

Mobile Live Video Broadcasting

Mini LVB

Product Introduction



Copyright Notice

©2013-2018 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Mini LVB

- Deployment

- Troubleshooting

- Error Codes and View Log

 - Source Code Parsing

 - Frontend/Backend Protocol Parsing

Mini LVB Deployment

Last updated : 2018-07-11 11:38:16

Activating Cloud Services

Activate LVB

1. Apply for LVB service

Log in to the [LVB Console](#). If the service has not been activated yet, the following page will appear:



Click **Apply**, and then go to the application approval step. The service is activated upon the approval of Tencent Cloud.

2. Configure LVB Code

After the LVB service is activated, enter [LVB Console](#) -> [LVB Code Access](#) -> [Access Configuration](#) to complete configurations, and then activate the LVB Code service:

The screenshot shows the '直播码接入' (Live Code Access) configuration page in the Tencent Cloud console. The left sidebar contains navigation links: 视频直播, 直播码接入 (selected), 频道托管, 质量监控, 截图鉴黄, 全局设置, and 小直播源码. The main content area has a header '直播码接入' and a descriptive text box explaining the service. Below this is the '申请直播码接入' (Apply for Live Code Access) section, which includes a '直播录制' (Live Recording) toggle switch, and three input fields: '推流防盗链key' (Push Stream Anti-Leakage Key), 'API鉴权key' (API Authentication Key), and '回调URL地址' (Callback URL Address). Each key field has a '随机生成' (Randomly Generate) button. The '回调URL地址' field has a note: '请输入包含http://的完整url地址, 目前暂不支持https的回调地址'. At the bottom are '确定接入' (Confirm Access) and '取消' (Cancel) buttons.

腾讯云 总览 云产品 直播 点播 对象存储服务 云通信

视频直播

直播码接入

直播码接入是指直播频道的控制和管理完全由客户自主掌控，除了对接难度低之外，同时带来较大的定制空间。在这套接入方案中，腾讯云将主要负责推流、转码、CDN分发、录制以及安全保护等功能，状态管理等部分则完全开放，您可另需注意：直播接入和频道托管方式不能同时使用！

申请直播码接入

直播录制 ☒

推流防盗链key 随机生成
共32个字符，可自定义字符范围为A-Z，a-z，0-9

API鉴权key 随机生成
共32个字符，可自定义字符范围为A-Z，a-z，0-9

回调URL地址
请输入包含http://的完整url地址，目前暂不支持https的回调地址

确定接入 取消

Click **Confirm Access**.

3. Obtain LVB configuration information

On the LVB console, get APP_ID , APP_BIZID and API_KEY used to configure the server later.



Activate Instant Messaging (IM)

1. Apply for IM service

Log in to the [IM Console](#). If you have not activated the service, click the **Activate IM** button. For a new Tencent Cloud account, the IM App list is empty, as shown below:



Click the **Create Application Access** button to create a new application access, that is, the name of the application for which you want to get the access to Tencent Cloud IM service, as shown below:

创建新应用

创建方式

新建应用

应用名称

小直播

应用类型

游戏

应用简介

小直播

不超过300字

确定

取消

Click **OK**, and then you can see in the application list the item you just added, as shown below:

腾讯云 总览 云产品 常用服务 备案 腾讯云小程序 应用

直播 点播 对象存储 关系型数据库 云服务器 云通信 SQL证书管理 域名管理 微信小程序 服务商管理

云通信 应用列表 查看录像 下载云通信SDK

SDKAPPID	应用名称	应用状态	创建时间	操作
	小直播演示	启用	2018-03-04 23:37:51	应用配置 统计分析 更多

2. Configure the standalone mode

Click the **Application Configuration** button in the list in the above figure to proceed with the configuration, as shown below.

腾讯云 总览 云产品 常用服务

直播 点播 对象存储 关系型数据库 云服务器 云通信 互动直播 +

云通信

应用列表

统计分析

帐号体系集成 编辑

帐号名称 RTMPRoom

accountType

集成模式 独立模式

验证方式 下载公私钥 什么是公私钥

帐号管理员 admin 什么是账号管理员

帐号体系集成

通过账号登录集成，我们支持您创建的应用采用自有账号，及QQ、微信等第三方开发账号

帐号名称 RTMPRoom

集成模式 独立模式 托管模式 了解集成模式

账号管理员 admin 什么是账号管理员

+ 添加管理员111

保存 取消

3. Obtain IM configuration information

On the IM console, get `IM_SDKAPPID`, `IM_ACCOUNTTYPE`, `ADMINISTRATOR`, `PRIVATEKEY`, `PUBLICKEY` used to configure the server later.

基础配置 功能配置 Crash 开发辅助工具

应用信息 编辑

SdkAppId [Redacted] ← IM_SDKAPPID

应用名称 小直播演示

应用类型 视频

应用简介

创建时间 2018-03-04 23:37:51

上次修改时间 2018-03-04 23:37:51

应用平台 编辑

所属平台

帐号体系集成 编辑

帐号名称 小直播演示

accountType [Redacted] ← IM_ACCOUNTTYPE

集成模式 独立模式

验证方式 下载公私钥 ← 什么是公私钥 ← PRIVATEKEY、PUBLICKEY

系统生成的公私钥便于开发者快速开发，每次下载不会重新生成密钥，所以请注意私钥的保密性。

账号管理员 admin ← 账号管理员 ← ADMINISTRATOR

Download and decompress the public and private keys from the Verification Method, and open private_key with a text editor, for example:

```
-----BEGIN PRIVATE KEY-----
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
-----END PRIVATE KEY-----
```

Copy the above content directly to the following configuration script to generate an IM signature in the backend of Mini LVB.

Then, convert it to strings as follows, which will be used in the configuration file (config.js) of the server.

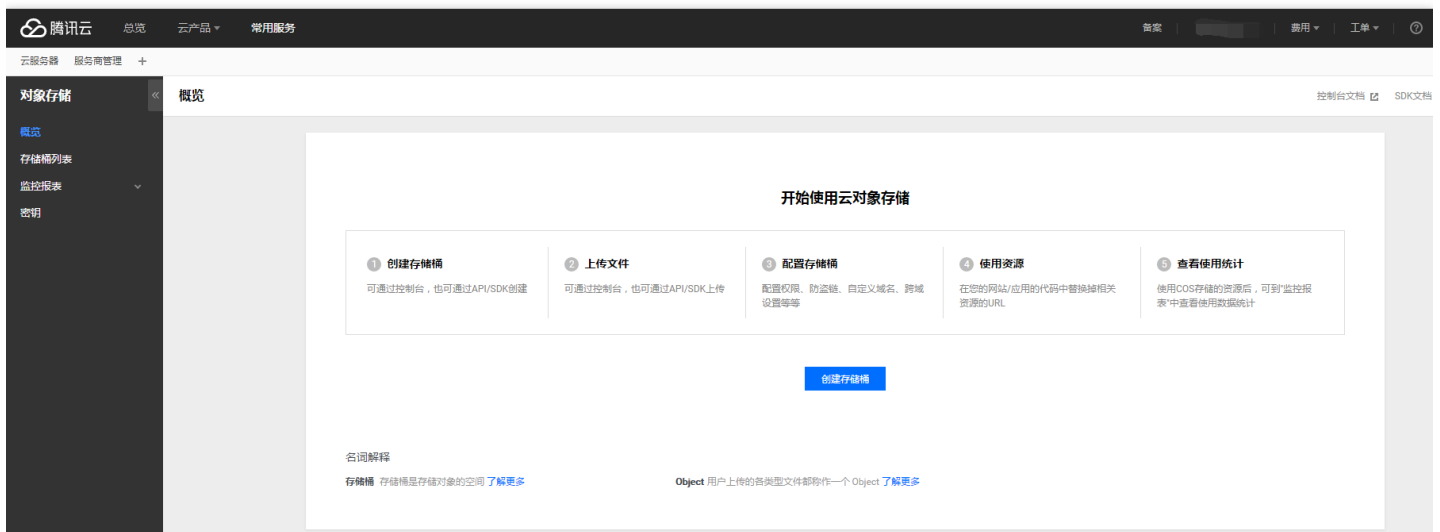
Note: Add \r\n at the end of each line:

```
"-----BEGIN PRIVATE KEY-----\r\n"+
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX\r\n"+
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX\r\n"+
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX\r\n"+
"-----END PRIVATE KEY-----\r\n"
```

Activate Cloud Object Storage (COS)

1. Apply for COS service

Log in to the [COS Console](#). If no bucket has been created, click the **Create Bucket** button as shown below:



2. Create a bucket and obtain its basic information

Enter the bucket name, select the region to which the bucket belongs, and configure the read and write permissions to create a bucket.

