Tencent Cloud

Video on Demand

Practical Tutorial

Product Documentation

Copyright Notice

Trademark Notice

Tencent Cloud

Service Statement

Contents

Tencent Cloud

Practical Tutorial
Media Upload
How to Pull Network Media Files to VOD

Last updated：2022-12-16 15:02:33

Overview

Introduction

This document describes how to pull an online video (provided in the form of URL) to VOD.

Fees

The code provided in this document is open-source and free of charge, but it may incur the following fees during use:
Fees for purchasing a Tencent Cloud CVM instance to run the API request script. For more information, please see Instance Billing Modes.
VOD storage space will be taken up by uploaded videos via pulling. For more information, please see Video Storage Pricing.

Limits

The pull from URL feature provided by VOD has the following limits:
The URL should directly point to a video file but cannot be a link to a video website page.
If the URL has a timestamp for hotlink protection, please make sure that the hotlink protection limits (such as the validity period and number of allowed access requests) are favorable; otherwise, the access may fail.
URLs with referer hotlink protection enabled are not supported.
DASH (MPD file type) is unsupported.
If the pull object is an HLS (.m3u8 file type) file, the URIs of media segments (generally in .ts file type) should be relative paths without parameters.
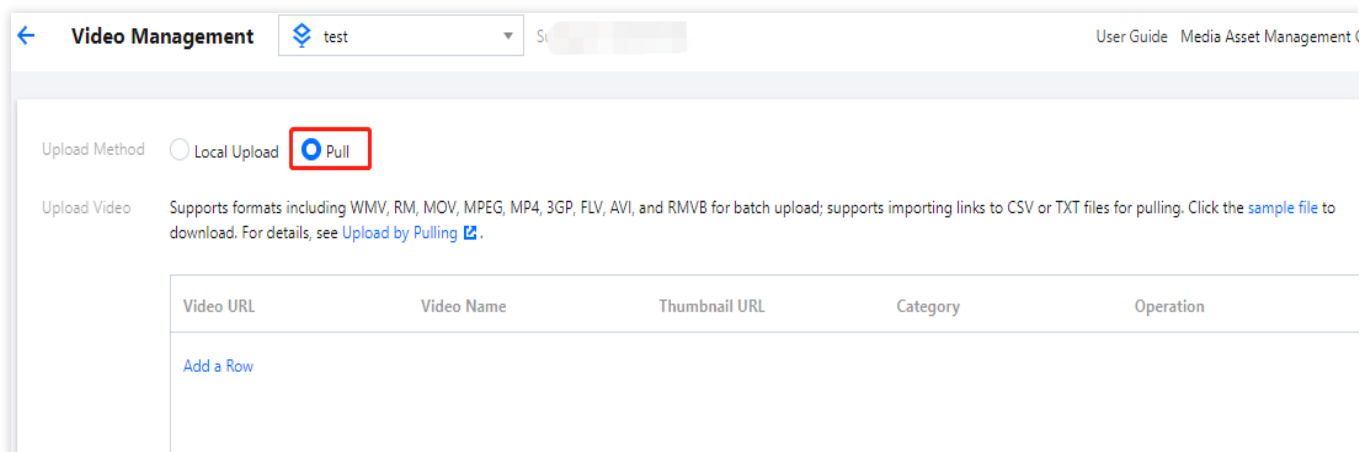
Upload by Pull in Console

Step 1. Activate VOD

Activate the VOD service as instructed in Getting Started - Step 1.

Step 2. Create a pull task

Go to the upload page in the VOD console, select **Pull** as the upload method, click **Add a Row**, enter the URL of the video to be pulled (the test URL is used here as an example. Other configuration items are optional, which can be entered as needed), and click **Pull Video** in the bottom-left corner:

Note:

The time it takes to pull a video is directly proportional to the video file size. We recommend you use a small video (of dozens of megabytes in size) for the test to avoid long wait.

Step 3. View the pull result

After waiting for one or two minutes (subject to the video file size), you can see the pulled video on the Media Assets page.

Note:

If the browser is stuck at the media assets page during the pull process, you need to refresh the page to view the pulled video.

Calling TencentCloud API for Pull

Step 1. Prepare a CVM instance

The TencentCloud API request script needs to be executed on a CVM instance meeting the following requirements:

Region: not limited.

Model: the minimum official configuration (1 CPU core and 1 GB memory) is sufficient.

Public network: a public IP is required, and the bandwidth should be at least 1 Mbps.

Operating system: official public image `Ubuntu Server 16.04.1 LTS 64-bit` or `Ubuntu Server 18.04.1 LTS 64-bit` .

For detailed directions on how to purchase a CVM instance and reinstall the system, please see Operation Guide - Creating Instances via CVM Purchase Page and Operation Guide - Reinstalling System, respectively.

Note:

If you do not have a CVM instance satisfying the above conditions, you can also run the script on another Linux (such as CentOS or Debian) or macOS server with public network access, but you need to modify certain commands in the script based on the operating system. Please search for the specific modification method by yourself.

Step 2. Get the API key

Your API key (i.e., `SecretId` and `SecretKey`) is required for TencentCloud API request. If you have not created an API key yet, please generate one as instructed in Root Account Access Key. If you have already created a key, please get it as instructed in the same document.

Step 3. Activate VOD

Activate the VOD service as instructed in Getting Started - Step 1.

Step 4. Initiate a pull task

Log in to the CVM instance prepared in step 1 as instructed in Logging into Linux Instance in Standard Login Method and enter and run the following command on the remote terminal:

```
ubuntu@VM-69-2-ubuntu:~$ export SECRET_ID=AKxxxxxxxxxxxxxxxxxxxxxxxx; export
SECRET_KEY=xxxxxxxxxxxxxxxxxxxxx;git clone https://github.com/tencentyun/vod-
server-demo.git ~/vod-server-demo; bash ~/vod-server-
demo/installer/pull_upload_api_en.sh
```

Note:

Please assign the corresponding values obtained in step 2 to `SECRET_ID` and `SECRET_KEY` in the command. This command will download the demo source code from GitHub and automatically run the installation script. The installation process will take several minutes (subject to the CVM network conditions), during which the remote terminal will print the following information:

```
[2020-07-15 17:40:13] Start installing pip3.
[2020-07-15 17:40:39] pip3 is successfully installed.
[2020-07-15 17:40:39] Start installing the TencentCloud API SDK for Python.
[2020-07-15 17:40:42] The TencentCloud API SDK for Python is successfully
installed.
[2020-07-15 17:40:42] Start configuring API parameters.
[2020-07-15 17:40:42] API parameter configuration is completed.
```

Run the `pull_upload.py` script to initiate upload:

```
ubuntu@VM-69-2-ubuntu:~$ cd ~/vod-server-demo/pull_upload_api/; python3
pull_upload.py
http://1400329073.vod2.myqcloud.com/ff439affvodcq1400329073/e968a7e552858908041
62014755/LKk92603oW0A.mp4 API-PullUpload
```

Note:

Please replace the URL in the command with the actual address of the video to be pulled.

This command will initiate a PullUpload request to the specified URL and print the response similar to the following:

```
{"TaskId": "1400329073-PullUpload-4ea60158fc6f8e611bbfa750eb1fd0a9t0",
"RequestId": "4e821b4a-9a29-409f-99cb-b703fa184e50"}
```

Step 5. View the pull result

After waiting for one or two minutes (subject to the video file size), you can see the pulled video on the Media Assets page.

Note:

If the browser is stuck at the media assets page during the pull process, you need to refresh the page to view the pulled video.

How to Upload Video from Server

Last updated：2021-05-13 10:10:39

Overview

Introduction

This document describes how to upload a video file on your local server to VOD.

Fees

The code provided in this document is open-source and free of charge, but it may incur the following fees during use:

Fees for purchasing a Tencent Cloud CVM instance to run the upload script. For more information, please see Instance Billing Modes.

VOD storage space will be taken up by uploaded videos via pulling. For more information, please see Video Storage Pricing.

Uploading Video in CVM to VOD

Step 1. Prepare a CVM instance

The upload script needs to be executed on a CVM instance meeting the following requirements:

Region: not limited.

Model: the minimum official configuration (1 CPU core and 1 GB memory) is sufficient.

Public network: a public IP is required, and the bandwidth should be at least 1 Mbps.

Operating system: official public image `Ubuntu Server 16.04.1 LTS 64-bit` or `Ubuntu Server 18.04.1 LTS 64-bit` .

For detailed directions on how to purchase a CVM instance and reinstall the system, please see Operation Guide - Creating Instances via CVM Purchase Page and Operation Guide - Reinstalling System, respectively.

Note:

If you do not have a CVM instance satisfying the above conditions, you can also run the script on another Linux (such as CentOS or Debian) or macOS server with public network access, but you need to modify certain commands in the script based on the operating system. Please search for the specific modification method by yourself.

Step 2. Activate VOD

Please activate the VOD service as instructed in Getting Started - Step 1.

Step 3. Get the API key

Your API key (i.e., `SecretId` and `SecretKey`) is required for video upload. If you have not created an API key yet, please generate one as instructed in Root Account Access Key. If you have already created a key, please get it as instructed in the same document.

Step 4. Download the code and install the SDK

Log in to the CVM instance prepared in step 1 as instructed in Logging into Linux Instance in Standard Login Method and enter and run the following command on the remote terminal:

```
ubuntu@VM-69-2-ubuntu:~$ export SECRET_ID=AKxxxxxxxxxxxxxxxxxxxxxx; export
SECRET_KEY=xxxxxxxxxxxxxxxxxxxxxxx;git clone https://github.com/tencentyun/vod-
server-demo.git ~/vod-server-demo; bash ~/vod-server-
demo/installer/server_upload.sh
```

Note:

Please assign the corresponding values obtained in step 3 to `SECRET_ID` and `SECRET_KEY` in the command. This command will download the demo source code from GitHub and automatically run the installation script. The installation process will take several minutes (subject to the CVM network conditions), during which the remote terminal will print information similar to the following:

```
[2020-06-23 19:56:31] Start installing pip3.
[2020-06-23 19:56:34] pip3 is successfully installed.
[2020-06-23 19:56:34] Start installing the VOD upload SDK for Python.
[2020-06-23 19:56:36] The VOD upload SDK for Python is successfully installed.
[2020-06-23 19:56:36] Start configuring SDK parameters.
[2020-06-23 19:56:36] SDK parameter configuration is completed.
```

Step 5. Upload a video

Before initiating upload, you need to prepare a video file and a cover image (optional) on your CVM instance. If it is inconvenient for you to upload a video to the CVM instance, you can run the following command on the remote terminal to download the test video and cover onto the CVM instance:

```
ubuntu@VM-69-2-ubuntu:~$ wget
http://1400329073.vod2.myqcloud.com/d62d88a7vodtranscq1400329073/7a9b2b56528589
0804459281865/v.f100010.mp4 -O ~/vod-server-
demo/server_upload/tencent_cloud.mp4; wget
http://1400329073.vod2.myqcloud.com/ff439affvodcq1400329073/8aa658d152858908044
59940822/5285890804459940825.jpg -O ~/vod-server-
demo/server_upload/tencent_cloud.jpg
```

Run the `server.upload.py` script to start uploading:

```
ubuntu@VM-69-2-ubuntu:~$ cd ~/vod-server-demo/server_upload/; python3
server_upload.py ./tencent_cloud.mp4 ./tencent_cloud.jpg
```

Note:

Please replace the paths of the video and cover image in the command with the actual file paths. Here, the cover image path parameter is optional, and if it is left empty, the uploaded video will have no cover.

This command will upload the `tencent_cloud.mp4` video to VOD and upload the `tencent_cloud.jpg` image as its cover. After the upload is completed, the remote terminal will print information similar to the following:

```
{"CoverUrl":
"http://1400329073.vod2.myqcloud.com/ff439affvodcq1400329073/8aa658d15285890804
459940822/5285890804459940825.jpg", "FileId": "5285890804459940822",
"MediaUrl":
"http://1400329073.vod2.myqcloud.com/ff439affvodcq1400329073/8aa658d15285890804
459940822/f0.mp4", "RequestId": "84a7fb42-9f05-4acd-9cc8-843690b188ce"}
```

Note:

If you want to use your own video for test, we recommend you use a small video file (of several megabytes in size) to avoid taking too much time for upload due to insufficient CVM bandwidth.

Step 6. View the result

On the Video Management page in the console, you can see the uploaded video file and cover.

Code Interpretation

1. `main()` is the script entry.
2. Call `parse_conf_file()` and read the configuration information from the `config.json` file. The configuration items are as described below:

Field	Data Type	Description
secret_id	String	API key
secret_key	String	API key
procedure	String	Task flow name. The specified task flow will be automatically triggered after video upload is completed. It is empty by default
subappid	String	Whether to upload the video to a VOD subapplication

Note:

This demo supports only the `procedure` and `subappid` upload parameters. For the complete features, please see SDK for Python.

3. Get the local path of the video file to be uploaded and the path of the cover image (if any) from the command line parameters and call `upload_media()` to initiate upload:

```
if len(sys.argv) < 2:
    usage()
    return
video_path = sys.argv[1]
cover_path = sys.argv[2] if len(sys.argv) > 2 else ""

# Initiate upload
rsp = upload_media(configuration, video_path, cover_path)
```

4. In `upload_media()` , use the method provided by the SDK for Python to construct an upload instance `client` , set upload parameters in `req` , and initiate upload:

```
client = VodUploadClient(conf["secret_id"], conf["secret_key"])
req = VodUploadRequest()

req.MediaFilePath = video
if cover != "":
    req.CoverFilePath = cover
if conf["procedure"] != "":
    req.Procedure = conf["procedure"]
req.SubAppId = int(conf["subappid"])

rsp = client.upload("ap-guangzhou", req)
return rsp
```

Note:

The first parameter (`"ap-guangzhou"`) in `client.upload()` is the access region of the upload instance rather than the storage region of the uploaded video. You can simply fix the parameter value as `"ap-guangzhou"` . If you want to specify the storage region for uploaded videos, please set the `req.StorageRegion` parameter.

Other Features

The VOD SDK for upload from server supports other features such as setting the video name, category, and expiration time. For more information, please see the SDK development guide for the corresponding programming language:

Java

C#

PHP

Python

Go

How to Upload Videos over the Web

Last updated：2022-12-16 15:12:27

Overview

Demo features

This demo shows you how to upload videos to VOD through the webpage. It builds two HTTP services based on SCF: The first service is used to receive a request from the browser to get the signature for upload from client, calculate the signature, and return it.
The second service uses VOD's web upload SDK to implement a page. You can access the page in a browser to upload local videos to VOD.

Architecture and process

The system mainly involves four components: browser, API Gateway, SCF, and VOD. Here, API Gateway and SCF are the deployment objects of this demo as shown below:
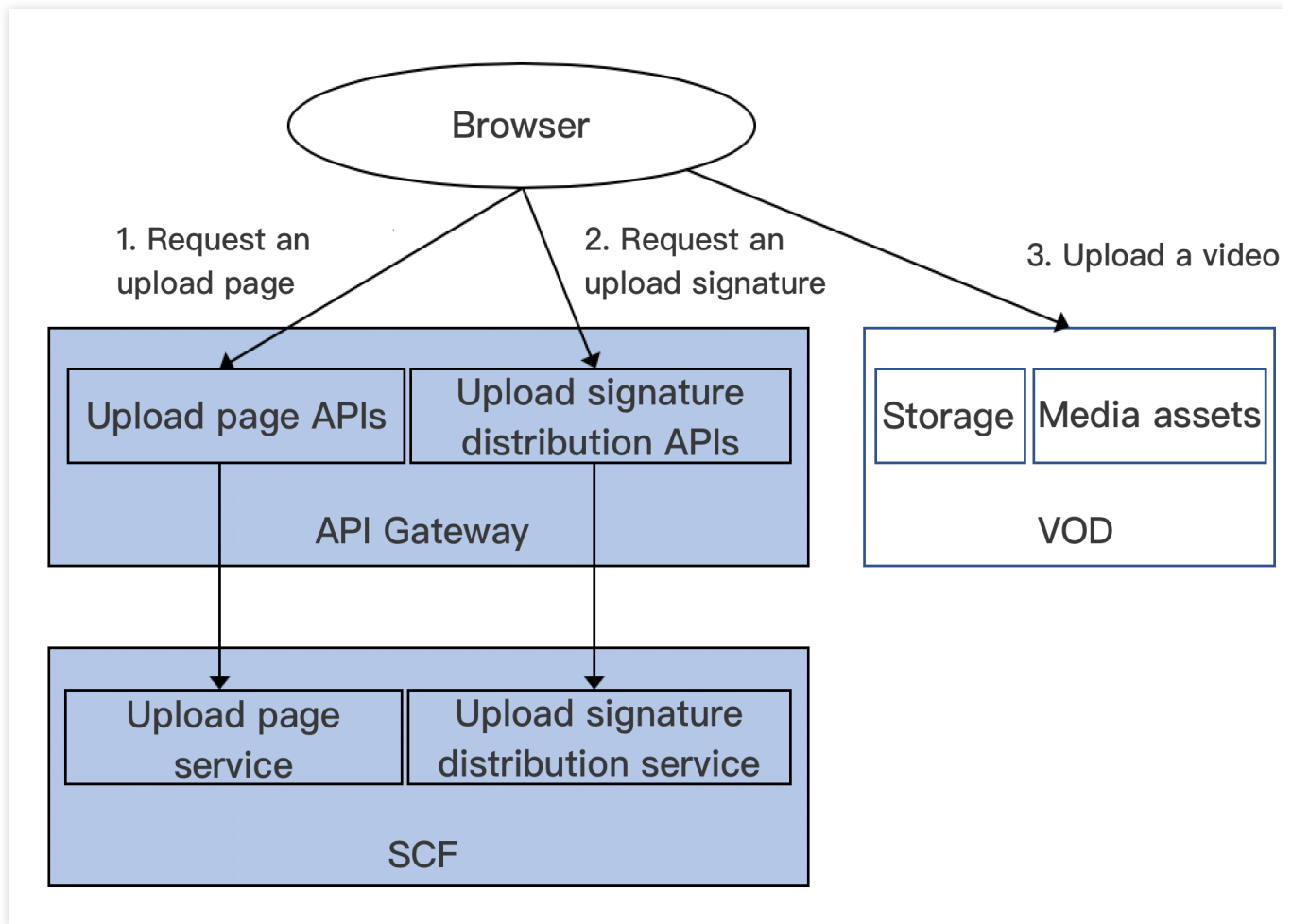
The main business process is as follows:

1. The browser requests SCF for an upload page.

2. You select a local video and click upload on the upload page, and the browser requests SCF for an upload signature.

3. The browser uses the upload signature to initiate an upload request to VOD and displays the upload result on the upload page after completion.

Note:

The SCF code in the demo is developed based on Python 3.6. SCF also supports other programming languages such as Python 2.7, Node.js, Go, PHP, and Java for your choice as needed. For more information, please see Development Guide.

Fees

The VOD web upload demo (including the webpage code and service backend code) provided in this document is open-source and free of charge, but it may incur the following fees during service building and use:
Fees for purchasing a Tencent Cloud CVM instance to run the service deployment script. For more information, please see Instance Billing Modes.

Fees for using the upload page and signature distribution service provided by SCF. For more information, please see Billing Mode and Free Tier.

Fees for using Tencent Cloud API Gateway to provide public network APIs for SCF. For more information, please see Billing Overview.

Fees for VOD storage of uploaded videos. For more information, please see Pay-as-You-Go (Postpaid Daily Billing Cycle).

Fees for VOD traffic consumed by video playback. For more information, please see Pay-as-You-Go (Postpaid Daily Billing Cycle).

Quick Deployment of Web Upload Demo

The web upload demo is deployed on SCF with a service entry provided by API Gateway. To make it easier for you to build services, we provide a quick deployment script as detailed below.

Step 1. Prepare a CVM instance

The deployment script needs to be executed on a CVM instance meeting the following requirements:

Region: not limited.

Model: the minimum official configuration (1 CPU core and 1 GB memory) is sufficient.

Public network: a public IP is required, and the bandwidth should be at least 1 Mbps.

Operating system: official public image `Ubuntu Server 16.04.1 LTS 64-bit` or `Ubuntu Server 18.04.1 LTS 64-bit`.

For detailed directions on how to purchase a CVM instance and reinstall the system, please see Operation Guide - Creating Instances via CVM Purchase Page and Operation Guide - Reinstalling System, respectively.

Note:

The web upload demo itself does not depend on CVM but only uses CVM to run the deployment script.

If you do not have a CVM instance satisfying the above conditions, you can also run the script on another Linux (such as CentOS or Debian) or macOS server with public network access, but you need to modify certain commands in the deployment script based on the operating system. Please search for the specific modification method by yourself.

Step 2. Activate VOD

Please activate the VOD service as instructed in Getting Started - Step 1.

Step 3. Get the API key and APPID

Your API key (i.e., `SecretId` and `SecretKey`) and `APPID` are required for deploying and running the web upload demo service.

If you have not created an API key yet, please generate one as instructed in Root Account Access Key. If you have already created a key, please get it as instructed in the same document.

You can view the `APPID` on the Account Information page in the console.

Step 4. Deploy the service backend and webpage

Log in to the CVM instance prepared in step 1 as instructed in Logging In to Linux Instance in Standard Login Method and enter and run the following command on the remote terminal:

```
ubuntu@VM-69-2-ubuntu:~$ export SECRET_ID=AKxxxxxxxxxxxxxxxxxxxxxxxx; export
SECRET_KEY=xxxxxxxxxxxxxxxxxxxxxx;export APPID=125xxxxxxx;git clone
https://github.com/tencentyun/vod-server-demo.git ~/vod-server-demo; bash
~/vod-server-demo/installer/web_upload_scf_en.sh
```

Note:

Please assign the corresponding values obtained in step 3 to `SECRET_ID` , `SECRET_KEY` , and `APPID` in the command.

This command will download the demo source code from GitHub and automatically run the installation script. The installation process will take several minutes (subject to the CVM network conditions), during which the remote terminal will print the following information:

```
[2020-04-25 23:03:20] Start installing pip3.
[2020-04-25 23:03:23] pip3 is successfully installed.
[2020-04-25 23:03:23] Start installing Tencent Cloud SCF.
[2020-04-25 23:03:26] SCF is successfully installed.
[2020-04-25 23:03:26] Start configuring SCF.
[2020-04-25 23:03:28] SCF configuration is completed.
[2020-04-25 23:03:28] Start deploying the VOD client upload client signature
distribution service.
[2020-04-25 23:03:40] The deployment of the VOD client upload signature
distribution service is completed.
[2020-04-25 23:03:44] Start deploying the VOD web upload page.
[2020-04-25 23:03:53] The deployment of the VOD web upload page is completed.
[2020-04-25 23:03:53] Please access the following address in your browser to
use the demo: https://service-xxxxxxxx-
125xxxxxxx.gz.apigw.tencentcs.com/release/web_upload_html
```

Copy the address of the webpage in the output log (which is `https://service-xxxxxxxx-` `125xxxxxxx.gz.apigw.tencentcs.com/release/web_upload_html` in this example).

Note:

If the following warning is displayed in the output log, it is generally because the CVM instance cannot immediately parse the service domain name deployed just now. You can ignore this warning.

```
[2020-04-25 17:18:44] Warning: the client upload signature distribution service
failed the test.
```

Step 5. Use the web upload demo

1. Access the address copied in step 4 in a browser to start using the web upload demo.

2. Perform video upload operations on this page:

2.1 Select a local video file (MP4 format is recommended).

2.2 (Optional) Select a local cover image (in JPG or PNG format).

2.3 (Optional) Enter the video name.

2.4 Click **Start Upload** to upload the video.

3. After the upload is completed, the VOD media IDs (i.e., `fileId`) and URLs of the uploaded video and cover will be displayed at the bottom of the page.

You can view the uploaded video in the VOD console.

Note:

You can try out other features on the upload page as prompted.

System Design Description

API protocol and test

Both the **upload page** and **upload signature distribution** functions use API Gateway to provide APIs. The specific API protocol is as detailed below:

Service	Function Name	API Form	Response Content
Upload page	web_upload_html	HTTP GET	HTML page
Upload signature distribution	ugc_upload_sign	HTTP POST	Upload signature

Upload page

You can access the SCF service list to view the details of the upload page service:

Note:

The two SCF functions used by the demo are deployed under the namespace `vod_demo` in the Guangzhou region. You need to select the corresponding region and namespace in the console to view the deployed SCF functions.

Click the function name, select **Trigger Management** on the left, and **Access Path** on the right is the URL of the upload page. Click **API Service Name** to redirect to the corresponding API Gateway page.

To test the service, directly access the page URL in a browser to check whether the upload page is displayed normally.

Upload signature distribution

You can access the SCF service list to view the details of the upload signature distribution service in the same way as detailed in Upload page.

Click the function name, select **Trigger Management** on the left, and **Access Path** on the right is the URL of the service. Click **API Service Name** to redirect to the corresponding API Gateway page.

To test the service, manually send an HTTP request and run the following command on a Linux or macOS device with public network access (please modify the service URL according to the actual situation):

```
curl -d '' https://service-xxxxxxxx-
125xxxxxxx.gz.apigw.tencentcs.com/release/ugc_upload_sign
```

If the service is normal, an upload signature will be returned. Below is a sample signature:

```
VYapc9EYdoZLzGx0CglRW4N6kuhzZWNyZXRJZD1BS0lEZk5xMzl6dG5tYW1tVzBMOXFvZERia25hUjd
Za0xPM1UmY3VycmVudFRpbWVTdGFtcD0xNTg4NTg4MDIzJmV4cGlyZVRpbWU9MTU4ODU4ODYyMyZyYW
5kb209MTUwNzc4JmNsYXNzSWQ9MCZvbmVUaW1lVmFsaWQ9MCZ2b2RTdWJBcHBJZD0w
```

You can also use third-party tools such as Postman to send HTTP requests. Please search for specific usage on the internet.

Upload page service code interpretation

1. `main_handler()` is the entry function.

