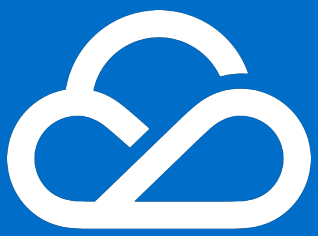
微信小程序云端解决方案

云端场景示例

产品文档



腾讯云

【版权声明】

【商标声明】

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

文件上传下载应用场景

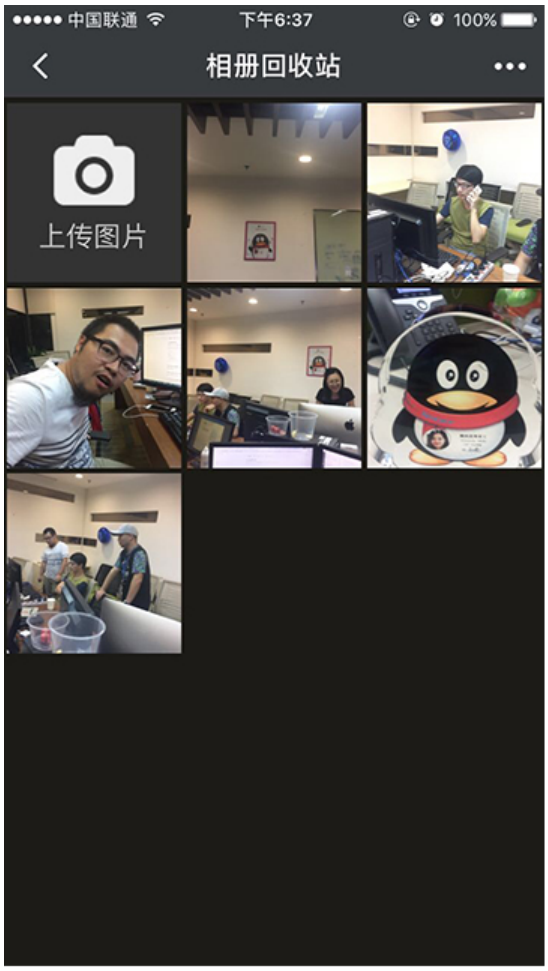
微信小程序提供了一套在微信上运行小程序的解决方案，有比较完整的框架、组件以及API，在这个平台上面的想象空间很大。

小相册是结合腾讯云对象存储服务（Cloud Object Service，简称COS）制作的一个微信小程序示例。在代码结构上包含如下两部分：

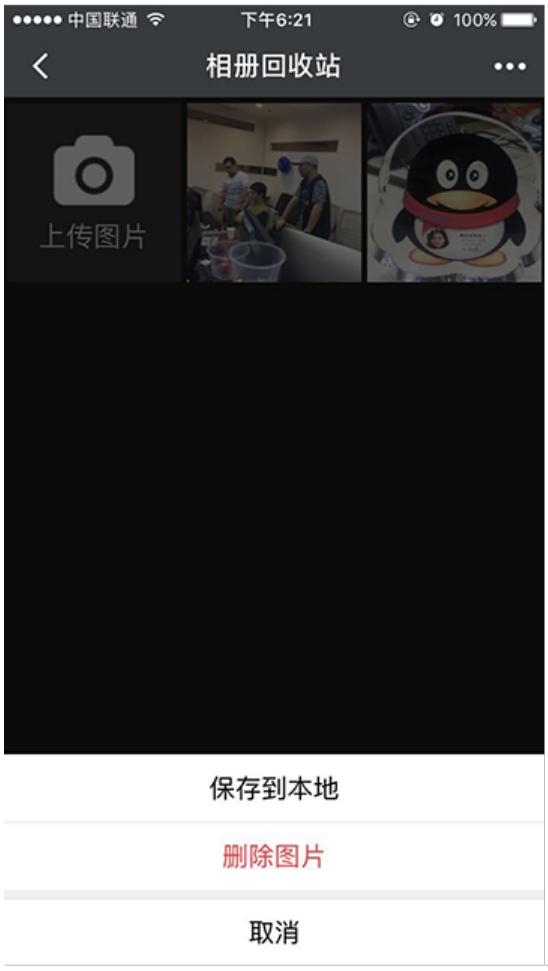
- applet: 小相册应用包代码，可直接在微信开发者工具中作为项目打开
- server: 搭建的 Node 服务端代码，作为服务器和applet通信，提供 CGI 接口示例用于拉取 COS 图片资源、上传图片到 COS、删除 COS 图片等。

小相册主要功能如下：

- 列出 COS 服务器中的图片列表
- 点击左上角上传图片图标，可以调用相机拍照或从手机相册选择图片，并将选中的图片上传到 COS 服务器中
- 轻按任意图片，可进入全屏图片预览模式，并可左右滑动切换预览图片
- 长按任意图片，可将其保存到本地，或从 COS 中删除


中国联通 下午6:37 100%
相册回收站
上传图片


中国联通 3G 上午9:41 43%
相册回收站
上传图片
拍照
从手机相册选择
取消

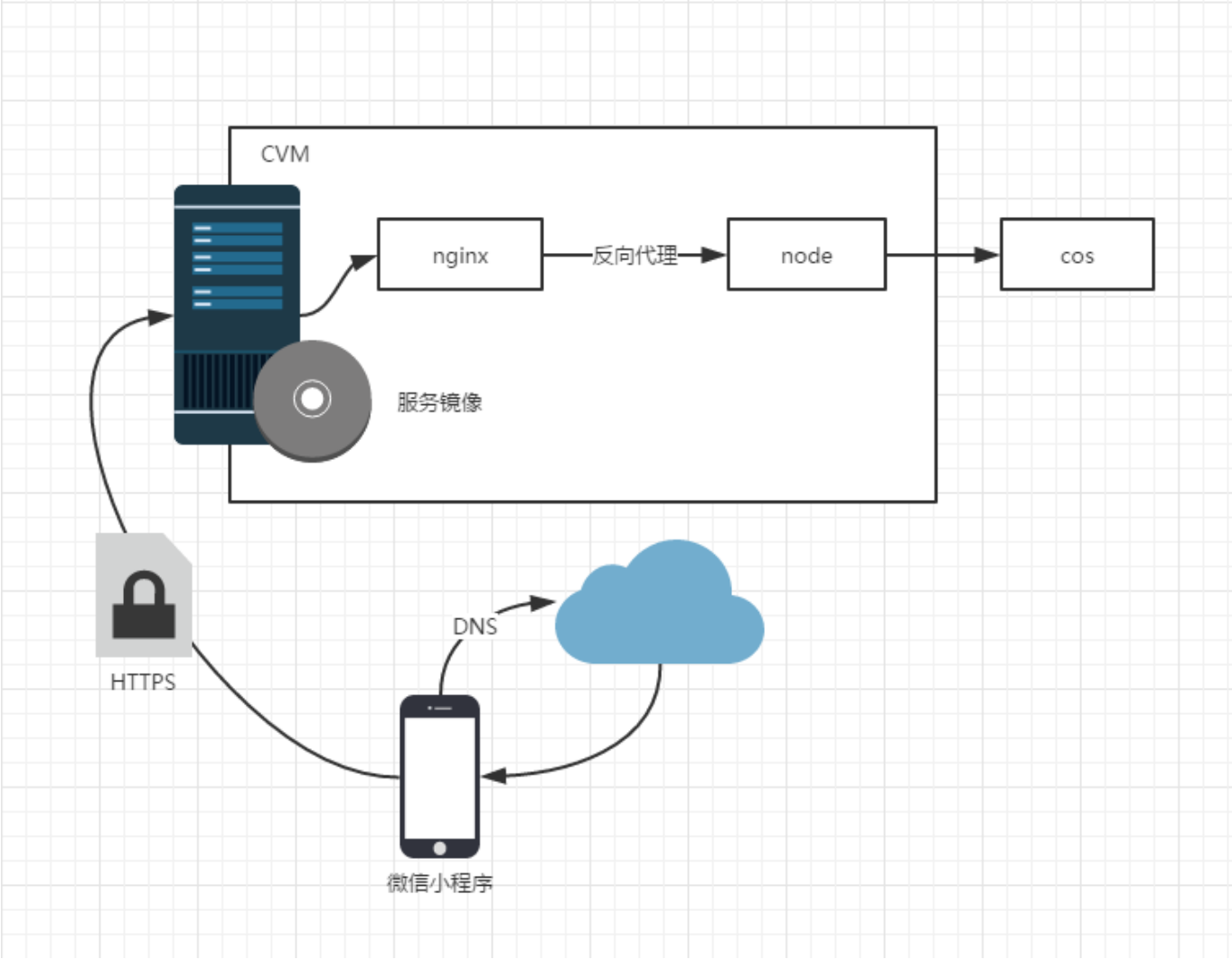

中国联通 下午6:21 100%
相册回收站
上传图片
保存到本地
删除图片
取消


中国联通 下午6:21 100%
相册回收站

部署和运行

拿到了本小程序源码的朋友可以尝试自己运行起来。

整体架构

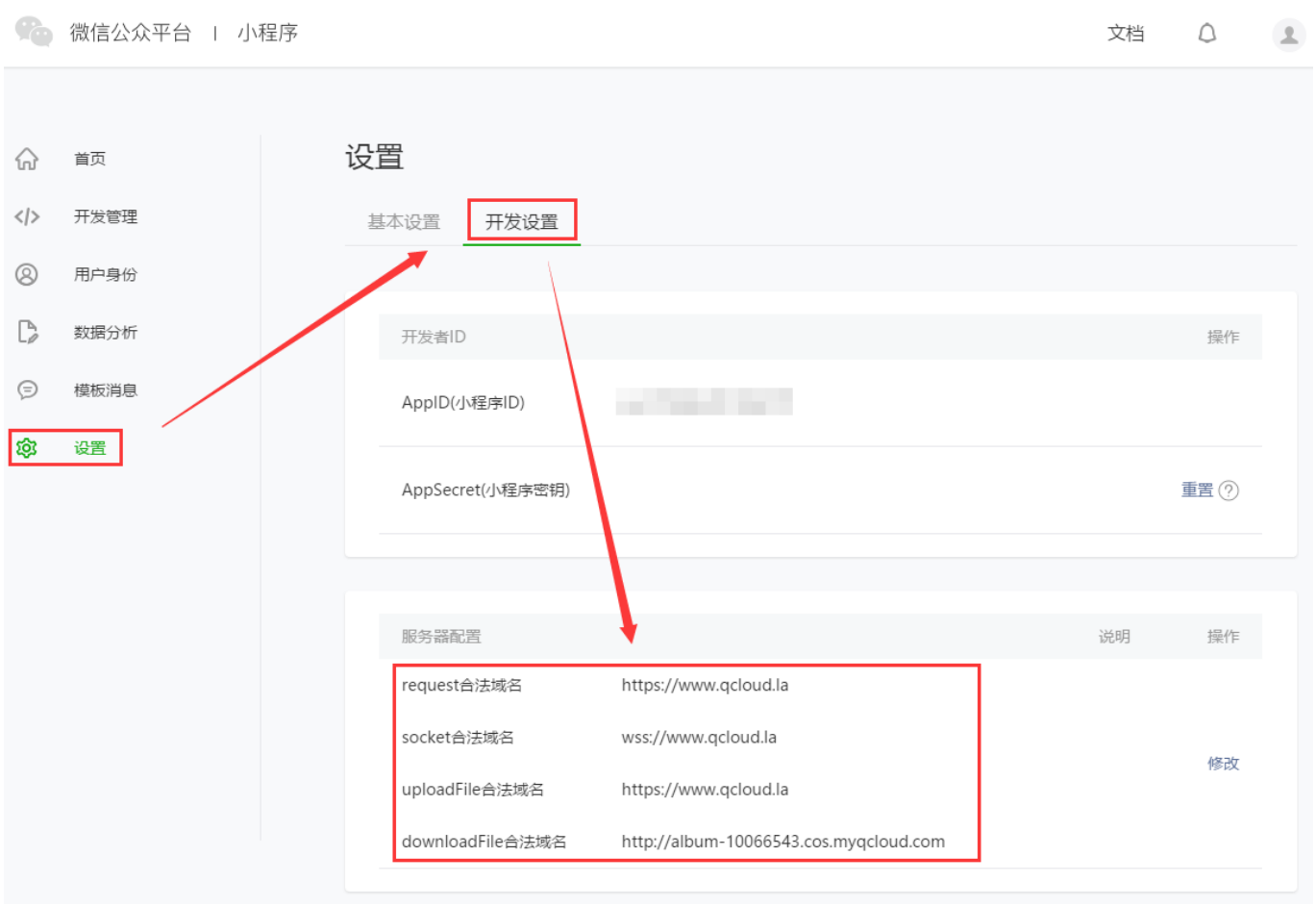


1. 准备域名和证书

在微信小程序中，所有的网络请求受到严格限制，不满足条件的域名和协议无法请求，具体包括：

- 只允许和在 MP 中配置好的域名进行通信，如果还没有域名，需要注册一个。
- 网络请求必须走 HTTPS 协议，所以你还需要为你的域名申请一个证书。

域名注册好之后，可以登录微信公众平台配置通信域名了。



2. 云主机和镜像部署

小相册的服务器运行代码和配置已经打包成腾讯云 CVM 镜像，大家可以直接使用。

腾讯云用户可以免费领取礼包，体验腾讯云小程序解决方案。

镜像已包含所有小程序的服务器环境与代码，需要体验其它小程序的朋友无需重复部署

3. 配置 HTTPS

镜像中已经部署了 nginx，需要在 /etc/nginx/conf.d 下修改配置中的域名、证书、私钥。

```
server {
    listen 80;
    listen 443 ssl;

    server_name  www.qcloud.la;
    charset      utf-8;

    access_log   logs/www.qcloud.la.access.log main;
    error_log    logs/www.qcloud.la.error.log;

    ssl on;
    ssl_certificate             ssl/1_www.qcloud.la_cert.crt;
    ssl_certificate_key         ssl/2_www.qcloud.la.key;
    ssl_session_timeout         5m;
    ssl_protocols               TLSv1;
    ssl_ciphers                 HIGH:!aggNULL:!MD5;
    ssl_prefer_server_ciphers   on;

    include conf.d/partials/*.conf;
}
```