创建Bucket

成都（西南）地域促销活动，最高免费10TB/月存储空间，详情参见 [【活动说明】](#)

所属项目

默认项目

* 名称

仅支持小写字母、数字和 - 的组合，不能超过40字符。

地域

广州(华南)

请根据您的业务就近存储，以提高访问速度。请注意，Bucket创建后不能修改所属地域，详见 [地域说明](#)

访问权限

☐ 私有读写

☒ 公有读私有写

公有读私有写: 可对object进行匿名读操作, 写操作需要进行身份验证。

CDN加速

☒ 开启

☐ 关闭

开通腾讯云 CDN 来加速您访问。 [CDN 免费额度](#)

确定

取消

Click **OK** to go to the management page of the bucket you just created. Select **Basic Configuration** to get `COSKEY_APPID` , `COSKEY_BUCKET` , `COSKEY_BUCKET_REGION` and other information used to configure

the server later.



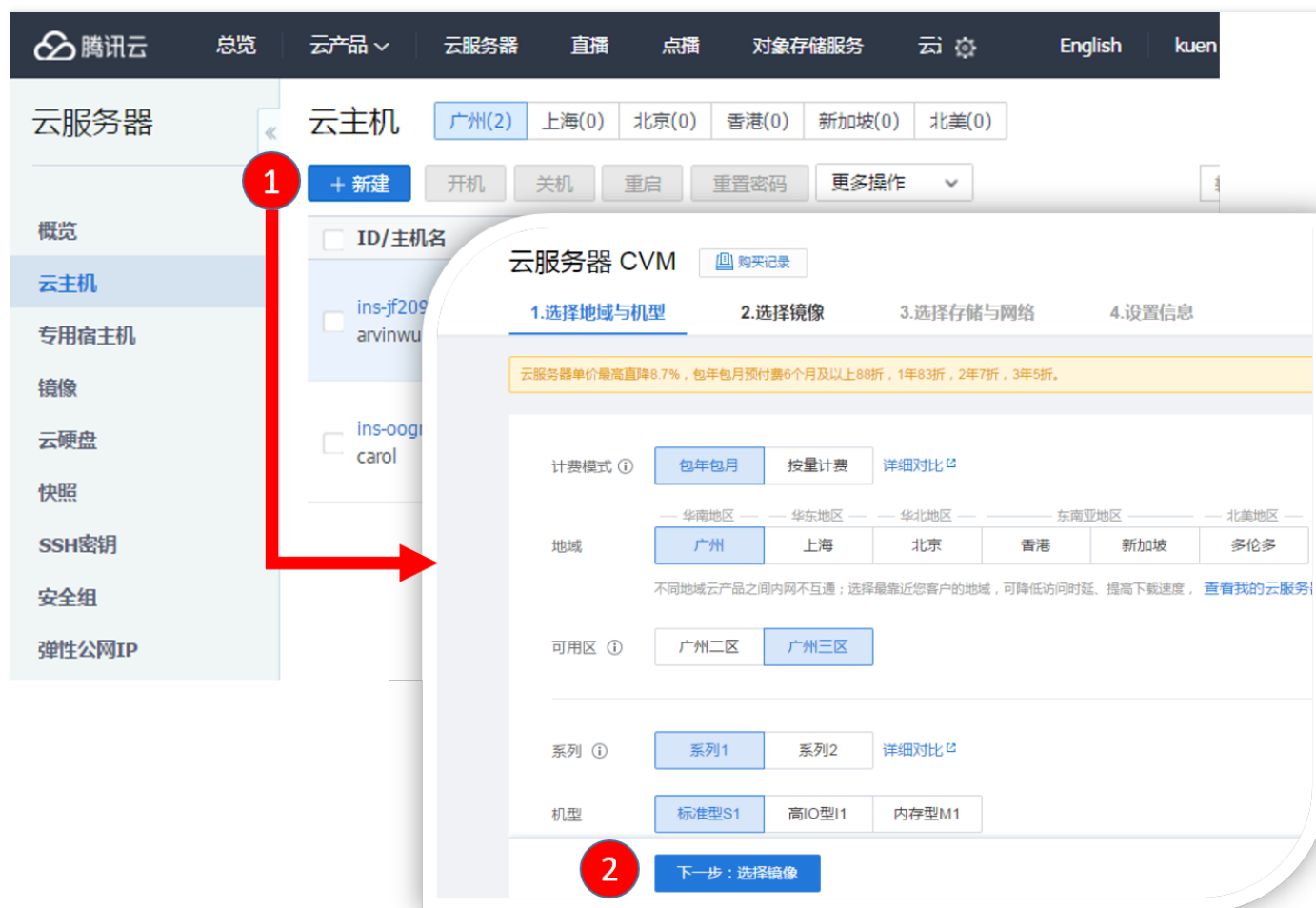
3. Obtain key information

Go to [COS Console](#) -> [Keys](#) -> [Cloud API Key](#) to get `COSKEY_SECRETID` and `COSKEY_SECRETKEY`.



Integration and Deployment at Backend

Deploy a Tencent Cloud CVM image

Step 1: Create a CVM

Step 2: Go to the **Service Marketplace** to select an image. **Mini LVB** image as shown in the figure is recommended.

云服务器 CVM [购买记录](#)

快速配置 自定义配置

新春采购节爆款2折秒杀，新购最高返1000，更有续费/升级大礼包 [立即购买](#)
云服务器新购特惠，超值活动价 0.73元/天起 [立即购买](#)
包年包月预付费6个月及以上88折，1年83折，2年7折，3年5折（注：金融专区不参加折扣）。

1. 选择地域与机型 2. 选择镜像 3. 选择存储与网络

已选配置

计费模式 包年包月

地域 华南地区（广州）

可用区 广州三区

机型 系列2、标准型S2、1核CPU、1G内存

镜像提供方 ☐ 公共镜像 ☐ 自定义镜像 ☐ 共享镜像 ☒ 服务市场

3 从服务市场选择

上一步 下一步：选择存储与网络

选择镜像

服务市场

基础环境 全能环境 管理与监控 建站模板 安全高可用 Docker容器 业务管理

小直播业务后台专用（nginx+php+mysql）
操作系统：CentOS 7.2 64位
集成软件：nginx1.10 + php5.4 + mysql5.5
集成软件：腾讯科技（深圳）有限公司
免费使用
[同意用户协议](#)

小直播业务后台专用V2.0（nginx+php+mysql）
操作系统：CentOS 7.4 64位
集成软件：nginx1.12、php5.4、mysql5.5、redis3.2
集成软件：腾讯科技（深圳）有限公司
4 免费使用
[同意用户协议](#)

小直播

上一页 1 下一页

Step 3: Configure the disk, network, and the access password for CVM (keep the password well to avoid leakage), and then configure the security group.