2. Read the content of the `web_upload.html` file, which is the upload page content.

```
html_file = open(HTML_FILE, encoding='utf-8')
html = html_file.read()
```

3. Read configuration items from `config.json`, which refer to the content that you cannot predict when you write the SCF service and need to determine during the deployment process. The content is written into `config.json` in real time by the deployment script before deploying the upload page service.

```
conf_file = open(CONF_FILE, encoding='utf-8')
conf = conf_file.read()
conf_json = json.loads(conf)
```

4. Call `render_template` and modify the upload page content according to the configuration information obtained in the previous step. The configuration items are expressed in the format of `"variable name": "value"` in the `config.json` file or in the format of `{variable name}` in the `web_upload.html` file. **When modifying them, please replace them with the specific values as detailed below.**

```
def render_template(html, keys):
  """Replace the variables (in the format of `${variable name}`) in HTML with
specific content."""
  for key, value in keys.items():
    html = html.replace("${" + key + "}", value)
  return html
```

Variable	Description	Value Type	Value Source
UGC_UPLOAD_SIGN_SERVER	Upload signature distribution service URL	String	Output by SCF CLI after the deployment of the upload signature distribution service is completed.

5. Return the modified content of the upload page. For the formats and descriptions of the returned data, please see Overview of API Gateway Trigger.

```
return {
    "isBase64Encoded": False,
    "statusCode": 200,
    "headers": {'Content-Type': 'text/html'},
    "body": html
}
```

Upload signature distribution service code interpretation

1. `main_handler()` is the entry function.

2. Call `parse_conf_file()` and read the configuration information from the `config.json` file. The configuration items are as described below (for specific parameters, please see Signature for Upload from Client):

Field	Data Type	Description
secret_id	String	API key
secret_key	String	API key
sign_expire_time	Integer	Signature validity period in seconds
class_id	Integer	Category ID of the uploaded video. 0 indicates the default category
otp	Integer	Whether the signature is one-time
subappid	Integer	Whether to upload to a VOD subapplication

3. Call `parse_source_context()` to parse the `sourceContext` field in the request body, which can be passed through to the event notification receipt service during video upload completion event notification (not used in this demo).

Note:

This field is optional during the upload process. If you don't need this feature, you can ignore this part of the code.

4. Call the `generate_sign()` function to calculate the signature. For more information, please see Signature for Upload from Client.

5. Return the signature. For the formats and descriptions of the returned data, please see Overview of API Gateway Trigger.

```
return {
    "isBase64Encoded": False,
    "statusCode": 200,
    "headers": {"Content-Type": "text/plain; charset=utf-8",
                "Access-Control-Allow-Origin": "*",
                "Access-Control-Allow-Methods": "POST,OPTIONS"},
    "body": str(signature, 'utf-8')
}
```

How to Increase the Speed and Success Rate of Media File Upload

Last updated：2021-10-28 14:38:10

Overview

No matter whether it is an upload from client or server, the following common quality problems may be encountered during file transfer:

1. Why is the file upload so slow?

2. How to speed up upload?

3. How to improve the upload success rate?

4. How to fix upload failures on mobile devices on weak networks?

The metrics that measure upload quality include upload speed and success rate.

The upload speed often affects the user experience in the most intuitive manner. For example, if users upload a video of 50 MB which is not completed after half an hour, they may lose patience, leading to the possibility of customer loss. The upload success rate is a guarantee of service quality. After the initial upload fails due to network problems, the possibility of initiating the upload again by the user will decrease. The immediate consequence will be user complaints. Therefore, ensuring high upload success rate is the most fundamental requirement.

This document describes the causes of and solutions to problems in VOD upload scenarios. You can make comparisons based on your actual business scenarios and choose appropriate solutions to improve the upload quality.

Things Affecting Upload Quality

Network bandwidth

Network bandwidth refers to the amount of data that can be transferred in a time unit. The higher the bandwidth, the greater the amount of data uploaded per time unit, and the faster the upload. Upload is an end-to-end activity, so the bandwidth at both ends has an impact on the upload quality. The backend servers of VOD currently have sufficient bandwidth, so the upload quality often depends on the user-side bandwidth.

Distance between user and storage center

The uploaded files eventually need to be stored in the storage center of VOD. After the user activates VOD, VOD will allocate **Chongqing** as the storage center by default. The distance between the user and the storage center affects the length of the network linkage.

For example, when a file is uploaded from Beijing to Chongqing, the linkage will be longer than that of uploading the same file from Chengdu to Chongqing, and the influencing factors will increase as the distance increases, eventually resulting in slower upload. Due to the long linkage, problems such as network jitter and packet loss during the transfer will also affect the upload success rate. For short linage, such problems may also occur, but their probability of occurrence is much lower than in long linkage. Shortening the distance between the user and the storage center is a key step in improving the upload quality.

Weak network

Weak network refers to the state of a network with high delay and packet loss rate, which is generally called "slow internet access". This problem is very common in real life, such as in elevators and subway trains. The main cause is poor reception in the affected environment, which results in slow or failing data packet transfer. This scenario is particularly common in uploads from client, especially in the current era of mobile internet. Weak network has been plaguing many developers as the most difficult issue to overcome in improving the upload success rate.

Solutions

Concurrent upload

For scenarios where network bandwidth is insufficient, a direct solution is to apply for more bandwidth. However, on a network with limited bandwidth, how to fully utilize the bandwidth for uploads is a problem that needs to be solved.
Concurrent upload can be divided into two levels:
File level, i.e., multiple files are uploaded at the same time.
Part level, i.e., multiple parts of a single file are uploaded at the same time.
At both levels, the bandwidth utilization can be improved by adjusting the corresponding number of concurrencies.

Concurrent file upload

Concurrent file upload refers to using multiple processes or threads to initiate upload operations simultaneously. At present, VOD does not provide related SDK packages for this mode. You can implement this feature by referring to the characteristics of specific programming languages. Below is a simple example based on the VOD SDK for Java.

```
import com.qcloud.vod.VodUploadClient;
import com.qcloud.vod.model.VodUploadRequest;
import com.qcloud.vod.model.VodUploadResponse;

import java.util.ArrayList;
import java.util.List;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;

public class Main {
```

```
    public static void main(String[] args) throws Exception {
        // Number of concurrencies
        Integer threadNumber = 20;
        // List of paths to files to be uploaded
        List<String> filePathList = new ArrayList<String>();

        // Add paths to files to be uploaded
        filePathList.add("/data/path1.mp4");
        filePathList.add("/data/path2.mp4");
        filePathList.add("/data/path3.mp4");

        // Create a thread pool
        ExecutorService pool = Executors.newFixedThreadPool(threadNumber);
        // Create an upload client
        VodUploadClient client = new VodUploadClient("your secretId", "your
secretKey");

        // Concurrent upload
        for (String path : filePathList) {
            // Submit an upload task
            pool.submit(new UploadThread(client, path));
        }
    }
}


// Upload thread
class UploadThread implements Runnable {
    // Upload client
    private VodUploadClient uploadClient;
    // File path
    private String filePath;

    public UploadThread(VodUploadClient uploadClient, String filePath) {
        this.uploadClient = uploadClient;
        this.filePath = filePath;
    }

    public void run() {
        VodUploadRequest request = new VodUploadRequest();
        request.setMediaFilePath(filePath);
        try {
            // Execute upload
            VodUploadResponse response = uploadClient.upload("ap-guangzhou",
request);
            System.out.println(response.getFileId());
        } catch (Exception e) {
            e.printStackTrace();
```

```
            }
        }
    }
```

Concurrent part upload

Concurrent part upload is applicable to uploading a large file in multiple parts simultaneously. The advantage of multipart upload lies in that a large file can be uploaded quickly. The SDK provided by VOD automatically selects simple upload or multipart upload based on the file size, eliminating your need to take care of every step in multipart upload. The number of concurrent parts of the file is specified by the `ConcurrentUploadNumber` parameter. For specific use cases, please see the corresponding SDK. The SDKs that currently support this parameter include:

SDK for Java

SDK for Python

SDK for Go

Nearby upload

Nearby upload refers to the ability to sense the location of the uploader and allocate the storage center closest to the uploader for file upload. For example, users in Chengdu will be allocated to the Chongqing rather than Shanghai region for upload.

The biggest benefit of the nearby upload capability is to shorten the transfer distance between the uploader and the server. This feature has the following advantages:

Shortened transfer distance and improved upload speed.

Improved stability and guaranteed success rate.

VOD natively supports nearby upload. You just need to simply confirm the following two points:

Activate multiple storage regions

The storage region provided by VOD is **Singapore** by default. If you want to take full advantage of the nearby upload capability, you need to activate regions desired for nearby upload in the console. For more information, please see Upload Storage Settings. After multi-region storage is enabled, when a user uploads a file, VOD will identify the user's region by IP and intelligently allocate the region closest to the user out of the activated regions for upload.

Check whether the scheduling is proper

If the storage regions of Chongqing and Shanghai are activated, and the user initiates an upload in Chengdu, the file will theoretically be uploaded to Chongqing through nearby scheduling. To confirm whether the scheduling is reasonable, you can get the `FileId` returned upon upload completion, and confirm it against the basic information (`basicInfo`) returned by the DescribeMediaInfos API, which contains a `StorageRegion` field representing the storage region of the uploaded media file.

If the transfer experiences proxy or forwarding, and the region identified by VOD by IP is incorrect, you can forcibly specify a storage region for file uploads. For specific directions, please see:

Guide for upload from client

Guide for upload from server

Pre-detection upload

Pre-detection upload is mainly a means to optimize various types of scenarios with network errors, such as network connection failure, timeout, and DNS hijacking. It is an effective mitigation solution offered by VOD for uploads on weak networks. The optimization strategy includes the following points:

HTTPDNS is used to resolve domain names and get backend addresses to prevent DNS hijacking.

The connectivity and upload speed in multiple regions are detected to determine the optimal upload target region.

Tencent Cloud CDN acceleration network is utilized to provide reliable and stable transfer tunnels.

The pre-detection upload capability is currently applied to uploads from client, and the connection method is simple. For specific directions, please see the description of `pre-upload` in the corresponding SDK:

Upload SDK for Android

Upload SDK for iOS

Smart Cold Storage of VOD Media Asset Files

Last updated：2022-05-26 12:13:36

Smart cold storage relies on the media asset management capabilities of VOD. To reduce the storage costs incurred when you use VOD, VOD enables media asset lifecycle management, with which you can change the storage class of VOD files from STANDARD to STANDARD_IA, ARCHIVE, or DEEP_ARCHIVE according to certain policies based on your business characteristics. In this way, you can manage your media asset files more flexibly.

Use Cases

Ecommerce live streaming: as stipulated by Measures for the Supervision and Administration of Online Trading issued by the State Administration for Market Regulation of China, live streaming service providers shall retain live streaming videos of online transactions for at least three years after the live streaming ends. Such videos are generally stored in VOD in STANDARD storage class, some of which will never or seldom be played back and will be only used for audit by applicable authorities. The smart cold storage feature can effectively help you reduce the storage costs of media assets.

Cold storage of infrequently accessed media: for video portals, streaming media platforms, and UGC management platforms, media assets that are infrequently accessed or watched by users cannot be directly removed for certain reasons, which incur high storage costs. The smart cold storage feature of VOD can store media files in a cold storage class according to the number of accesses, which effectively reduces the storage costs of media assets while still allowing infrequent watches.

Media asset archive: in news, media, radio, and TV industries, some media asset files are highly sensitive to time and generally stored for a long period as historical materials and will be searched and watched in the future only when required. In such scenarios, you can change their storage class to ARCHIVE or DEEP_ARCHIVE to reduce the storage costs.

Prerequisites

1. You have signed up for and logged in to your Tencent Cloud account and completed identity verification.
2. You have activated the VOD service.
3. You have created relevant policies. For more information, see Media Asset Cold Storage.

Use Instructions

To use the media asset cold storage feature, you should understand the concepts of storage class, data retrieval and retrieval mode, and policy management.

Storage class

VOD provides the following storage classes for storing your media asset files: STANDARD, STANDARD_IA, ARCHIVE, and DEEP_ARCHIVE. Their attributes are as detailed below:

Storage Class	STANDARD	STANDARD_IA	ARCHIVE	DEEP_ARCHIVE
Default in VOD	Yes	No	No	No
Storage costs	High	Medium	Low	Very low
Access performance	High	Low	Access not supported	Access not supported
Data retrieval fees	No	No	Yes	Yes
Supported regions	All	All	All	Beijing, Shanghai, and Chongqing

STANDARD is the default storage class in VOD. Live recording files as well as files generated in various upload methods or video processing tasks are stored in STANDARD by default.

Attribute	Ranking from High to Low
Storage costs	STANDARD > STANDARD_IA > ARCHIVE > DEEP_ARCHIVE
Access performance	STANDARD > STANDARD_IA

Note:

ARCHIVE and DEEP_ARCHIVE storage classes don't support direct access, and you should retrieve the data first before you can access it. VOD only allows you to retrieve data to **STANDARD**.

Access performance will affect video watch metrics such as **time to first frame (TTFF) and lag rate**; therefore, **we recommend you not change the storage class for frequently accessed businesses in the production environment**.

VOD allows you to change between storage classes as follows:

Source Storage Class	Target Storage Class
STANDARD	STANDARD_IA, ARCHIVE, and DEEP_ARCHIVE
STANDARD_IA	STANDARD, ARCHIVE, and DEEP_ARCHIVE
ARCHIVE	STANDARD

DEEP_ARCHIVE	STANDARD

Note:

To switch ARCHIVE and DEEP_ARCHIVE to another storage class other than STANDARD, you should switch them to STANDARD first and then switch STANDARD to the target storage class.

The granularity of storage class change is `FileId`, that is, the storage classes of the original and video processing files are the same, and you cannot specify the storage class for part of the files under the `FileId` separately.

Data retrieval and retrieval mode

Data retrieval

After the storage class of a media asset is changed to the ARCHIVE or DEEP_ARCHIVE storage class, the file cannot be directly accessed, that is, operations such as video playback and processing initiation cannot be performed. If you want to access it, you can change its storage class as instructed in the previous section. However, sometimes you only want to access a media asset for a short period of time and still retain it in ARCHIVE or DEEP_ARCHIVE after access. In this case, the above method is not applicable. To meet such needs, VOD provides the data retrieval capabilities.

VOD supports two retrieval operations for media in ARCHIVE and DEEP_ARCHIVE: permanent retrieval and retrieval for specified period.

Operation on ARCHIVE/DEEP_ARCHIVE	Storage Class Change	Retrieval
Real-Timeness	Async	Async
Mode	Expedited/Standard/Bulk	Expedited/Standard/Bulk
Validity period	Permanent	Specified period
Copy generation	No	Yes

1. After retrieval, VOD will generate a copy for the media asset in STANDARD storage class.

2. A retrieval copy is valid for a specific number of days, within which the media asset can be accessed. Once expired, it will be automatically cleared, and then the media asset will become inaccessible.

3. Within the copy's validity period, additional STANDARD storage fees will be incurred.

4. Within the copy's validity period, the media asset cannot be retrieved again.

Retrieval mode

Storage class change and retrieval for ARCHIVE and DEEP_ARCHIVE have multiple modes, which have the same final effect but are different in speed and cost (i.e., retrieval fees).

Retrieval Mode	Time of Retrieval from ARCHIVE	Time of Retrieval from DEEP_ARCHIVE

Expedited mode	5 minutes	Not supported
Standard mode	5 hours	24 hours
Bulk mode	12 hours	48 hours

Note:

A media asset (represented as one `FileId`) can contain many files in storage, such as original file, transcoded file, and screenshot file, each of which may have different actual retrieval completion time. VOD doesn't maintain the specific retrieval completion status of each file but takes the media asset as a whole to calculate its retrieval completion time based on the longest possible time uniformly. Before that time arrives, even if all files under the media asset are retrieved, the media asset will still be marked as not retrieved and be inaccessible.

As the time point when a media asset is marked as unretrieved will be after the actual retrieval time point, the available duration of the retrieved copy of the media asset will be shorter than the expected period. To ensure that the copy has a sufficient available duration, we recommend you add one more day to the validity period when retrieving a media asset from DEEP_ARCHIVE.

Policy management

To help you uniformly manage the lifecycle of a large number of media assets, VOD combines media asset information and media asset file playback statistics to offer the policy-based smart management system.
The VOD backend executes management tasks every day and changes the storage class of media assets that meet the conditions of the specific policy.

Policy capabilities

Combined filtering

A policy allows you to specify a series of conditions to change the storage class of media assets that meet all conditions **at the same time**. The specific filter conditions are as detailed below:

Media asset type, such as video, audio, and image.

Media asset creation date, which offers the following options:

Files created before the specified date.

Files created after the specified date.

Files created between two specified dates.

Files created before a certain number of days (which dynamically changes with the current date).

Unlimited (that is, all media assets meet this condition).

Category ID. You can specify multiple category IDs or none (that is, all media assets meet this condition).

Media asset source type, including live recording, upload, video editing, and other. You can specify multiple source types or none (that is, all media assets meet this condition).

Number of recent playbacks. You can filter media assets whose number of playbacks is less than the specified value in the specified number of days. You can also set no limit (that is, all media assets meet this condition).

Disabling policy

Once created, a policy will be started automatically, and you can disable it as needed. Once disabled, it will be ignored in management tasks created on the next day and afterwards until it is enabled again.

Note:

Once disabled, a policy may still take effect or not take effect in the management tasks on the current day.

Restrictions

Up to 10 policies can be configured.

You cannot specify the policy priority. If a media asset hits multiple policies at the same time, the policy priorities will be automatically determined by the target storage class as follows: DEEP_ARCHIVE > ARCHIVE > STANDARD_IA.

Configuring VOD Smart Cold Storage Policy

Step 1. Create a cold storage policy

Log in to the VOD console (non-admin), click **Media Assets** > **Cold Storage** on the left sidebar, and click **Create Policy**.
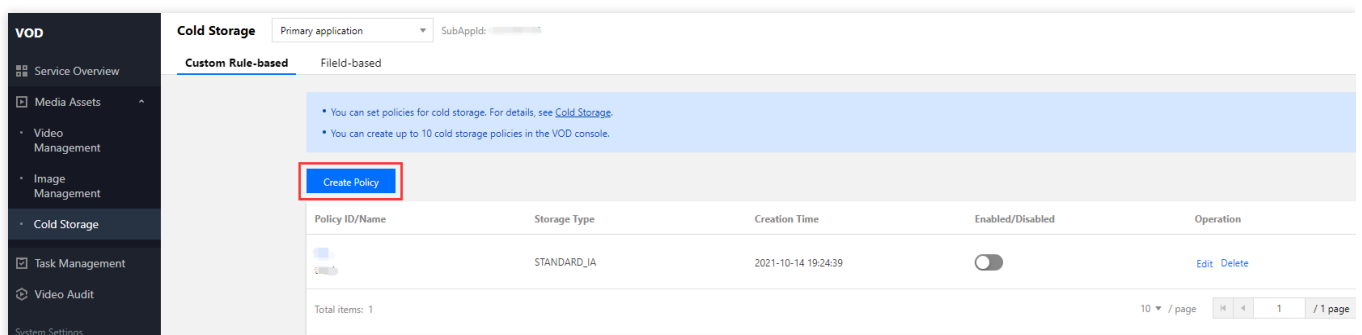
Step 2. Configure the cold storage policy

Configure the cold storage policy as needed.

Example: as shown above, if the number of video playbacks is less than 300 in 30 days for a video, the STANDARD_IA storage policy will be triggered, and VOD will store all media assets meeting the rule in STANDARD_IA.

Step 3. Confirm the configuration for it to take effect
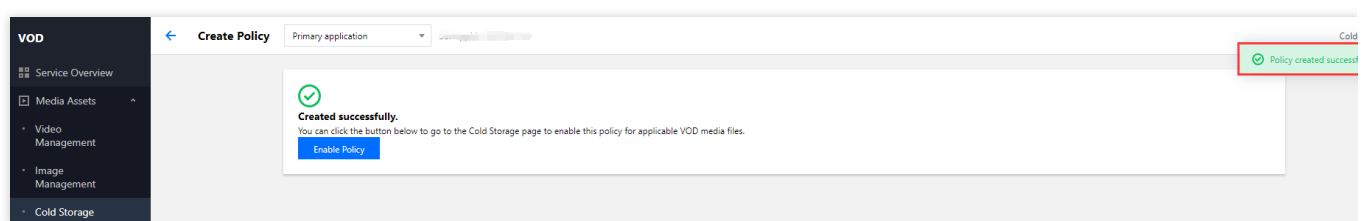
Video Processing

How to Add Subtitles to a Video

Last updated：2024-11-11 16:28:30

VOD supports adding (or deleting) standard multilingual subtitle files to HLS videos outputted from Adaptive Bitrate Streaming. During playback, users can switch between different language subtitles to enhance the viewing experience.

The process includes two steps: **Add subtitles to the media files** and **Associate subtitles with the media files**. For ease of description, this article assumes the following specific scenario: add Chinese (zh) and English (en) subtitles to the output file from the adaptive streaming template ID (10) for FileId (1397757888586312345), so that when playing the file, you can choose to display the added Chinese or English subtitles.

Step 1. Add Subtitles to Media Files

This step adds subtitles to the media asset with FileId (1397757888586312345) and obtains a subtitle ID. Next, this subtitle ID will be used for subtitle association.
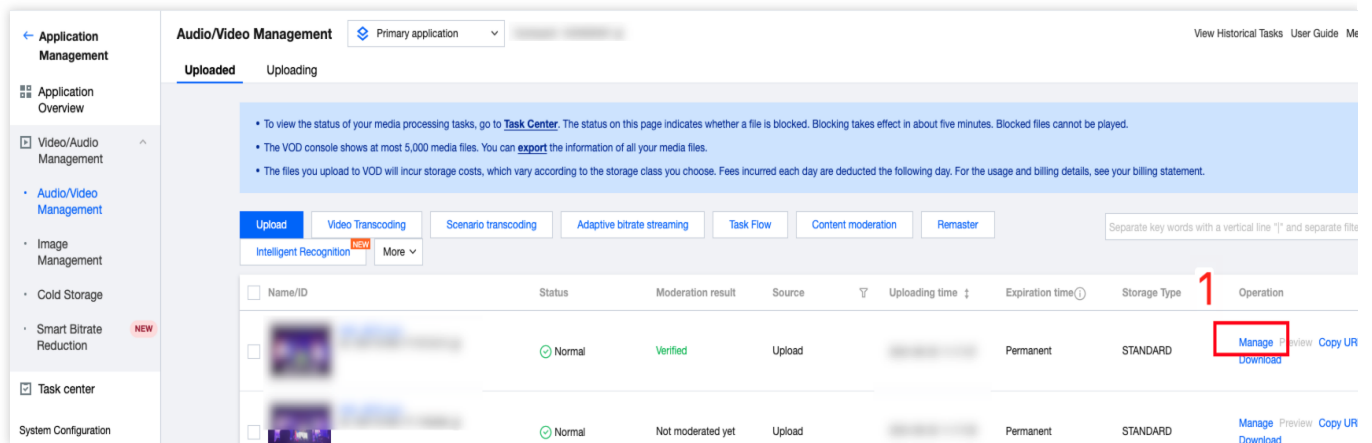
VOD provides two ways to add subtitles to media files: uploading subtitles and automatic subtitle generation.

Upload Subtitles

Upload existing subtitle files. After a successful upload, the subtitle information is automatically added to the media asset. There are two upload methods: console upload and server API upload.

Upload through console

1. log in to VOD console, go to the audio and video management page of the target application, and click **manage** the media asset you need to process.

2. Click **Subtitle > Add Subtitles.**

3. Upload the required subtitle file, and edit **language** and **stream name** (for display in the player), then click **Confirm.**

4. The Chinese and English subtitle IDs are qFXJrv and PPGQep, respectively.

Server API Upload

1. Invoke the server API ModifyMediaInfo, input parameters AddSubtitles to specify the subtitles to be uploaded:

```
"AddSubtitles": [
    {
      "Name": "demo_cn.vtt",
```

```
      "Language": "zh",
      "Format": "vtt",
      "Content": "V0VCVlRUCgowMDowMDowMC4wMDAgLS0+IDAwOjAwOjAyLjA0MArkvaDlpb3lkJf
    },
    {
      "Name": "demo_en.vtt",
      "Language": "en",
      "Format": "vtt",
      "Content": "V0VCVlRUCgowMDowMDowMC4wMDAgLS0+IDAwOjAwOjAyLjA0MApIb3cgYXJlIHl
    }
  ]
```

2. Example of interface output (other fields omitted), from which the Chinese and English subtitle IDs are qFXJrv and PPGQep, respectively.

```
{
    "Response": {
        "AddedSubtitleSet": [
            {
                "Id": "qFXJrv",
                "Name": "demo_cn",
                "Language": "en",
                "Format": "vtt",
                "Url": "http://123.vod2.myqcloud.com/vodgzp123/1397757888586312345/
            },
            {
                "Id": "PPGQep",
                "Name": "demo_en",
                "Language": "en",
                "Format": "vtt",
                "Url": "http://123.vod2.myqcloud.com/vodgzp123/1397757888586312345/
            }
        ],
        "RequestId": "90b40581-bf44-4455-a238-c8fe614c20ca"
    }
}
```

Intelligent Recognition Subtitle Generation

Through Audio and Video CR, automatically generate subtitles and add subtitle information to media assets. The process includes: preparing Audio and Video CR Templates, Task Initiation, and Result Acquisition. The key points are as follows:

1. Preparing Audio and Video CR Templates

There are two ways to create templates: console creation and server API creation.