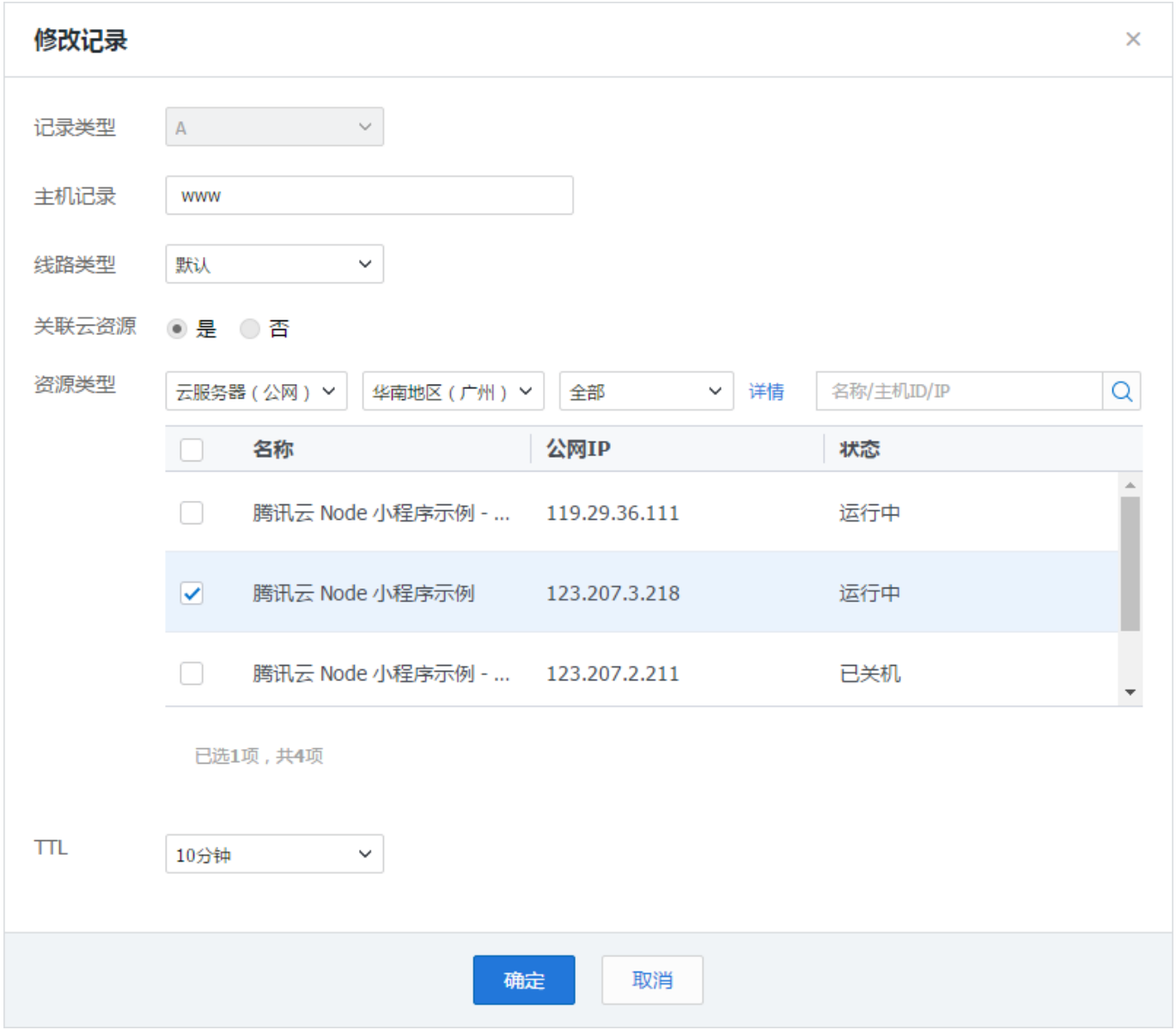
配置完成后，即可启动 nginx。

```
nginx
```

4. 域名解析

我们还需要添加域名记录解析到我们的云服务器上，这样才可以使用域名进行 HTTPS 服务。

在腾讯云注册的域名，可以直接使用云解析控制台来添加主机记录，直接选择上面购买的 CVM。



解析生效后，我们在浏览器使用域名就可以进行 HTTPS 访问。

5. 开通和配置 COS

小相册示例的图片资源是存储在 COS 上的，要使用 COS 服务，需要登录 COS 管理控制台，然后在其中完成以下操作：

- 开通 COS 服务分配得到唯一的APP ID
- 使用密钥管理生成一对SecretID和SecretKey（用于调用 COS API）
- 在 Bucket 列表中创建公有读私有写访问权限、CDN加速的 bucket（存储图片的目标容器）

6. 启动小相册示例 Node 服务

在镜像中，小相册示例的 Node 服务代码已部署在目录/data/release/qcloud-applet-album下：

进入该目录：

```
cd /data/release/qcloud-applet-album
```

在该目录下有个名为config.js的配置文件（如下所示），按注释修改对应的 COS 配置：

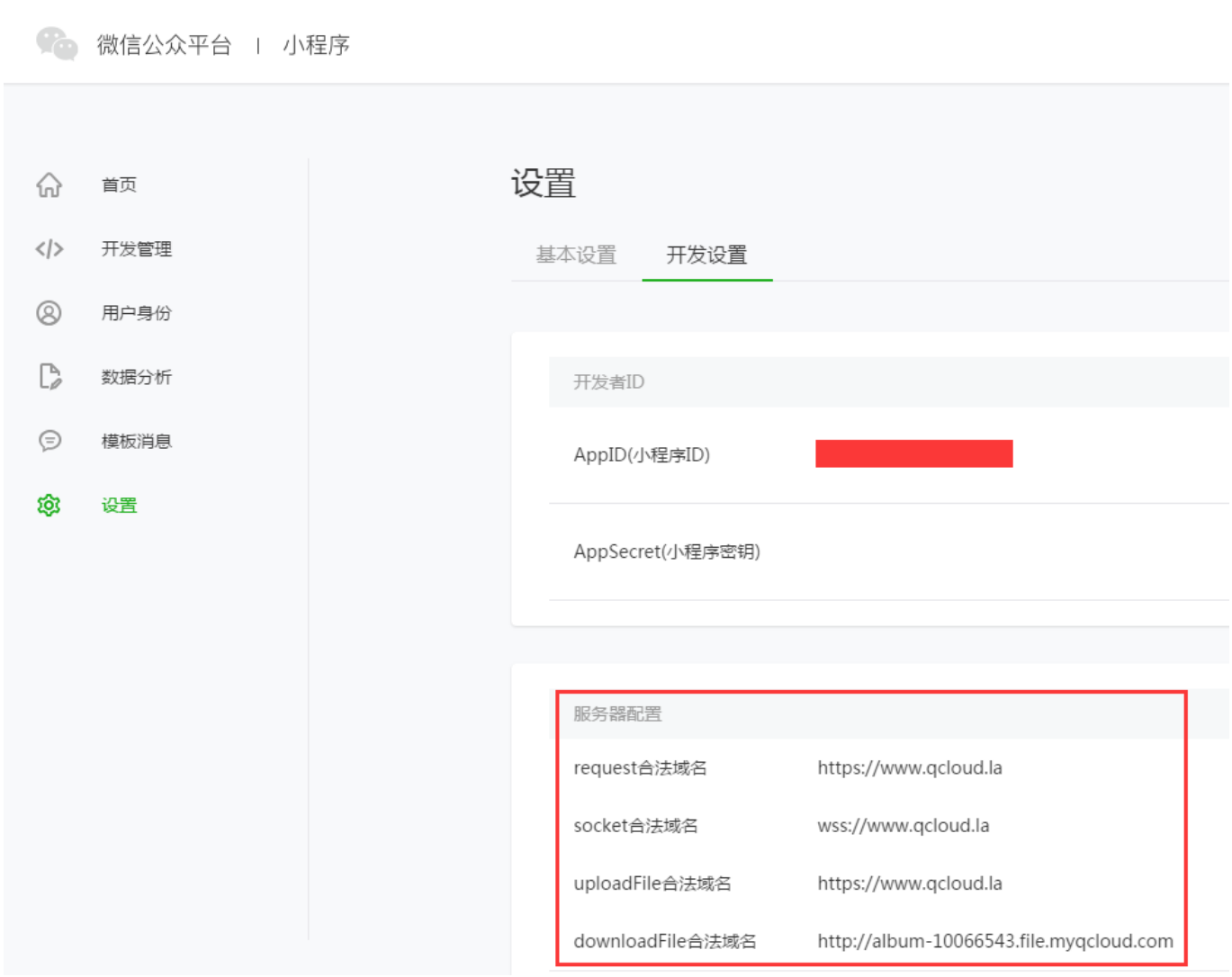
```
module.exports = {
    // Node 监听的端口号
    port: '9993',
    ROUTE_BASE_PATH: '/applet',

    cosAppId: '填写开通 COS 时分配的 APP ID',
    cosSecretId: '填写密钥 SecretID',
    cosSecretKey: '填写密钥 SecretKey',
    cosFileBucket: '填写创建的公有读私有写的bucket名称',
};
```

小相册示例使用pm2管理 Node 进程，执行以下命令启动 node 服务：

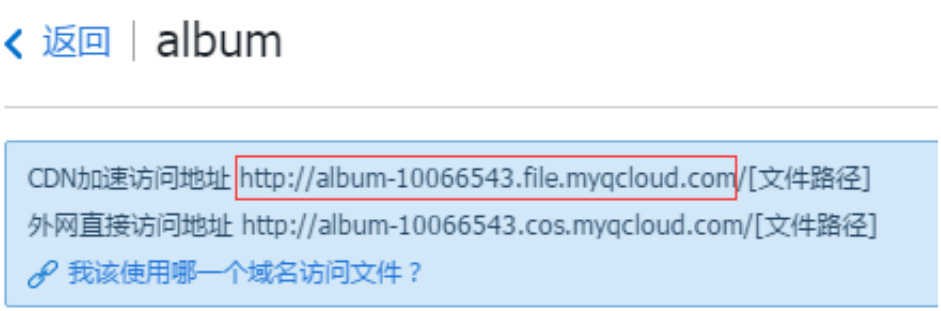
```
pm2 start process.json
```

7. 微信小程序服务器配置

进入微信公众平台管理后台设置服务器配置，配置类似如下设置：

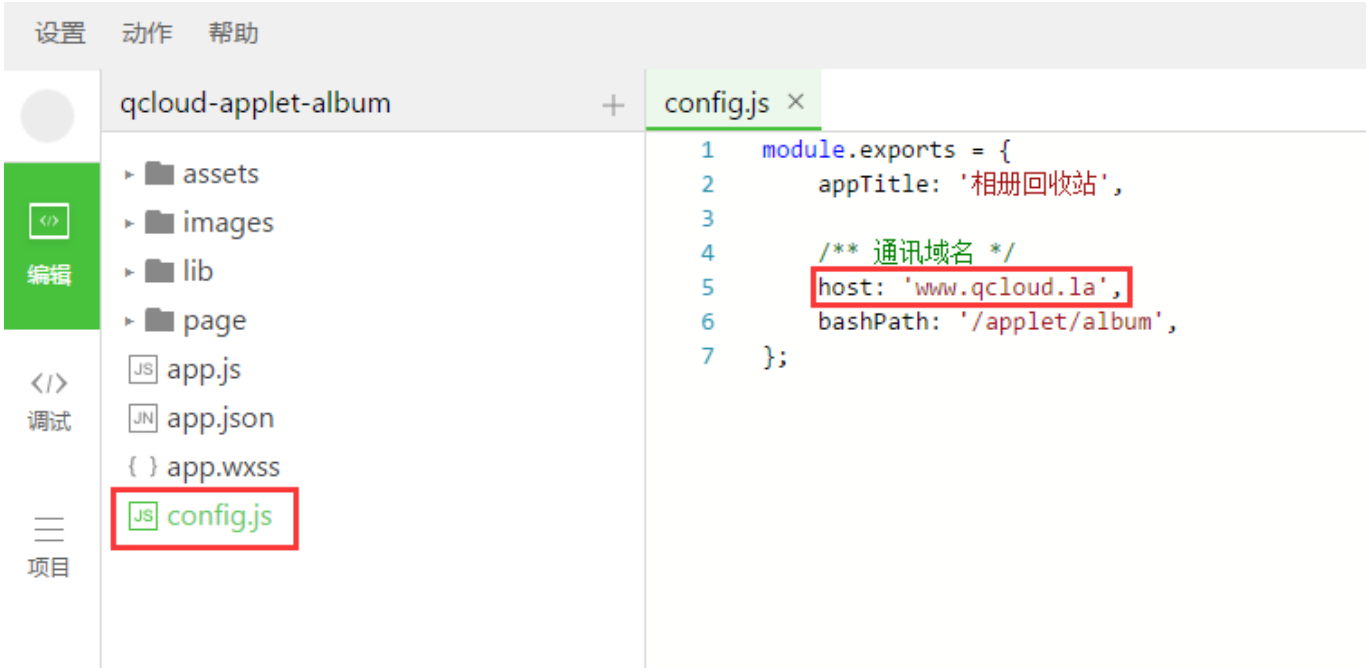


注意：需要将 www.qcloud.la 设置为上面申请的域名，将 downloadFile 合法域名设置为在 COS 管理控制台中自己创建的 bucket 的相应 CDN 加速访问地址，如下图所示：

