购买点播SaaS预付费套餐三或套餐四。套餐三对应基础版SDK、套餐四对应标准版SDK。【立刻购买】
2. 购买套餐后请提供订单号、bundleID、包名和签名，并发送至videosdk@service.aliyun.com，以便为您开通短视频SDK License。
3. license不需要集成到SDK里面，只需确保提交申请的bundleID、包名、签名和自己工程中的完全一致。测试时可以直接使用demo提供的bundleID、包名、签名来体验。
4. 如果成功获取了license，在运行app时还显示license无效，请先清除下工程再运行app。

[点击了解产品详情](#)

端SDK相关汇总

前言

我们发现现在非常多的客户对阿里视频云服务了解不多，本文档的核心目标是将用户使用阿里视频云服务（点播，直播，连麦）的常见问题做一个系统的归纳.这可能不仅仅是一个FAQ，我们会试图将具体实现给开发者讲清楚，希望开发者能够更加深入的了解视频云的核心团队的架构设计方案.并且提供一系列的解决方案.个人认为端SDK是最接近客户的，所以所有的问题出发点都是从端SDK来讲解的，但是会试图结合云端服务的设计将问题讲清楚。

汇总常见问题方案

1.1 点播服务相关

- 【视频点播】点播服务是什么？为什么需要点播服务？点播服务由哪些部分组成？
- 【视频点播】怎么样快速使用视频点播服务？（快速入门）

1.1.1 短视频端SDK

- 【短视频SDK】短视频基础版、标准版、专业版的差异
- 【短视频SDK】短视频SDK Demo模块：魔法相机、拍摄、导入裁剪、导入编辑的差异
- 【视频云-端SDK】关于安卓ABI的兼容
- 【视频云-端SDK】iOS库冲突的问题---动态库导入
- 【短视频SDK】如何做到视频原始比例裁剪
- 【短视频SDK】短视频SDK整体License的策略
- 【短视频SDK - 参数解析】对焦模式、裁剪模式、视频质量、分辨率、视频比例、帧率、关键帧间隔等参数解析
- 【视频云端-科普】什么是视频封装格式、编码格式？目前SDK对封装格式，编码格式的支持？
- 【短视频SDK】如何获取单视频的缩略图？
- 【短视频SDK】如何实现变速录制做一个像抖音、muse这样的短视频应用
- 【短视频SDK】如何开启人脸识别做到像faceu一样的人脸动图效果？如何集成第三方人脸识别库？
- 【短视频SDK】合成时长对比数据---给参数设置提供参考.
- 【短视频SDK】Android如何使用硬编硬解？
- 短视频SDK的资源自定义如何做(音乐、动图、字幕动图等)
- 关于编辑界面MVC的设计--暴露View 让开发者自由修改
- 【短视频SDK的FAQ】关于iOS的pod导入，Android的maven导入

- 【短视频SDK】如何导入Android的AAR? 动态库so文件到底怎么样放呢?
- 【短视频SDK】关于iOS提供的SDK中模拟器版本(debug)和release 版本

1.1.2 播放器端SDK

- 【播放器端SDK】视频点播播放器如何实现加密下载?
- 【播放器端SDK】点播和播放器下载需要的参数的区别
- 【播放器端SDK】加密下载的安全文件是什么? 如何获取?
- 【播放器端SDK】点播播放器和直播播放器冲突如何解决?
- 【播放器端SDK】点播带皮肤播放器, 不带皮肤播放器的区别
- 【播放器端SDK】视频点播如何获取视频播放URL
- 【播放器端SDK】Authinfo模式、url模式、AK模式的区别
- 【播放器端SDK】重播视频
- 【播放器端SDK】纯音频播放
- 【播放器SDK】播放器支持边缓存边下载
- 【播放器SDK】Android如何实现固定竖屏播放视频
- 支持多实例播放器
- 音量调节, 可静音播放
- web播放器加密视频播放

1.2 直播服务相关

- 阿里视频直播服务解决什么问题?
- 直播录制转为媒体文件(直播录制, 录制回看)
- 直播安全如何保证?

1.2.1 推流端SDK

- 如何开通视频直播, 直播的核心流程
- 如何设置推流端的参数
- 如何在推流端添加水印
- 如何实现后台推流
- 支持的推流格式和播放格式

1.2.2 播放器端SDK

- 首屏秒开如何实现
- 播放视频怎么做到尽量少的延迟
- 后台播放
- 支持弱网条件下的丢帧策略setMaxBufferDuration
- 网络超时的策略, setTimeout

1.2.3 连麦端SDK

- 如何开启连麦
- 连麦SDK跟推流SDK的区别
- 实现一个最基础的连麦流程



云技术服务大学
云产品干货高频分享



云技术服务课堂
和大牛零距离沟通



阿里云开发者“藏经阁”
海量电子书免费下载