1.1. Console Creation of Templates

1.1.1. Log in to VOD Console, after entering the target application, click **Media Processing > Template Settings > Intelligent Recognition Template,** click **Create Intelligent Recognition Template.**
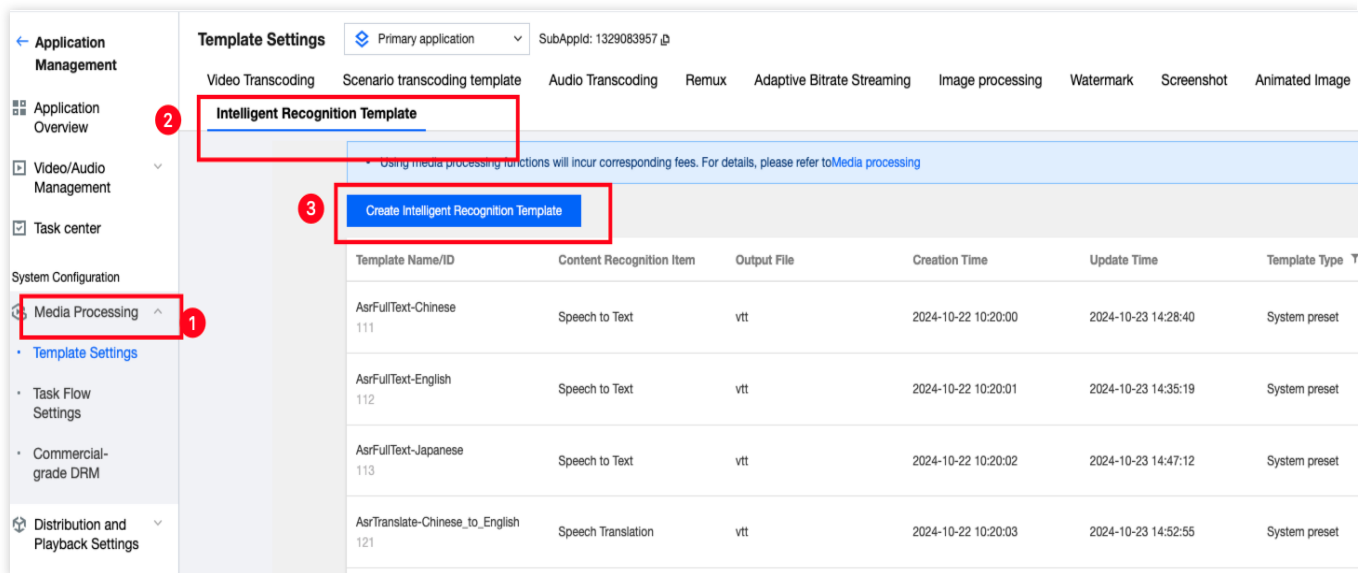
1.1.2. Supports selecting "Speech to Text" to output the source language subtitle file, or "Speech Translation" to output the subtitle file in the target translation language. The following screenshot scenario describes converting source video speech to CR and translating it to the target language subtitle file.

1.1.3. After creating, you can see the newly created template ID 72314 in the template list.

1.2. Create template via Server API

1.2.1. Invoke the server API to create an audio/video CR template. The input parameter AsrTranslateConfigure specifies the speech translation task parameters:

```
"AsrTranslateConfigure": {
    "Switch": "ON",
    "SrcLanguage": "zh",
    "DstLanguage": "en",
    "SubtitleFormats": ["vtt"]
}
```

Note, if subtitle translation is not needed, you can use Speech Full Text Recognition (AsrFullTextConfigure) to specify generating subtitles in the video source language:

```
"AsrFullTextConfigure": {
    "Switch": "ON",
    "SrcLanguage": "zh",
    "SubtitleFormats": ["vtt"]
}
```

1.2.2. According to this scenario, we specified Speech Translation (AsrTranslateConfigure). The interface output example (other fields omitted) provides the template ID (72314).

```
{
    "Response": {
        "Definition": 72314,
        "RequestId": "12ae8d8e-dce3-4151-9d4b-5594145287e1"
    }
}
```

2. Task Initiation

Multiple methods are supported to initiate intelligent recognition tasks:

Invoke server API

Initiate task flow from the console

Specify task flow upon upload

Upload from server

Upload from client

Upload through console

For the above methods, please refer to the Developer Guide Task Initiation. Additionally, you can directly initiate an II task through the console as follows:

2.1. Log in to VOD console, go to the Audio and Video Management page of the target application, select the FileId for subtitle generation, and click **Intelligent Recognition.**

2.2. Click **Select template.**

2.3. Select the recognition template you want to use and then click **Confirm.**

Select Intelligent Recognition Template   ✕

When the content recognition item is "Speech to Text," it automatically generates subtitle files by intelligently recognizing the audio in videos. When the content recognition item is "Speech Translation," it recognizes and translates into specified languages, generating multilingual subtitle files.

Template name/ID	Content Recognition Item	Subtitle Format	Creation Time	Update Time	Template Type
○ Malay_to_English 123	Speech Translation	vtt	2024-10-22 10:20:05	2024-10-23 15:02:26	Preset
○ AsrTranslate-English_to_Malay 124	Speech Translation	vtt	2024-10-22 10:20:06	2024-10-23 15:07:44	Preset
○ AsrTranslate-Chinese_to_Japanese 125	Speech Translation	vtt	2024-10-22 10:20:07	2024-10-23 15:12:52	Preset
○ AsrTranslate-Chinese_to_Korean 126	Speech Translation	vtt	2024-10-22 10:20:08	2024-10-23 15:18:08	Preset
○ **①** demo11 72314	Speech Translation	vtt	2024-11-01 15:45:11	2024-11-01 15:45:11	Custom

Total items: 10   20 ∨ / page |◄ ◄ 1 / 1 page ► ►|

②   [Confirm] [Cancel]

Using media processing (transcoding, ultra-fast HD, adaptive streaming, task flow, audio/video review, intelligent recognition) functions will incur corresponding fees. For details, please refer to **Media Processing**.

Processing Type   ○ Video Transcoding   ○ Scenario templates ^NEW   ○ Adaptive bitrate streaming   ○ Task Flow   ○ Content moderation   ○ Remaster   ● Intelligent Recognition ^NEW

Intelligent Recognition *   [Select template]

Template name/ID	Content Recognition Item	Subtitle Format	Creation Time	Update Time	Template Type	Ope
demo11 72314	Speech Translation	vtt	2024-11-01 15:45:11	2024-11-01 15:45:11	Custom	Rem

[Confirm] Cancel

The page will indicate that the task has been successfully created:
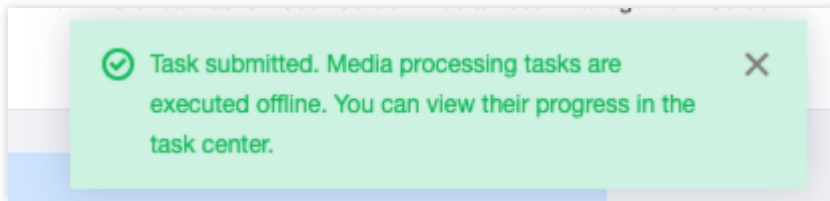
2.4. On the **Task Center** page, you can see the task ID and task status of the subtitle generation task.
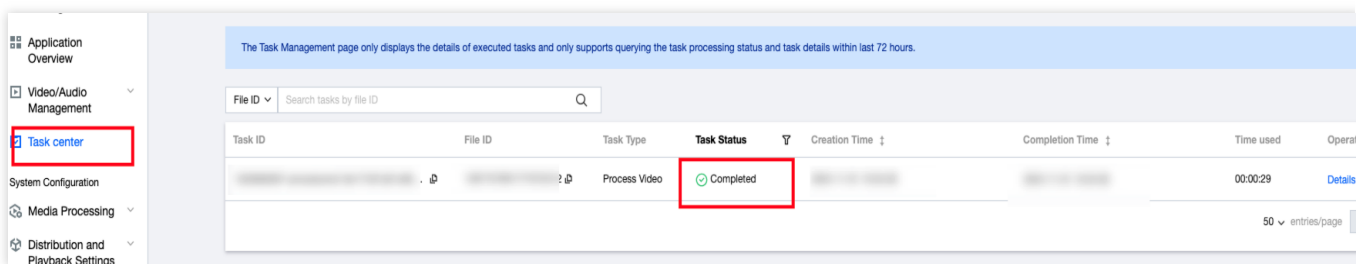
When the task status shows Completed, the subtitles have been successfully generated and automatically added to the media assets.

3. Result Acquisition

Using the task ID obtained from Back 'Task Initiation' and following the development guide Result Acquisition, you can obtain the subtitle generation results, which are located in the AiRecognitionResultSet array (type AiRecognitionResult) within the ProcedureTask structure:

When Type is AsrFullTextRecognition, it indicates that the full text translation result is saved in AsrFullTextTask.

When Type is AsrTranslateRecognition, it indicates that the translation result is saved in AsrTranslateTask.

Example of AsrTranslateTask:

```
{
    "Status": "SUCCESS",
    "ErrCode": 0,
    "ErrCodeExt": "",
    "Message": "SUCCESS",
    "Progress": 100,
    "BeginProcessTime": "2024-10-16T12:21:04Z",
    "FinishTime": "2024-10-16T12:21:47Z",
    "Input": {
        "Definition": 102270
    },
    "Output": {
        "SegmentSet": [
```

```
                {
                    "Text": "How are you?",
                    "Translation": "How are you?",
                    "StartTimeOffset": 0,
                    "EndTimeOffset": 2.04,
                    "Confidence": 100
                },
                {
                    "Text": "I am good, thank you. How about you?",
                    "Translation": "Fine, thank you, and you?",
                    "StartTimeOffset": 2.1,
                    "EndTimeOffset": 4.95,
                    "Confidence": 100
                },
                {
                    "Text": "I am good too.",
                    "Translation": "I'm fine too.",
                    "StartTimeOffset": 4.98,
                    "EndTimeOffset": 6.9,
                    "Confidence": 100
                }
            ],
            "SegmentSetFileUrl": "http://251000800.vod2.myqcloud.com/6c0f30dfvodgzp2510
            "SegmentSetFileUrlExpireTime": "2024-10-19T12:21:44.525Z",
            "SubtitleSet": [
                {
                    "Id": "qFXJrv",
                    "Name": "zh",
                    "Language": "zh",
                    "Format": "vtt",
                    "Url": "https://123.vod-qcloud.com/vodgzp123/1397757888586312345/su
                },
                {
                    "Id": "PPGQep",
                    "Name": "en",
                    "Language": "en",
                    "Format": "vtt",
                    "Url": "https://123.vod-qcloud.com/vodgzp123/1397757888586312345/su
                }
            ]
        }
    }
```

The example above shows that the Chinese and English subtitle IDs are qFXJrv and PPGQep, respectively.

When the task is successful, the generated subtitle information has been automatically added to the media assets.

Step 2. Associate subtitles with the media files

There are two ways to associate subtitles with the media files: associating subtitles via the console and associating subtitles via server APIs. The former only supports associating subtitles with already generated Adaptive Bitrate output files, while the latter supports associating subtitles with both stock and incremental files.

Console Subtitle Association

1. log in to VOD console, go to the audio and video management page of the target application. Click **Manage > Subtitles > Edit** for the Media Asset with FileId (1397757888586312345).

2. After clicking **Confirm**, an Operation Successful prompt will pop up in the upper right corner of the page, indicating that the subtitle association was successful.

Server API Associated Subtitles

Including associating subtitles with existing files and incremental files.

1. Associating subtitles with existing files

To associate subtitles with the output files of adaptive bitrate conversion, please invoke the server API Associated Media Subtitles. Input parameters are as follows:

```
{
    "Action": "AttachMediaSubtitles",
    "Version": "2018-07-17",
    "FileId": "1397757888586312345",
    "Operation": "Attach",
    "AdaptiveDynamicStreamingDefinition": 10,
    "SubtitleIds": ["qFXJrv", "PPGQep"]
}
```

2. Associating subtitles with incremental files

When initiating an Adaptive Bitrate Streaming task, specify the subtitle ID list in the input parameters of the Adaptive Bitrate Streaming task: MediaProcessTask -> AdaptiveDynamicStreamingTaskSet -> `SubtitleSet` .

The input parameter example for MediaProcessTask is as follows:

```
{
    "AdaptiveDynamicStreamingTaskSet": [
      {
        "Definition": 10,
        "SubtitleSet": ["qFXJrv", "PPGQep"]
      }
    ]
}
```

Once the task is completed, the output file of the adaptive bitrate streaming will be associated with subtitles.

Step 3. Playback Effect

After successfully associating subtitles, you can choose to display subtitles in different languages while playing the video. Refer to the effect in the image below:
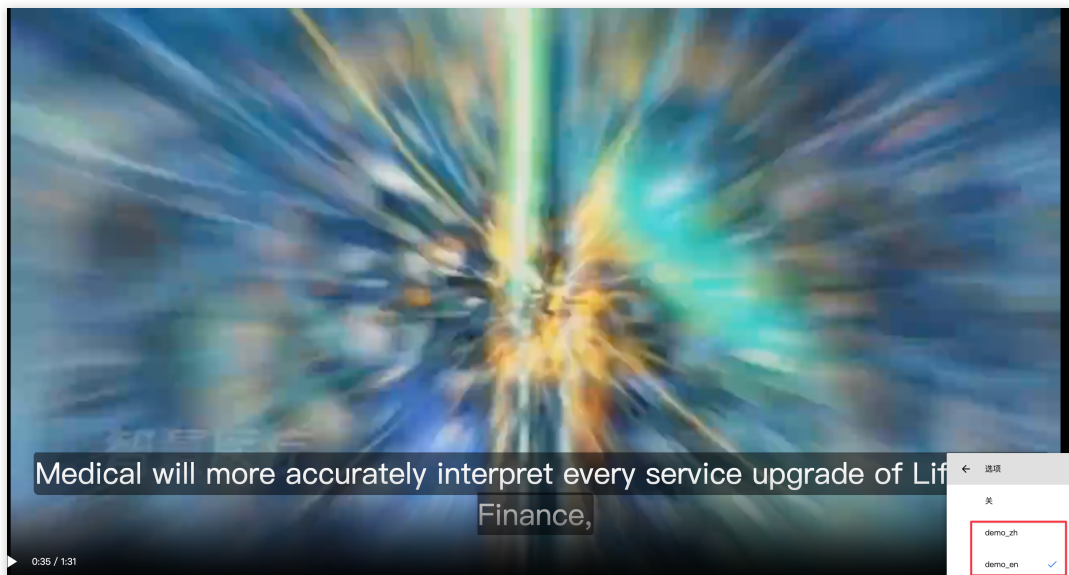
How to Transcode Videos

Last updated：2023-05-15 17:39:49

Instructions

Overview

This document describes how to transcode videos stored in VOD and how to get the outputs.

Costs

The open-source code provided in this document is free, but the following costs may incur.

Fees for purchasing a Tencent Cloud CVM instance to execute TencentCloud API requests. For more information, see CVM Billing Mode.

Fees for storing videos in VOD. For details, see Pay-As-You-Go and Prepaid Packages.

Fees for transcoding videos stored in VOD. For details, see Pay-As-You-Go and Prepaid Packages.

Fees for playing videos stored in VOD. For details, see Pay-As-You-Go and Prepaid Packages.

Parameters

VOD supports the following formats for transcoding:

Type	Parameter	Description
Input	Container format	WMV, RM, MOV, MPEG, MP4, 3GP, FLV, AVI, RMVB, TS, ASF, MPG, WEBM, MKV, M3U8, WM, ASX, RAM, MPE, VOB, DAT, MP4V, M4V, F4V, MXF, QT, and OGG.
	Video codec	AV1, AVS2, H.264/AVC, H.263, H.263+, H.265, MPEG-1, MPEG-2, MPEG-4, MJPEG, VP8, VP9, QuickTime, RealVideo, Windows Media Video
Output	Container format	Video: FLV, MP4, HLS (M3U8 + TS)
		Audio: MP3, MP4, Ogg, FLAC, M4A
		Image: GIF, WebP.
	Video codec	H.264/AVC, H.265/HEVC, AV1

The target specifications of a transcoding task include codec, resolution, bitrate, and others. VOD uses templates to represent different combinations of these parameters. For details, see Video Processing - Overview.

Category	Parameter	Description

Video encoding	Codec	H.264, H.265, and AV1 codecs are supported.
	Bitrate	Supported bitrate range: 10 Kbps - 35 Mbps.
	Frame rate	Supported frame rate range: 1-60 fps; common values: 24, 25, and 30.
	Resolution	Supported width range: 128 px - 4096 px. Supported height range: 128 px - 4096 px.
	GOP length	Supported GOP length range: 1-10s
	Profile	When the video codec is H.264, the baseline, main, and high profiles are supported. When the video codec is H.265, only the main profile is supported.
	Color space	YUV420p is supported.

Note:

Codec: A method of converting video files from a certain format into another using specific compression technology. Compared with H.264, H.265 uses more advanced encoding techniques and can transcode videos at much lower bitrates (which means lower bandwidth costs) without compromising video quality.

Bitrate: The size of data encoded by the encoder per second, in kbps. For example, 800 kbps indicates that the encoder generates 800 KB of data per second.

Frame rate: The number of frames per second.

Resolution: The number of pixels per inch.

GOP: The number of frames between two I-frames.

For general transcoding, we recommend the following resolution and bitrate combinations:

Definition	Recommended Bitrate	Recommended Resolution	Resolution Range
SD	600	640 x 480	SD (short side ≤ 480 px)
HD	2000	1280 x 720	HD (short side ≤ 720 px)
FHD	4000	1920 x 1080	FHD (short side ≤ 1080 px)
2K	6000	2560 x 1440	2K (short side ≤ 1440 px)
4K	8000	3840 x 2160	4K (short side ≤ 2160 px)

VOD's Top Speed Codec (TSC) solution integrates image quality remaster and enhancement, adaptive parameter selection, and V265 encoder, among other video processing features. It can transcode videos to higher quality at

lower bitrates, helping you save network resource costs while delivering a better viewing experience. For TSC transcoding, we recommend the following resolution and bitrate combinations:

Definition	Recommended Bitrate	Recommended Resolution	Resolution Range
SD	350 or leave it empty	640 x 480	SD (short side ≤ 480 px)
HD	1350 or leave it empty	1280 x 720	HD (short side ≤ 720 px)
FHD	2700 or leave it empty	1920 x 1080	FHD (short side ≤ 1080 px)
2K	3500 or leave it empty	2560 x 1440	2K (short side ≤ 1440 px)
4K	7500 or leave it empty	3840 x 2160	4K (short side ≤ 2160 px)

Note:

If bitrate is left empty, VOD will set the minimum bitrate automatically based on intelligent analysis of the source video.
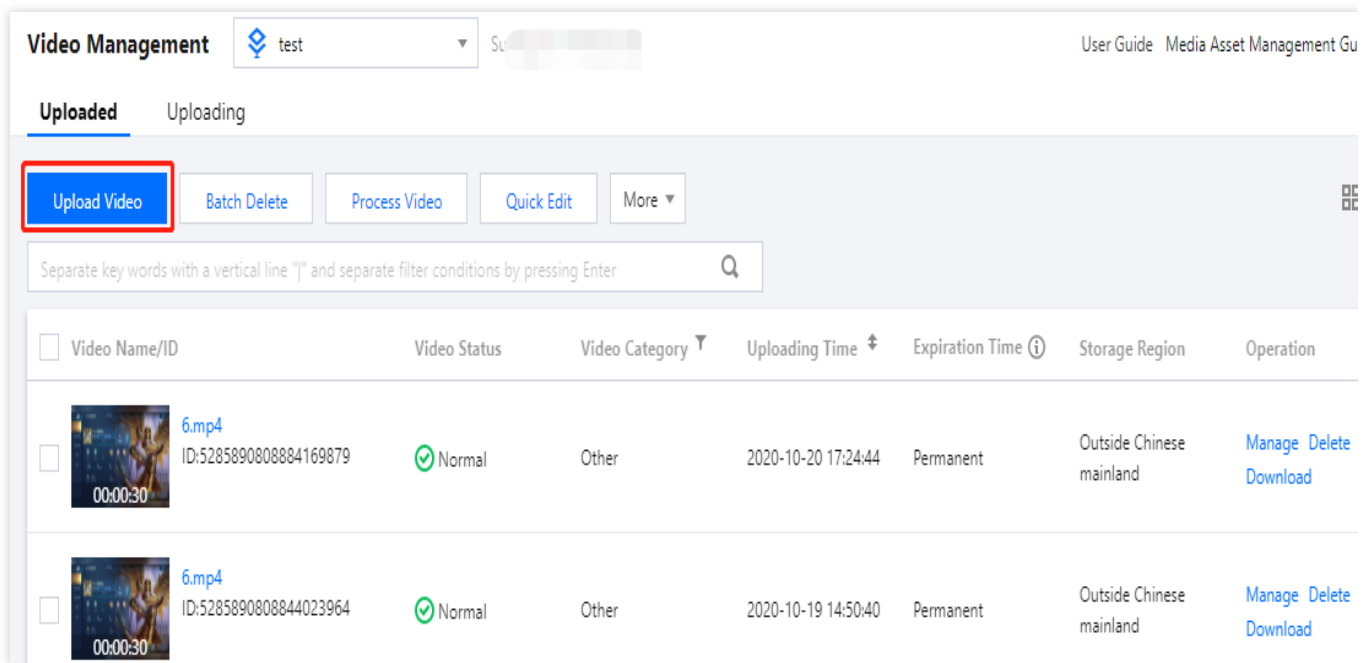
Initiating Transcoding Through the Console

Step 1. Activate VOD

Activate VOD. For detailed directions, see Getting Started - Step 1. Activate VOD.

Step 2. Upload a video

Upload a test video. For detailed directions, see Getting Started - Step 2. You can download the video used in this document here. The file ID generated is `3270835008936537687` .

Note:

To avoid the transcoding taking too much time, we recommend you use a short video (dozens of seconds) for test .

Step 3. Initiate transcoding

1. On the Video/Audio Management page of the VOD console, select the video you uploaded, and click **Transcoding**.

2. Select **Transcoding** as the processing type and select a **transcoding template**.

3. Click **Confirm**. As an example, the preset templates `STD-H264-MP4-360P` (ID 100010) and `STD-H264-MP4-540P` (ID 100020) are selected in this document. For directions on how to use a custom transcoding template, see Template Configuration.

4. Click **Confirm**.

©2013-2025 Tencent Cloud International Pte. Ltd.   Page 52 of 130

5. Go to **Task Center**. If the status of the task is "Completed", the video has been successfully transcoded.

Step 4. Get the transcoding output

1. In the VOD console, select the target application. On the **Media Assets > Video/Audio Management** page, if the status of your test video is "Normal", transcoding is finished. Click **Manage** on the right to enter the details page.
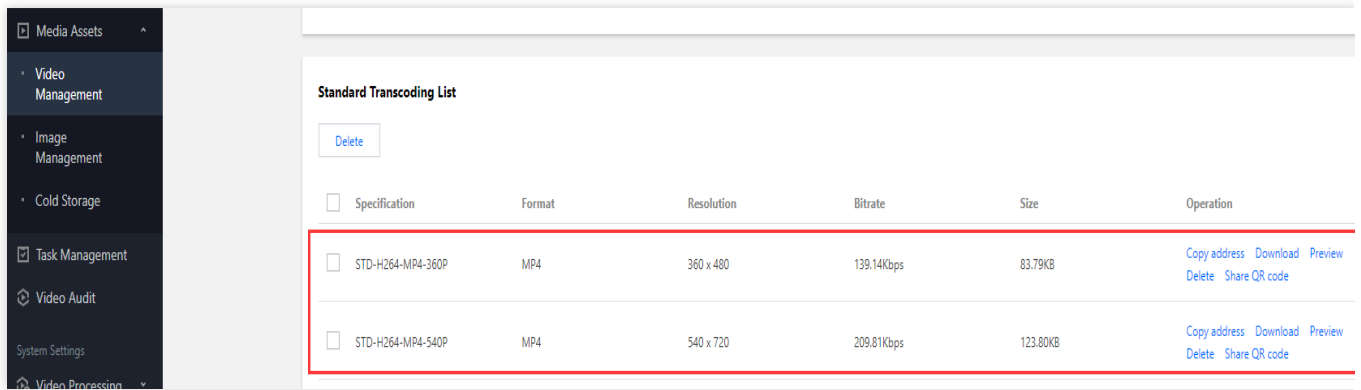
2. Under the **Transcoding outputs** tab, you will find the transcoding outputs generated by the `TD-H264-MP4-360P` and `STD-H264-MP4-540P` templates. You can click **Preview** on the right to play the video or copy the playback URL and share it to others.

Calling TencentCloud APIs to initiate transcoding

Step 1. Prepare a CVM instance

TencentCloud API script needs to be executed on a CVM instance that meets the following requirements:

Region: No limit.

Model: The minimum specification (1 CPU core and 1 GB memory) or higher.

Public network: A public IP address is required, and the bandwidth should be at least 1 Mbps.

Operating system: Official public image `Ubuntu Server 16.04.1 LTS 64-bit` or `Ubuntu Server 18.04.1 LTS 64-bit` .

For detailed directions on how to purchase a CVM instance and reinstall the system, see Operation Guide - Creating Instances via CVM Purchase Page and Operation Guide - Reinstalling System.

Note:

If you do not have a CVM instance that meets the above conditions, you can also run the script on other Linux (such as CentOS or Debian) or macOS servers with public network access, but you need to modify certain commands in the deployment script based on the operating system.

Step 2. Obtain the TencentCloud API key

An API key (`SecretId` and `SecretKey`) is required for calling TencentCloud APIs. If you have not created an API key yet, create one as instructed in Creating an API key for a root account. If you have already created a key, follow the steps in Viewing an API key of a root account to view the key.

Step 3. Activate VOD

Activate VOD. For detailed directions, see Getting Started - Step 1. Activate VOD.