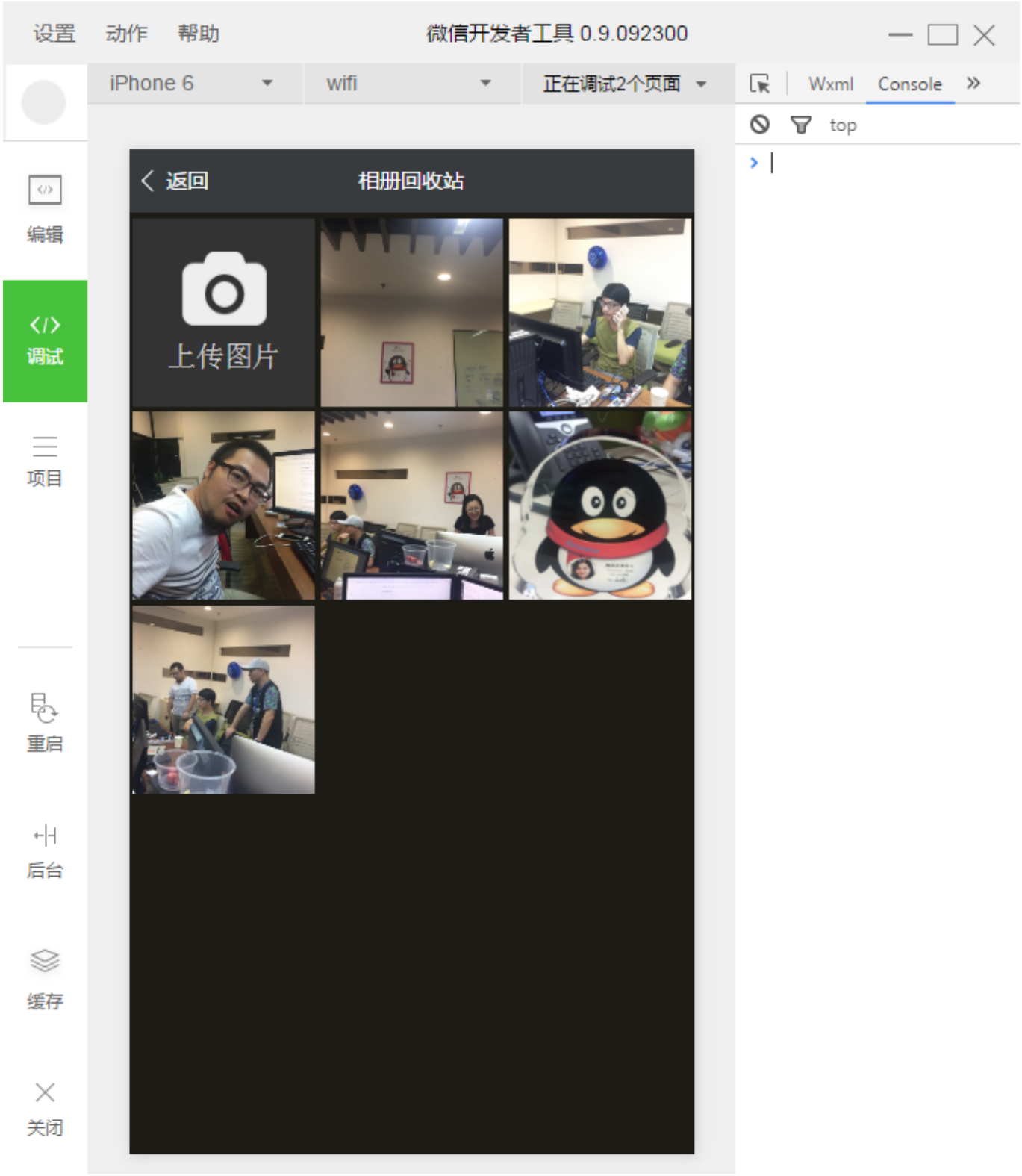


8. 启动小相册 Demo

在微信开发者工具将小相册应用包源码添加为项目，并把源文件config.js中的通讯域名修改成上面申请的域名。



然后点击调试即可打开小相册Demo开始体验。



这里有个问题。截止目前为止，微信小程序提供的上传和下载 API
无法在调试工具中正常工作，需要用手机微信扫码预览体验。

主要功能实现

上传图片

上传图片使用了微信小程序提供的wx.chooseImage(OBJECT)获取需要上传的文件路径，然后调用上传文件接口wx.request(OBJECT)发送 HTTPS POST 请求到自己指定的后台服务器。和传统表单文件上传一样，请求头 Content-Type也是multipart/form-data。后台服务器收到请求后，使用 npm 模块 multiparty 解析 multipart/form-data
请求，将解析后的数据保存为指定目录下的临时文件。拿到临时文件的路径后，就可直接调用 COS SDK 提供的文件上传 API 进行图片存储，最后得到图片的存储路径及访问地址（存储的图片路径也可以直接在 COS 管理控制台看到）。

获取图片列表

调用列举目录下文件&目录 API可以获取到在 COS 服务端指定 bucket 和该 bucket 指定路径下存储的图片。

下载和保存图片

指定图片的访问地址，然后调用微信小程序提供的wx.downloadFile(OBJECT)和wx.saveFile(OBJECT)接口可以直接将图片下载和保存本地。这里要注意图片访问地址的域名需要和服务器配置的 dowmloadFile 合法域名一致，否则无法下载。

删除图片

删除图片也十分简单，直接调用文件删除 API 就可以将存储在 COS 服务端的图片删除。

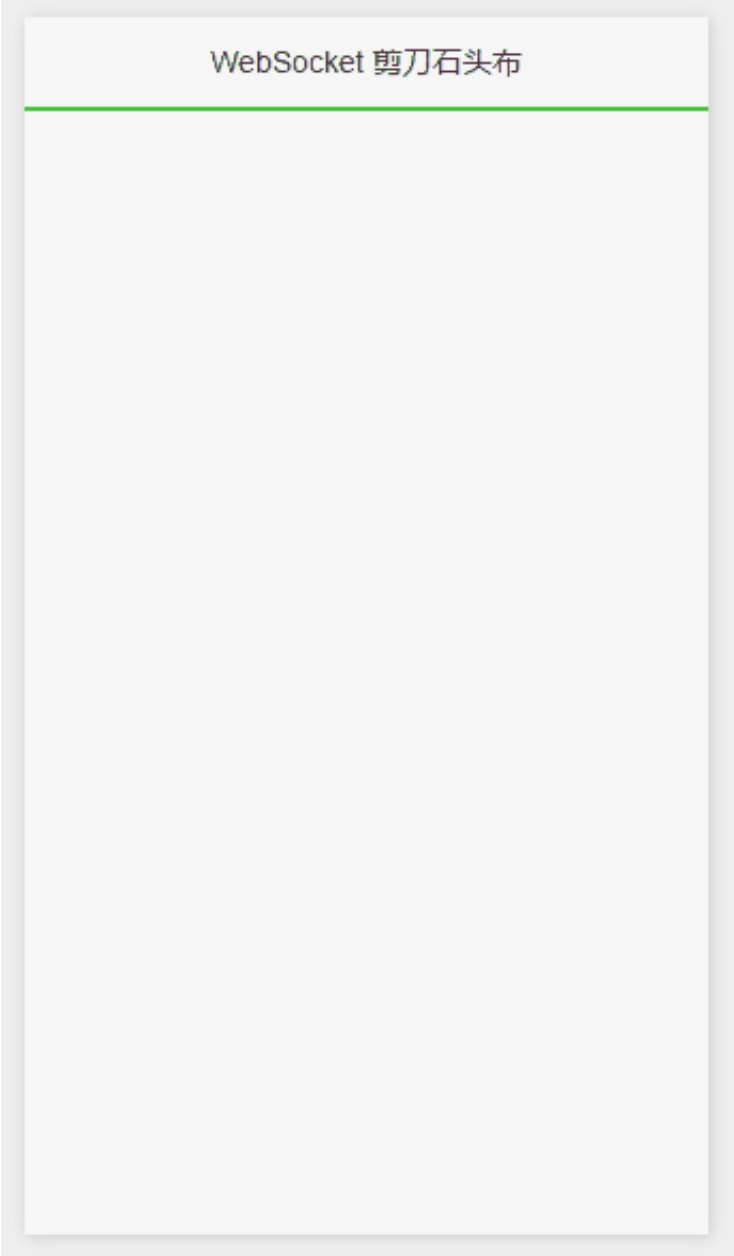
WebSocket长连接应用场景

没事打开小程序，和附近的人剪刀石头布，想来就来，想走就走。谁能成为武林高手？！

微信小程序提供了一套在微信上运行小程序的解决方案，有比较完整的框架、组件以及API，在这个平台上面的想象空间很大。

腾讯云拿到了小程序内测资格，研究了一番之后，发现微信支持 WebSocket 还是很值得玩味的。这个特性意味着我们可以做一些实时同步或者协作的小程序。

这篇文章分享一个简单的剪刀石头布的小游戏的制作，希望能对想要在小程序中使用 WebSocket 的开发者有帮助。

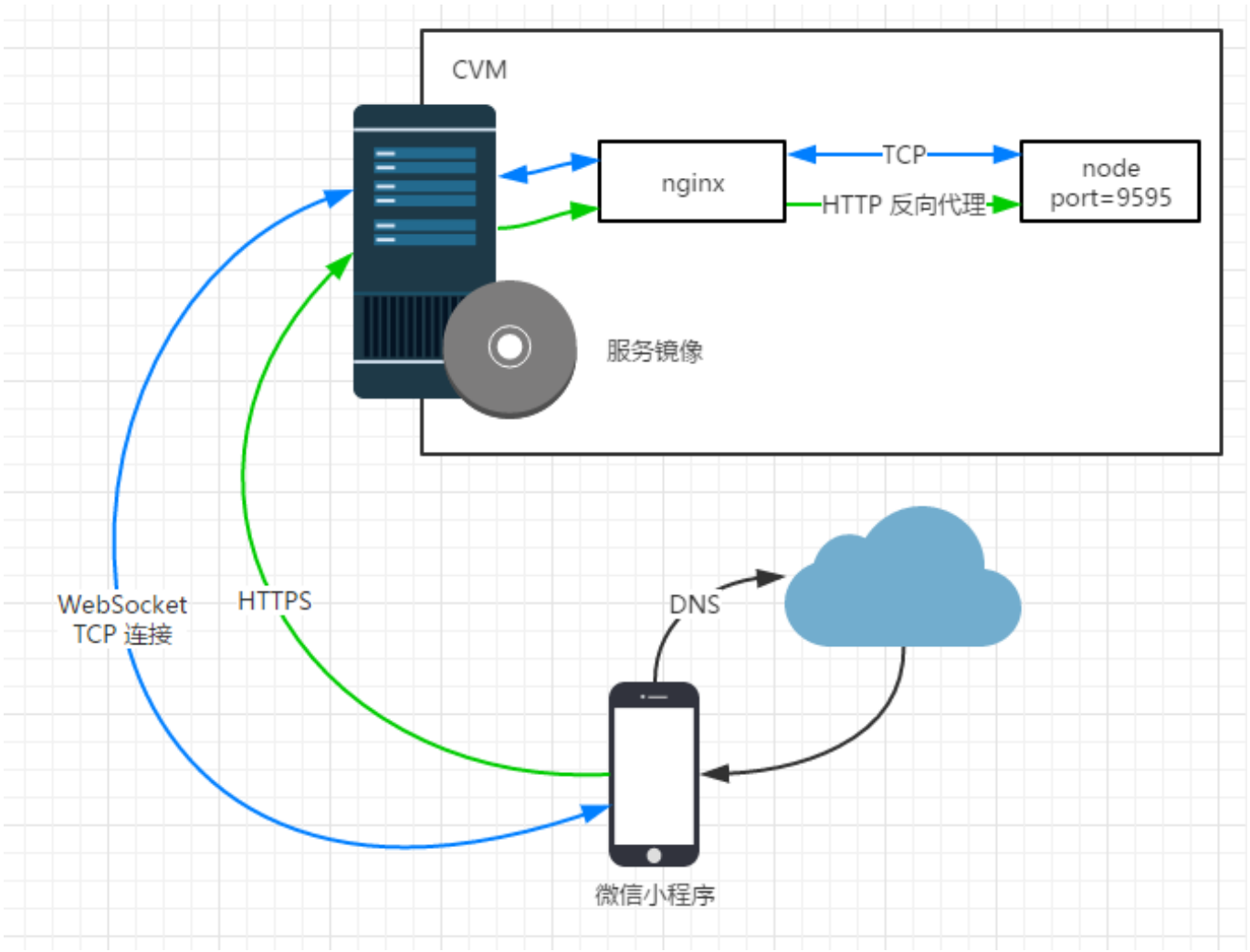


整个游戏非常简单，连接到服务器后自动匹配在线玩家（没有则分配一个机器人），然后两人进行剪刀石头布的对抗游戏。当对方进行拳头选择的时候，头像会旋转，这个过程使用 WebSocket 会变得简单快速。