1.选择地域与机型

2.选择镜像

3.选择存储与网络

4.设置信息

用户名

root

密码

请输入主机密码

linux机器密码需8到16位，至少包括两项（[a-z,A-Z],[0-9]和[!@#\$%^&*~+=_{}|;':<>.,?/]的特殊符号）

确认密码

请再次输入主机密码

安全组 ?

新建安全组

已有安全组

安全加固

请选择安全组

请选择安全组

sg-ogq8i4i3 | Windows安全组放通3389端口

sg-7hrmnp8h | Linux安全组放通22端口

sg-ak1ijhgr | 默认安全组放通全部端口

使用指引

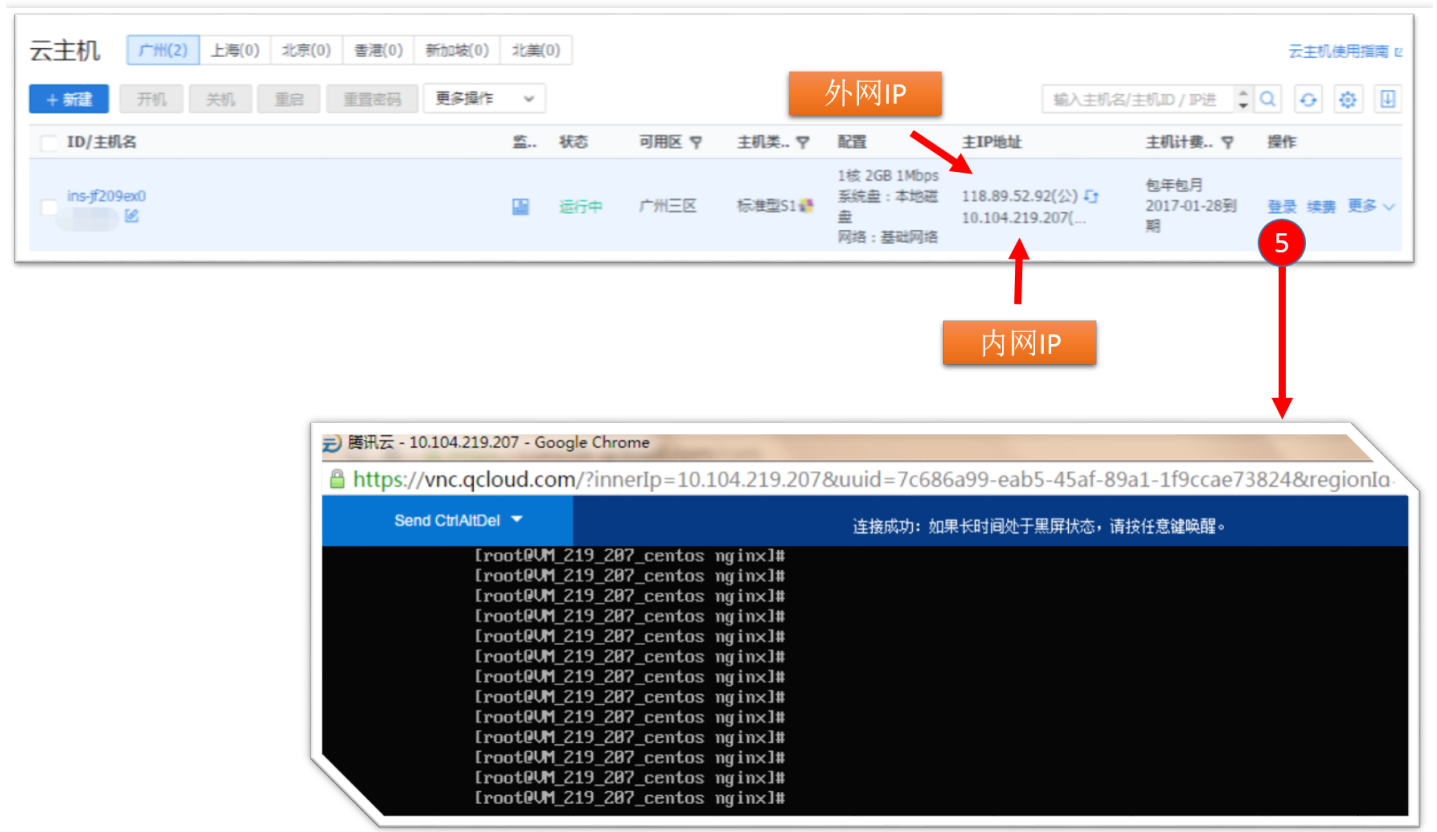
介绍

云监控

免费开通云产品监控、分析和实施告警，安装组件获取主机监控指标

详细介绍

Step 4: Make the payment to generate the CVM. You can click **Log In** to access the CVM via Tencent Cloud's webpage shell, or use **putty** or **SecretCRT** to log in to the CVM via SSH.



Step 5: Modify CVM configurations

Configure the `APP_ID`, `APP_BIZID`, `API_KEY`, `COSKEY_BUCKET`, `COSKEY_BUCKET_REGION`, `COSKEY_SECRECTKEY`, `COSKEY_APPID`, `COSKEY_SECRECTID`, `IM_SDKAPPID`, `IM_ACCOUNTTYPE` in the script below to the values generated in the COS service above and save the script. **Then log in to the CVM and execute the modified script directly on the CVM.**

The content in double quotation marks following the first echo in the code below is the IM private key. Open the IM private_key with a text editor tool and then copy it into the double quotation marks.

Note: Modify the following values locally and copy the modified script. Log in to the CVM, paste the script in the console, and then press Enter to execute the script.

```
#!/bin/bash
echo "-----BEGIN PRIVATE KEY-----
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
-----END PRIVATE KEY-----" > /data/live_demo_service/deps/sig/private_key;
echo "<?php
define('APP_ID','123456'); //Replace with the appid of the LVB service you applied for
define('APP_BIZID','1234'); //Replace with the bizid of the LVB service you applied for
define('API_KEY','xxxxxxx'); //api key
```