Step 4. Upload a video

Upload a test video. For detailed directions, see Getting Started - Step 2. You can download the video used in this document here.
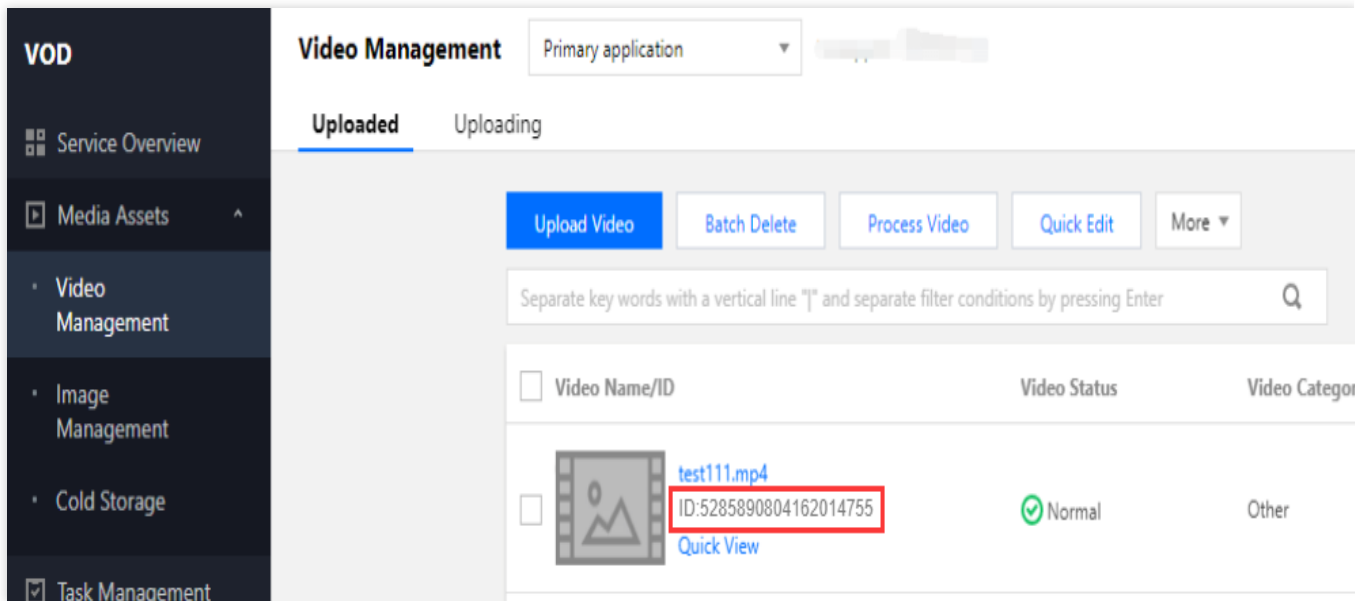
Note:

To avoid the transcoding taking too much time, we recommend you use a short video (dozens of seconds) for test .

Step 5. Initiate transcoding

1. Log in to the CVM instance prepared in Step 1 and run the following command on the remote terminal:

```
ubuntu@VM-69-2-ubuntu:~$ export SECRET_ID=AKxxxxxxxxxxxxxxxxxxxxxxxx; export SECRET_
```

Note:

You need to assign values to `SECRET_ID` and `SECRET_KEY` according to the key obtained in Step 2.

2. The above command will download the demo source code from GitHub and automatically run the installation script. The installation process will take several minutes (subject to the CVM network conditions), during which the remote terminal will print the following information:

```
[2020-06-15 20:39:56] Start installing pip3.
[2020-06-15 20:40:06] pip3 is successfully installed.
[2020-06-15 20:40:06] Start installing TencentCloud API Python SDK.
[2020-06-15 20:40:07] TencentCloud API Python SDK is installed.
[2020-06-15 20:40:07] Start configuring API parameters.
[2020-06-15 20:40:07] API parameters are configured.
```

3. Execute the `process_media.py` script to initiate transcoding.

```
ubuntu@VM-69-2-ubuntu:~$ cd ~/vod-server-demo/transcode_api/; python3 process_media
```

Note:

You need to replace `243791581340253754` in the command with the file ID generated in Step 4.

4. This command will initiate a ProcessMedia request for the video `243791581340253754` , transcode the video according to the preset templates `100010` and `100020` , and print the response:

```
{"TaskId": "1400329073-procedurev2-f6bf6f01612369b6db30f2224792a2aft0", "RequestId"
```

Step 6. Get the transcoding output
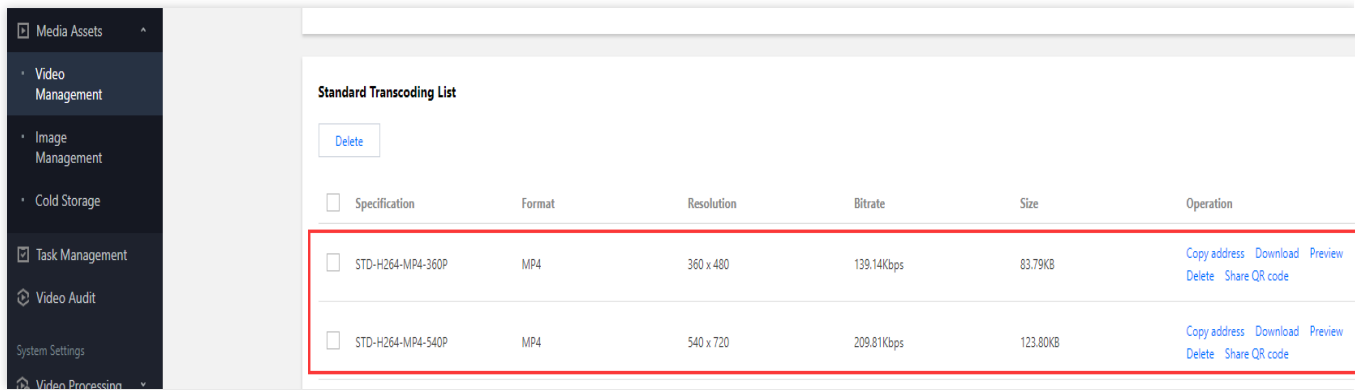
1. Go to **Task Center**. If the status of the task is "Completed", the video has been successfully transcoded.
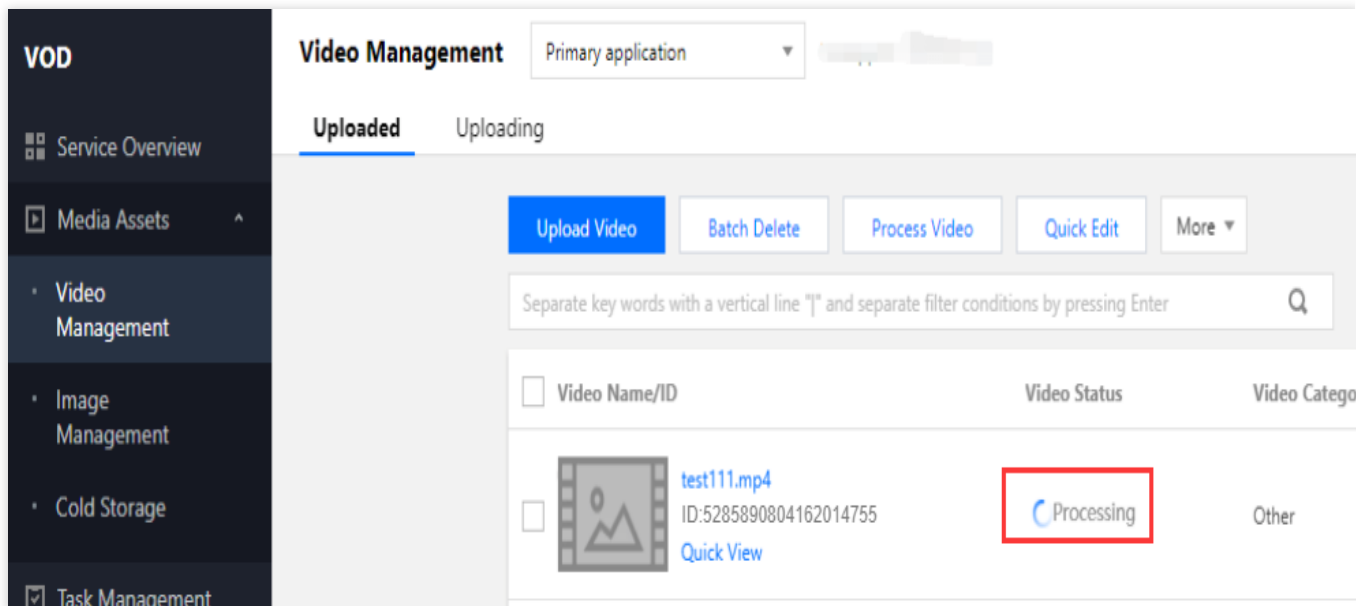
2. When the status of the test video changes to "Normal", the transcoding is finished. Click **Manage** on the right to enter the details page.
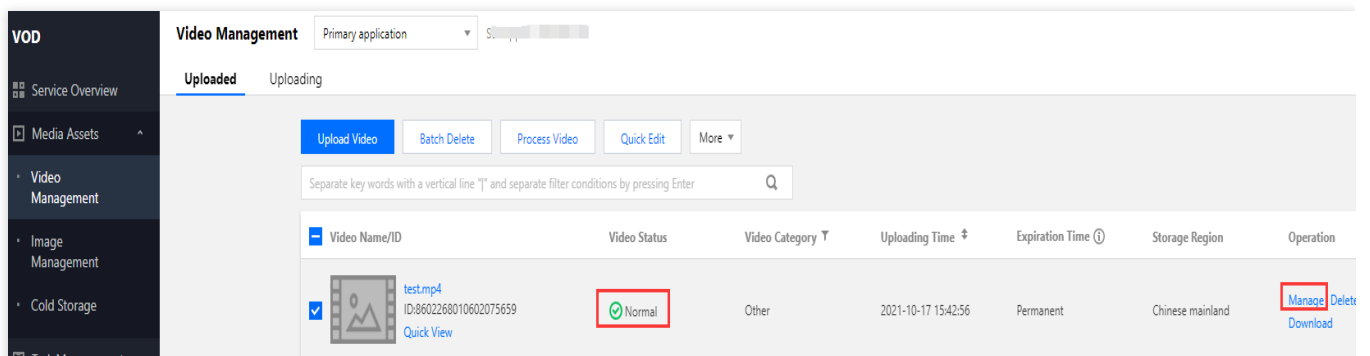
3. Under the **Transcoding outputs** tab, you will find the transcoding outputs generated. You can click **Preview** on the right to play the video or copy the playback URL and share it to others.

Auto Transcoding After Uploading (Task Flow)

VOD supports multiple upload methods, including upload through the console, upload from the server, upload from a client and upload by pulling from URLs. Whatever method you use, you can specify a task flow to automatically transcode the video after it is uploaded.

Auto Transcoding After Uploading (Event Notification)

VOD will send notification requests for both uploading and transcoding events. You can initiate transcoding tasks for newly uploaded videos based on the event notifications as well as get the transcoding results from notifications (or from the console as described above). For details about how to use event notifications, see How to Receive Event Notification.

How to Moderate Media Content

Last updated：2023-04-27 15:00:32

Non-compliant content may expose your business to legal risks and hurt your brand. VOD offers audio/video moderation and image moderation capabilities to help you ensure the compliance of your content.

Audio/Video Moderation

The audio/video moderation feature of VOD analyzes your video and audio content and gives suggestions on whether to publish or block the content.

You can initiate audio/video content moderation in the following two ways:

1. Initiate it in the VOD console.

2. Call a server-side API.

Using a server-side API to initiate audio/video moderation

The figure below shows how a video application moderates the videos uploaded by users using the server-side API.
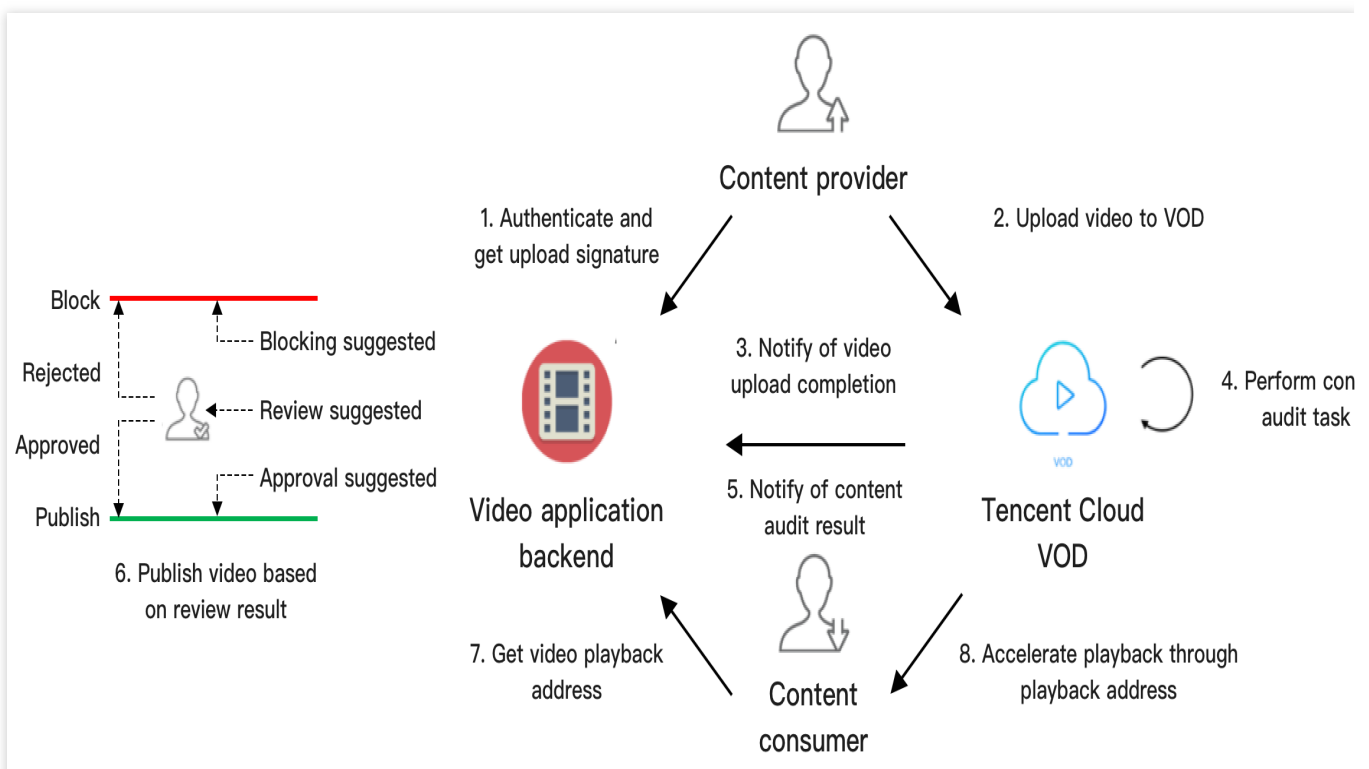
1. The video application backend authenticates the content provider and distributes a signature for upload from client.

2. The content provider uploads the content to VOD.

3. VOD sends information including the file ID and playback URL of the uploaded video to the application backend.

4. VOD executes a moderation task (`ReviewAudioVideoTask`) according to the `procedure` parameter passed in when the video is uploaded.

5. VOD sends the result to the application backend via the ReviewAudioVideoComplete notification.

If the moderation result is "block", there's a high chance that the content is non-compliant. We recommend you block the content.

If the moderation result is "pass", the probability of the content being non-compliant is low. We recommend you allow the content to pass.

If the moderation result is "review", the probability of the content being non-compliant is high; in this case, manual verification is recommended.

6. The application backend publishes videos whose suggestion is "pass" as well as those that are given a "review" suggestion and have passed manual verification.

7. Viewers request a playback URL for a published video from the application backend.

8. Viewers play the video via the URL (VOD offers acceleration services).

Steps 4-6 ensure that the video obtained in step 7 is compliant.

Note:

In the above process, videos are moderated before being published (only videos that pass the moderation or review are published). You can also use the post-moderation mode, in which videos are published first before moderation is performed (non-compliant content detected will be removed).

Image Moderation

The image moderation feature of VOD analyzes images and gives suggestions on whether to pass or block them.
Unlike audio/video moderation, image moderation is performed synchronously, and results are generated immediately.

If the moderation result is "block", there's a high chance that the content is non-compliant. We recommend you block the content.

If the moderation result is "pass", the probability of the content being non-compliant is low. We recommend you allow the content to pass.

If the moderation result is "review", the probability of the content being non-compliant is high; in this case, manual verification is recommended.

You can initiate image moderation in two ways:

1. Initiate it in the VOD console.

2. Call the server-side API ReviewImage.

Billing

For details about how content moderation is billed, see Purchase Guide.

How to Add Complex Watermarks to Videos

Last updated：2021-03-25 17:52:05

VOD's watermarking feature allows the adding of simple image or text watermarks, but cannot meet more sophisticated needs such as adding text-image watermarks or applying filters to watermarks. Given this, VOD has supported Scalable Vector Graphics (SVG) watermarks, allowing you to design text and image layout as you like, draw your own images, and apply filters, gradients, and other special effects.

What is SVG

SVG is an XML-based markup language for describing two-dimensional vector graphics. It has been widely applied to web standards including CSS, DOM, and JavaScript.
VOD does not set height or width limits for SVG. It can identify the smallest rectangle that contains all the elements of an image and use it as the original dimension of an SVG watermark. In the figure below, the dotted frame is the smallest rectangle identified.
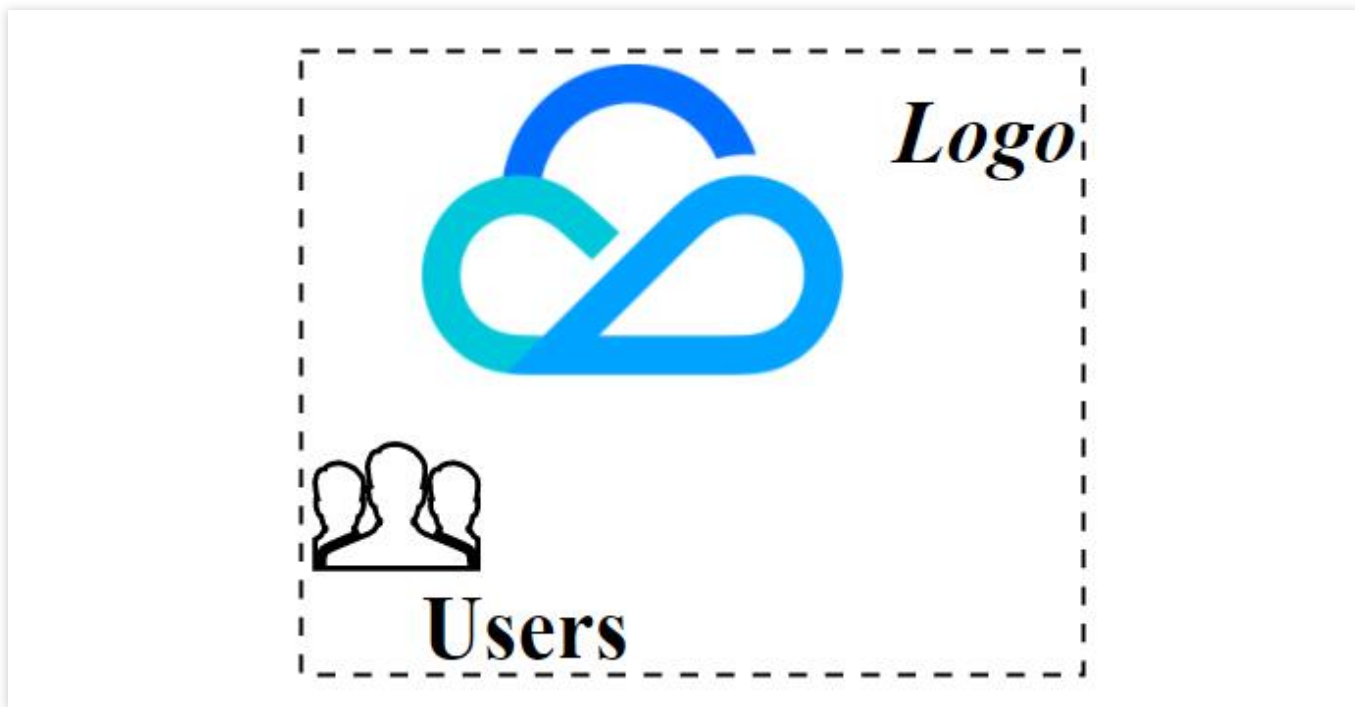
Note:

You can find on the internet different free, web-based SVG editors. Use them to draw the graphics you want and export them as XML files.
Before getting the final XML code, you can manually fine-tune your image, for example, align element attributes or change font size, and check the result in an editor.

How to Add SVG Watermarks

Step 1. Edit an SVG watermark.

1. Use an SVG editor (e.g., the simple tool Tryit Editor, or Online SVG Editor, which incorporates more advanced features) to draw your graphics. The graphics are displayed in real time as you edit. After editing, you can save the code as an HTML file, which you can open with a browser later to check the effect.

Note:

If you need to align different elements, we recommend that you make flexible use of the elements' alignable attributes and check if the alignment is effective by changing the values of the elements, for example, placing images next to text, or reducing/increasing text length. For detailed instructions, see this SVG tutorial. An example is offered below.

Step 2. Create an SVG watermark template.

Call the `CreateWatermarkTemplate` API and specify parameters including the watermark coordinates and dimensions. After the creation, you get a watermark template ID.

Step 3. Add the SVG watermark to a video.

For how to start a video processing task and get the result, see Video Processing Task System.

Take the `ProcessMedia` API as an example. In Example2 Initiating a transcoding task, you should set `WatermarkSet.N.Definition` to the template ID obtained in step 2 and, as an SVG watermark is used, pass the XML code obtained in step 1 in `WatermarkSet.N.SvgContent`.

Example

This section uses an example to guide you through the process of adding a complex watermark that contains an image and replaceable text.

Use case

A client wants to watermark videos and has the following requirements:

The watermark consists of a brand logo and ID of the logged in account (text).

The watermark appears in the top right corner of videos. The distance between its origin and the right and top border of videos is 2% of video width and 5% of video height respectively.

The width of the watermark is 30% of video width, and the height scales proportionally.

The text is beneath the logo and aligns to right of the logo.

The font for the text is SimHei, the background in white, with Gaussian blur and shadow in black. The font size is 50 px when the brand logo is not scaled.

Step 1. Edit the SVG watermark.

Below is the XML source code for the SVG watermark:

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"
    <defs>
        <filter id="filter" x="0" y="0">
            <feGaussianBlur stdDeviation="2"/>
            <feOffset dx="0" dy="3"/>
        </filter>
    </defs>

    <image id="img_watermark"  xlink:href="data:image/png;base64,iVBORw0KGgoAAAANSU
        x="74.4%" y="0px" height="185px" width="256px"/>
    <text id="text_watermark_shadow" text-anchor="end" font-family="SimHei" font-st
        x="100%" y="245px" style="opacity:1;" fill="black" filter="url(#filter)">@T
    <text id="text_watermark" text-anchor="end" font-family="SimHei" font-style="Re
        x="100%" y="245px" fill="white">@Tencent</text>
</svg>
```
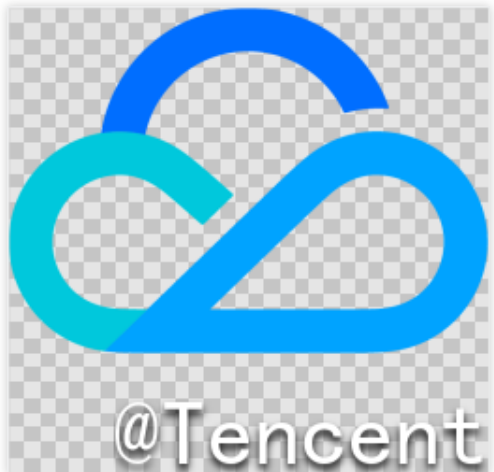
Result checking:

Method 1: copy the above code to the input field of Tryit Editor, and click **Run** at the top. The watermark appears on the right side:

Method 2: save the code as an HTML file and open it with a browser. The watermark (white background) in figure 2 below appears.

Figure 1: Transparent Background	Figure 2: White Background	F

Code explanation

Parameter	Description	Attribute
`<svg>`	Define the SVG canvas	`width="1000px" height="1000px"` : both height and are 1000 px. Just make sure that the canvas is large enough to include all the elements of the watermark.
`<filter id="filter">`	Define the filter to be used	-
`<feGaussianBlur>`	Set the Gaussian blur filter	`stdDeviation="2"`
`<feOffset>`	Set the filter offset	`dx="0" dy="3"` : offset the filter by 3 px **down**.
`<image id="img_watermark">`	Brand logo	`xlink:href="data:image/png;base64, {Base64-encoded image data}"` : reference a local image. `height="185px" width="256px"` : set the original aspect ratio.

		`x="74.4%" y="0px"` : as the logo is above the text, set the Y-axis offset to 0 px. Keep *adjusting* the X-axis offset until the expected result is achieved (`x="74.4%"`).
`<text id="text_watermark_shadow">`	Add a shadow to text	`font-family="SimHei" font-style="Regular" font-size="50px"` : set font and font size. `text-anchor="end"` : set the end of the text as its original position for alignment. `x="100%"` : set `text-anchor` and `x` to **pin** the last character of the text to the **right** of the canvas so as to adjust the position of the brand logo (`x` attribute of `<image id="img_watermark">`). `style="opacity:1;"` : set the opacity. `fill="black"` : set the shadow-fill-color to black. `filter="url(#filter)` : apply the filter whose id is filter. `y="245px"` : set the Y-axis offset. The height of the brand logo is 185 px, and the font size 50 px, which is basically equivalent to the font height, so `y` must not be smaller than `185+50=235` . After **testing** the value is set to 245.
`<text id="text_watermark">`	Text	Most of this parameter's attributes are the same as those of `<text id="text_watermark_shadow">` , but it does not use a filter（without the `filter` attribute), and the text-fill-color is white (`fill="white"`).

When checking the final result, you are advised to change the length of the text, for example, by replacing `@Tencent` with `@Tencent is a kind person` to see if the result still meets the requirements.

To use a different user ID, just replace the value of `<text>` in the SVG code.

Step 2. Create an SVG watermark template.