部署和运行

拿到了本小程序源码的朋友可以尝试自己运行起来。

整体架构



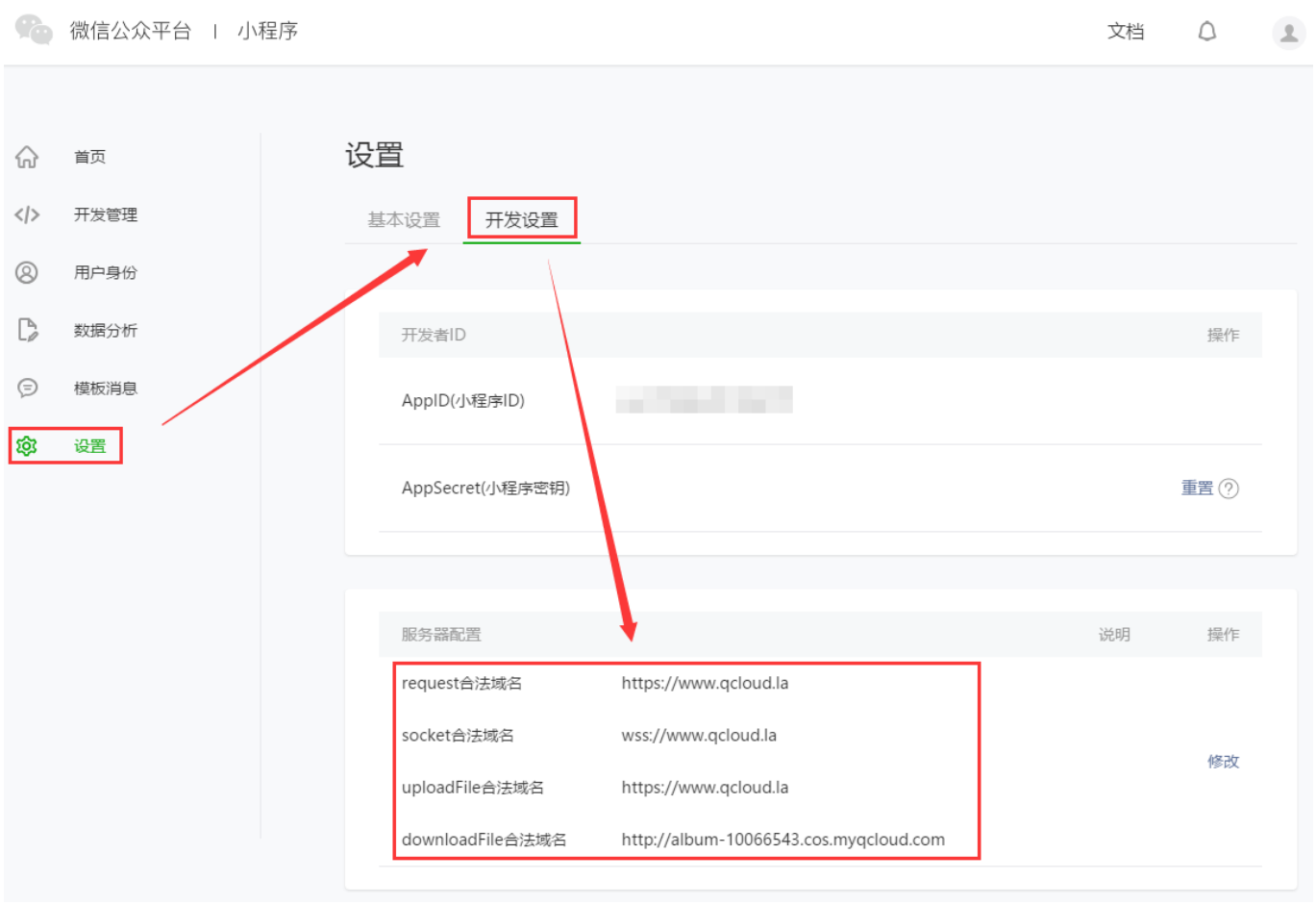
小程序的架构非常简单，这里有两条网络同步，一条是 HTTPS 通路，用于常规请求。对于 WebSocket 请求，会先走 HTTPS 后再切换协议到 WebSocket 的 TCP 连接，从而实现全双工通信。

1. 准备域名和证书

在微信小程序中，所有的网路请求受到严格限制，不满足条件的域名和协议无法请求，具体包括：

- 只允许和在 MP 中配置好的域名进行通信，如果还没有域名，需要注册一个。
- 网络请求必须走 HTTPS 协议，所以你还需要为你的域名申请一个证书。

域名注册好之后，可以登录微信公众平台配置通信域名了。



2. 云主机和镜像部署

剪刀石头布的服务器运行代码和配置已经打包成腾讯云 CVM 镜像，大家可以直接使用。

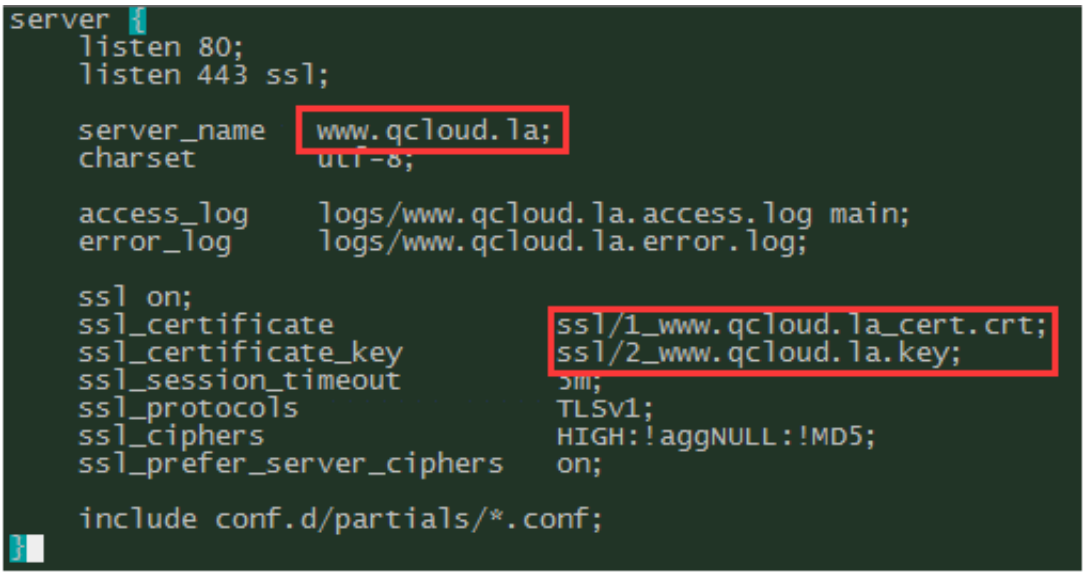
腾讯云用户可以免费领取礼包，体验腾讯云小程序解决方案。

镜像部署完成之后，云主机上就有运行 WebSocket 服务的基本环境、代码和配置了。

> 镜像已包含所有小程序的服务器环境与代码，需要体验其它小程序的朋友无需重复部署

3. 配置 HTTPS

镜像中已经部署了 nginx，需要在 /etc/nginx/conf.d 下修改配置中的域名、证书、私钥。



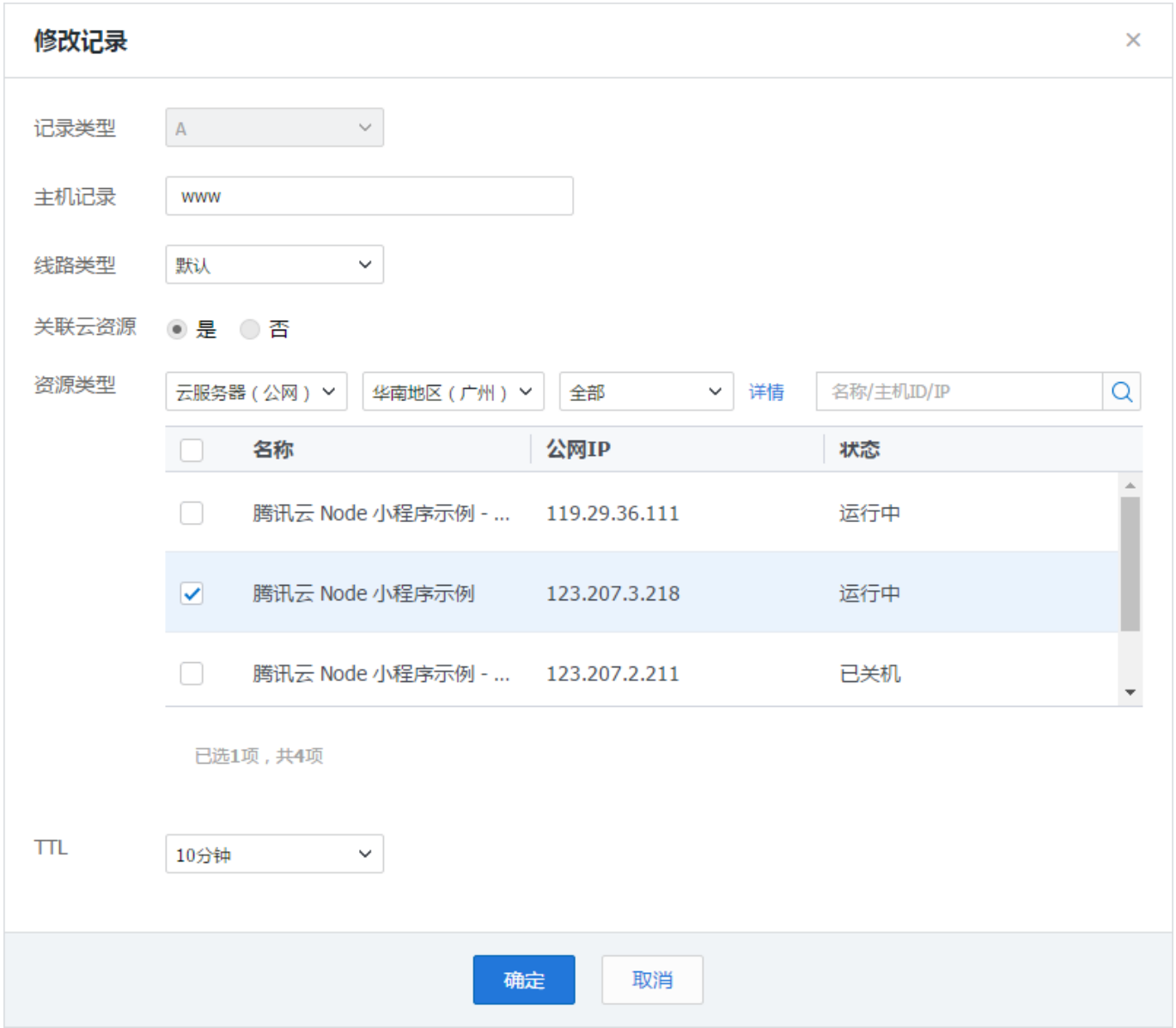
配置完成后，即可启动 nginx。

```
nginx
```

4. 域名解析

我们还需要添加域名记录解析到我们的云服务器上，这样才可以使用域名进行 HTTPS 服务。

在腾讯云注册的域名，可以直接使用云解析控制台来添加主机记录，直接选择上面购买的 CVM。



解析生效后，我们在浏览器使用域名就可以进行 HTTPS 访问。

5. 启动 WebSocket 服务

在镜像的 nginx 配置中（/etc/nginx/conf.d），已经把 /applet/websocket 的请求转发到 http://127.0.0.1:9595 处理。我们需要把 Node 实现的 WebSocket 服务在这个端口里运行起来。

进入镜像中源码位置：

```
cd /data/release/qcloud-applet-websocket
```

使用 pm2 启动服务：

```
pm2 start process.json
```

```
# pm2 start process.json
[PM2][WARN] Applications websocket not running, starting...
[PM2] App [websocket] launched (1 instances)
```

App name	id	mode	pid	status	restart	uptime	cpu	mem	watching
websocket	0	fork	17313	online	0	0s	0%	7.9 MB	enabled

6. 启动微信小程序

在微信开发者工具中修改小程序源码中的 config.js

配置，把通讯域名修改成上面申请的域名。完成后点击调试即可连接到 WebSocket 服务进行游戏。

配置完成后，运行小程序就可以看到成功搭建的提示！

为什么要用 WebSocket

使用传统的 HTTP 轮询或者长连接的方式也可以实现类似服务器推送的效果，但是这类方式都存在资源消耗过大或推送延迟等问题。而 WebSocket 直接使用 TCP 连接保持全双工的传输，可以有效地减少连接的建立，实现真正的服务器通信，对于有低延迟有要求的应用是一个很好的选择。

目前浏览器对 WebSocket 的支持程度已经很好，加上微信小程序的平台支持，这种可以极大提高客户端体验的通信方式将会变得更加主流。

Server 端需要实现 WebSocket 协议，才能支持微信小程序的 WebSocket 请求。鉴于 SocketIO 被广泛使用，剪刀石头布的小程序，我们选用了比较著名的 SocketIO 作为服务端的实现。

Socket IO 的使用比较简单，仅需几行代码就可启动服务。

```
export class Server {

    init(path: string) {
        /** Port that server listen on */
        this.port = process.env.PORT;

        /** HTTP Server instance for both express and socket io */
        this.http = http.createServer();
```

```
        /** Socket io instance */
        this.io = SocketIO(this.http, { path });

        /** Handle incomming connection */
        this.io.on("connection", socket => {
            // handle connection
        });
    }

    start() {
        this.http.listen(this.port);
        console.log(`---- server started. listen : ${this.port} ----`);
    }
}

const server = new Server();
server.init("/applet/ws/socket.io");
server.start();
```