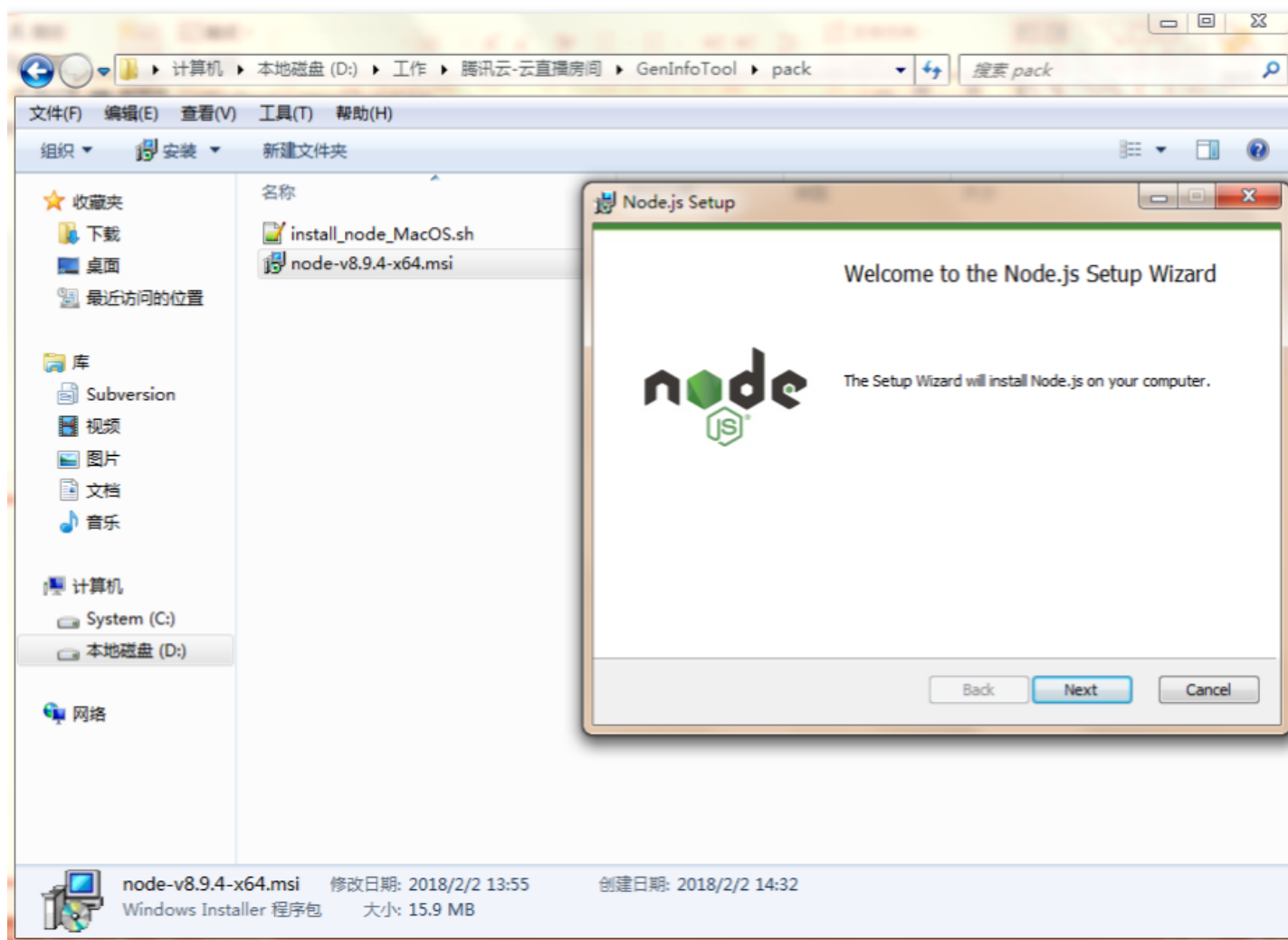
```
define('COSKEY_BUCKET','xxxxxxx');//Replace with the bucket created in COS
define('COSKEY_BUCKET_REGION','xxxxxxx');//Replace with the region of the bucket created in COS
define('COSKEY_SECRETKEY','xxxxxxx');//Replace with the secretkey created in COS
define('COSKEY_APPID',123456);//Replace with the appid generated in COS
define('COSKEY_SECRECTID','xxxxxxx');//Replace with the secrectid (paired with secretkey) generated in COS
define('COSKEY_EXPIRED_TIME',30);

define('IM_SDKAPPID',123456);//IM SDK
define('IM_ACCOUNTTYPE','1234');//IM account integration type
?>" > /data/live_demo_service/conf/OutDefine.php;
```

Now, the deployment at the backend is completed.

Configuring RoomService Service

Download the [RoomTool](#) first and decompress it.

Step 1: Install Nodejs environment

Step 2: Replace the parameters in the config.js file under the root directory of the toolkit with the values generated in the above LVB and IM services.

```

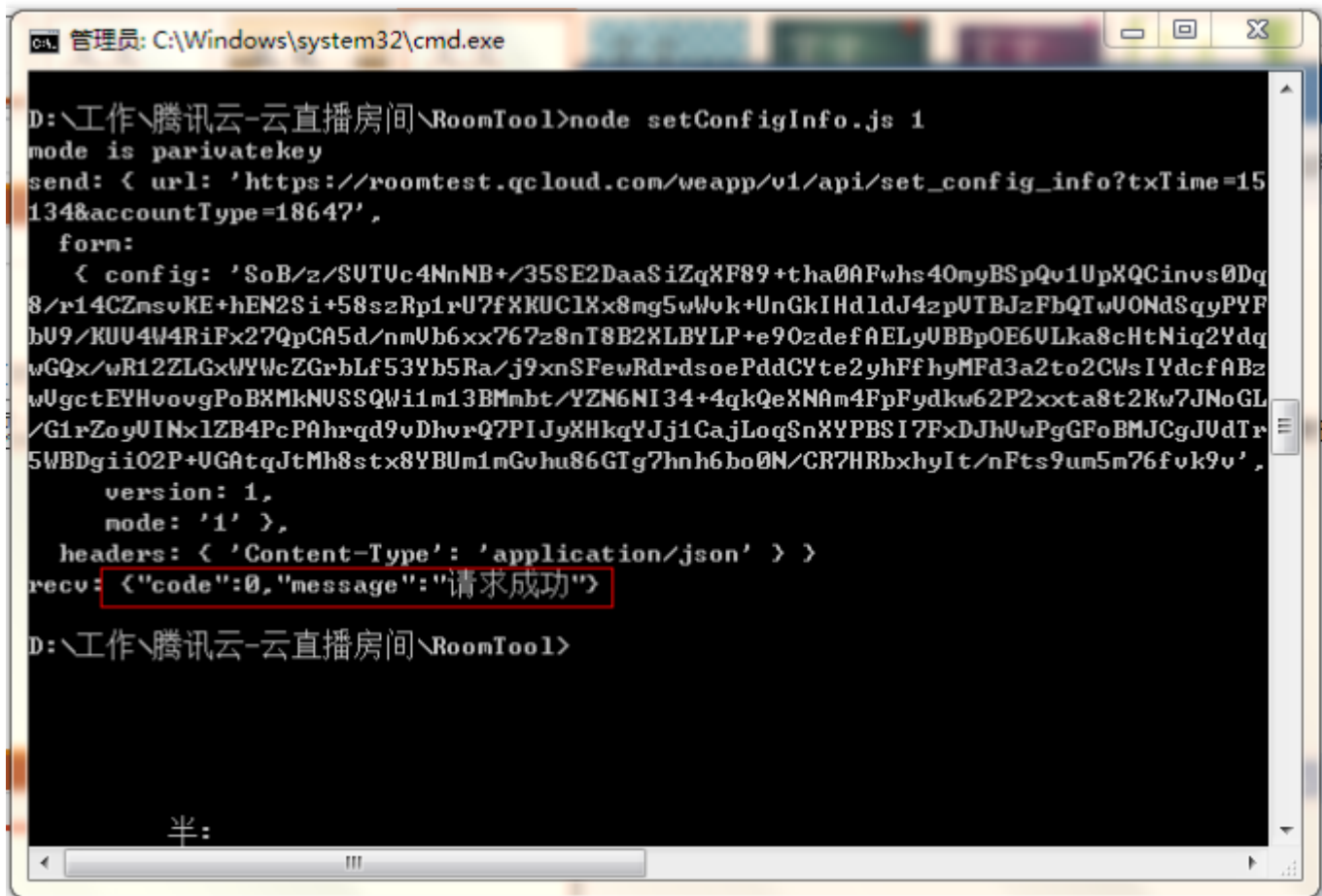
4  * [live 需要开通云直播服务]
5  * 具体填写, 可以参考, "使用说明.pptx"
6  */
7  live: {
8
9      /**
10     * [APP_ID 云直播 appID]
11     * @type {Number}
12     */
13     APP_ID: 0,
14
15     /**
16     * [API_KEY 云直播 API鉴权key]
17     * @type {String}
18     */
19     API_KEY: '',
20 },
21
22 /**
23 * 需要开通云通信服务
24 * 参考指引 @https://cloud.tencent.com/document/product/454/7953#3.-.E4.BA.91.E9.80.9A.E8.AE.AF.E6.9C.8D.E5.8A.A1.EF.BC.88im.EF.BC.89
25 * 有介绍appid 和 accType的获取方法。以及私钥文件的下载方法。
26 * 也可以打开文件 "使用说明.pptx"
27 */
28 im: {
29     /**
30     * 云通信 sdkAppID: accountType 和 privateKey 是云通信独立模式下, 为您的独立账号 identifier,
31     * 派发访问云通信服务的userSig票据的重要信息, 填写错误会导致IM登录失败, IM功能不可用
32     */
33     IM_SDKAPPID: 0,
34
35     /**
36     * 云通信 账号集成类型 accountType: sdkAppID 和 privateKey 是云通信独立模式下, 为您的独立账号identifier,
37     * 派发访问云通信服务的userSig票据的重要信息, 填写错误会导致IM登录失败, IM功能不可用
38     */
39     IM_ACCOUNTTYPE: "",
40
41     /**
42     * [ADMINISTRATOR 云通信管理员账号, 用于后台调用云通信的REST API]
43     * @type {String}
44     */
45     ADMINISTRATOR: "",
46
47     /**
48     * 云通信 派发usersig 采用非对称加密算法RSA, 用私钥生成签名。privateKey就是用于生成签名的私钥, 私钥文件可以在互动直播控制台获取
49     * 配置privateKey
50     * 将private_key文件的内容按下面的方式填写到 privateKey字段。
51     * 特别提醒, 换行需要用 \r\n 代替。
52     */
53     PRIVATEKEY: "-----BEGIN PRIVATE KEY-----\r\n" + "xxxxxxxxxx\r\n" + "xxxxxxxxxx\r\n" + "xxxxxxxxxx\r\n" + "-----END PRIVATE KEY-----\r\n",
54
55     /**
56     * 云通信 和privateKey对应的公钥
57     */
58     PUBLICKEY: "-----BEGIN PUBLIC KEY-----\r\n" + "xxxxxxxxxx\r\n" + "xxxxxxxxxx\r\n" + "-----END PUBLIC KEY-----\r\n",
59 }