```
https://vod.tencentcloudapi.com/?Action=CreateWatermarkTemplate
&Type=svg
&Name=Test
&CoordinateOrigin=TopRight
&XPos=2%
&YPos=5%
```

```
&SvgTemplate.Width=30S%
&SvgTemplate.Height=0px
&<Common request parameters>
```

Parameter description:

Parameter	Description
Type=svg	Set the watermark type to SVG.
Name=Test	Set the name of the template (optional).
CoordinateOrigin=TopRight	Pin the watermark to the top right of the video.
XPos=2%	Set the distance between the watermark origin and the right border of videos to 2% of video width.
YPos=5%	Set the distance between the watermark origin and the top border of videos to 5% of video height.
SvgTemplate.Width=30S%	Set the watermark width to 30% of video width.
SvgTemplate.Height=0px	The height of the watermark scales proportionally.

Suppose the ID of the SVG watermark template created is 12345.

Step 3. Add the SVG watermark to a video.

Initiates a video processing task:

```
https://vod.tencentcloudapi.com/?Action=ProcessMedia
&FileId=5285485487985271487
&MediaProcessTask.TranscodeTaskSet.0.Definition=30
&MediaProcessTask.TranscodeTaskSet.0.WatermarkSet.0.Definition=12345
&MediaProcessTask.TranscodeTaskSet.0.WatermarkSet.0.SvgContent={the XML code in ste
&<Common request parameters>
```

Below is an example of the result

Appendix

Supported fonts

SimHei:style=Regular

Roboto:style=Bold

Contact us if you want other fonts supported.

Distribution and Playback
How to Use Key Hotlink Protection

Last updated：2022-12-16 14:16:08

Overview

Demo features

This demo describes how to use the key hotlink protection mechanism of VOD, including enabling key hotlink protection in the console, building a hotlink protection signature distribution service, and playing back a video with a hotlink protection signature.

Architecture and process

An HTTP service is built based on SCF in the demo to receive the requests for getting hotlink protection signatures from clients. It gets the original URL of a video in VOD from the request body, calculates a signature, and returns the URL with the signature to the client.

The system mainly involves four components: developer (you), API Gateway, SCF, and VOD. Here, API Gateway and SCF are the deployment objects of this demo as shown below:
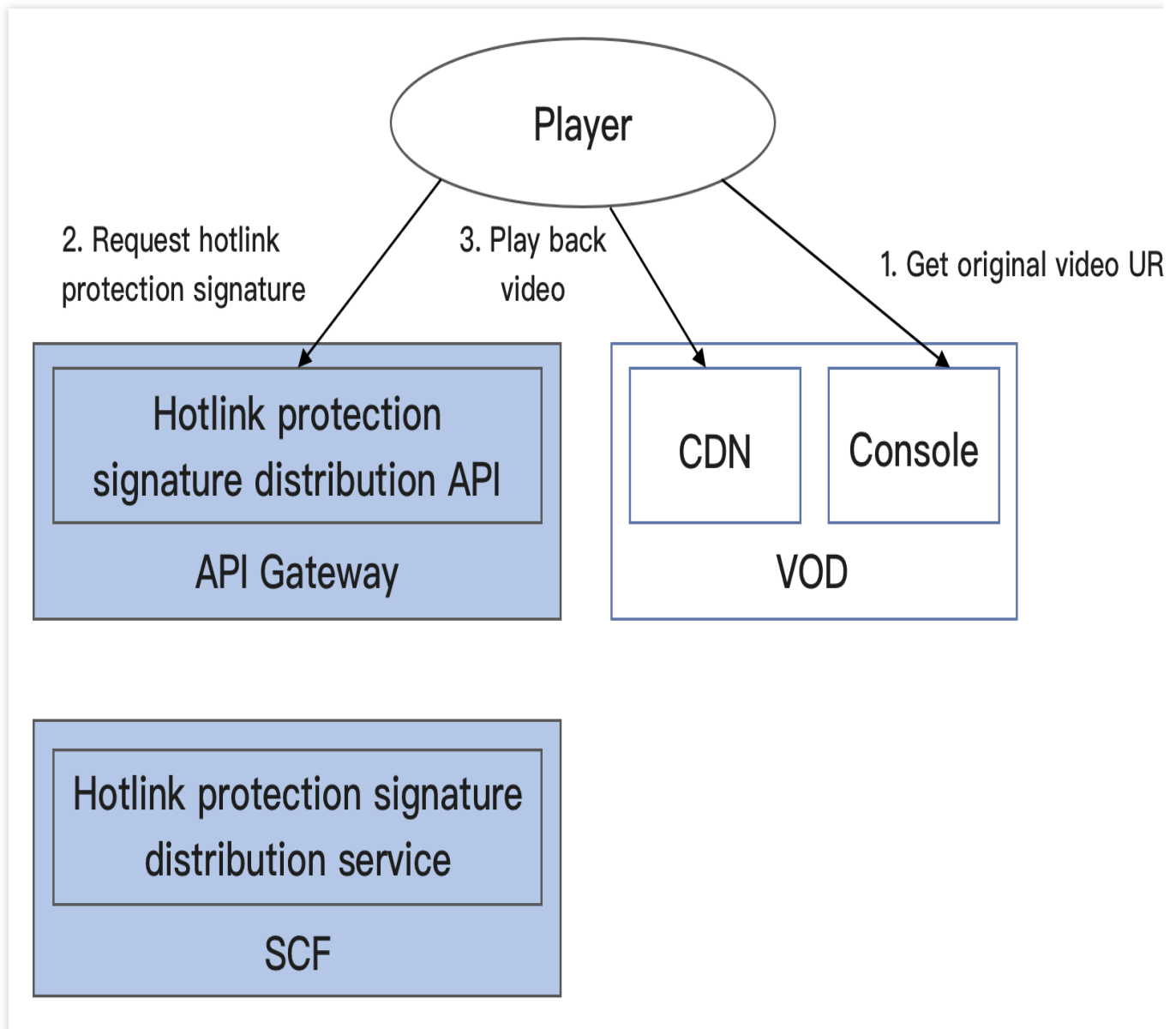
The specific business process is as follows:

1. Get the original URL of the video in the VOD console (in the actual production environment, it should be the player that requests the video URL from the business backend; to simplify the process, this document takes your operation as an example to simulate such business behavior here).

2. Use the original URL of the video to request a hotlink protection signature from SCF.

3. Use the video URL with the hotlink protection signature to request VOD CDN to play back the video.

Note:

The SCF code in the demo is developed based on Python 3.6. SCF also supports other programming languages such as Python 2.7, Node.js, Go, PHP, and Java for your choice as needed. For more information, please see Development Guide.

Fees

The VOD key hotlink protection signature distribution service demo provided in this document is open-source and free of charge, but it may incur the following fees during service building and use:

Fees for purchasing a Tencent Cloud CVM instance to run the service deployment script. For more information, please see Instance Billing Modes.

Fees for using signature distribution service provided by SCF. For more information, please see Billing Mode and Free Tier.

Fees for using Tencent Cloud API Gateway to provide public network APIs for SCF. For more information, please see Billing Overview.

Fees for VOD storage of uploaded videos.

VOD storage space will be taken up by uploaded videos. For more information, please see Video Storage Pricing.

Quickly Deploying Key Hotlink Protection Signature Distribution Service

Step 1. Prepare a CVM instance

The deployment script needs to be executed on a CVM instance meeting the following requirements:

Region: not limited.

Model: the minimum official configuration (1 CPU core and 1 GB memory) is sufficient.

Public network: a public IP is required, and the bandwidth should be at least 1 Mbps.

Operating system: official public image `Ubuntu Server 16.04.1 LTS 64-bit` or `Ubuntu Server 18.04.1 LTS 64-bit` .

For detailed directions on how to purchase a CVM instance and reinstall the system, please see Operation Guide - Creating Instances via CVM Purchase Page and Operation Guide - Reinstalling System, respectively.

Note:

The key hotlink protection signature distribution service demo itself does not depend on CVM but only uses CVM to run the deployment script.

If you do not have a CVM instance satisfying the above conditions, you can also run the script on another Linux (such as CentOS or Debian) or macOS server with public network access, but you need to modify certain commands in the deployment script based on the operating system. Please search for the specific modification method by yourself.

Step 2. Activate VOD and configure key hotlink protection

1. Activate the VOD service as instructed in Getting Started - Step 1.

2. After activation, enable key hotlink protection as instructed in Setting Hotlink Protection and record the hotlink protection key:
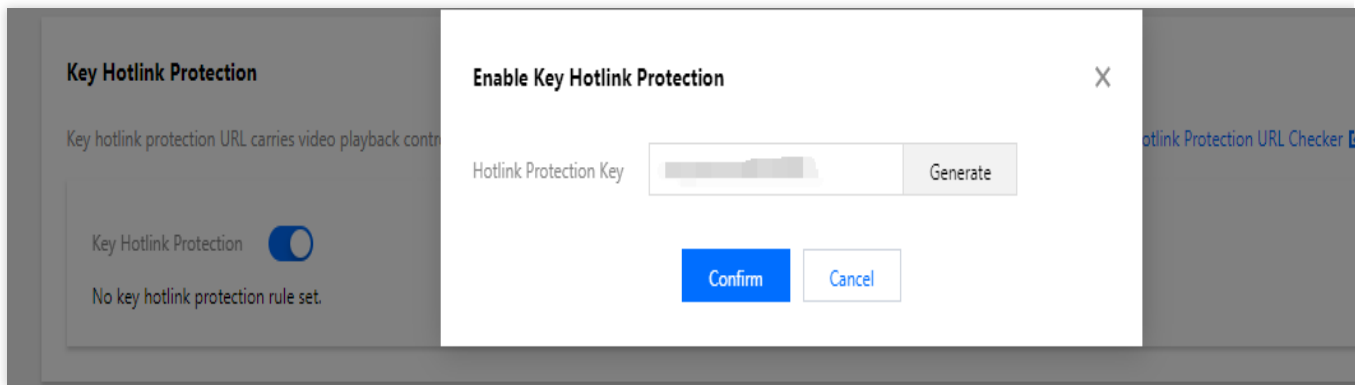
Note:

Here, enable key hotlink protection instead of referer hotlink protection. If you enable referer hotlink protection at the same time, the request may fail as the test method below does not meet the corresponding requirements.

Step 3. Get the API key and APPID

Your API key (i.e., `SecretId` and `SecretKey`) and `APPID` are required for deploying and running the key hotlink protection signature distribution service demo.

If you have not created an API key yet, please generate one as instructed in Root Account Access Key. If you have already created a key, please get it as instructed in the same document.

You can view the `APPID` on the Account Information page in the console as shown below:
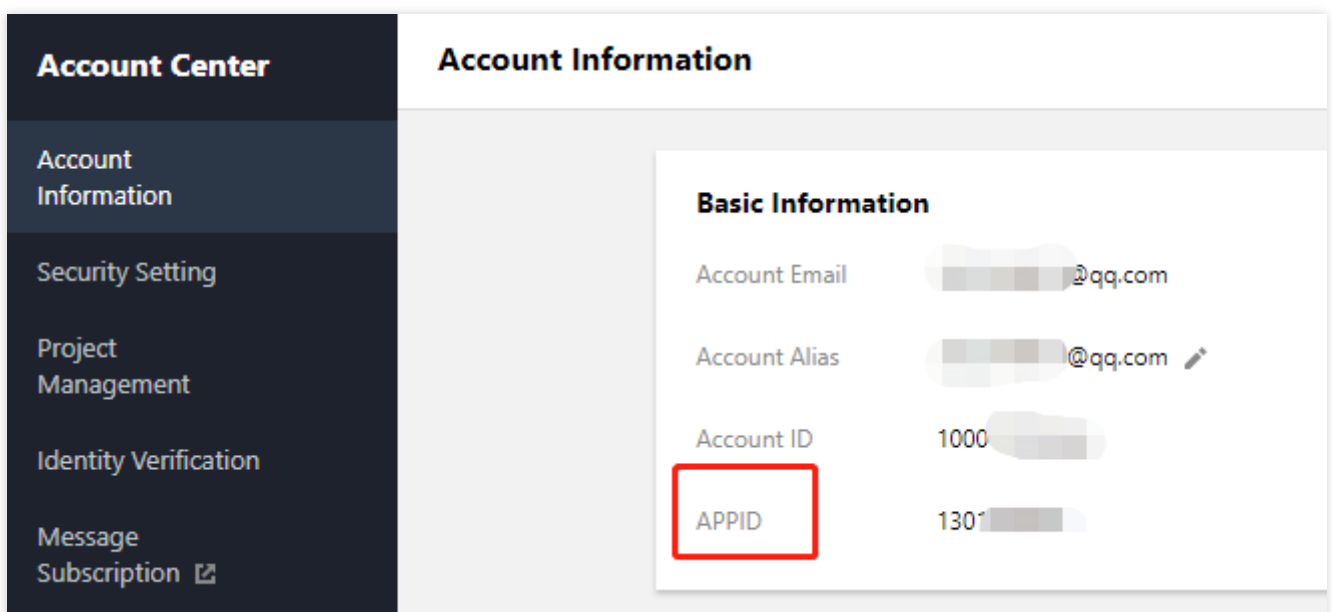
Step 4. Deploy the hotlink protection signature distribution service

Log in to the CVM instance prepared in step 1 as instructed in Logging into Linux Instance in Standard Login Method and enter and run the following command on the remote terminal:

```
ubuntu@VM-69-2-ubuntu:~$ export SECRET_ID=AKxxxxxxxxxxxxxxxxxxxxxxxx; export
SECRET_KEY=xxxxxxxxxxxxxxxxxxxxxx;export APPID=125xxxxxxx;export
ANTI_LEECH_KEY=xxxx;git clone https://github.com/tencentyun/vod-server-demo.git
~/vod-server-demo; bash ~/vod-server-demo/installer/ anti_leech_sign_scf_en.sh
```

Note:

Please assign the corresponding values obtained in step 3 to `SECRET_ID` , `SECRET_KEY` , and `APPID` in the command and assign the hotlink protection key obtained in step 2 to `ANTI_LEECH_KEY` .

This command will download the demo source code from GitHub and automatically run the installation script. The installation process will take several minutes (subject to the CVM network conditions), during which the remote terminal will print the following information:

```
[2020-06-04 15:57:10] Start checking npm.
[2020-06-04 15:57:18] npm is successfully installed.
[2020-06-04 15:57:18] Start installing ServerLess.
[2020-06-04 15:57:19] ServerLess is successfully installed.
[2020-06-04 15:57:20] Start deploying the VOD key hotlink protection signature
distribution service.
[2020-06-04 15:57:30] The deployment of the VOD key hotlink protection
signature distribution service is completed.
[2020-06-04 15:57:32] Service address: https://service-xxxxxxxx-
125xxxxxxx.gz.apigw.tencentcs.com/release/anti_leech_sign
```

Copy the address of the signature distribution service in the output log (which is `https://service-xxxxxxxx-125xxxxxxx.gz.apigw.tencentcs.com/release/anti_leech_sign` in this example).

Note:

If the following warning is displayed in the output log, it is generally because the CVM instance cannot immediately parse the service domain name deployed just now. You can ignore this warning.

```
[2020-04-25 17:18:44] Warning: the key hotlink protection signature
distribution service failed the test.
```

Step 5. Test key hotlink protection

Upload a test video to VOD as instructed in Uploading Video - Local Upload. After the video is uploaded, click **Quick View** and click **Copy URL** on the right to copy the video URL.

On the command line on the CVM instance, run the `curl` command to try directly accessing this URL. The access will be rejected by the server for non-compliance with the key hotlink protection rule, and the HTTP return code will be 403 (during the test, please replace the URL in the command with the actual URL, which also applies below):

```
ubuntu@VM-69-2-ubuntu:~$ curl -I
"http://125xxxxxxx.vod2.myqcloud.com/f888c998vodcq125xxxxxxx/c849148f528xxxxxxx
xxxxxxxxxx/xxxxxxxxxx.mp4"
HTTP/1.1 403 Forbidden
Server: NWS_VP
Connection: keep-alive
Date: Thu, 04 Jun 2020 08:27:54 GMT
Content-Type: text/plain
Content-Length: 14
```

On the command line on the CVM instance, run the `curl` command to request the URL with the hotlink protection signature from the service deployed in step 4 (`-d` means to initiate the request in POST method, and the carried parameter is the video URL):

```
ubuntu@VM-69-2-ubuntu:~$ curl -d
'http://125xxxxxxx.vod2.myqcloud.com/f888c998vodcq125xxxxxxx/c849148f528xxxxxxx
xxxxxxxxxx/xxxxxxxxxx.mp4' https://service-xxxxxxxx-
125xxxxxxx.gz.apigw.tencentcs.com/release/anti_leech_sign; echo
http://125xxxxxxx.vod2.myqcloud.com/f888c998vodcq125xxxxxxx/c849148f528xxxxxxx
xxxxxxxx/xxxxxxxxxx.mp4?
t=5ed8b8d2&exper=0&rlimit=0&us=455041&sign=fe6394007c2e7aef39fc70a02e897f69
```

Run the `curl` command again to access the URL with the hotlink protection signature obtained in the previous step, and the URL can be accessed normally (the HTTP return code will be 200):

```
ubuntu@VM-69-2-ubuntu:~$ curl -I
"http://125xxxxxxx.vod2.myqcloud.com/f888c998vodcq125xxxxxxx/c849148f528xxxxxxx
xxxxxxxxx/xxxxxxxxxx.mp4?
t=5ed8b8d2&exper=0&rlimit=0&us=455041&sign=fe6394007c2e7aef39fc70a02e897f69"
HTTP/1.1 200 OK
Server: tencent-cos
Connection: keep-alive
Date: Thu, 04 Jun 2020 08:37:17 GMT
Last-Modified: Fri, 22 May 2020 15:06:15 GMT
Content-Type: video/mp4
Content-Length: 232952632
Accept-Ranges: bytes
ETag: "1da6be3a0d1da5edae4ff0b1feff02cf-223"
x-cos-hash-crc64ecma: 16209801220610226954
x-cos-request-id: NWVkOGIyYmVfZDUyMzYyNjRfYWMwMF85YjkyNzA=
X-Daa-Tunnel: hop_count=4
X-NWS-LOG-UUID: b404f43e-3c86-4c54-8a78-fb78e4e85cf2
add71e19fb08c6d9dbe1b21a2fb157bf
Access-Control-Allow-Credentials: true
Access-Control-Allow-Headers: Origin,No-Cache,X-Requested-With,If-Modified-
Since,Pragma,Last-Modified,Cache-Control,Expires,Content-
Type,X_Requested_With,Range
Access-Control-Allow-Methods: GET,POST,OPTIONS
Access-Control-Allow-Origin: *
```

Note:

You can access the URL with the hotlink protection signature in a browser and verify the signature by playing back the video. However, this method has certain requirements for the video format. Generally, H.264-encoded .mp4 videos have high compatibility, which are recommended. You can also use a third-party tool such as Postman to send HTTP requests for test. Please search for the specific usage by yourself.

System Design Description

API protocol

The key hotlink protection signature distribution function uses API Gateway to provide APIs. The specific API protocol is as detailed below:

Service	Function Name	API Form	Request Content	Response Content
Key hotlink protection	anti_leech_sign	HTTP POST	Original	URL with hotlink protection

signature distribution			video URL	signature

Signature distribution service code interpretation

1. `main_handler()` is the entry function.

2. Call `parse_conf_file()` and read the configuration information from the `config.json` file. The configuration items are as described below (for specific parameters, please see Key Hotlink Protection):

Field	Data Type	Description
key	String	Key hotlink protection key
t	Integer	Signature validity period in seconds. When a request is being processed, this parameter plus the current time on the SCF server will be `t` in the hotlink protection parameters
exper	Integer	Preview duration
rlimit	Integer	Maximum number of client IPs that can access the signature

3. Parse the `Dir` parameter from the request body, generate the `t` and `us` parameters locally, and read the `exper` and `rlimit` parameters from the configuration file:

```
original_url = event["body"]
parse_result = urlparse(original_url)
directory = path.split(parse_result.path)[0] + '/'
# Signature parameters
timestamp = int(time.time())
rand = random.randint(0, 999999)
sign_para = {
    "t": hex(timestamp + configuration['t'])[2:],
    "exper": configuration['exper'],
    "rlimit": configuration['rlimit'],
    "us": rand
}
```

4. Call `generate_sign()` to calculate the hotlink protection signature. For the specific algorithm, please see Key Hotlink Protection.

5. Generate the `QueryString` and add it at the end of the original URL to concatenate a URL with the hotlink protection signature:

```
sign_para["sign"] = signature
query_string = urlencode(sign_para)
new_parse_result = parse_result._replace(query=query_string)
signed_url = urlunparse(new_parse_result)
```

6. Return the signature. For the formats and descriptions of the returned data, please see Overview of API Gateway Trigger.