但是，SocketIO 和一些其它的服务器端实现，都有其配套的客户端来完成上层协议的编码解码。但是由于微信的限制（不能使用 window 等对象），SocketIO 的客户端代码在微信小程序平台上是无法运行的。

经过对 SocketIO 通信进行抓包以及研究其客户端源码，笔者封装了一个大约 100 行适用于微信小程序平台的 WxSocketIO 类，可以帮助开发者快速使用 SocketIO 来进行 WebSocket 通信。

```
const socket = new WxSocketIO();
socket.on('hi', packet => console.log('server say hi: ' + packet.message));
socket.emit('hello', { from: 'techird' });
```

如果想要使用微信原生的 API，那么在服务器端也可以直接使用 ws 来实现 W3C 标准的接口。不过 SocketIO 支持多进程的特性，对于后续做横向扩张是很有帮助的。腾讯云在后面也会有计划推出支持大规模业务需求的 WebSocket 连接服务，减小业务的部署成本。

通信协议设计

实现一个多客户端交互的服务，是需要把中间涉及到所有的消息类型都设计清楚的，就像是类似剪刀石头布这样一个小程序，都有下面这些消息类型。

消息	方向	说明
hello	c => s	客户端连上后发送 hello 信息，告知服务器自己身份以及位置。
hi	s => c	服务器响应客户端打招呼，并且反馈附近有多少人
join	c => s	客户端请求加入一个房间进行游戏
leave	c => s	客户端请求退出房间
start	s => c	房间里面全部人都 ready 后，会发送游戏开始的信号，并且告知客户端游戏时间。
choice	c => s	客户端选择出剪刀、石头还是布
face	c => s	客户端更新自己的表情
movement	s => c	有用户更新选择或者更新表情会通知其它用户
result	s => c	超过选择时间后，游戏结束，广播游戏结果

具体每个消息的参数可以参考源码里的 server/protocol.brief.md

服务器逻辑

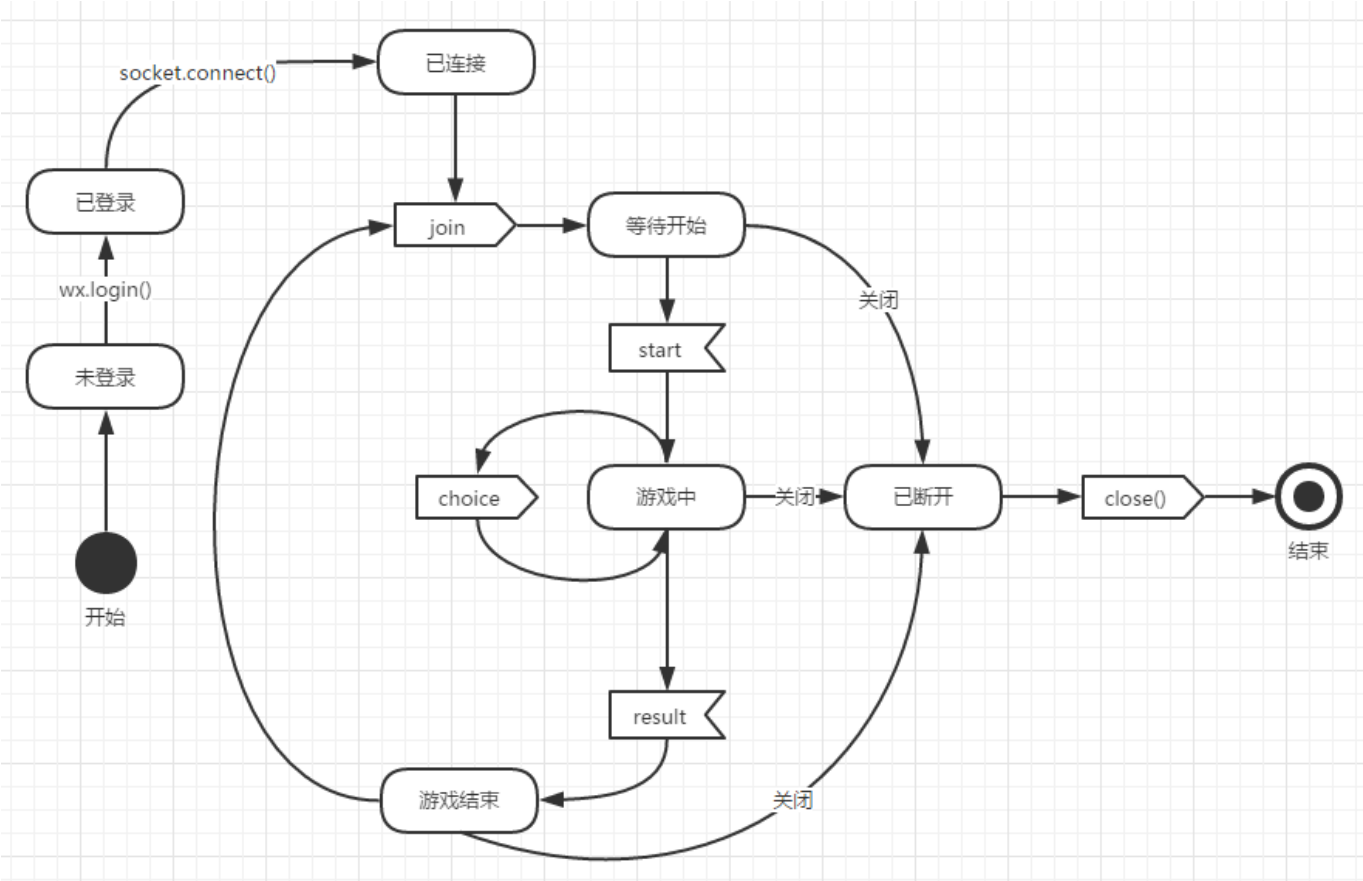
服务器的逻辑很简单：

- 收到用户请求加入房间（join），就寻找还没满的房间
    - 找到房间，则加入

    - 没找到房间，创建新房间
- 有用户加入的房间检查是否已满，如果已满，则：
    - 给房间里每个用户发送开始游戏的信号（start）
    - 启动计时器，计时器结束后进行游戏结算
- 游戏结算
    - 两两之间 PK，赢方分数加一，输方减一，最终得每个玩家基本得分 x
    - 对于每个玩家，如果分数 x 大于 0，则视为胜利，连胜次数加一，否则连胜次数归零
    - 本局得分为分数 x 乘以连胜次数
- 发送本局游戏结果给房间里的每位玩家

微信端实现

微信小程序直接使用上面的协议，针对不同的场景进行渲染。整体的状态机如下。



状态机整理清楚后，就是根据状态机来控制什么时候发送消息，接到消息后如何处理的问题了。具体实现请参照 app/pages/game/game.js 里的源码。

会话管理场景

微信小程序示例 - 一笔到底

微信小程序提供了一套在微信上运行小程序的解决方案，有比较完整的框架、组件以及API，在这个平台上面的想象空间很大。

微信的定位并不是 HTML5，这里很多人都有误解。在一些实现上，并不能想当然地用 HTML5 的思路来思考。比如，微信的请求接口 wx.request 并不支持 cookie 传递，所以会话层不能使用传统的 Session 方式。

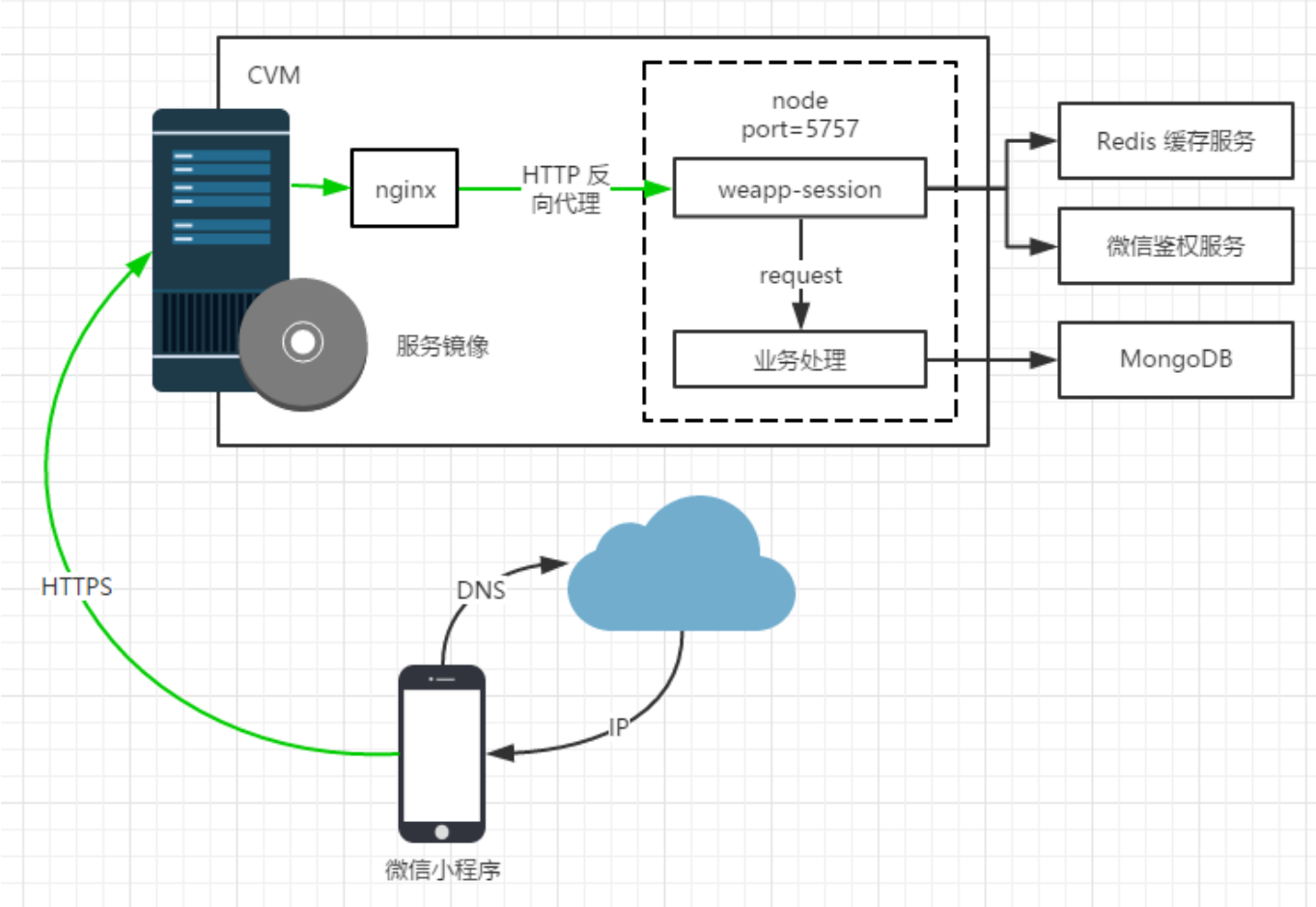
这篇文章分享一个简单的画图应用，使用自己新鲜出炉的小程序会话管理能力来判断并当前用户的身份。