```

Step 3: Submit configuration parameters

Go to the RoomTool directory and execute the following command to submit configuration parameters:

```
node setConfigInfo.js 1 //1 means to send the private key to the backend of Tencent Cloud RoomService
```



```
管理员: C:\Windows\system32\cmd.exe
D:\工作\腾讯云-云直播房间\RoomTool>node setConfigInfo.js 1
node is privatekey
send: < url: 'https://roomtest.qcloud.com/weapp/v1/api/set_config_info?txTime=15134&accountType=18647' ,
  form:
    < config: 'SoB/z/SUTUc4NnNB+/35SE2DaaSiZqXF89+tha0AFwhs40myBSpQv1UpXQCinvs0Dq8/r14CZmsvKE+hEN2Si+58szRp1rU7fXKUC1Xx8mg5wWvk+UnGkIHdldJ4zpUTBJzFbQTWUONdSqyPYFbU9/KUU4W4RiF+27QpCA5d/nmUb6xx767z8nT8B2XLBVLP+e90zdefAELyUBBpOE6ULka8cHtNiQ2YdqWgQx/wR12ZLGxWYwCZGrbLf53Yb5Ra/j9xnSFewRdrdsOEddCYte2yhFfhyMFd3a2to2CWsIYdcfABzwUgctEYHvougPoBXMkNUSSQWi1m13BMmbt/YZN6NI34+4qkQeXNAm4FpFydkw62P2xxta8t2Kw7JNoGL/GlrZoyUINx1ZB4PcPAhrqd9vDhvrQ7PIJyXHkqYJj1CaJLoqSnXYPBSI7FxDJhUwPgGFoBMJCgJUdTr5WBDgii02P+UGAtqJtMh8stx8YBUm1mGvhu86GTg7hnh6bo0N/CR7HRbxhyIt/nPts9um5m76fuk9v' ,
    version: 1,
    node: '1' >,
  headers: < 'Content-Type': 'application/json' > >
recv: <{"code":0,"message":"请求成功"}>
D:\工作\腾讯云-云直播房间\RoomTool>
```

After the submission succeeds, execute the `node genLoginInfo.js` command to verify whether the parameters are configured successfully.

Integration at Mobile End and Callback Setting

The integration at mobile end mainly refers to the integration of Mini LVB source code and involves the following steps:

Download Mini LVB source code

[Click here](#) to download **Mini LVB IOS** and **Mini LVB Android** source codes.

Change the address of Mini LVB backend server

- iOS

After the source code package is decompressed, you can find a **TCConstants.h** file in the `TCLVBIMDemo/Classes/LVB/Base` directory. Change the `kHttpServerAddr` in the file to the address

of your CVM.

- Android

After the source code package is decompressed, you can find a **TCConstants.java** file in the `app/src/main/java/com/tencent/qcloud/xiaozhibo/common/utils` directory. Change the `APP_SVR_URL` in the file to the address of your CVM.

Note: If no certificate is configured for the CVM, then HTTP, instead of HTTPS, must be used in the CVM address.

Set a callback address

Set a callback address on the LVB console. When such event as stream status change, video recording completion, screenshot completion occurs, the Tencent Cloud backend gives a callback to the business server via this address for handling the event. For more information, please see [Event Notification Messages](#).

Go to **LVB Console -> LVB Code Access -> Access Configuration** to configure the callback URL. If you do not modify the code of Mini LVB business server, the callback URL format is:

`http://您的云主机服务器地址/callback/tape_callback.php`

The screenshot shows the Tencent Cloud LVB console interface. The top navigation bar includes '腾讯云', '总览', '云产品', '直播', '点播', '对象存储服务', '云通信', '关系型数据库', and '云服务器'. The left sidebar shows '视频直播' with sub-items like '概览', '接入管理', '直播码接入 (推荐)', '频道托管', '质量监控', '截图鉴黄', '全局设置', and '小直播源码'. The main content area is titled '直播管理' and shows 'appid: 1252463788' and 'bizid : 3891'. Below this are tabs for '接入配置', '推流生成器', '房间列表', and '房间监控 (公测中)'. The '应用信息' section contains fields for '直播防盗链Key', 'API鉴权Key', and '回调URL'. The '回调URL' field is highlighted with a red box and contains the value 'http://[redacted]/callback/Live_callback.php'. There is also a '直播录制' toggle switch which is currently turned on.