```
return {
    "isBase64Encoded": False,
    "statusCode": 200,
    "headers": {"Content-Type": "text/plain; charset=utf-8",
                "Access-Control-Allow-Origin": "*",
                "Access-Control-Allow-Methods": "POST,OPTIONS"},
    "body": signed_url
}
```

How to Prevent Malicious Video Hosting

Last updated：2022-03-24 15:32:21

User-generated content (UGC) and professionally generated Content (PGC) platforms are two common scenarios in the video industry where video content can be freely uploaded and shared.

However, third-party video platforms may impersonate normal users to upload videos to your platform and then put the URLs of those videos on their own platforms. In this way, they can "live" on your platform like a parasite and get "free" access to video storage and playback acceleration. As a result, your video platform is used maliciously by others as a free video hosting service, which is called "malicious video hosting".

Malicious video hosting can cause serious economic losses since all the storage, bandwidth, and traffic fees incurred by the parasites have to be borne by you.

Causes of Malicious Video Hosting

How UGC/PGC platforms interact with others normally
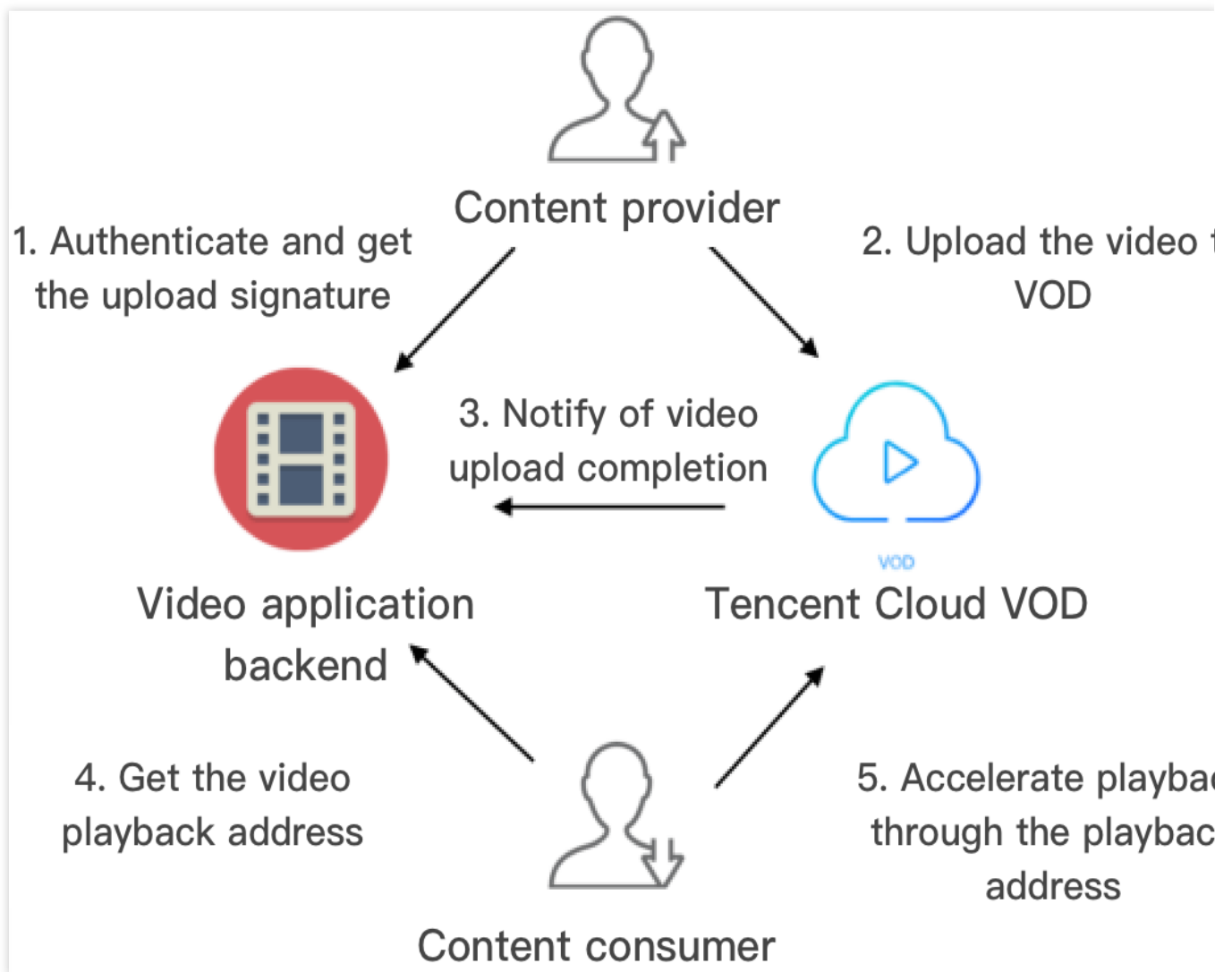
Note:

Arrows in the figure represent the network request directions.

Generally, a UGC or PGC video platform interacts with content providers, content consumers, and VOD in the following ways (for more information on steps 1–3, please see Upload from Client):

1. The application backend authenticates the content provider and distribute to them a video upload signature after authentication is passed.

2. The content provider uploads the content to be shared to VOD.

3. VOD notifies the application backend of relevant information such as the `fileId` and playback URL of the uploaded video.

4. The content consumer requests the playback URL from the application backend.

5. The content consumer accelerates video playback via VOD through the URL.

How malicious video hosting is implemented
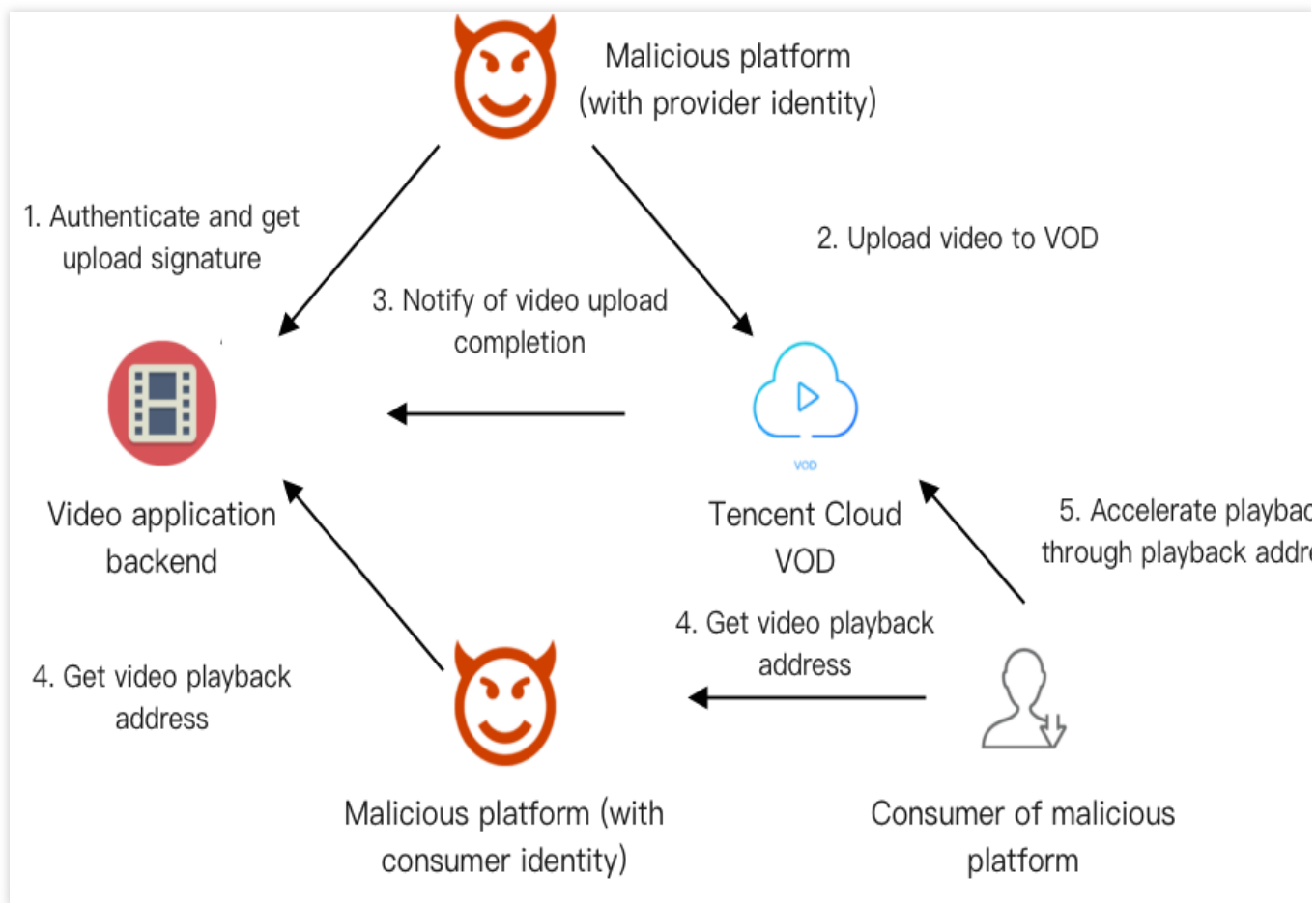
Note:

Arrows in the figure represent the network request directions.

A malicious third-party video platform impersonates a normal user of your platform:

It upload its own videos to VOD as a video provider (steps 1 and 2).

Then, it gets the playback URLs of these videos from your platform as a consumer (step 4).

The users of the malicious platform get these URLs (step 4) and accelerate playback through VOD (step 5).

Core causes of problem

The ultimate purpose of malicious video hosting is to steal others' CDN bandwidth resources (while taking advantage of their storage resources). Malicious users do so mainly for the following:

Step 4. The malicious platform can quickly get the playback URLs from the application, store them, and distribute them to its own users.

Step 5. After the users of the malicious platform get the URLs, they can accelerate video playback unlimitedly.

Malicious Video Hosting Prevention

In view of the core causes of malicious video hosting listed above, the key solutions lie in:

Preventing **unrestricted access to playback URLs** as mentioned in step 4.

Preventing **unrestricted playback acceleration** as mentioned in step 5.

The following describes how VOD helps you restrict **playback** of and **access** to URLs.

Restricting playback of URLs

VOD's key hotlink protection provides the ability to limit the number of devices allowed to play back a video from a URL, so as to prevent the URL from being distributed to any number of devices for playback.
In order to implement effective control of the playback URL, you need to enable hotlink protection in the console; in step 4, a hotlink protection-enabled URL needs to be generated on the application backend according to the key hotlink protection URL generation rules (please see the example of "maximum number of IPs allowed for playback at a video playback address") in order to limit the validity period of the URL and the number of IPs allowed for playback.

Restricting access to URLs

Restricting playback of URLs alone cannot effectively prevention malicious video hosting. This is because in step 4, a malicious platform can make countless requests for different hotlink protection-enabled URLs for the same video and then distribute those URLs to its own users, thus bypassing the restriction on the number of IPs allowed for playback. In response, the application backend needs to verify user identity in step 4 and impose frequency control, i.e., how many times an individual user can get the same playback URL within a specified period of time. This can prevent malicious users from getting a large number of video playback addresses in a short period of time.

How to Receive Event Notification

Last updated：2023-05-15 17:39:11

Overview

About this document

In this document, we will run you through the process of uploading and transcoding a video and show you how to use VOD's event notifications.

Architecture and workflow

In the demo, an HTTP server was built based on SCF to receive event notification requests from VOD. The server initiates transcoding tasks and gets the transcoding results based on the NewFileUpload and ProcedureStateChanged notifications it receives.

The system consists of four main components: the console, API Gateway, SCF, and VOD. API Gateway and SCF are the deployment objects of the demo.
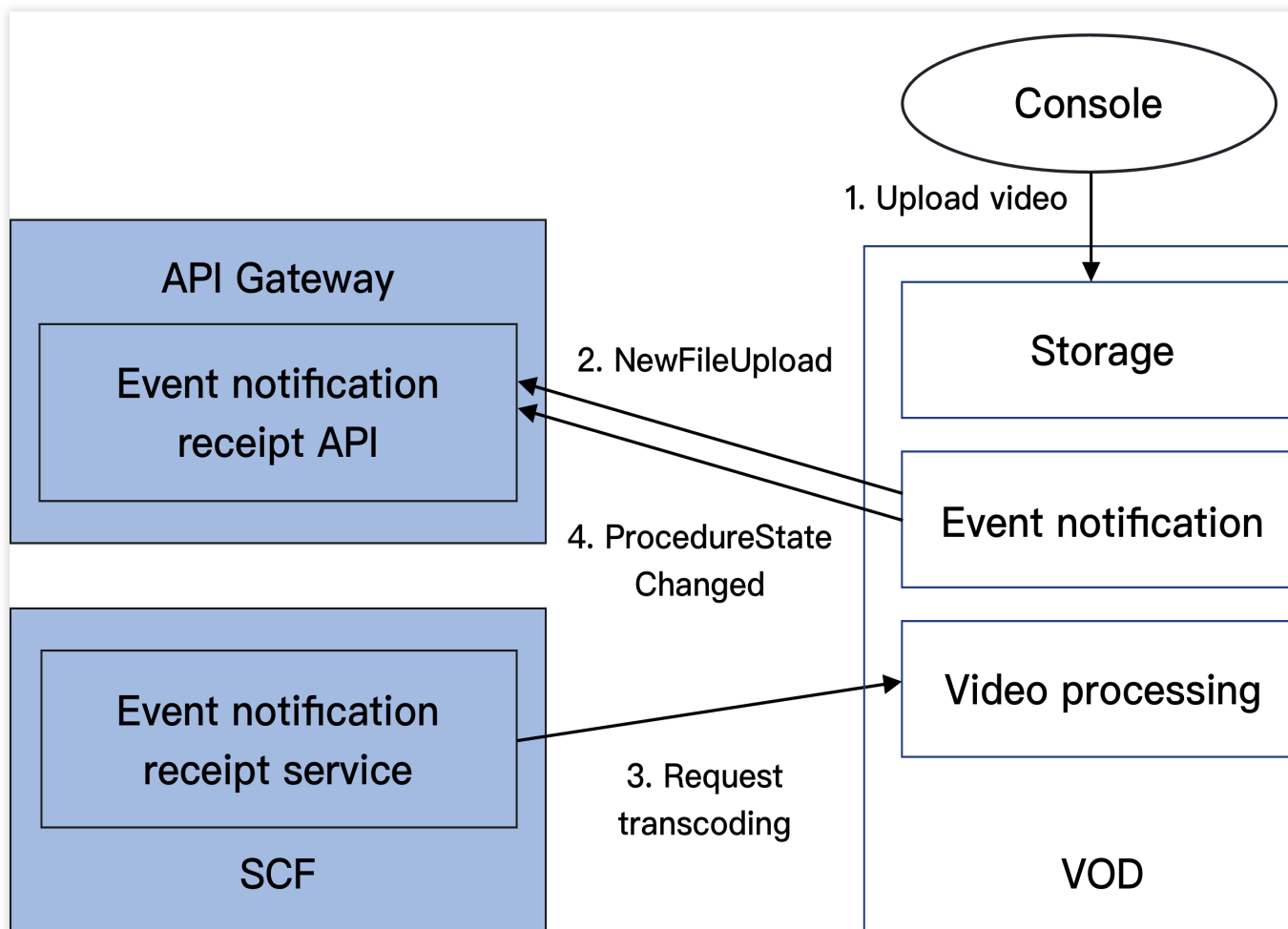
The workflow is as follows:

1. Upload a video to the VOD console.

2. VOD sends a `NewFileUpload` notification request to the demo.

3. The demo parses the notification and calls VOD's ProcessMedia API to transcode the uploaded video using the preset transcoding templates `100010` and `100020` .

4. After completing the transcoding task, VOD sends a `ProcedureStateChanged` notification request to the demo.

5. The demo parses the notification and prints the URL of the transcoding output in SCF logs.

Note:

The SCF code in the demo is developed based on Python 3.6. SCF also supports other programming languages such as Python 2.7, Node.js, Go, PHP, and Java. For more information, see Code Development.

Fees

The demo for receiving VOD event notifications provided in this document is open-source and free of charge, but to build the demo, you may incur the following fees:

Fees for purchasing a Tencent Cloud CVM instance to execute the script. For more information, see CVM Billing Mode.

Fees for using SCF to distribute signatures. For more information, see Pay-As-You-Go and Free Tier.

Fees for using Tencent Cloud API Gateway to provide public network APIs for SCF. For more information, see Shared Instance Billing.

Fees for storing videos uploaded to VOD. For details, see Pay-As-You-Go and Prepaid Packages.

Fees for transcoding videos stored in VOD. For details, see Pay-As-You-Go and Prepaid Packages.

Impact on your production environment

The demo uses VOD's event notification mechanism, for which you need to configure an event notification address. If there is already a VOD-based production environment under your account, changing the event notification address may affect your active business. Therefore, **before doing so, make sure it will not affect your production environment. If you are not sure, please use a new account to deploy the demo.**

Quickly Deploying Event Notification Receipt Service

Step 1. Prepare a CVM instance

The deployment script needs to be executed on a CVM instance that mees the following requirements:
Region: No limit.
Model: The minimum specification (1 CPU core and 1 GB memory) or higher.
Public network: A public IP address is required, and the bandwidth should be at least 1 Mbps.
Operating system: Official public image `Ubuntu Server 16.04.1 LTS 64-bit` or `Ubuntu Server 18.04.1 LTS 64-bit` .
For detailed directions on how to purchase a CVM instance and reinstall the system, see Operation Guide - Creating Instances via CVM Purchase Page and Operation Guide - Reinstalling System.
Note:
The demo itself does not depend on CVM, but it needs a CVM instance to run the deployment script.
If you do not have a CVM instance that meets the above conditions, you can also run the script on other Linux (such as CentOS or Debian) or macOS servers with public network access, but you need to modify certain commands in the deployment script based on the operating system.

Step 2. Activate VOD

Activate VOD. For detailed directions, see Getting Started - Step 1. Activate VOD.

Step 3. Get the API key and APPID

Your API key (i.e., `SecretId` and `SecretKey`) and `APPID` are required for deploying and running the demo. If you have not created an API key yet, create one as instructed in Creating an API key for a root account. If you have already created a key, follow the steps in Viewing an API key of a root account to view the key.
You can view your APPID on the Account Information page of the console.
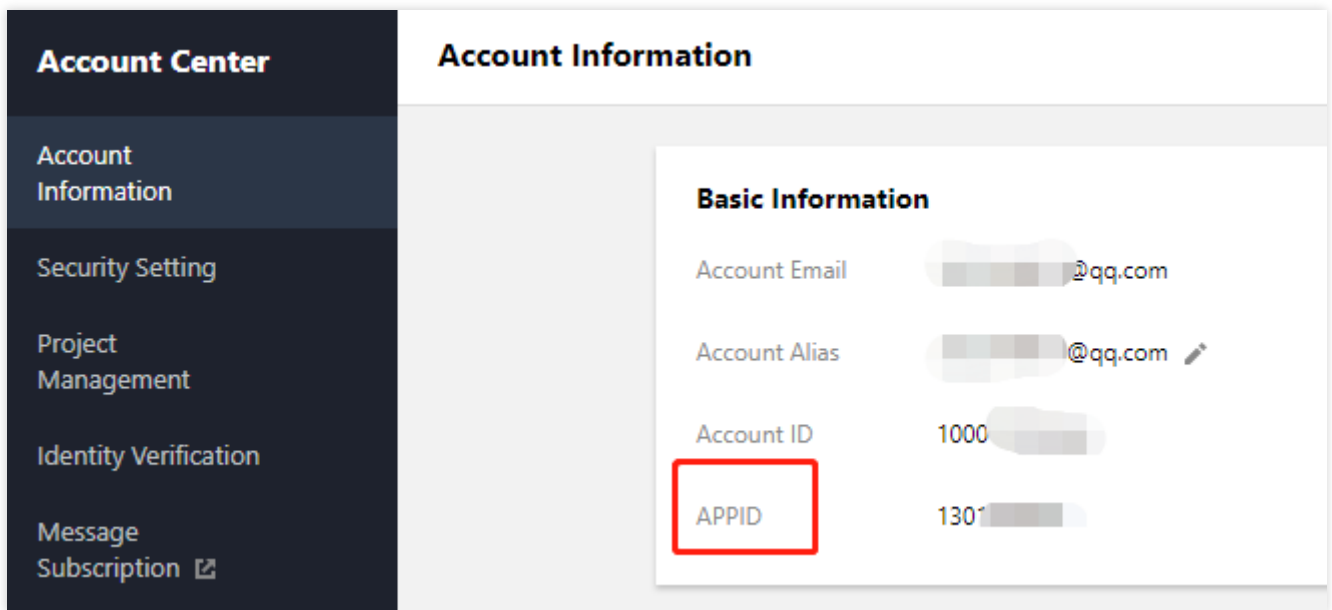
Step 4. Deploy the notification receipt service

Log in to the CVM instance prepared in Step 1 and run the following command on the remote terminal:

```
ubuntu@VM-69-2-ubuntu:~$ export SECRET_ID=AKxxxxxxxxxxxxxxxxxxxxxxxx; export SECRET_
```

Note:

You need to assign values to `SECRET_ID` , `SECRET_KEY` , and `APPID` according to the key and APPID obtained in Step 3.

This command will download the demo source code from GitHub and automatically run the installation script. The installation process will take several minutes (subject to the CVM network conditions), during which the remote device will print the following information:

```
[2020-06-05 17:16:08]Start checking npm.
[2020-06-05 17:16:12] npm is successfully installed.
[2020-06-05 17:16:12]Start installing serverless.
[2020-06-05 17:16:13]Serverless is successfully installed.
[2020-06-05 17:16:14] Start deploying the event notification receipt service.
[2020-06-05 17:16:24] The event notification receipt service is deployed.
[2020-06-05 17:16:26] Service address: https://service-xxxxxxxx-125xxxxxxx.gz.apigw
```

Copy the address of the event notification receipt service in the output log (which is `https://service-xxxxxxxx-125xxxxxxx.gz.apigw.tencentcs.com/release/callback` in this example).

Note:

If the following warning is printed in the output log, it is probably because the CVM instance cannot parse the domain deployed immediately. You can ignore this warning.

```
[2020-04-25 17:18:44] Warning: The event notification receipt service failed the te
```

Step 5. Configure the event notification address

Note:

As mentioned in Impact on your production environment, before you perform the following operations, please make sure that your active business does not depend on VOD event notifications.

1. Log in to the VOD console and select **Application Management** on the left sidebar.

2. Select the target application.

3. Click **Callback Settings** on the left sidebar and click **Set**. For **Event Notification Method**, select "Normal Callback", and enter the address obtained in Step 4. Select all event notification types, and click **Confirm**.

Note:

If you see two callback URL configurations (v2.0 and v3.0) in the console, configure v3.0.

Step 6. Test the demo

1. Upload a test video to VOD (select **No processing after upload**). After the video is uploaded, if the status of the video on the **Uploaded** page is "Processing", it indicates that the demo has received the `NewFileUpload` notification and initiated a transcoding task.

2. After the transcoding is finished (status becomes "Normal"), click **Quick View**, and you will see the two transcoding outputs for the file on the right.

3. Log in to the SCF console and go to the log page. In the latest log, you can see that the URLs of the two output files have been printed. You can use SCF to record the URLs in your database or send them to viewers.

Note:

There may be a delay in log generation. If you can't find the URLs, please wait for a minute or two and click **Reset** to refresh the page.

System Design

API protocol

The event notification receipt function uses API Gateway to provide APIs. For the API protocols, see Video Upload Completion and Task Flow Status Change.

Code interpretation

1. `main_handler()` is the entry function.

2. Call `parse_conf_file()` to read configuration information from the `config.json` file.

Field	Type	Description
secret_id	String	The API key.
secret_key	String	The API key.
region	String	The TencentCloud API request region, which can be any region.
definitions	Array of Integer	The transcoding template.
subappid	Integer	The VOD subapplication the event notification is from.

3. For `NewFileUpload` event notifications, call `deal_new_file_event()` to parse the request and get the file ID of the uploaded video.

```
if event_type == "NewFileUpload":
    fileid = deal_new_file_event(body)
    if fileid is None:
        return ERR_RETURN
```

4. Call `trans_media()` to initiate transcoding, output the API's response packets to SCF logs, and send the response packets to the event notification service of VOD.

```
rsp = trans_media(configuration, fileid)
if rsp is None:
    return ERR_RETURN
print(rsp)
```

5. In `trans_media()` , call the TencentCloud API `ProcessMedia` :

```
cred = credential.Credential(conf["secret_id"], conf["secret_key"])
client = vod_client.VodClient(cred, conf["region"])

method = getattr(models, API_NAME + "Request")
req = method()
req.from_json_string(json.dumps(params))

method = getattr(client, API_NAME)
rsp = method(req)
return rsp
```

6. For `ProcedureStateChanged` event notifications, call `deal_procedure_event()` to parse the request, get the URL of the transcoding output, and print it in SCF logs:

```
elif event_type == "ProcedureStateChanged":
    rsp = deal_procedure_event(body)
    if rsp is None:
        return ERR_RETURN
```

How to Migrate Files from Origin Server to VOD

Last updated：2022-08-03 10:37:15

Overview

VOD Migrate Tool is an all-in-one tool that integrates data migration features. By writing a simple configuration file, you can quickly migrate media files at the source address to VOD.

Supported Data Sources

Local folder
URL list
Tencent Cloud COS
AWS S3
Alibaba Cloud OSS
Qiniu Kodo

Operating Environments

System environment

Windows, Linux, and macOS

Software dependency

Python 2.7, 3.4 and above
Latest version of pip

Installation

Installing via pip (recommended)

You can install the SDK into your project through pip. If you haven't installed pip in your project environment yet, install it first as instructed at pip's official website.

```
pip install vodmigrate
```

Installing by using source package

You can click here to download the source code.

Download the latest code and decompress:

```
git clone https://github.com/tencentyun/vod-migrate.git
cd vod-migrate
python setup.py install
```

Example

Run the following command:

```
vodmigrate config.toml
```

Note:

After the migration is completed, the result will be output to the directory corresponding to the configuration item `migrateResultOutputPath` , and the filename will be `vod_migrate_result.txt` .

Configuration Files

The configuration file is in TOML format (for more information, please see config_template.toml. Make sure that the file is encoded in UTF-8). Configure the following file content:

1. Configure the migration type

`type` indicates the migration type, which is filled in by users based on their migration needs. For example, to migrate local data to VOD, users need to configure `type=migrateLocal` for `[migrateType]` .

```
[migrateType]
type="migrateLocal"
```

Currently, the following migration types are supported:

migrateType	Description
migrateLocal	From local system to VOD
migrateUrl	From download URL to VOD

migrateCos	From Tencent Cloud COS to VOD
migrateAws	From AWS S3 to VOD
migrateAli	From Alibaba Cloud OSS to VOD
migrateQiniu	From Qiniu Kodo to VOD

2. Configure the migration task

You can configure a migration task based on your actual needs, including information for the destination VOD and task-related configurations.

```
# Common configuration for the migration tool
[common]
secretId = "SECRETID"
secretKey = "SECRETKEY"
region = 'REGION'
subAppId = 0
concurrency = 5
supportMediaClassification = [ 'video', 'audio', 'image' ]
excludeMediaType = [  ]
migrateDbStoragePath = ''
migrateResultOutputPath = ''
```

Name	Description
secretId	`SecretId` of your key. Replace `SECRETID` with your real key information, which can be obtained on the TencentCloud API key page in the CAM console.
secretKey	`SecretKey` of your key. Replace `SECRETKEY` with your real key information, which can be obtained on the TencentCloud API key page in the CAM console.
region	Access point region, i.e., the region where to request a VOD server. This is different from the storage region. For more information, please see Region List.
subAppId	ID of a subapplication in VOD. If you need to migrate a file into a subapplication, enter the subapplication ID in this field; otherwise, leave it empty.
concurrency	Number of concurrently migrated files. Maximum value: 50
supportMediaClassification	List of media types supported for migration. Valid values: video, audio, image
excludeMediaType	List of file types to be excluded
migrateDbStoragePath	Save path of the migrated `db` . If this parameter is left empty, it means the

	current directory.
migrateResultOutputPath	Save path of the migration result (one migration record corresponds to one line of JSON string). If this parameter is left empty, it means the current directory.

File type description:

Video: MP4, TS, FLV, WMV, ASF, RM, RMVB, MPG, MPEG, 3GP, MOV, WEBM, MKV, and AVI. **HLS and DASH are not supported**.

Audio: MP3, M4A, FLAC, OGG, and WAV

Image: JPG, JPEG, PNG, GIF, BMP, TIFF, AI, CDR, and EPS

3. Configure the data source

Configure each section according to the migration type described in `[migrateType]` . For example, if the configuration item of `[migrateType]` is `type=migrateLocal` , you only need to configure the `[migrateLocal]` section.

3.1 Configure a local data source `migrateLocal`

If you migrate a local file to VOD, configure this section. The specific configuration items and descriptions are as follows:

```
# Configuration section for migration from a local system to VOD
[migrateLocal]
localPath = ''
excludes = [ ]
```

Configuration Item	Description
localPath	Local path, which should be an absolute path
excludes	Absolute path of the directory to be excluded, which means that some files in the directory at `localPath` are not to be migrated

3.2 Configure a URL list data source `migrateUrl`

If you migrate files from a specified URL list to VOD, configure this section. The specific configuration items and descriptions are as follows:

```
# Configuration section for migration from a URL list to VOD
[migrateUrl]
urllistPath=D:\\\\folder\\\\urllist.txt
```


Configuration Item	Description
urllistPath	Absolute path of the file storing the URL list. The file content is URL text containing one original URL address per line.

Note:

To migrate large local files to VOD, you're advised to use the pullUpload API.

3.3. Configure a COS data source (migrateCos)

If you migrate files from Tencent Cloud COS to VOD, configure this section. The specific configuration items and descriptions are as follows:

```
# Configuration section for migration from Tencent Cloud COS to VOD
[migrateCos]
region = 'ap-shanghai'
bucket = 'examplebucket-1250000000'
secretId = 'COS_SECRETID'
secretKey = 'COS_SECRETKEY'
prefix = ''
```

Configuration Item	Description
region	Region information of the source bucket. For more information, see Regions and Access Endpoints.
bucket	Name of the source bucket in the format of `<BucketName-APPID>` . The bucket name must include `APPID` , such as `examplebucket-1250000000` .
secretId	`SecretId` of the key of the account to which the source bucket belongs. You can view this parameter in API Keys.
secretKey	`SecretKey` of the key of the account to which the source bucket belongs. You can view this parameter in API Keys.
prefix	Prefix of the path to be migrated. If all data in the bucket needs to be migrated, leave the prefix empty.

3.4 Configure an AWS data source `migrateAws`

If you migrate files from AWS to VOD, configure this section. The specific configuration items and descriptions are as follows:

```
# Configuration section for migration from AWS S3 to VOD
```

```
[migrateAws]
region = 'ap-northeast-2'
bucket = 'bucket-aws'
accessKeyId = 'AccessKeyId'
accessKeySecret = 'AccessKeySecret'
prefix = ''
```

Configuration Item	Description
region	AWS S3 region
bucket	AWS S3 bucket name
accessKeyId	Replace `AccessKeyId` with your real key information
accessKeySecret	Replace `AccessKeySecret` with your real key information
prefix	Prefix of the path to be migrated. If all data in the bucket needs to be migrated, leave the prefix empty

3.5 Configure an Alibaba Cloud OSS data source `migrateAli`

If you migrate files from Alibaba Cloud OSS to VOD, configure this section. The specific configuration items and descriptions are as follows:

```
# Configuration section for migration from Alibaba Cloud OSS to VOD
[migrateAli]
bucket = 'bucket-aliyun'
accessKeyId = 'yourAccessKeyId'
accessKeySecret = 'yourAccessKeySecret'
endPoint = 'oss-cn-hangzhou.aliyuncs.com'
prefix = ''
```

Configuration Item	Description
bucket	Alibaba Cloud OSS bucket name
accessKeyId	Replace `yourAccessKeyId` with your real key information
accessKeySecret	Replace `yourAccessKeySecret` with your real key information
endPoint	Alibaba Cloud endpoint address
prefix	Prefix of the path to be migrated. If all data in the bucket needs to be migrated, leave the prefix empty

3.6 Configure a Qiniu data source `migrateQiniu`

If you migrate from Qiniu to VOD, configure this section. The specific configuration items and descriptions are as follows:

```
# Configuration section for migration from Qiniu Kodo to VOD
[migrateQiniu]
bucket = 'bucket-qiniu'
accessKeyId = 'AccessKey'
accessKeySecret = 'SecretKey'
endPoint = 'www.bkt.clouddn.com'
prefix = ''
```

Configuration Item	Description
bucket	Qiniu Kodo bucket name
accessKeyId	Replace `AccessKey` with your real key information
accessKeySecret	Replace `SecretKey` with your real key information
endPoint	Download URL of Qiniu Kodo, which corresponds to `downloadDomain`
prefix	Prefix of the path to be migrated. If all data in the bucket needs to be migrated, leave the prefix empty

Restrictions

The tool is designed as a one-time migration tool. The migration has three stages: **origin server file scanning**, **migrating**, and **migration completed**. After the file scan is completed, if the configuration needs to be changed, the `db` file must be cleared (i.e., deleting `migrate.db` or modifying the `db` storage path) to avoid errors with configuration file MD5 verification.
The migrated files must be displayed with the file extension.
HLS/DASH files cannot be migrated currently.
After the migration, the directory relationship between the original videos cannot be maintained, and each video has an independent `FileId`, all of which are not related to each other.

Migration Process

1. The configuration file is read, the corresponding configuration section is read according to the migration type, and parameters are checked.

2. The origin server is scanned according to the migration type, and migration tasks are generated.

3. After the scan is completed, the migration is performed, and the results of each task and the overall progress are printed.

4. After the migration is completed, the details are output to the result file.