小程序非常简单，使用 Canvas 绘图后，把序列化的 actions
提交给服务器保存。下次加载的时候，再列出用户曾经绘制过的图。

部署和运行

拿到了本小程序源码的朋友可以尝试自己运行起来。

整体架构

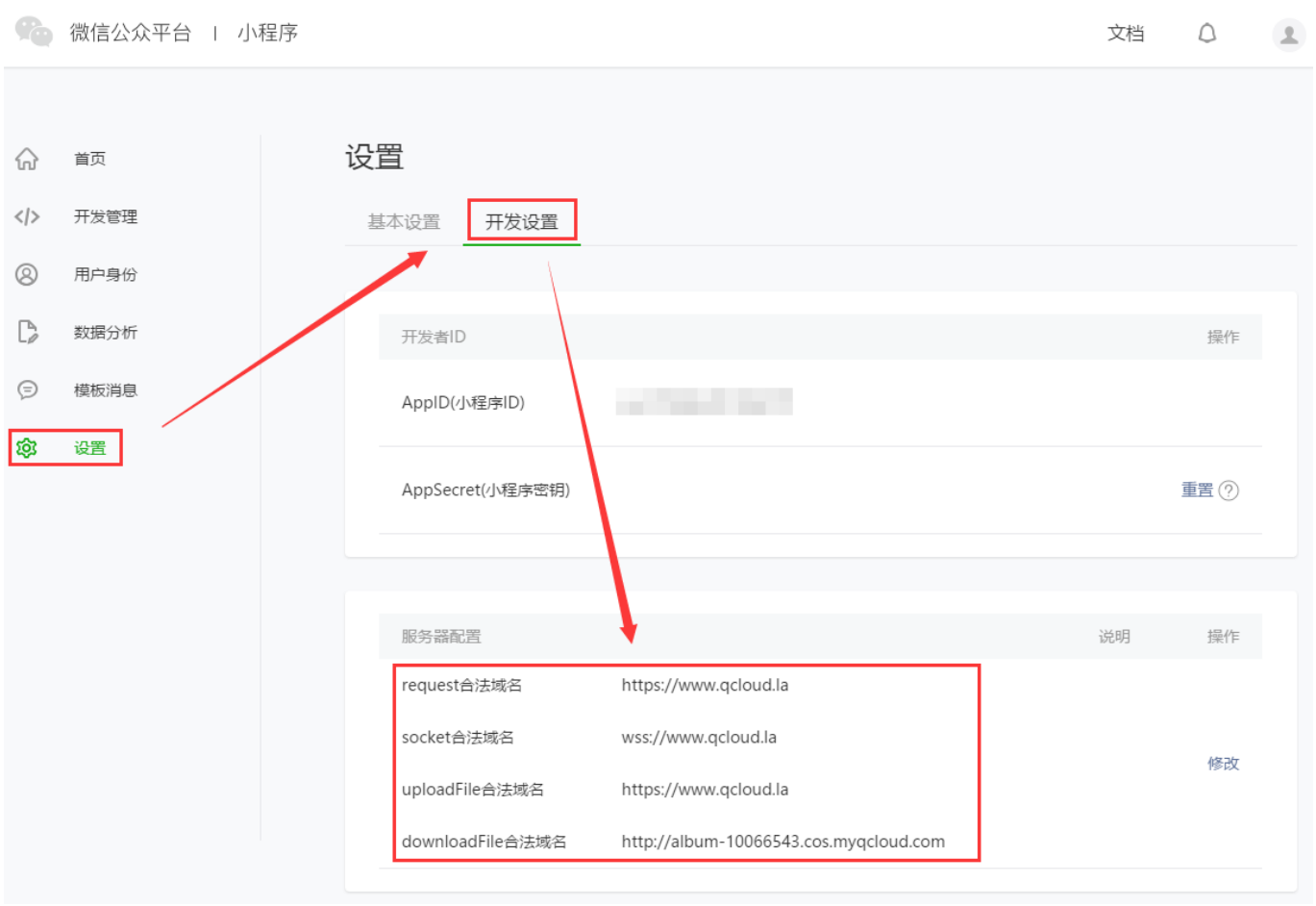


1. 准备域名和证书

在微信小程序中，所有的网路请求受到严格限制，不满足条件的域名和协议无法请求，具体包括：

- 只允许和在 MP 中配置好的域名进行通信，如果还没有域名，需要注册一个。
- 网络请求必须走 HTTPS 协议，所以你还需要为你的域名申请一个证书。

域名注册好之后，可以登录微信公众平台配置通信域名了。



2. 云主机和镜像部署

一笔到底的服务器运行代码和配置已经打包成腾讯云 CVM 镜像，大家可以直接使用。

腾讯云用户可以免费领取礼包，体验腾讯云小程序解决方案。

镜像已包含所有小程序的服务器环境与代码，需要体验其它小程序的朋友无需重复部署

3. 配置 HTTPS

镜像中已经部署了 nginx，需要在 /etc/nginx/conf.d 下修改配置中的域名、证书、私钥。

```
server {
    listen 80;
    listen 443 ssl;

    server_name  www.qcloud.la;
    charset      utf-8;

    access_log   logs/www.qcloud.la.access.log main;
    error_log    logs/www.qcloud.la.error.log;

    ssl on;
    ssl_certificate             ssl/1_www.qcloud.la_cert.crt;
    ssl_certificate_key         ssl/2_www.qcloud.la.key;
    ssl_session_timeout         5m;
    ssl_protocols               TLSv1;
    ssl_ciphers                 HIGH:!aggNULL:!MD5;
    ssl_prefer_server_ciphers   on;

    include conf.d/partials/*.conf;
}
```

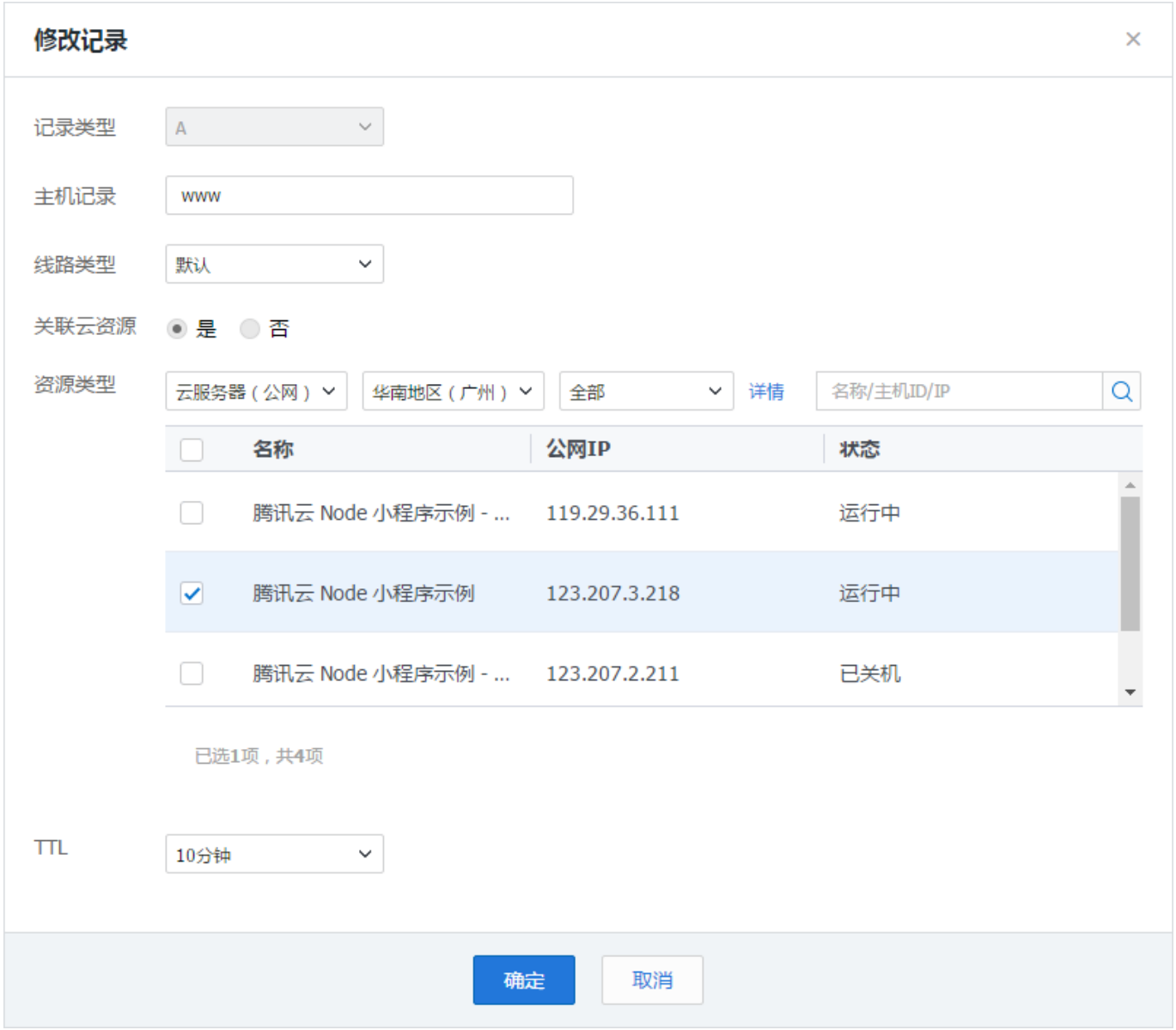
配置完成后，即可启动 nginx。

```
nginx
```

4. 域名解析

我们还需要添加域名记录解析到我们的云服务器上，这样才可以使用域名进行 HTTPS 服务。

在腾讯云注册的域名，可以直接使用云解析控制台来添加主机记录，直接选择上面购买的 CVM。



解析生效后，我们在浏览器使用域名就可以进行 HTTPS 访问。

5. 配置云存储 Redis

会话管理依赖 Redis 进行作为缓存管理，开发者可以选择自行搭建 Redis 服务或者直接购买云存储 Redis 服务。

6. 配置云数据库 MongoDB

一笔到底小程序使用 MongoDB 来存储用户绘制的图像路径，要运行小程序开发者需要自行搭建 MongoDB 服务或者直接购买云数据库 MongoDB。

7. 启动一笔到底示例 Node 服务

在镜像的 nginx 配置中（/etc/nginx/conf.d），已经把/applet/session的请求转发到http://127.0.0.1:5757处理。我们需要把 Node 服务运行起来。Node 代码部署在目录/data/release/qcloud-applet-session下。

进入该目录：

```
cd /data/release/qcloud-applet-session
```

在该目录下有个名为config.js的配置文件（如下所示），根据注释将appId、appSecret、redisConfig、mongoConfig修改成自己的配置。

```
module.exports = {
    port: '5757',
    ROUTE_BASE_PATH: '/applet',

    // □□□□□ App ID
    appId: '',

    // □□□□□ App Secret
    appSecret: '',

    // Redis □□
    // @see https://www.npmjs.com/package/redis#options-object-properties
    redisConfig: {
        host: '',
        port: '',
        password: '',
    },

    // MongoDB □□
    // @see https://www.qcloud.com/doc/product/240/3979
```

```
    mongoConfig: {
        username: '',
        password: '',
        host: '',
        port: '',
        query: '?authMechanism=MONGODB-CR&authSource=admin',
        database: 'qcloud-applet-session',
    },
};
```

一笔到底示例使用 pm2 管理 Node 进程，执行以下命令启动 node 服务：

```
pm2 start process.json
```

实现

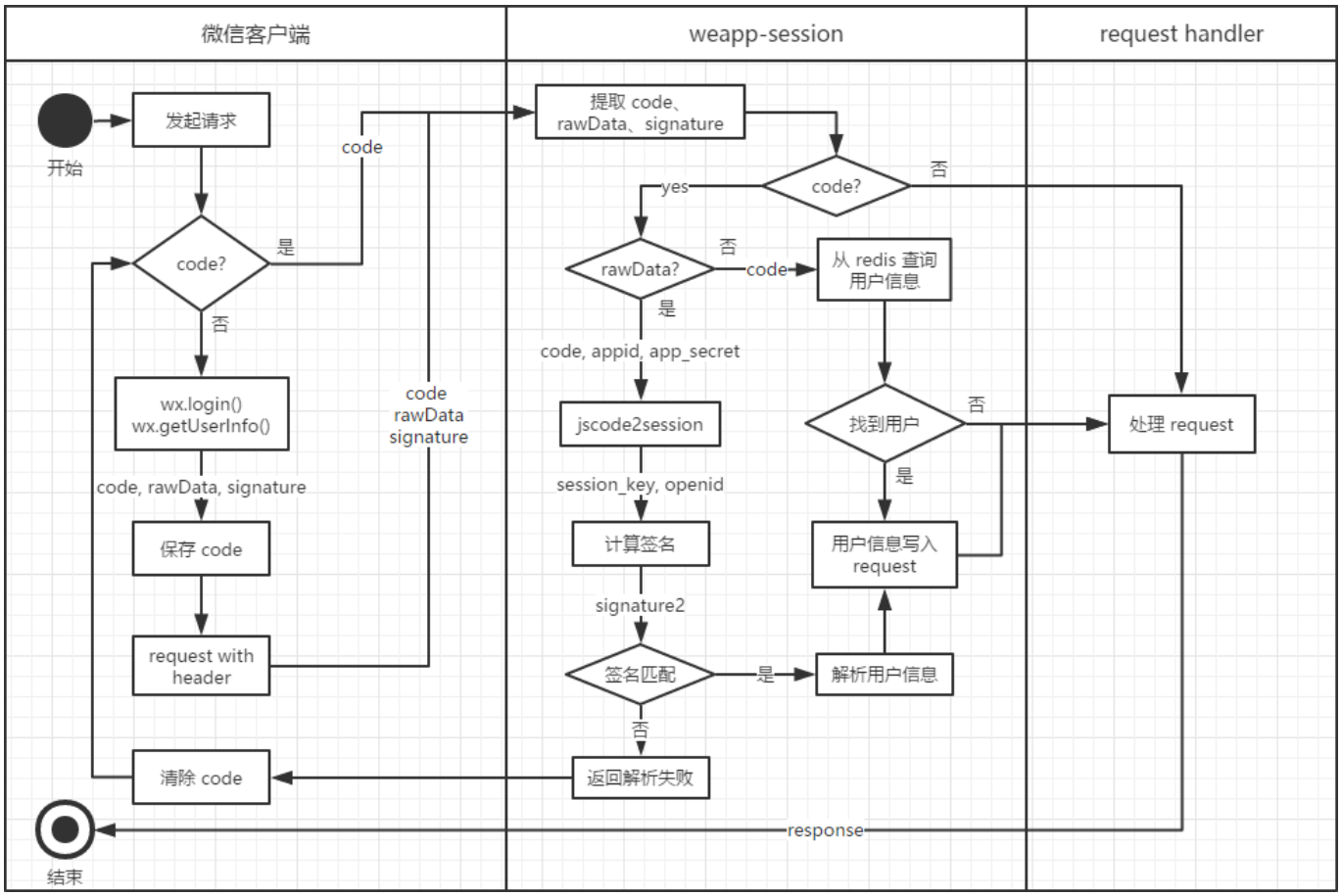
会话层实现

会话层实现包含两个部分：

- 服务器端：https://github.com/CFETeam/weapp-session
- 客户端：https://github.com/CFETeam/weapp-session-client

我们的 Demo 直接使用这两个仓库，可以快速地拥有会话层的能力。

会话层的实现和传统 Cookie 的实现方式类似，都是在 Header
上使用特殊的字段跟踪。一个请求的完整流程如下：



1. 客户端（微信小程序）发起请求 request
2. [weapp-session-client](#) 包装 request
    - 首次请求
        - 调用 wx.login() 和 wx.getUserInfo() 接口获得 code、rawData 和 signature
        - requset 的头部带上 code、rawData 和 signature
        - 保存 code 供下次调用
    - 非首次请求
        - request 的头部带上保存的 code
3. 服务器收到请求 request，中间件从头部提取 code、rawData 和 signature 字段
    - 如果 code 为空，跳到第 4 步
    - 如果 code 不为空，且 rawData 不为空，需要进行签名校验
        - 使用 code，appid、app_secret 请求微信接口获得 session_key 和 openid
            - 如果接口失败，响应 ERR_SESSION_KEY_EXCHANGE_FAILED
        - 使用签名算法通过 rawData 和 session_key 计算签名 signature2
        - 对比 signature 和 signature2
            - 签名一致，解析 rawData 为 wxUserInfo
                - 把 openid 写入到 wxUserInfo