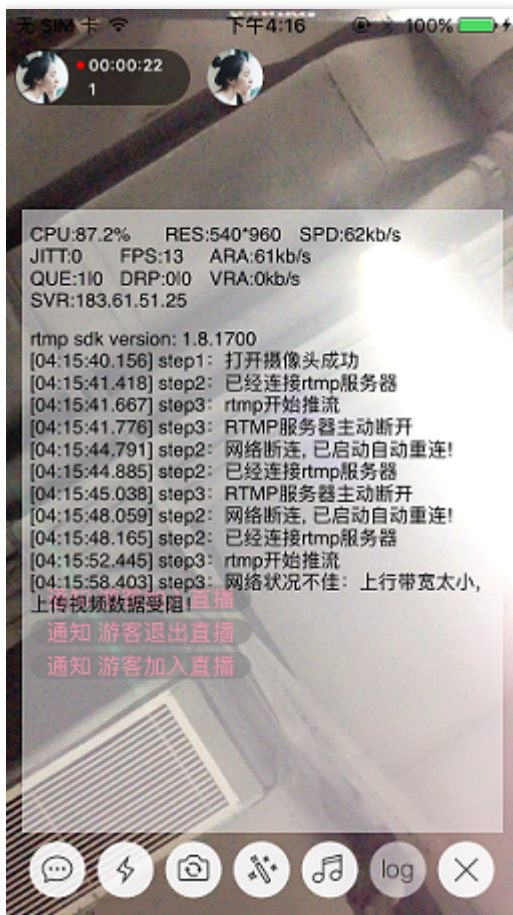
Troubleshooting

Last updated : 2018-08-27 12:03:21

1. What if the backend configuration parameters are incorrect?

1.1 appid or bizid

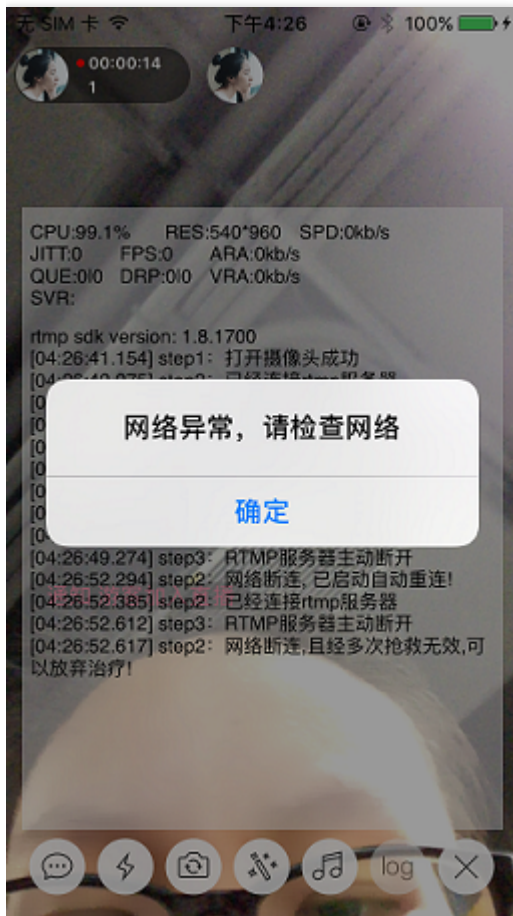
The main indication is that the VJ push always fails. The main reason is that the generated push address is invalid, and Tencent Cloud rejects the push request. In the log, you can see the message **RTMP server actively disconnects**.



1.2 Push hotlink protection key

The main indication is that the LVB push always fails. The main reason is that the push hotlink protection key involves in the txSecret computing. And push hotlink protection key error leads to txSecret error,

eventually resulting in the txSecret verification failure from the server, and Tencent Cloud rejects the request. In the log, you can see the message **RTMP server actively disconnects**.



1.3 API authentication key

The main indication is that the push playback is normal, but it does not appear in the playback list of the terminal.

Generation process of playback record:

- At the end of the LVB, Tencent Cloud completes the recording and notifies the mini LVB backend using the callback URL configured on the console.
- The mini LVB backend verifies the validity of the callback via API authentication key. If the verification fails, the playback record will not be inserted in the database.
- If the callback passes the verification, a playback record will be written to the tape_data table.
- You can only have a playback record after the record is written to the database successfully.

The main reason is that the configuration of the API authentication key is incorrect, and it leads to the callback authentication failure in the service backend, thus no playback record is generated.

1.4 COS APPID

The main indication is the upload failure of profile photo and cover. The main reason is that the signature used by the COS upload request is issued by the business backend. Due to the signature error, the COS upload request fails. This can be confirmed by the terminal's log keyword **"ERROR_PROXY_APPID_USERID_NOTMATCH"**.

```
2016-12-30 16:43:15.438223  
TCLVBIMDemo[1779:551233] applog:upload image  
failed, code:-70,  
msg:ERROR_PROXY_APPID_USERID_NOTMATCH
```

1.5 COS bucket name

The main indication is the upload failure of profile photo and cover. The main reason is that the COS bucket can be considered as a virtual disk, and the COS upload fails if the disk is specified incorrectly, and the error "Bucket cannot be found" will be prompted. This can be confirmed by the terminal's log keyword **"ERROR_PROXY_SIGN_BUCKET_NOTMATCH"**.

```
2016-12-30 16:44:29.748708  
TCLVBIMDemo[1779:551233] applog:upload image  
failed, code:-61,  
msg:ERROR_PROXY_SIGN_BUCKET_NOTMATCH
```

1.6 COS SecretId

The main indication is the upload failure of profile photo and cover. The main reason is that the signature used by the COS upload request is issued by the business backend. COS SecretId is used to specify the key used by the signature. It must be paired with the COS SecretKey. This can be confirmed by the terminal's log keyword **"PROXY_AUTH_SECRETID_NOEXIST"**.

```
TCLVBIMDemo[1779:551233] applog:upload image  
failed, code:-79,  
msg:PROXY_AUTH_SECRETID_NOEXIST
```

1.7 COS SecretKey

The main indication is the upload failure of profile photo and cover. The main reason is that the signature used by the COS upload request is issued by the business backend. The COS SecretKey error causes the signature failure. This can be confirmed by the terminal's log keyword **"ERROR_PROXY_AUTH_FAILED"**.


```
2016-12-30 16:39:50.076097
TCLVBIMDemo[1779:551233] applog:upload image
failed, code:-97, msg:ERROR_PROXY_AUTH_FAILED
```

2. What if the terminal (taking IOS as an example) parameters are incorrect?

2.1 kTCIMSDKAppId or kTCIMSDKAccountType

The main indication is login failure.



2.2 kTCCOSAppId or kTCCOSBucket

The main indication is the upload failure of profile photo or cover. The log is shown as follows:

```
2016-12-30 17:00:03.236158
TCLVBIMDemo[1817:566639] applog:upload image
failed, code:-61,
msg:ERROR_PROXY_SIGN_BUCKET_NOTMATCH
```

2.3 kTCCOSRegion

The main indication is the upload failure of profile photo or cover. The main reason is that kTCCOSRegion is a new parameter of COS 4.0, which is used to specify the location of the COS data center, and if it is configured incorrectly, it prompts that the bucket cannot be found. The log is shown as follows:

```
2016-12-30 17:01:33.233488
TCLVBIMDemo[1823:567345] applog:upload image
failed, code:-133, msg:ERROR_CMD_BUCKET_NOTEXIST
```

2.4 kHttpServerAddr

The main indication is that related features such as pulling list are exceptional, and it prompts that the request times out. The main reason is that the terminal does not access the correct backend service due to the incorrect server address.



3. Why do I fail to pull the playback list?

The generation process of playback list is described in **1.3 API Authentication Key**. The playback list is stored in the data table `tape_data`. Pull failure causes can be checked from the following aspects.



3.1 Whether the database write operation after callback is normal

Generally, if you do not change our backend source code, there will be no problem. If you have changed the createDB script, then it is necessary to check here. Log is a good tool for troubleshooting. To enable log debugging at the backend is simply to create a log directory under the live_demo_service/ directory. Check mysql_XXXX.log. Failure of the database inserting operation may be caused by field attribute modification.

3.2 Whether the API authentication key is correct

Make sure that the value of CALL_BACK_KEY in OutDefine.php is the same as the console API authentication key. Its role has been explained above.