```
[2020-06-23 11:03:15,077] build tasks
[2020-06-23 11:03:16,611] add migrate task: E:\cloud\vod_migrate\media\fancybg.jpg
[2020-06-23 11:03:16,613] add migrate task: E:\cloud\vod_migrate\media\Wildlife - 副本 (2).mp4
[2020-06-23 11:03:16,616] add migrate task: E:\cloud\vod_migrate\media\Wildlife - 副本.mp4
[2020-06-23 11:03:16,625] add migrate task: E:\cloud\vod_migrate\media\Wildlife.mp4
[2020-06-23 11:03:16,626] add migrate task: E:\cloud\vod_migrate\media\中文.jpg
[2020-06-23 11:03:16,627] add migrate task: E:\cloud\vod_migrate\media\测试.png
[2020-06-23 11:03:17,772] 当前迁移进度：16.67%, 总量：6, 成功数：1, 失败数：0
[2020-06-23 11:03:17,852] 当前迁移进度：33.33%, 总量：6, 成功数：2, 失败数：0
[2020-06-23 11:03:17,957] 当前迁移进度：50.00%, 总量：6, 成功数：3, 失败数：0
[2020-06-23 11:03:18,068] 当前迁移进度：66.67%, 总量：6, 成功数：4, 失败数：0
[2020-06-23 11:03:18,218] 当前迁移进度：83.33%, 总量：6, 成功数：5, 失败数：0
[2020-06-23 11:03:18,292] 当前迁移进度：100.00%, 总量：6, 成功数：6, 失败数：0
[2020-06-23 11:03:19,295] tasks finished
[2020-06-23 11:03:19,454] migrate result: E:\cloud\vod_migrate\vod_migrate_result.txt
```

Live Recording
Recording to VOD and Processing Video

Last updated：2023-05-15 17:37:30

Live recording is a solution that records and saves live streams to VOD after remuxing (without modifying audio/video data, timestamp etc.). The recording files can be further processed and distributed. For details, see Live Recording.

Strengths

Based on the capabilities of CSS, the live recording feature can quickly record and store live streaming content to VOD for secondary production and distribution.

Relying on Tencent Cloud's leading AI technologies in audio and video and globally deployed cache nodes, VOD provides top-notch audio/video services including stream publishing, transcoding, distribution, and playback, delivering ultra-low latency and ultra-high image quality while being capable of handling large numbers of concurrent requests.

With the live recording feature, you can easily spread videos of your live events in various scenarios and to different applications.

The feature is useful in cases such as corporate live streaming, live shopping, and online classes. It allows you to distribute your content through different channels.

Prerequisites

You have signed up for a Tencent Cloud account and logged in.
You have activated CSS and VOD.

Directions

Step 1. Create a recording template

To use the live recording feature, you need to create a recording template first. The configuration of live recording is saved in the recording template. You can create recording templates with different configurations to record files into different formats or different durations.

Creating a template in the console:

1.1 Log in to the CSS console and select **Feature Configuration** > **Live Recording**.

1.2 Click **Create Recording Template** and select at least one recording format. For detailed directions, see Creating a Recording Template.

1.3 Click **Save**.

Creating a template using an API:

You can also call the CreateLiveRecordTemplate API to create a recording template. The template ID will be returned after the template is created successfully.

Step 2. Select a recording scheme

CSS offers the following recording schemes for different scenarios:

Scheme 1. Global recording for a domain name

You can bind your live recording template to a push domain in the CSS console or by calling an API, and streams pushed through the domain will be recorded automatically.

Use cases: Showroom streaming, live shopping, online classroom, and video surveillance.

Steps:

1.1 After creating a recording template, you will be prompted to bind a domain name. Click **Bind Domain Name** and select a push domain.
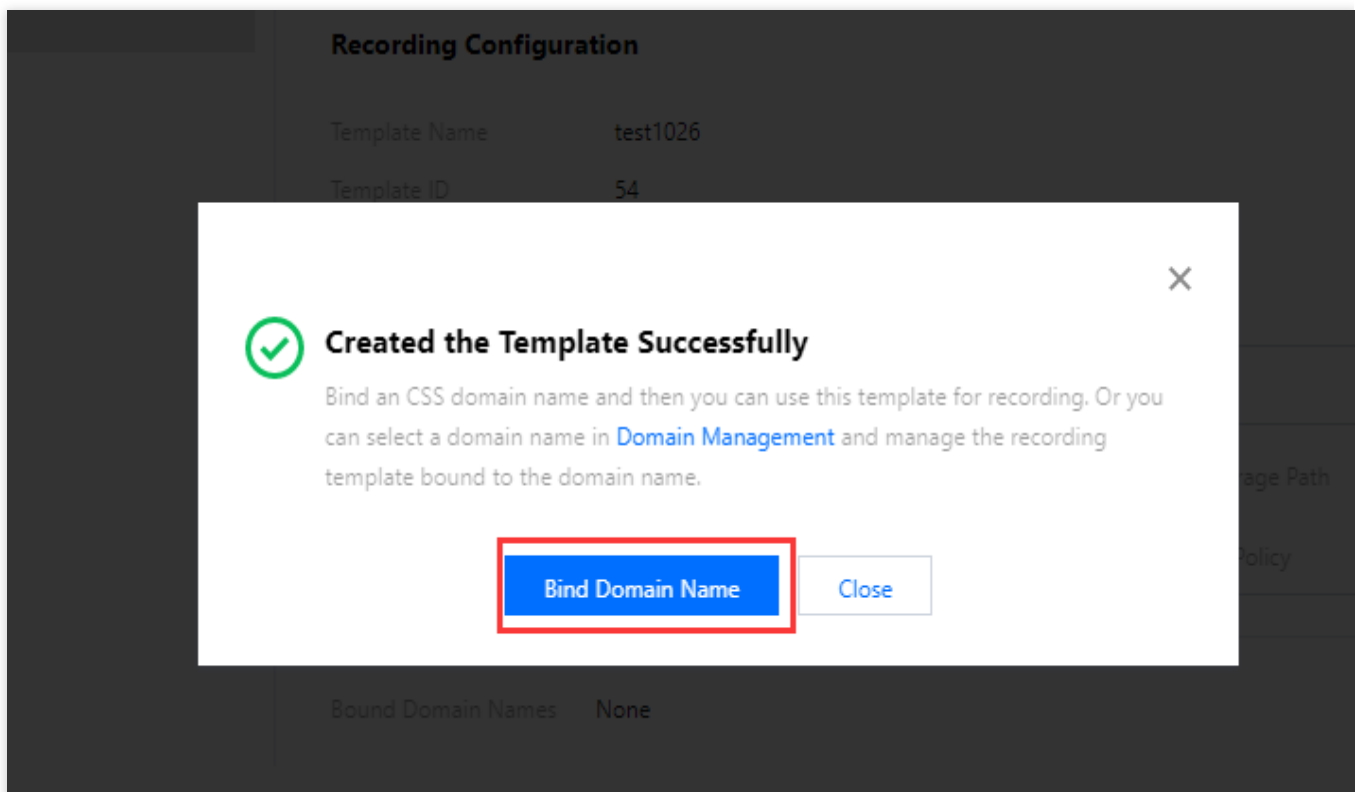
1.2 You can also bind a push domain on the **Domain Management**. Click your push domain. On the domain details page, select **Template Configuration**. In the **Recording Configuration** area, click **Edit** to bind the push domain. For more information, see Binding a Recording Template.
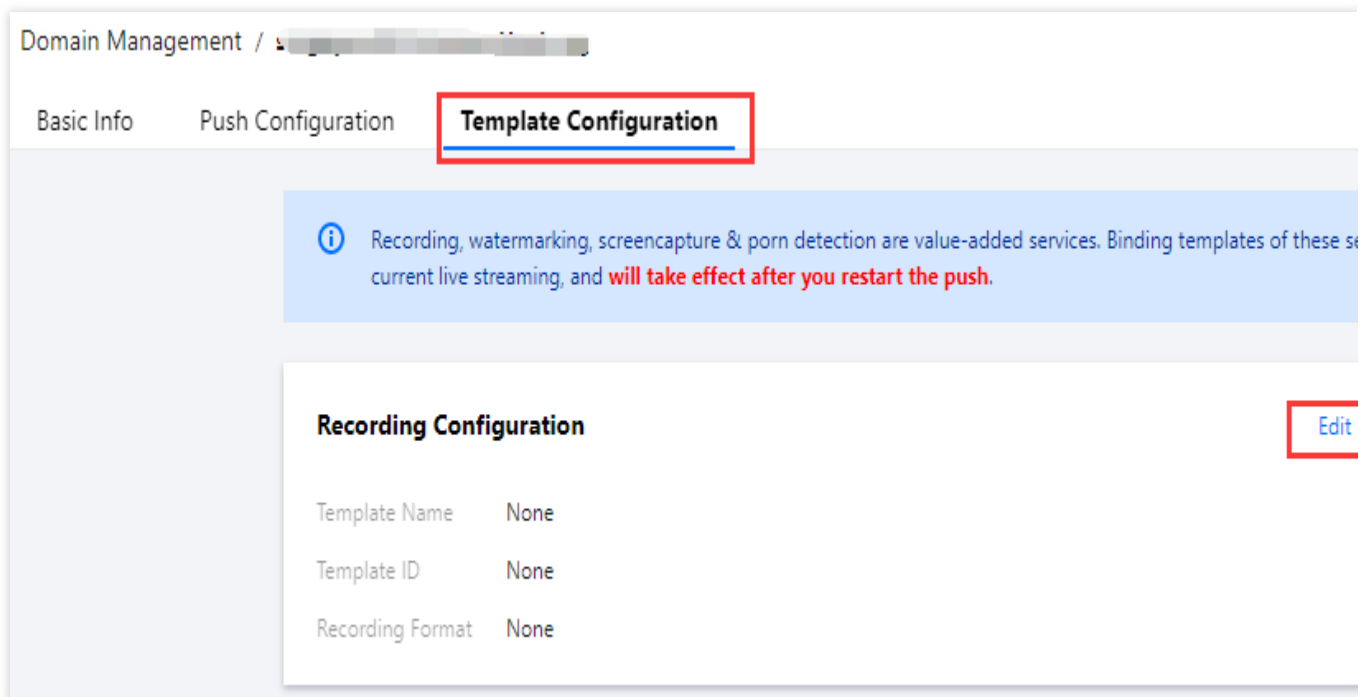
1.3 Alternatively, you can call CreateLiveRecordRule, passing in the template ID and push domain to bind them.

Scheme 2. Recording a specified stream

You can record a specific live stream by binding your live recording template to the specified live stream through an API.

Use cases: Live events, live exhibitions, sports live streaming, and co-streaming.

Steps: To bind a recording template to a specific live stream, call CreateLiveRecordRule, passing in the recording template ID, the target push domain, push path, and `StreamName` (these three parameters must match).

Scheme 3. Recording a specified time period

You can call an API to specify the time to start and end recording.

Use cases: News live streaming and live events.

Steps: Call CreateRecordTask to create a recording task, specifying the recording template ID, the target push domain, push path, and `StreamName` (these three parameters must match), as well as the start and end time.

Example:

1. In the simplest case, you only need to specify `StreamName`, `DomainName`, `AppName`, and `EndTime` to record live streams.

The following sample code creates a video recording task from 8 AM to 10 AM on August 10, 2020. The recording files are in FLV format and are saved permanently. Each segment is 30 minutes long.

Sample request:

```
https://live.tencentcloudapi.com/?Action=CreateRecordTask&AppName=live&DomainName=m
```

2. You can also specify the recording format, recording type, and storage parameters.

The following sample code creates a video recording task from 8 AM to 10 AM on August 10, 2020. The recording

files are in MP4 format and are saved permanently. Each segment is one hour long.

3. Call CreateLiveRecordTemplate to create a recording template.

Sample request:

```
https://live.tencentcloudapi.com/?Action=CreateLiveRecordTemplate&TemplateName=temp
```

Sample response:

```
{"Response": {"RequestId": "839d12da-95a9-43b2-a9a0-03366d01b532","TemplateId": 170
```

4. Call CreateRecordTask to create a recording task.

Sample request:

```
https://live.tencentcloudapi.com/?Action=CreateRecordTask&StreamName=livetest&AppNa
```

Scheme 4. Real-time recording (mixed-stream recording is supported)

You can record any segment of a stream in real time through API calls.

Use cases: Scenarios where only segments need to be recorded, such as sporting events or game live streaming (you can also record streams globally and clip out the segments you need afterward).

Steps: Call CreateRecordTask to create a recording task, specifying the recording template ID, the target push domain, push path, and `StreamName` (these three parameters must match), as well as the end time. The task will start immediately after creation.

Example:

```
https://live.tencentcloudapi.com/?Action=CreateRecordTask&StreamName=test&AppName=l
```

Scheme 5. Audio-only recording

If only audio is published, you can record it in AAC format.

Use cases: Audio live streaming and audio co-streaming.

Steps: When creating the recording template, select AAC as the format, and bind the push domain.

Note:

A binding rule will take effect in about **5-10 minutes** after creation. Any change of the binding rule will not affect live streams that are being pushed and will apply only to new streams.

Step 3. Publish a stream

After binding the recording template with your push domain as instructed in Step 2, generate a push address using the push domain, and push you stream via the address.

After the live stream ends, the recording file will be stored to VOD.

Note:

If you have selected a subapplication when creating the recording template, the recording file will be stored in the specified subapplication.

If you want to receive the URL of the recording file via a callback, you need to create a callback template (enter the callback URL and bind the push domain) before publishing your stream. For details, see Recording Event Notification.

Step 4. Get the recording file

You can query and get a recording file using the following methods:

Recording callback: If you have configured a callback template (you need to provide a callback URL) before publishing a live stream, when a recording file is generated, a callback will be sent to your callback server. To learn more, see Recording Event Notification.

VOD console: You can query a recording file in the VOD console. For more information, see View Audio/Video Files.

VOD API: You can also call the SearchMedia API to query files.

Step 5. Process the recording file

Scheme 1. Live recording + automatic transcoding + video playback acceleration

Use cases: The recording file of a live stream can be immediately transcoded and accelerated automatically for viewers to play back on demand. This scheme is suitable for most live streaming scenarios where no video processing is required.

Steps:

1.1 When creating a recording template before stream publishing, click **Advanced Configuration** to configure a task flow.

1.2 Bind the template to a video processing task flow you created in the VOD console.

1.3 Publish the live stream. For details, see Live Push.

1.4 After recording is finished, get the file ID from the callback.

1.5 Get the URL of the recording file for playback.

Scheme 2. Live recording + manual transcoding + video playback acceleration

Use cases: If you want to record live streams to VOD without transcoding the recording files right away, do not add a task flow when creating the recording template. When you want to transcode the videos later, you can do it manually. You can also use the on-cloud clipping feature to clip the recording files.

Steps:

1.1 Publish the live stream. For details, see Live Push.

1.2 The stream will be automatically recorded to VOD.

1.3 Get the file ID.

1.4 Configure a transcoding template or task flow manually to transcode the recording file. For details, see Template Settings.

1.5 You can further edit the videos if necessary.

1.6 After transcoding and processing, get the URL of the output file for playback.

Scheme 3. Live recording + adaptive bitrate streaming + video delivery acceleration + player

Use cases: If you have high security requirements that cannot be met by HLS encryption, you can use adaptive bitrate streaming together with our Player SDK. This scheme is suitable for online education and corporate training.

Steps:

1.1 Publish the live stream. For details, see Live Push.

1.2 The stream will be automatically recorded to VOD.

1.3 Get the file ID.

1.4 Configure a task flow to generate adaptive bitrate streams. For details, see Task Flow Settings.

1.5 Configure the player: Select the adaptive bitrate stream generated for playback.

1.6 Play the video by specifying the file ID.

Live Streaming Time Shifting

Last updated：2024-10-12 10:22:55

Combining the time shifting feature of CSS and the delivery acceleration capability of VOD, time shifting for live stream playback allows users to rewind and play back earlier parts of a live stream. This is commonly used to play back highlights during live streamed sports events. Users can drag the progress bar to view earlier parts of a live stream, and they can also switch back to continue viewing the currently ongoing live stream.

Strengths

Users can specify the delay for playing live content, i.e., the difference between the playback start time and current time.

Users can specify the bitrate for time shifting if a live stream is recorded in multiple bitrates.

Prerequisites

You have signed up for a Tencent Cloud account and logged in.

You have activated CSS and added a push domain.

Must-Knows

Before you can use the time shifting feature, you must first enable the feature for a stream.

Currently, the smallest time-shift offset is 90 seconds, which means there must be a delay of at least 90 seconds between live content and time-shifted content.

Time shifting relies on the live recording feature. Therefore, you may also incur VOD storage and traffic costs.

Directions

Step 1. Activate VOD

1. Go to the VOD console. Click **Activate Now**.

2. Agree to the service agreement, and click **OK** to activate VOD.

Step 2. Add a domain for time shifting

Follow the steps below to add a VOD domain for time shifting:

1. Log in to the VOD console. Select **Application Management** on the left sidebar and select the target application. Go to **Distribution and Playback > Domain Name**.

2. Click **Add Domain** and enter a domain name. For detailed directions, see Distribution and Playback Settings.

3. Add a CNAME record for the domain.

Step 3. Bind a recording template

1. Go to the CSS console and select **Feature Configuration** > **Live Recording**.

2. Click **Create Template**. For detailed directions, see Live Recording.

Note:

Select HLS as the file format.

Set the file storage period, which cannot be shorter than the time-shift period.

3. Bind the recording template to the push domain. For detailed directions, see Recording Configuration.

Step 4. Enable time shifting

You need to submit a ticket to enable the time shifting feature: Select "CSS", make your request, and provide the following parameters:

The domain added in Step 2.

The ID of the recording template created in Step 3.

The time-shift period (`timeshift_dur`) in seconds.

Note:

Video on Demand

Time-shift period indicates how further back you can go in terms of time shifting. The maximum time-shift period allowed currently is 7 days.

This feature may not work exactly according to the period configured. We recommend you set the period a little longer than you actually need.

If the time-shift period is set to `7200` (two hours), you will be able to request content generated as early as two hours ago. (i.e., the value range for playback delay is 90 seconds to 2 hours). If content generated earlier than two hours ago is requested, `HTTP 404` will be returned even if there is live streaming content at the requested time.

Playback Request

Request URL format

```
http://[Domain]/timeshift/[AppName]/[StreamName]/timeshift.m3u8?delay=xxx
```

Parameter description

Parameter	Description
[Domain]	The domain you registered for time shifting (the domain you added in the VOD console).
timeshift	This is a fixed field. You don't need to modify it.
[AppName]	The application name. For example, if your application is named `live`, pass in `live`.
[StreamName]	The stream name.
timeshift.m3u8	This is a fixed field. You don't need to modify it.
delay	The playback delay (seconds). This parameter cannot be smaller than 90. If you pass in a value smaller than 90, 90 will be used.

Example

Suppose the time-shift domain name is `testtimeshift.com`, application name `live`, and stream name `SLPUrIFzGPE`. If you need to play back live content generated 5 minutes ago, the request URL should be:

```
http://testtimeshift.com/timeshift/live/SLPUrIFzGPE/timeshift.m3u8?delay=300
```

Using VOD and WordPress to Build a Website

Last updated：2022-11-02 16:27:12

Overview

WordPress is a popular open-source blog framework and content management system. It can significantly reduce the difficulty and increase the efficiency of website building. Using WordPress plans, you can get a small website up and running in just a few minutes. However, as you add images, audios, and videos to your WordPress website, it will become more and more demanding on bandwidth and storage, which may start to affect your website's performance. The management of media files is crucial to the long-term success of a website.

Why VOD

We do not recommend using WordPress' default features to upload videos directly to a WordPress website that you intend to run long term. This is mainly because the media files are stored in a web server, which has the following disadvantages:

No.	Disadvantages
1	When users access media files hosted in a web server, videos may stutter and images may be slow to load.
2	Accessing media files hosted in a web server consumes bandwidth resources and may take up all your bandwidth.
3	As your website grows, media files may take up all the server's storage space.
4	When you migrate to a different server, the transfer of media files can be cumbersome.

VOD offers media storage, transcoding, and distribution/playback acceleration services. Using it to build your website has the following advantages:

No.	Advantages
1	Using our WordPress plugin or SDK, you can upload and store your website's media files to VOD. This can free up storage space in the server and facilitate easier server migration. It also avoids the problem of deleting files by mistake.
2	VOD offers CDN addresses for your media files, which makes accessing images, audios, and videos faster and does not consume your server's bandwidth.

VOD also offers access control, media processing, and content moderation capabilities. For details, see Feature Overview.

VOD plugin

VOD offers a WordPress plugin. Using this plugin, you can bind a VOD subapplication to host your WordPress media files in VOD. Visitors of your website can access the files via the CDN addresses provided by VOD. Currently, the plugin can achieve the following:

You can bind a VOD subapplication to WordPress.

The video files you upload to WordPress can be automatically saved to the VOD subapplication.

If you embed a video on a webpage, the video will be played via the address offered by VOD.

Prerequisites

1. Create a VOD subapplication

VOD supplications help you achieve resource isolation. You can bind different websites to different VOD subapplications so that you can manage their media files separately. To learn about how to create a subapplication, see Subapplication System.

2. Use WordPress to build a website

You can build a WordPress website in one of the following two ways:

Method 1 (recommended)

Method 2

Use the WordPress image of TencentCloud Lighthouse or Cloud Virtual Machine to build a website.

Download and install the latest version of WordPress. For detailed directions, see How to install WordPress.

Directions

Step 1. Install the plugin

Install online

Download and install

Type **tencentcloud-vod** in the search box, click **Install**, and then click **Activate**.
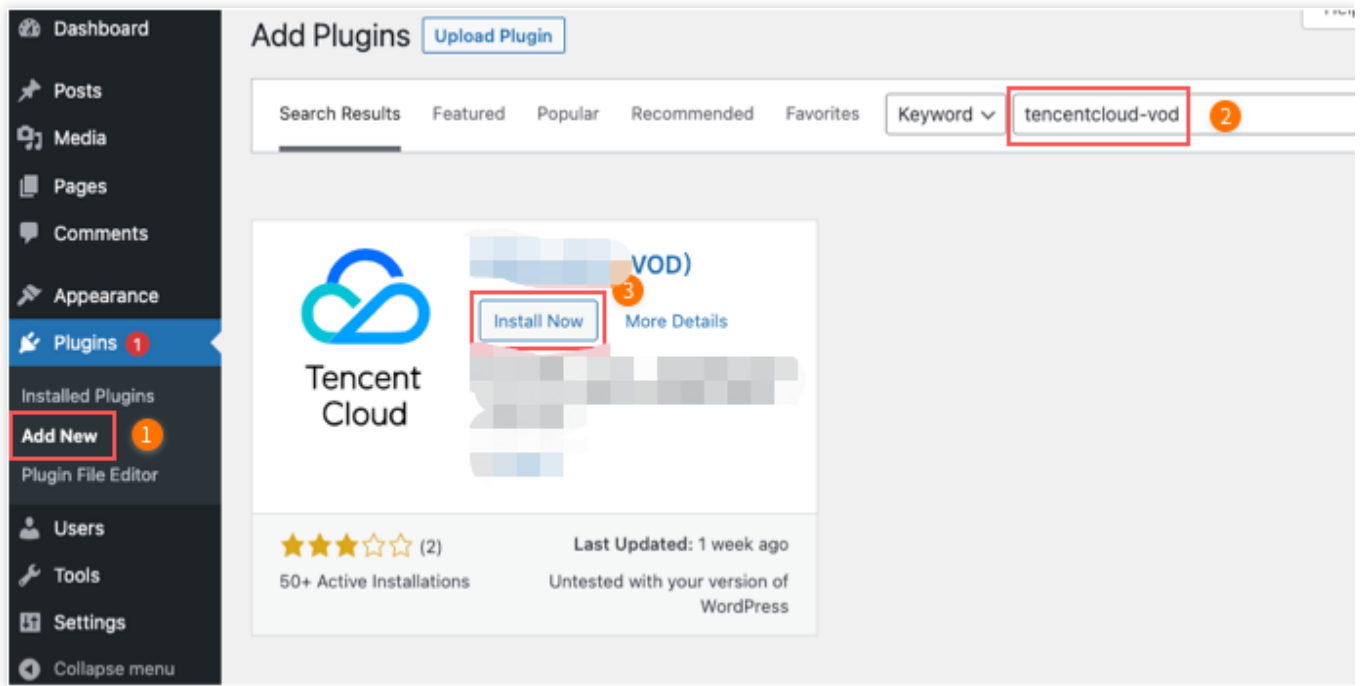
Download a ZIP file of the latest version of the plugin. In WordPress, click **Upload Plugin > Choose File > Install Now**. After the installation, click **Activate**.
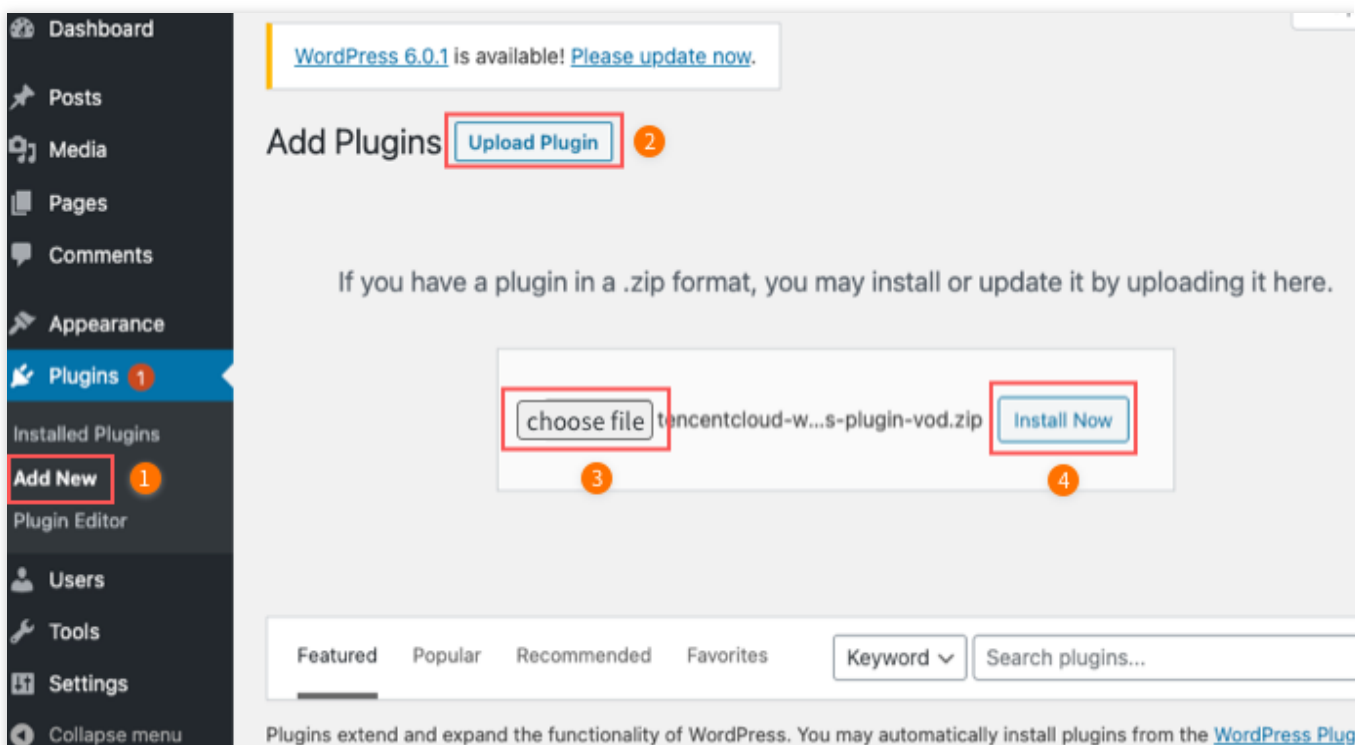
Step 2. Configure the plugin

In WordPress, select **Tencent Cloud Settings >** VOD on the left sidebar.

Complete the following settings:

Item	Description
Custom key	If you toggle this off, the key configured for **Tencent Cloud Settings** will be used, which means the VOD plugin will use the same key as other Tencent Cloud plugins. If you toggle this on, the key configured below will be used.
SecretId, SecretKey	The access key. You can go to the API Keys page of the console to create and view access keys.
SubAppID	The ID of the VOD subapplication you want to bind. You can view your VOD subapplications in Application Management.
Adaptive bitrate	If you toggle this on, videos uploaded will be encoded at multiple bitrates. For details, see Adaptive Bitrate Streaming.

Click **Save**.

3. Upload videos and use VOD videos in your webpage

1. After you finish configuring the plugin, the videos in the media library of WordPress as well as the videos you upload while editing a post or webpage will be automatically saved to VOD.

2. For example, when you edit a post in WordPress, click the plus icon, select **Video**, and click **Upload** to upload a video. After the video is uploaded, click **Preview** in the top right corner. You will see that the video is played via VOD's URL.
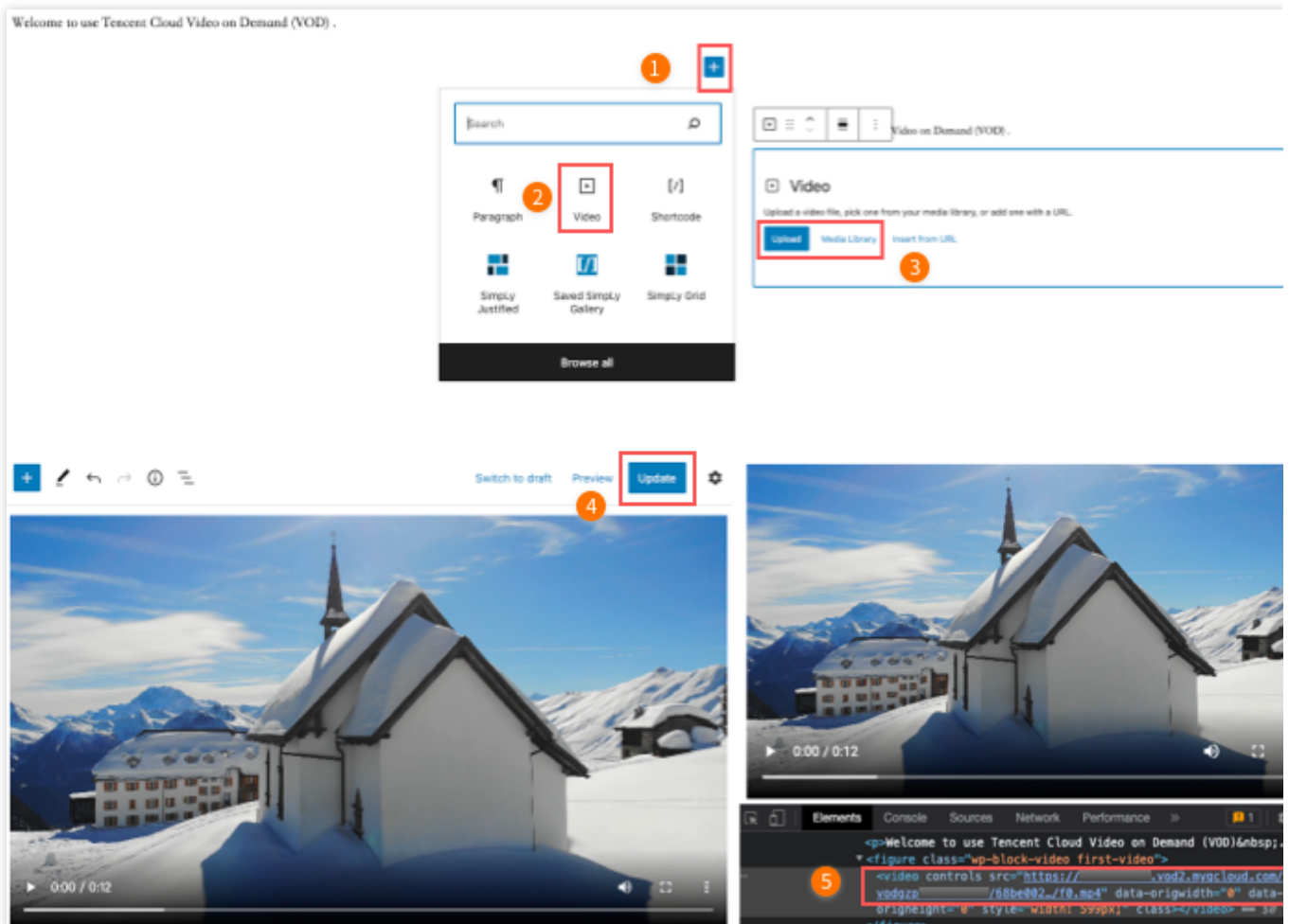
3. As you can see below, the video on the left is stored in a web server, which stutters due to limited bandwidth. This is not a problem if you use VOD, which uses its CDN to accelerate video distribution.
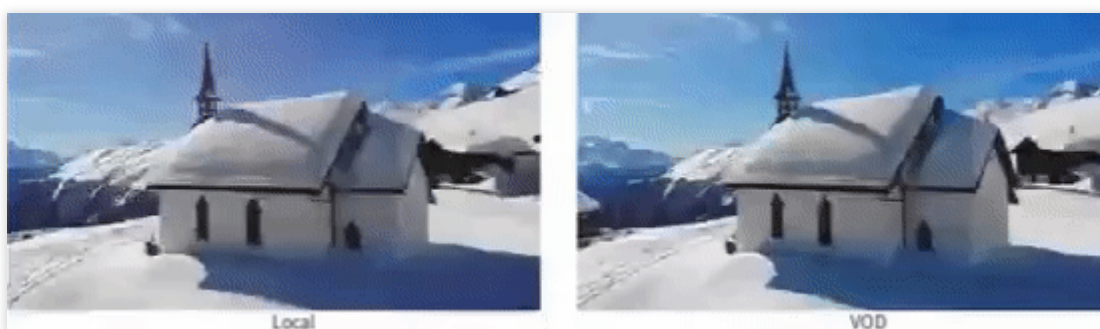
Others

Using HTTPS

VOD uses HTTP by default. If you want to use HTTPS, select your subapplication in the VOD console, go to **System Settings > Distribution and Playback > Default Distribution Domain**, click **Edit**, and select **HTTPS** for **Default Distribution Protocol**.

How to Pull from Custom Origin Servers

Last updated：2023-03-07 15:49:29

Overview

You can configure a third-party origin server for your domain in VOD to pull media files from the origin server and distribute the media via VOD. This multi-cloud media distribution solution is enabled by VOD's CDN capability. This document shows you how to configure and use an origin server with VOD.

Use cases

Your media assets are stored with a third-party cloud vendor, and **the cost of migrating them is high**. You can use VOD's custom origin server feature to deliver your content without migrating your media assets.
You have **high requirements on latency and stuttering** and your existing service provider is unable to meet them. You can use VOD's custom origin server feature to deliver smoother playback and improve your user experience.
You need **more than one channel** to distribute your content to ensure the reliability and improve the disaster recovery capability of your business.

Prerequisites

1. You have signed up for a Tencent Cloud account, completed identity verification, and logged in.
2. You have activated VOD.

Supported Origin Server Types

Self-owned origin servers
Third-party storage services

Directions

Step 1. Add a domain and configure origin server information

1. Log in to the VOD console and select **Distribution and Playback** > Domain Name on the left sidebar.
2. Select the **Custom origin server domains** tab at the top.

3. Click **Add Domain** and enter a domain. For more information, see Customizing Domain Names.

4. Enter your origin server information. Currently, **self-owned servers** and **third-party storage** are supported.

Self-owned servers

If you want to use your existing server as the origin server to distribute media files stored in it via VOD, configure as follows:

Item	Description
Origin server type	Select Self-owned server
Protocol	Select the protocol, which can be HTTP, HTTPS, or "Same as request".
Origin server address	You can enter multiple IP addresses (separate with commas) or one domain.
Host	For self-owned origin servers, you can specify the host of a domain or IP address that VOD pulls from. If you do not specify it, VOD will pull from the current domain.

Note:

You cannot use your VOD default domain as the origin server address.

If you select **Same as request**, HTTP will be used for HTTP requests, and HTTPS will be used for HTTPS requests (your origin server must support HTTPS).

You can use a domain as the origin server address, but this domain cannot be the same as the domain you use for acceleration.

Third-party storage

If you want to distribute the media files you store in a third-party storage service via VOD's CDN, configure as follows:

Item	Description
Origin server type	Select Third-party storage
Third-party storage	Currently, Alibaba Cloud OSS and AWS S3 are supported.

Protocol	HTTP or HTTPS
Origin server address	Enter a valid bucket address (cannot contain "http://" or http:// headers).

If you use a private bucket as the origin server, you need to enter a valid access ID and key to authorize VOD to access the bucket.

Step 2. Configure CNAME for your domain

After you add a custom domain in VOD, you need to add a CNAME record for it at your DNS provider before users can access your media files via the domain. For detailed directions, see Configuring CNAME.

Step 3. View and modify origin server information

1. Go to **Domain Name**, select the **Custom origin server domains** tab, find the domain you added, and click **Set**.

2. Click **Basic Configuration** to view or modify the origin server settings of the domain.
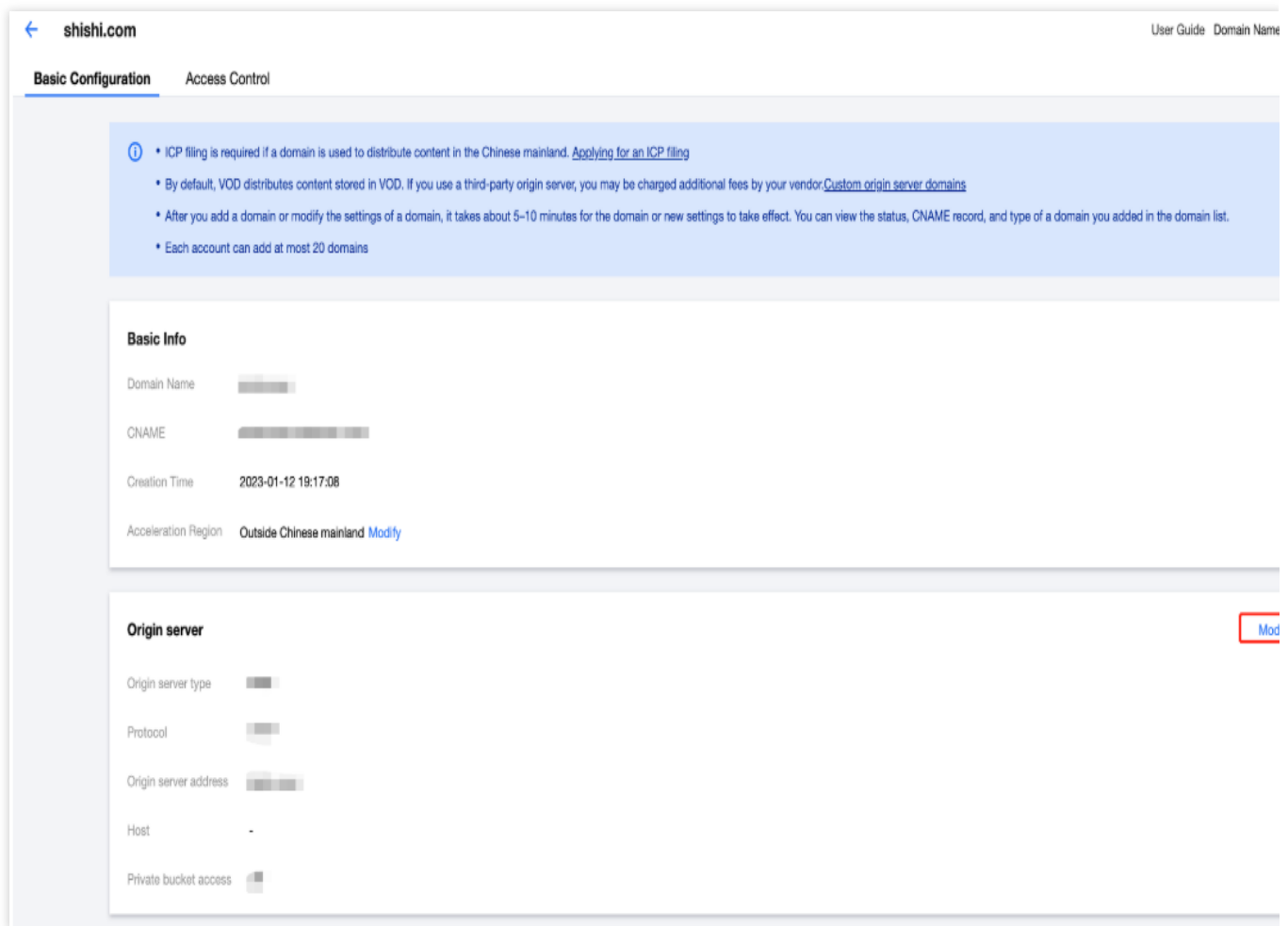
After completing the above steps, you will be able to distribute media files to end users from a self-owned server or third-party storage service via VOD's CDN. The process is as follows:

1. After you complete the above settings for a custom domain (for example `test.com`), when an end user opens the address of a media file under the domain (for example `http://www.test.com/test.mp4`) in a browser, a request will be sent to the local DNS resolver.

2. Because a CNAME record (`www.test.com.cdn.dnsv1.com`) has been added for the domain, the request will be referred to Tencent's proprietary load balancing system Global Server Load Balance (GSLB).

3. GSLB will return a list of IP addresses to the local DNS resolver.

4. The user gets the IP address of the optimal node to access the media file.

5. The end user sends a request to the IP address to visit the media file `http://www.test.com/test.mp4` .

6. If the `test.mp4` file has been cached in the CDN node, the file will be returned to the user. If not, the node will request the file from the **origin server you configured**, cache the file, and return it to the user.

Note:

Using VOD's origin server feature to distribute media files will incur **playback traffic costs** and **origin server traffic costs**. Playback traffic costs are charged by VOD. For details, see Daily Pay-As-You-Go and Prepaid Packages. Origin server traffic costs are charged by your origin server.

How to Use EdgeOne to Distribute Content in VOD

Last updated：2024-11-29 17:58:49

This document will introduce how to use EdgeOne to distribute files stored in VOD. While enabling EdgeOne domain protection, it also utilizes VOD's copyright protection capabilities to further enhance the protection of high-quality content.

Overview

If your audio and video content is of high value (e.g., courses, movies, short series, etc.), to prevent it from being pirated, you typically need to upload the content to cloud VOD and use private encryption or DRM encryption solutions to secure the content. Additionally, to avoid traffic abuse, you have configured anti-fraud settings such as usage cap and BOT protection in EdgeOne. Now, you can use EdgeOne domains with anti-fraud configurations to accelerate the distribution of encrypted videos in VOD, achieving a stronger content protection effect.

Prerequisites

1. You have signed up for and logged in to your Tencent Cloud account and completed identity verification.
2. You have activated the VOD service.
3. Refer to the guide "Site Acceleration Quick Start" to complete the site and domain integration.

Directions

Step 1. Add the EdgeOne domain for VOD origin site

1. Log in to the EdgeOne console, click **Site List** in the left menu bar, and click the site to be configured on the site list.
2. On the left navigation bar, click **Domain Name Service** > **Domain Management** to enter the domain name management page, and click **Add domain name** to add a new domain name.

3. In the domain configuration, enter your acceleration domain, select **VOD** as the origin type, choose VOD application you want to use, and click **Next**.

4. In the recommended configuration, select the **audio and video on demand** and click **Next**.

5. In the CNAME access mode, EdgeOne will assign a CNAME address for the domain. You need to configure the CNAME to enable the secure acceleration for the domain. For configuration instructions, please refer to: Modifying CNAME Records. Once the configuration is complete, click **Finish**.

Step 2. Enable Access Control for the EdgeOne Domain (Optional)

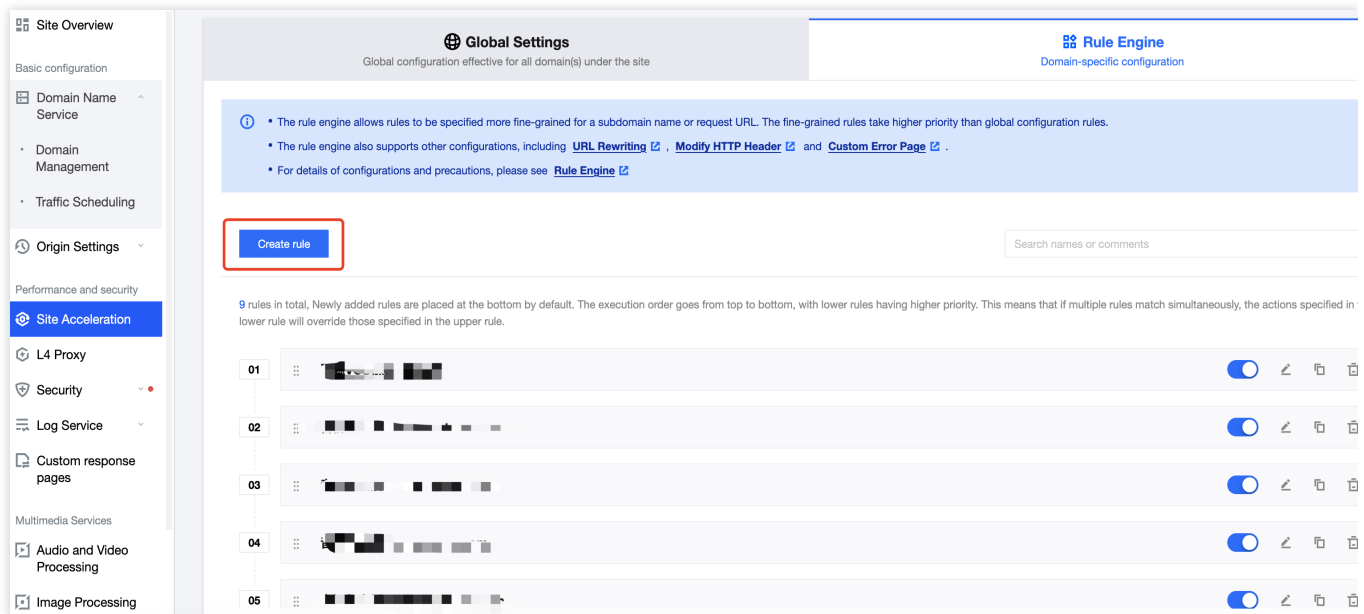**Note**：

If you do not need to enable access control, you can skip this step.

1. Log in to the EdgeOne console. In the left sidebar, click **Site List**. In the site list, click the site to be configured.

2. On the site details page, click **Site Acceleration** to go to the global site configuration page. Then click the **Rule Engine** > **Create Rule** tab, and select **Add Blank Rule**.
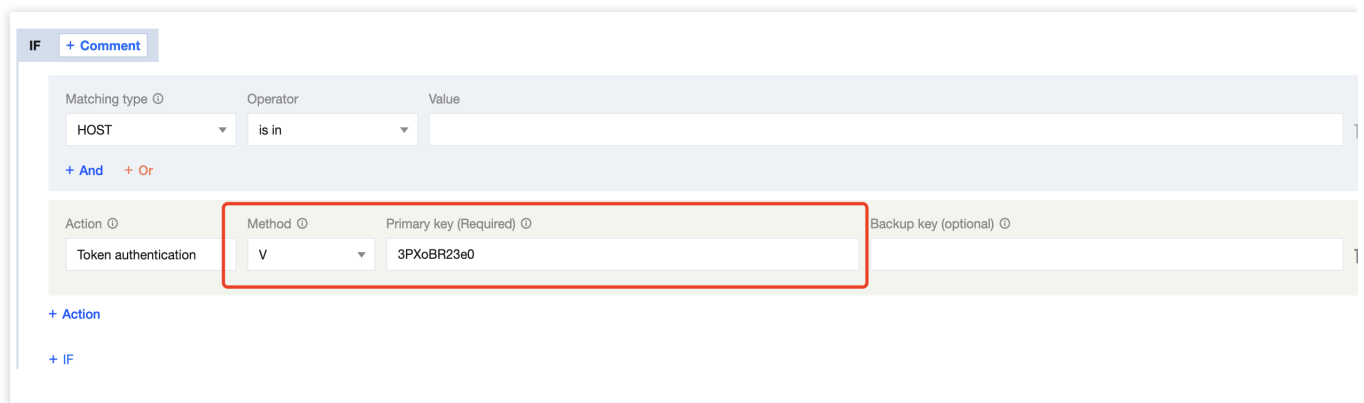
3. On the rule editing page, set the matching conditions that trigger this rule. Click **Action > Select Box**, and select **Token Authentication** in the pop-up action list.

4. In the Token Authentication operation, select "V" as the authentication method, enter your authentication key, and save the authentication key for use in subsequent steps.

5. Finally, click **Save** to enable access control.

Step 3. Import the EdgeOne domain into VOD.

1. Log in to the VOD console and select Application Management on the left sidebar.

2. Select the target application. By default, you will enter the **Media Assets > Video Management > Uploaded** page.

3. Select Distribution and **Playback > Domain Name** on the left sidebar, click **Add Domain.**

4. On the Add Domain page, select **Import existing EdgeOne domain** as the domain type, then choose the EdgeOne domain added in step 1. Click **Confirm** to import the EdgeOne domain into Cloud VOD.

Step 4. Add the Authentication Key to Cloud VOD (Optional)

Note：

If you have not enabled access control in EdgeOne, you can skip this step.

1. Log in to the VOD console and select Application Management on the left sidebar.

2. Select the target application. By default, you will enter the **Media Assets > Video Management > Uploaded** page.

3. Select Distribution and **Playback > Domain Name** on the left sidebar, select the EdgeOne domain you need to configure and click **Set**.

4. On the Domain Basic Configuration page, turn on the **Authentication Key** switch.

5. In the pop-up window, enter the authentication key saved in step 2 and click **Confirm.**

Notes

1. VOD currently only supports Type V authentication in EdgeOne.

2. If you modify the authentication key in EdgeOne, please promptly update the key in the domain settings of VOD to avoid any playback issues.

Related Documentation

If you want to learn more about VOD copyright protection, please refer to Media Encryption and Copyright Protection.

If you want to learn more about Type V in EdgeOne, please refer to Token Authentication.