        - 把 (code, wxUserInfo) 缓存到 Redis
        - 把 wxUserInfo 存放在 request.$wxUserInfo 里
        - 跳到第 4 步
      - 签名不一致，响应 ERR_UNTRUSTED_RAW_DATA
    - 如果 code 不为空，但 rawData 为空，从 Redis 根据 code 查询缓存的用户信息
      - 找到用户信息，存放在 request.$wxUserInfo 字段里，跳到第 4 步
      - 没找到用户信息（可能是过期），响应 ERR_SESSION_EXPIRED
4. request 被业务处理，可以使用 request.$wxUserInfo 来获取用户信息（request.$wxUserInfo 可能为空，业务需要自行处理）

源代码

可从 Github 获取 https://github.com/CFETeam/weapp-session

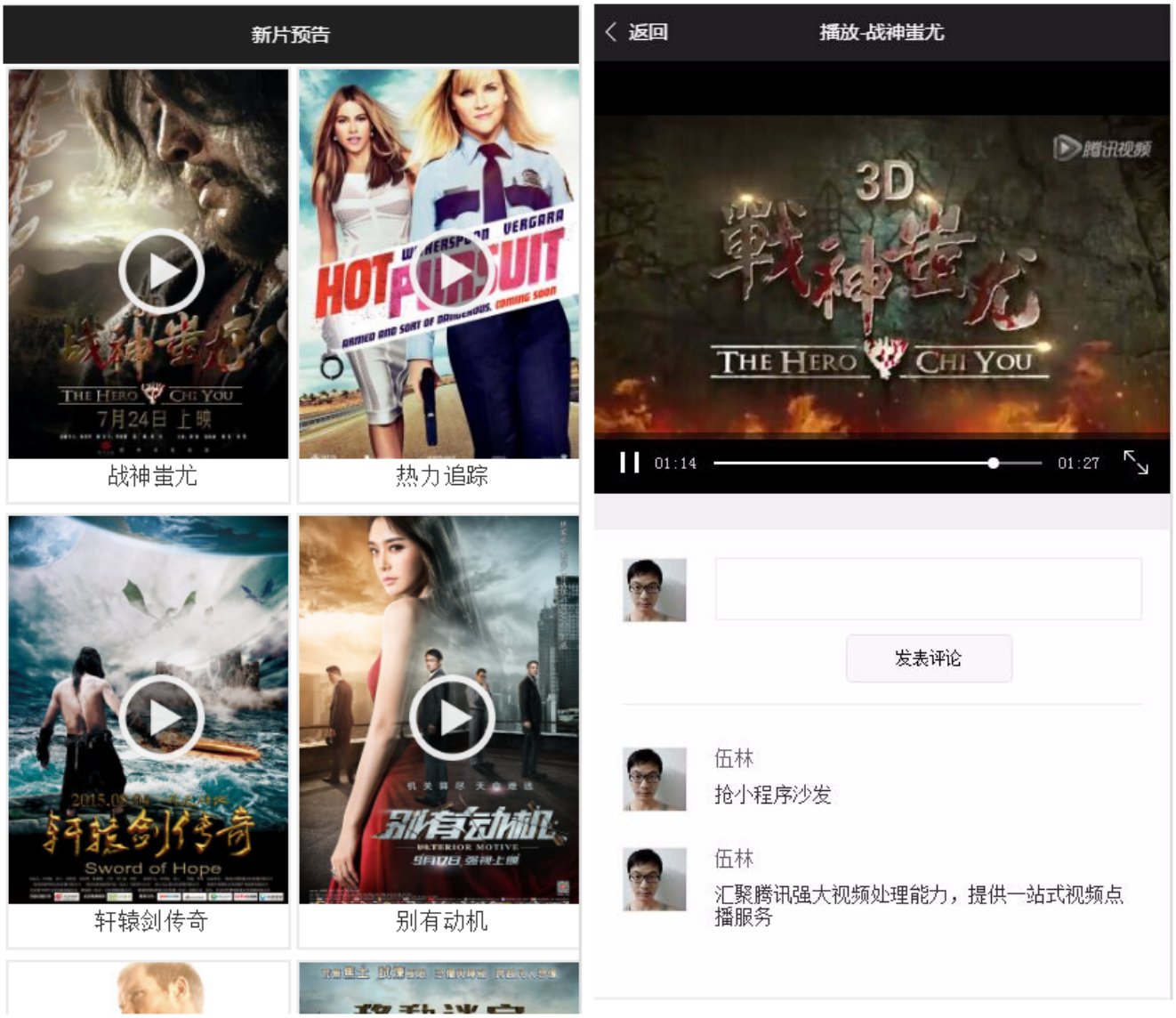
视频应用场景

新片预告是结合腾讯云点播 VOD和云数据库 MySQL
制作的一个微信小程序示例。在代码结构上包含如下两部分：

- app: 新片预告应用包代码，可直接在微信开发者工具中作为项目打开
- server: 搭建的Node服务端代码，作为服务器和app通信，提供 CGI
  接口示例用于拉取云数据库上的视频列表、评论列表，将评论数据提交到云数据库

新片预告主要功能如下：

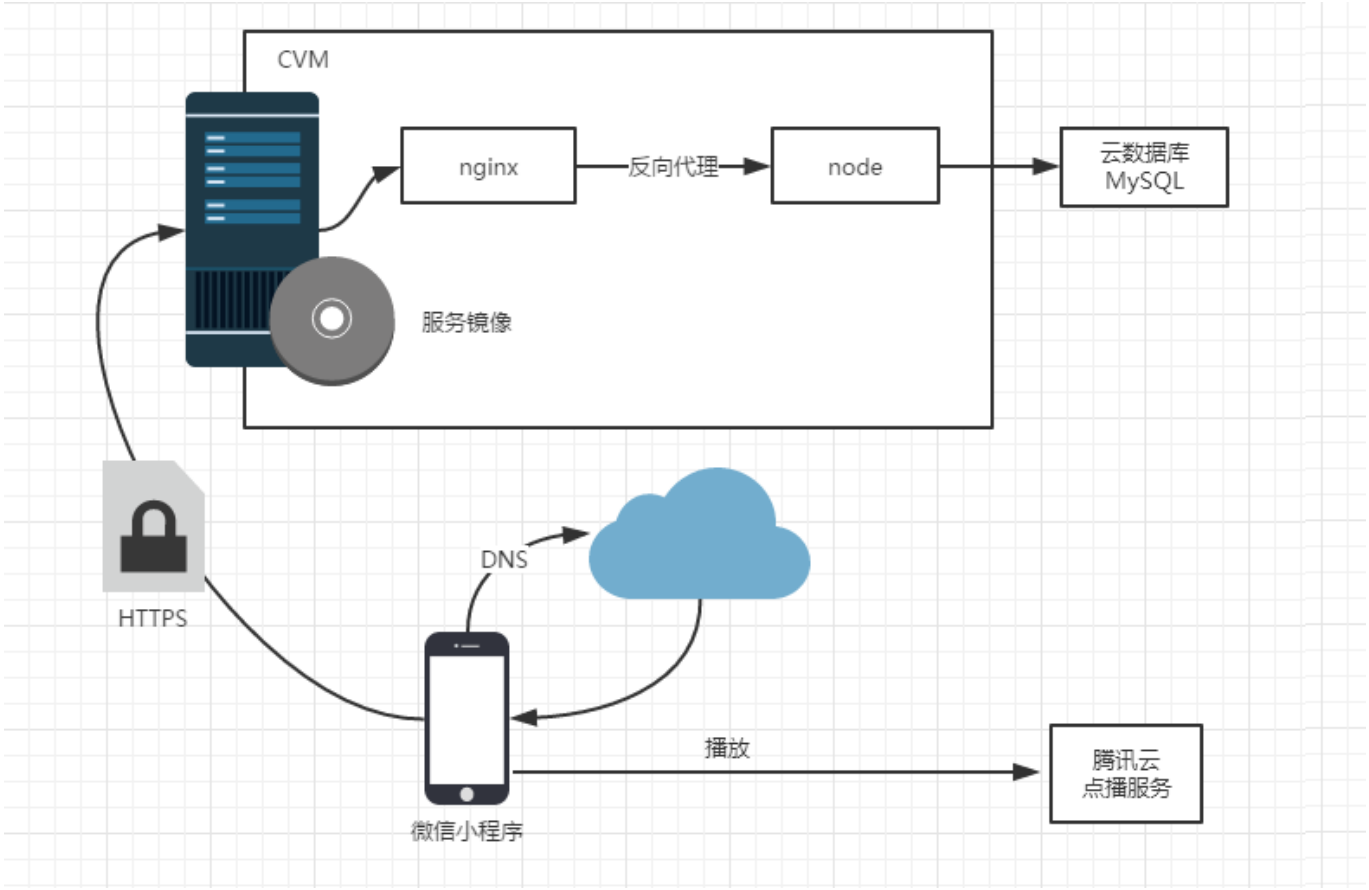
- 支持分页滚动加载视频列表
- 点击海报跳转至详情页播放视频
- 对视频进行评论
- 展示视频的评论列表



部署和运行

拿到了本小程序源码的朋友可以尝试自己运行起来。

整体架构



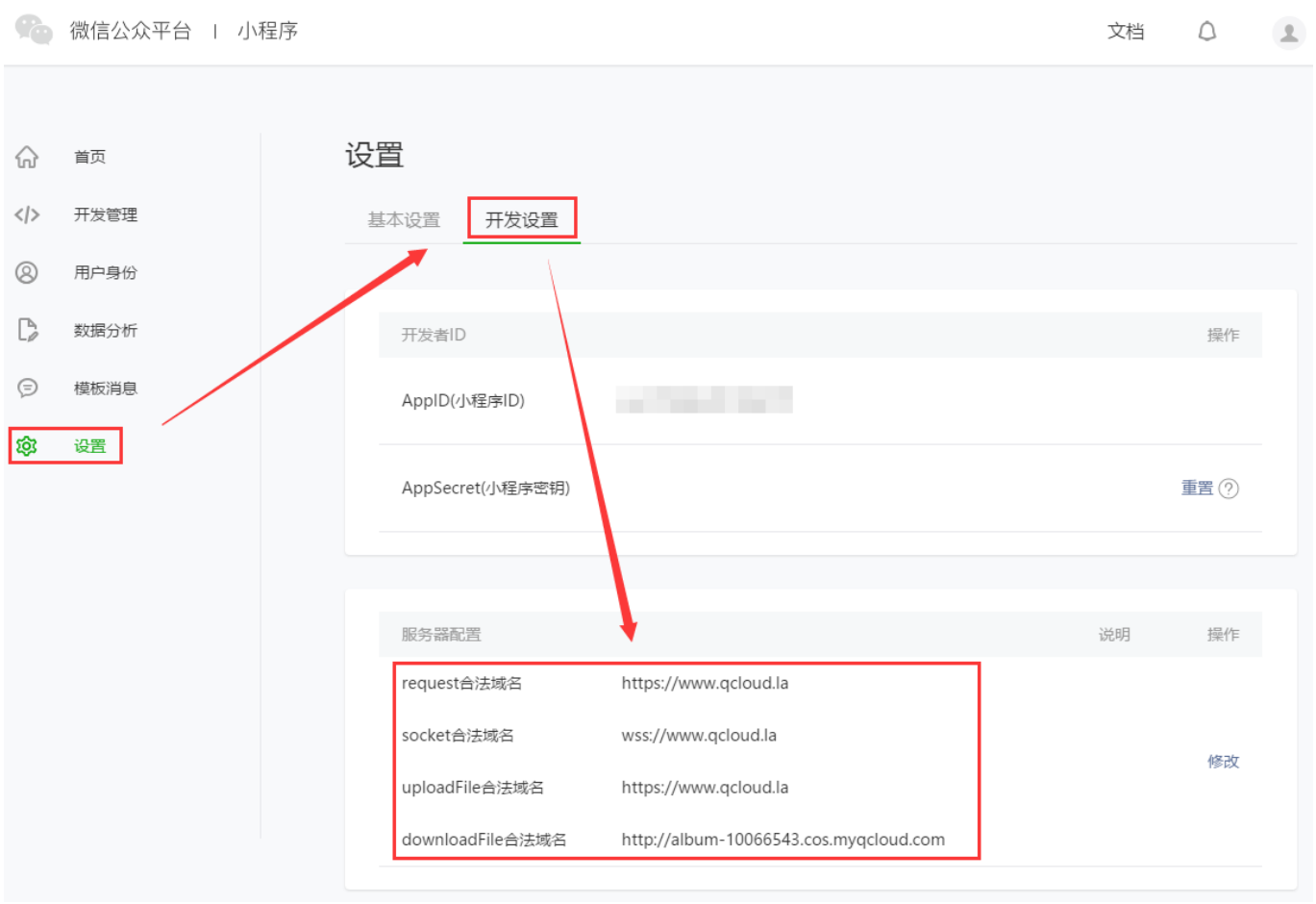
1. 准备域名和证书

在微信小程序中，所有的网路请求受到严格限制，不满足条件的域名和协议无法请求，具体包括：

- 只允许和在 MP 中配置好的域名进行通信，如果还没有域名，需要注册一个。
- 网络请求必须走 HTTPS 协议，所以你还需要为你的域名申请一个 SSL 证书。