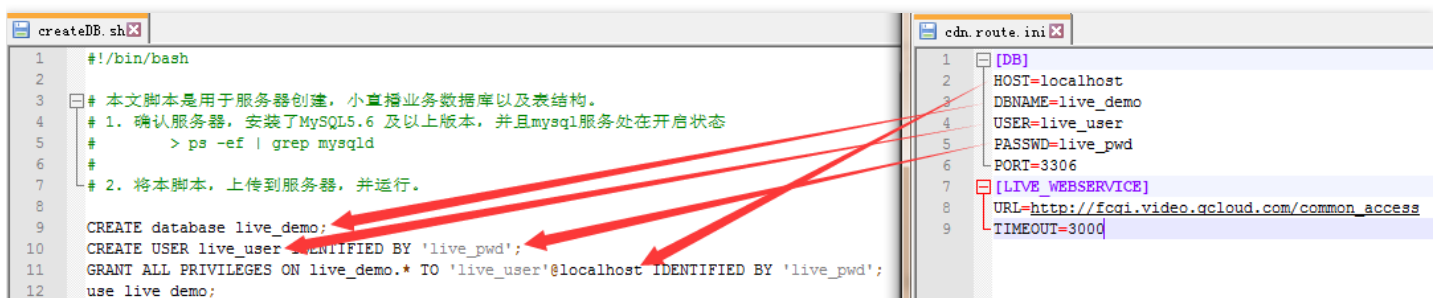
3.3 Whether callback URL is configured correctly

Check whether the callback URL is correctly entered in Tencent Cloud Official Website -> Console -> LVB -> Access Management -> LVB Code Access -> Access Configuration. If it is incorrect, the business backend will not receive a callback notification from Tencent Cloud Server after the LVB ends, and no playback record will be generated.



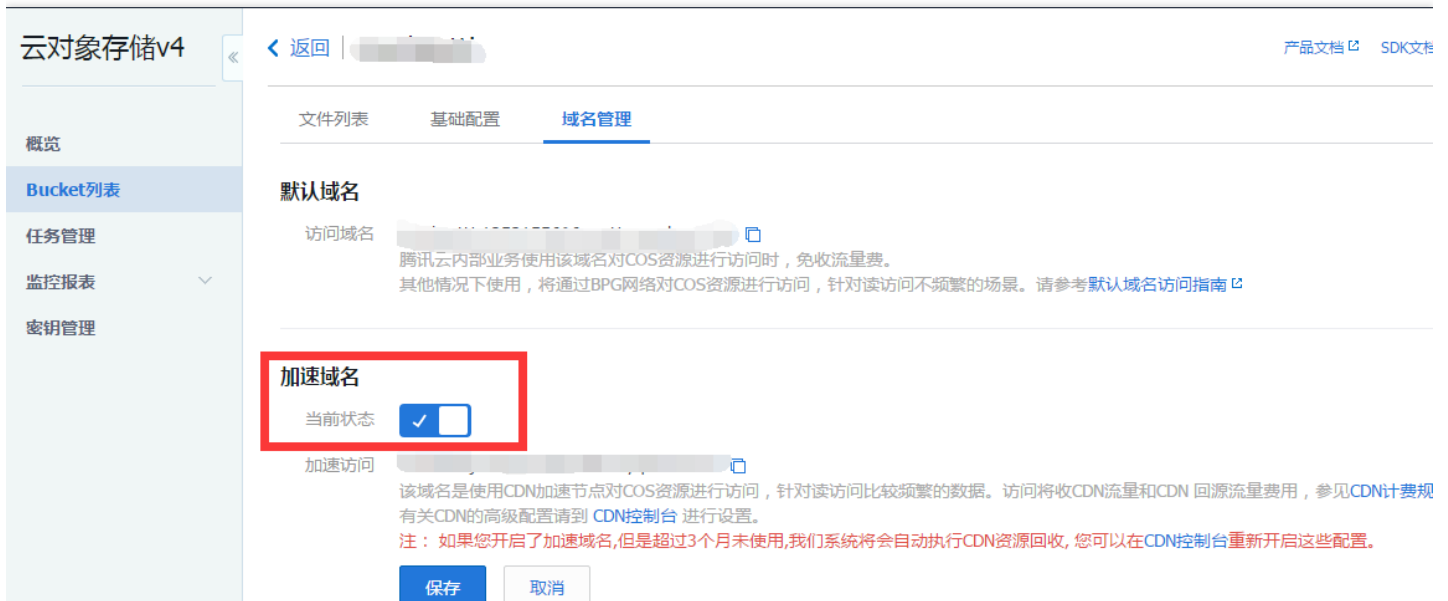
4. Why do I fail to pull the playback list?

It is generated mainly based on the live_data (LVB list) and tape_data (playback list) of the database. Make sure **kHttpServerAddr** is entered correctly if the terminal network is normal. You can check the server. After the App is logged in on Android, if it prompts a failure of list pull, you can see the message **"HTTP Req error, error code:500"** in logcat. Similarly, after the App is logged in on iPhone, it prompts **internal server error**. Open the mysql_errorxxxxxxx.log in the backend log directory, and you can see the message **mysqli_connect failed, error:Access denied for user 'live_user'@'localhost' to database 'live']**. The API failure is caused by database access failure. The confirmation method is to open the cdn.route.ini file in the live_demo_service/conf directory, and ensure that the DB parameters are the same as those you specified when creating the database. PHP accesses the local database via the parameters specified by cdn.route.ini. The mapping relationship is shown as in the figure:



5. Why do I fail to pull the profile photo or cover?

The main indication is that the profile photo or the cover are successfully uploaded, but I failed to download them. The main reason is that the domain name acceleration of COSv4 is disabled by default. The COS upload returns the address of CDN. You can set the domain name acceleration to solve the problem.



6. Why does the upload still fail with the COS parameters configured correctly?

- **Main indication:** COS parameters, terminal and mini LVB backend settings are correct, but the upload of the profile photo and cover still fails.
- **Main reason:** In November 2016, the COS server was upgraded with the region parameter added. The new system and the old one are completely independent and need to be used in combination with the cos sdk version of the corresponding terminal. All newly activated COS services use cos sdk v4. For old versions of COS, you can [submit a ticket](#) to apply to switch it to the new COS version. From December 30, 2016, mini LVB source code package is also equipped with the terminal cos sdk v4 version. A bucket can be considered as a virtual disk in the COS. The upload of a bucket created in the old version of COS server using the new version of cos sdk v4 will fail, with the error **bucket notexist** prompted.
- **The solution** is to create a bucket on the cos v4 platform, and update backend and terminal COS related parameters to those of the new bucket.

7. Why does it prompt "Login failed. Registration operation is rejected for security reason." during registration or login?

Generally, this is because that the registration operations are too frequent on the same network, and the backend rejected the request. Reduce the registration frequency.

What if other problems that are not in the list?

You can [submit a ticket](#) to contact us.

Error Codes and View Log

Last updated : 2018-07-10 14:16:15

How to check the logs of business servers

You can focus your attention on the following two logs:

- Log of Nginx: In case of an error code returned by HTTP (such as 404, 500, etc.), check Nginx's log at the error level, which is located in the subdirectory /logs under Nginx's installation directory. The reasons for the error are generally the configuration problems of Nginx, PHP or MySQL.
- Log in business server code: If a response is returned for the request, but the code in the JSON of the response package is not 200, this means the request failed. In this case, check the file in the /log under the directory where the PHP code is located. If the log directory does not exist, create a log directory and then add read/write permissions (it is recommended to execute `chmod 777` to enable all permissions).