腾讯云提供域名注册和证书申请服务，还没有域名或者证书的可以去使用

域名注册好之后，可以登录微信公众平台配置通信域名了。



注意：需要将 www.qcloud.la 设置为上面申请的域名

2. Nginx 和 Node 代码部署

小程序服务要运行，需要进行以下几步：

- 部署 Nginx，Nginx 的安装和部署请大家自行搜索（注意需要把 SSL 模块也编译进去）
- 配置 Nginx 反向代理到 http://127.0.0.1:9994
- Node 运行环境，可以安装 Node V6.6.0
- 部署 server 目录的代码到服务器上，如 /data/release/qcloud-applet-video
- 使用 npm install 安装依赖模块
- 使用 npm install pm2 -g 安装 pm2

上述环境配置比较麻烦，新片预告的服务器运行代码和配置已经打包成腾讯云 CVM 镜像，推荐大家直接使用。

- 镜像部署完成之后，云主机上就有运行 WebSocket 服务的基本环境、代码和配置了。

- 腾讯云用户可以免费领取礼包，体验腾讯云小程序解决方案。
- 镜像已包含所有小程序的服务器环境与代码，需要体验小程序的朋友无需重复部署

3. 配置 HTTPS

镜像中已经部署了 nginx，需要在 /etc/nginx/conf.d 下修改配置中的域名、证书、私钥。

```
server {
    listen 80;
    listen 443 ssl;

    server_name   www.qcloud.la;
    charset       utf-8;

    access_log    logs/www.qcloud.la.access.log main;
    error_log     logs/www.qcloud.la.error.log;

    ssl on;
    ssl_certificate             ssl/1_www.qcloud.la_cert.crt;
    ssl_certificate_key         ssl/2_www.qcloud.la.key;
    ssl_session_timeout         5m;
    ssl_protocols               TLSv1;
    ssl_ciphers                 HIGH:!aggNULL:!MD5;
    ssl_prefer_server_ciphers   on;

    include conf.d/partials/*.conf;
}
```

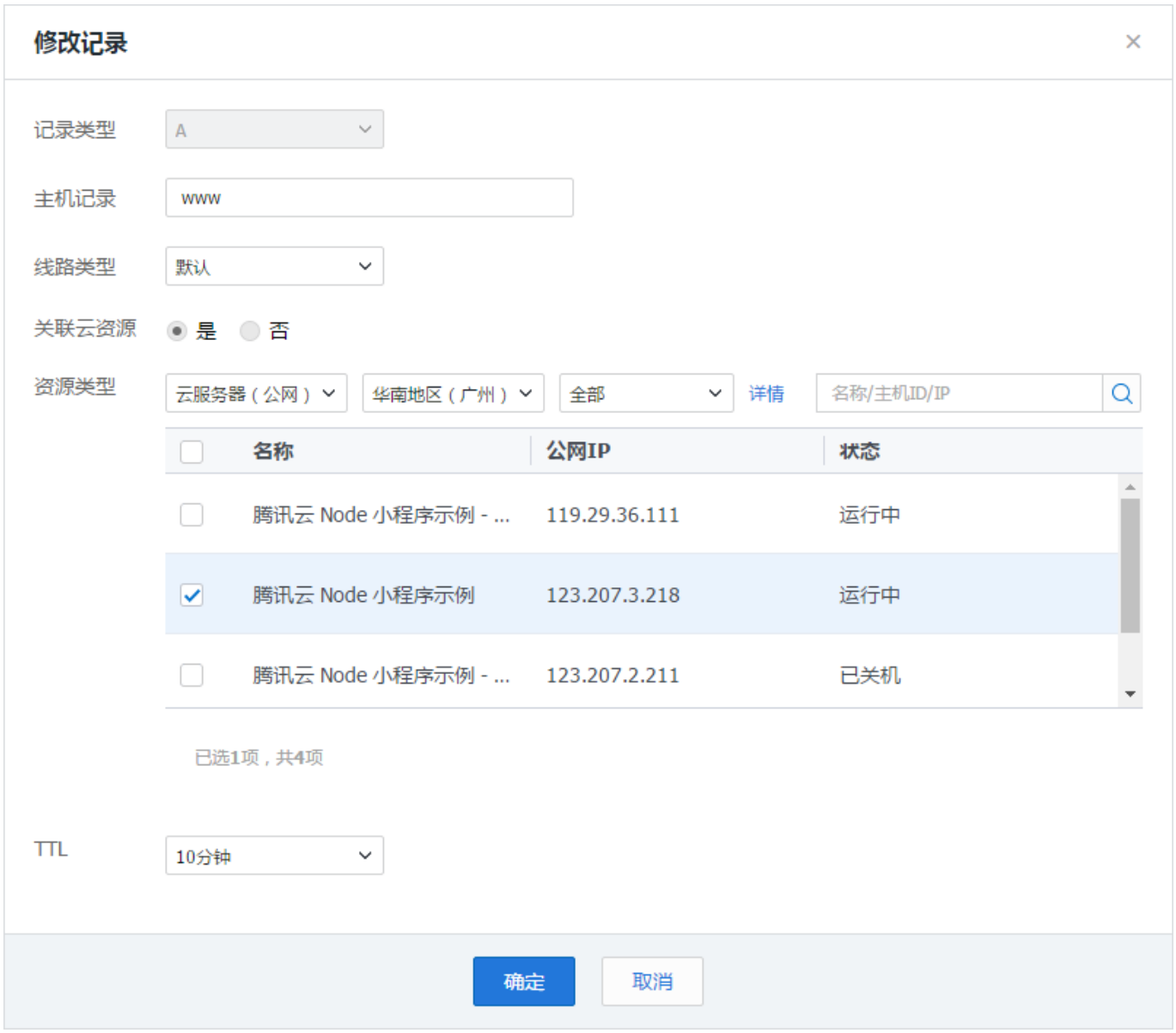
配置完成后，即可启动 nginx。

```
nginx
```

4. 域名解析

我们还需要添加域名记录解析到我们的云服务器上，这样才可以使用域名进行 HTTPS 服务。

在腾讯云注册的域名，可以直接使用云解析控制台来添加主机记录，直接选择上面购买的 CVM。



解析生效后，我们在浏览器使用域名就可以进行 HTTPS 访问。

5. 开通 点播服务

新片预告示例的播放资源是存储在 腾讯云点播 上的mp4文件，要使用 点播 服务，需要登录 点播 管理控制台 ，然后在其中完成以下操作：

- 上传视频资源，点播几乎支持所有主流的视频格式上传
- 转码成功后获取mp4或m3u8源地址

目前微信小程序video组件经测试支持mp4和m3u8格式，其中 m3u8 格式只能在手机上使用，开发者可以使用腾讯云点播控制台将视频源转码成 mp4 或 m3u8 格式，并且腾讯云点播会对播放的资源进行CDN加速。

6. 准备 云数据库MySQL

示例中拉取的视频和评论列表都是存储在 云数据库 上，要使用 云数据库 服务需要完成以下操作

- 购买，注意购买的云数据库需要与云服务器同在一个地域分区
- 初始化流程，本示例选用的是utf8编码
- 点击云数据库 控制台
  操作栏的登录按钮，登录到phpMyAdmin创建数据库并在当前数据库中导入本示例中的SQL文件

  注意：导入SQL文件中包含了 点播 上传的视频列表，开发者可以基于云数据库自行开发维护一个视频发布管理系统，因为此内容跟本示例暂不相关，所以不再详述。

7. 启动新片预告示例 Node 服务

在镜像中，新片预告示例的 Node 服务代码已部署在目录/data/release/qcloud-applet-video下：

进入该目录：

```
cd /data/release/qcloud-applet-video
```

在该目录下有个名为config.js的配置文件（如下所示），按注释修改对应的 MySQL 配置：

```
module.exports = {
    // Node 监听的端口号
    port: '9994',
    ROUTE_BASE_PATH: '/applet',

    host: '填写开通 MySQL 时分配的内网IP',
    user: '填写MySQL用户名',
    password: '填写MySQL密码',
    database: '填写上一步中创建的MySQL数据名',
};
```

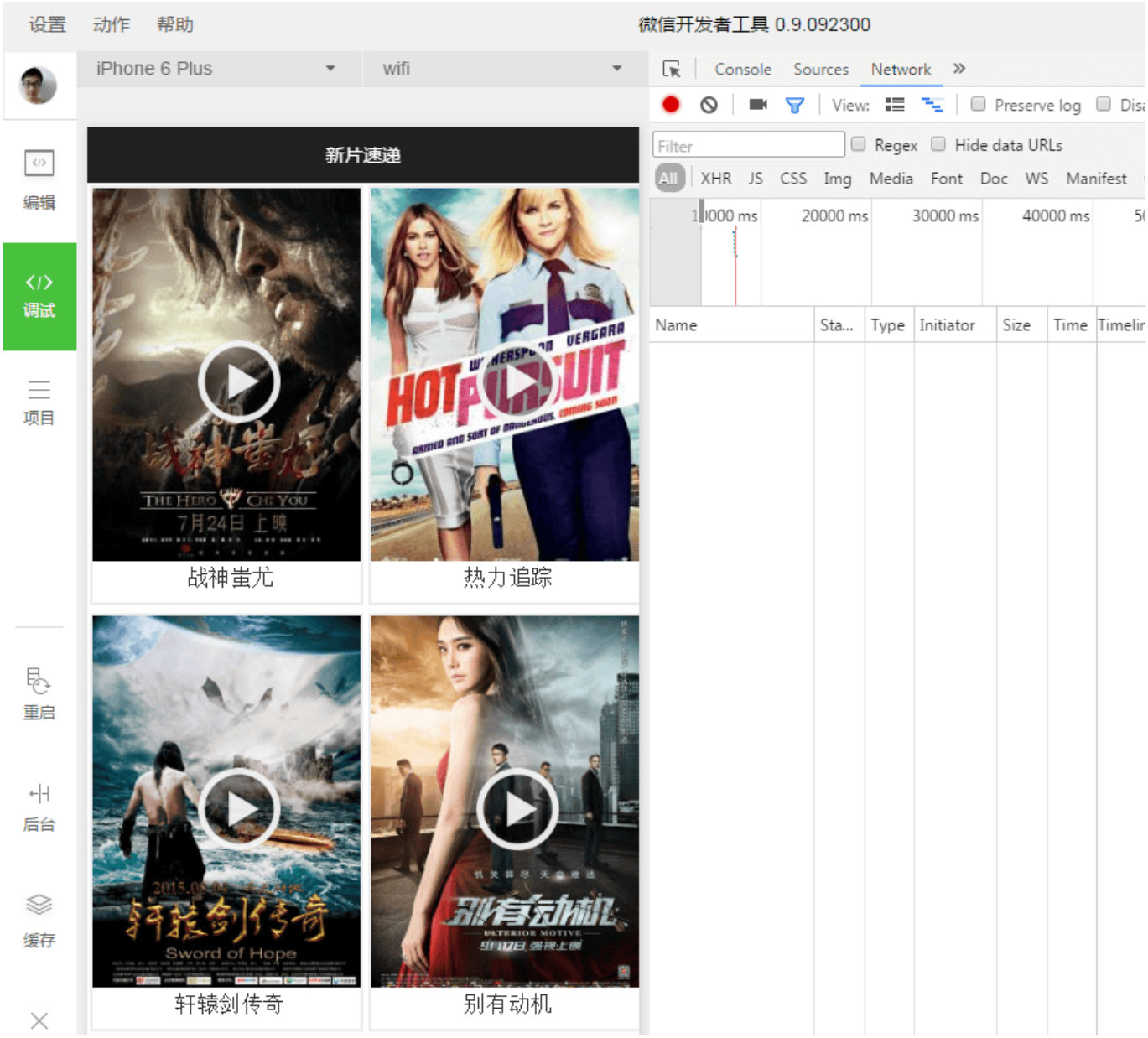
示例使用pm2管理 Node 进程，执行以下命令启动 node 服务：

```
pm2 start process.json
```

8. 启动新片预告 Demo

在微信开发者工具将新片预告应用包源码添加为项目，并把源文件config.js中的通讯域名修改成上面申请的域名。

然后点击调试即可打开新片预告Demo开始体验。



主要功能实现

获取视频列表、展示评论、提交评论

通过node的mysql模块连接mysql，进行查询，插入操作

以下是查询评论列表的示例代码

```
const mysql = require('mysql');
const config = require('../../../config');

let vid = this.req.query.vid;
```

```
if (!vid) {
    this.res.json({ code: -1, msg: 'failed', data: {} });
    return;
}

//CDB Mysql□□
let connection = mysql.createConnection({
    host: config.host,
    password: config.password,
    user: config.user,
    database: config.database
});

//□□□□□□□
connection.connect((err) => {
    if (err) {
        this.res.json({ code: -1, msg: 'failed', data: {} });
    }
});

//□□□□
connection.query(
'SELECT * from comment whe
re vid = ? order by id desc', [vid], (err, result) => {
    if (err) {
        this.res.json({ code: -1, msg: 'failed', data: {} });
        return;
    }

    this.res.json({
        code: 0,
        msg: 'ok',
        data: result,
    });
```

```
});

//□□□□□□□□□
connection.end();
```

播放视频

```
<video src="{{videoUrl}}" binderror="videoErrorCallback"></video>
```

属性名	类型	说明
src	String	要播放视频的资源地址
binderror	EventHandle	当发生错误时触发error事件，event.detail = {errMsg: 'something wrong'}

播放视频使用的是video标签，目前官方文档上只给出了两个参数说明，笔者测试了src支持加载mp4和m3u8格式视频，
video标签的控制条暂时没办法自定义样式以及隐藏