How to check the logs of mobile devices

The path to iOS mobile device's log: `Document/Caches/rtmpsdk_date.log`

The path to Android mobile device's log:

`sdcrad/ tencent/imsdklogs/com/tencent/qcloud/xiaozhibo/rtmpsdk_date.log`

Error Codes

Error Code	Description
498	Verification failed
500	The database operation failed. Verify whether the database table has been created correctly. For more information on the error code, please check the log file <code>mysql_errorxxx</code> (xxx is the date of error) in the /log under the directory where PHP code is located
601	Update failed
602	Invalid parameter
610	Incorrect format of user name

Error Code	Description
611	Incorrect format of password
612	User already exists
621	Wrong password
620	User does not exist

For errors related to IM, please see [IM Error Codes](#)

For error codes related to COS (for uploading images, covers), please see [COS Error Codes](#)

Source Code Parsing

Frontend/Backend Protocol Parsing

Last updated : 2018-08-27 11:58:04

Mini LVB communicates with business server using HTTP protocol. The data in request and response packages are in JSON format. API name is specified using Action, for example, Action=RequestLVBAddr. POST method is used to send a request.

Protocol Descriptions

1. Request LVB push URL

This API is used to submit LVB-related data (for example, user information and LVB data such as title, position, etc.) and return push URL. After receiving the request, the business server will store the LVB-related data in database, and return the data when it receives a request for LVB list from a viewer

Request package format:

Parameter	Type	Description
Action	string	Action of this API is RequestLVBAddr
userid	string	User id
groupid	string	Group id
title	string	LVB title
userinfo	object	User information

userinfo is defined as follows:

Parameter	Type	Description
nickname	string	Nickname
headpic	string	Avatar address
frontcover	string	Front cover address
location	string	Geographical position

Response package format:

Parameter	Type	Description
returnValue	int	Error code. 0: Successful; other values: Failed
returnMsg	string	Description of error codes
returnData	object	Returned data in JSON format

returnData format:

Parameter	Type	Description
pushurl	string	Push URL
timestamp	int	Time stamp

2. Modify online status

VJ starts push, and calls this API when it receives the push event (PUSH_EVT_PUSH_BEGIN), to set the push status to online. After stopping push, VJ calls this API to set the push status to offline

Request package format:

Parameter	Type	Description
Action	string	ChangeStatus
userid	string	User id
status	int	0: Online; 1: Offline

Response package format:

Parameter	Type	Description
returnValue	int	Error code. 0: Successful; other values: Failed
returnMsg	string	Description of error codes
returnData	object	Null

3. Modify counter

This protocol is used to modify the number of likes. When a viewer gives a like, this protocol is sent to the business server to modify the like counter

Request package format:

Parameter	Type	Description
Action	string	ChangeCount
userid	string	User id
type	int	0: Modify the number of viewers; 1: Modify the number of likes (0 is deprecated. The number of viewers is modified using EnterGroup and QuitGroup protocols)
optype	int	0: Increase; 1: Decrease
flag	int	0: LVB; 1: VOD
fileid	string	It is used in VOD scenarios to determine which video is played

Response package format:

Parameter	Type	Description
returnValue	int	Error code. 0: Successful; other values: Failed
returnMsg	string	Description of error codes
returnData	object	Null

4. Pull the list

Pull the list from business server. Paged pull is supported

Request package format:

Parameter	Type	Description
Action	string	FetchList
flag	int	1: Pull online LVB list; 2: Pull VOD list of last 7 days; 3: Pull online LVB list and VOD list of last 7 days with LVB list followed by VOD list
pageno	int	Page number
pagesize	int	Page size

Response package format:

Parameter	Type	Description
returnValue	int	Error code. 0: Successful; other values: Failed
returnMsg	string	Description of error codes
returnData	object	List data

returnData format:

Parameter	Type	Description
totalcount	int	Total number of lists
pusherlist	array	LVB/VOD list data

pusherlist is the array of pusherinfo which is defined as follows:

Parameter	Type	Description
userid	string	User id
groupid	string	Group id
timestamp	int	Time stamp where push starts
type	int	0: LVB; 1: Recording
viewercount	int	Number of viewers
likecount	int	Number of likes
title	string	LVB title
playurl	string	Playback URL
fileid	string	VOD file id
status	string	0: Offline; 1: Online
hls_play_url	string	HLS playback URL
userinfo	object	User information, which is the same as that defined in RequestLVBAddr

5. Obtain signature required to upload COS files

Cloud Object Storage (COS) is a service provided by Tencent Cloud for file storage. You need to provide a signature when uploading files. Since the signature is generated using encryption key, it is not applicable to be generated locally at the client side, but by business server according to specified rules

Request package format:

Parameter	Type	Description
Action	string	GetCOSSign

Response package format:

Parameter	Type	Description
returnValue	int	Error code. 0: Successful; other values: Failed
returnMsg	string	Description of error codes
returnData	object	Returned data in JSON format

returnData format:

Parameter	Type	Description
sign	string	Upload signature

6. Notify business server that a member joins the group

Since the group list of imsdk cannot be customized, for example sorting by levels, it is maintained by the business server, and you can modify the sorting rules based on your demands. When a viewer joins the group, you can call this protocol to notify the business server of a new member:

Request package format:

Parameter	Type	Description
Action	string	EnterGroup
userid	string	User id
flag	int	0: LVB; 1: VOD
liveuserid	string	VJ's user id
groupid	string	Enter group id if flag is 0; enter fileid if flag is 1

Parameter	Type	Description
nickname	string	Nickname
headpic	string	Avatar address

Response package format:

Parameter	Type	Description
returnValue	int	Error code. 0: Successful; other values: Failed
returnMsg	string	Description of error codes
returnData	object	Null

7. Notify business server that a member exits the group

Since the group list of imsdk cannot be customized, for example sorting by levels, it is maintained by the business server, and you can modify the sorting rules based on your demands. When a viewer exits the group, you can call this protocol to notify the business server of the exit of a member:

Request package format:

Parameter	Type	Description
Action	string	QuitGroup
userid	string	User id
flag	int	0: LVB; 1: VOD
liveuserid	string	VJ's user id
groupid	string	Enter group id if flag is 0; enter fileid if flag is 1

Response package format:

Parameter	Type	Description
returnValue	int	Error code. 0: Successful; other values: Failed
returnMsg	string	Description of error codes
returnData	object	Null

8. Pull the list of group members

Since the group list of imsdk cannot be customized, for example sorting by levels, it is maintained by the business server, and you can modify the sorting rules based on your demands. When a viewer joins an Live room, you can call this protocol to pull the list of group members, display the list of viewer profile photos, and update the total number of members on the interface.

Request package format:

Parameter	Type	Description
Action	string	FetchGroupMemberList
liveuserid	string	VJ's user id
groupid	string	Enter group id if flag is 0; enter fileid if flag is 1
pageno	int	Page number
pagesize	int	Page size

Response package format:

Parameter	Type	Description
returnValue	int	Error code. 0: Successful; other values: Failed
returnMsg	string	Description of error codes
returnData	object	Returned data in JSON format

returnData format:

Parameter	Type	Description
totalcount	int	Total number of members
memberlist	array	List of members

memberlist is a list of meminfo which is defined as follows:

Parameter	Type	Description
userid	string	User id
nickname	string	Nickname

Parameter	Type	Description
headpic	string	Avatar address

9. Obtain details of specified VJ

Request package format:

Parameter	Type	Description
Action	string	GetUserInfo
userid	string	User id
type	int	0: LVB; 1: Recording
fileid	string	VOD file id, which can be ignored if type is 0

Response package format:

Parameter	Type	Description
returnValue	int	Error code. 0: Successful; other values: Failed
returnMsg	string	Description of error codes
returnData	object	Details of VJ, which is the same as pusherinfo defined in protocol 4 (pull the list)