OpenResty 在直播场景的应用

罗玉杰

lyj_william@126.com

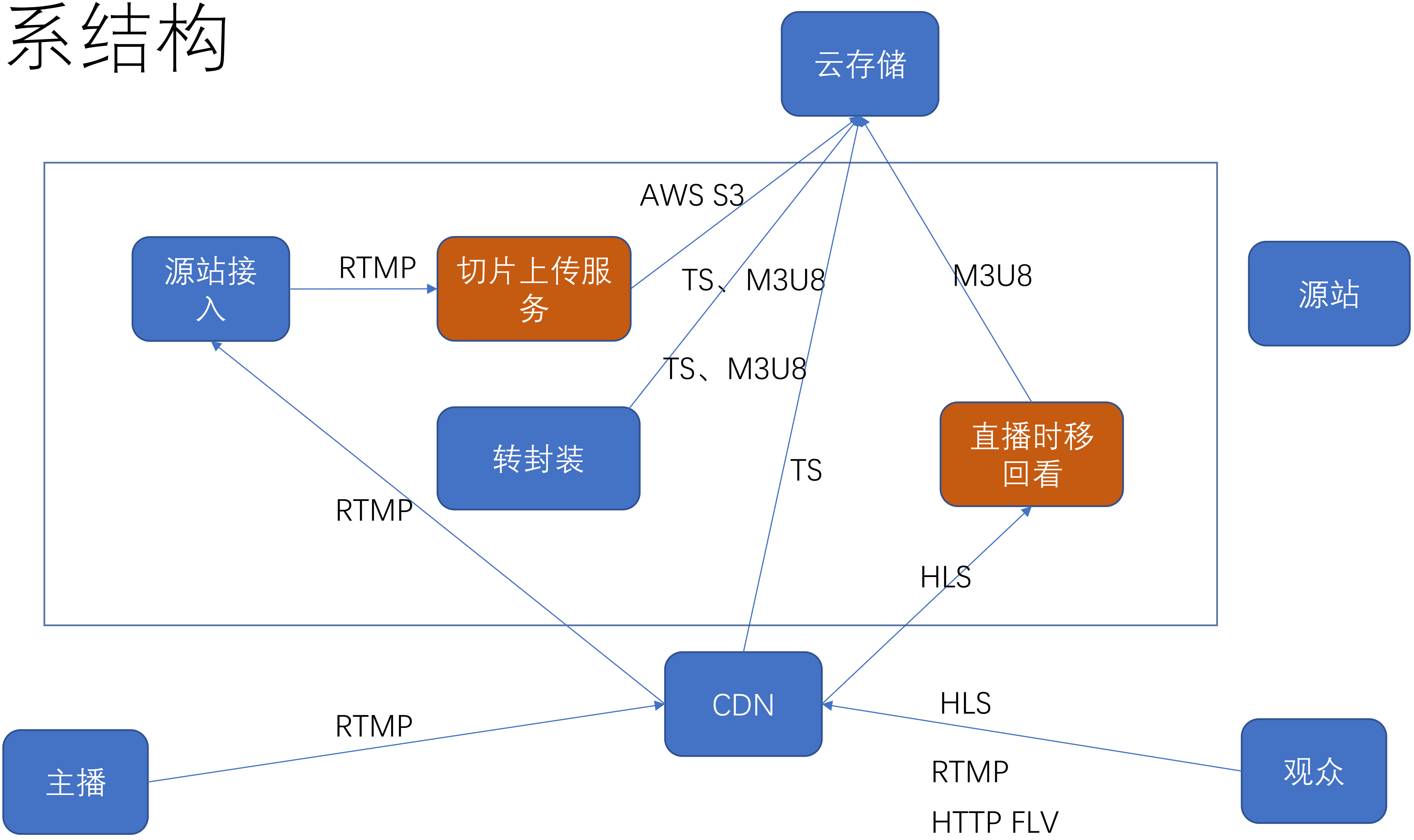
大纲

- 需求、背景
- 体系结构
- 功能
- 问题及优化

项目需求

- 视频数据上云
- 提供基础数据 （转封装、转码、点播服务）
- 提供HLS 服务

体系结构


云存储
AWS S3
源站接入
RTMP
切片上传服务
TS、M3U8
M3U8
源站
TS、M3U8
转封装
RTMP
TS
直播时移回看
HLS
CDN
HLS
RTMP
主播
RTMP
HTTP FLV
观众

功能

- 基于s3 PUT 将TS文件上传至云存储
- 基于S3 multi 分片上传大文件，支持断点续
- 基于redis实现任务队列、存储任务元数据、点播M3U8
- Nginx worker负载调度，后端服务保护，连接，请求量控制
- 失败重试，异常处理，降级策略
- 生成直播、点播M3U8，上传云存储
- 直播时移回看服务

AWS S3

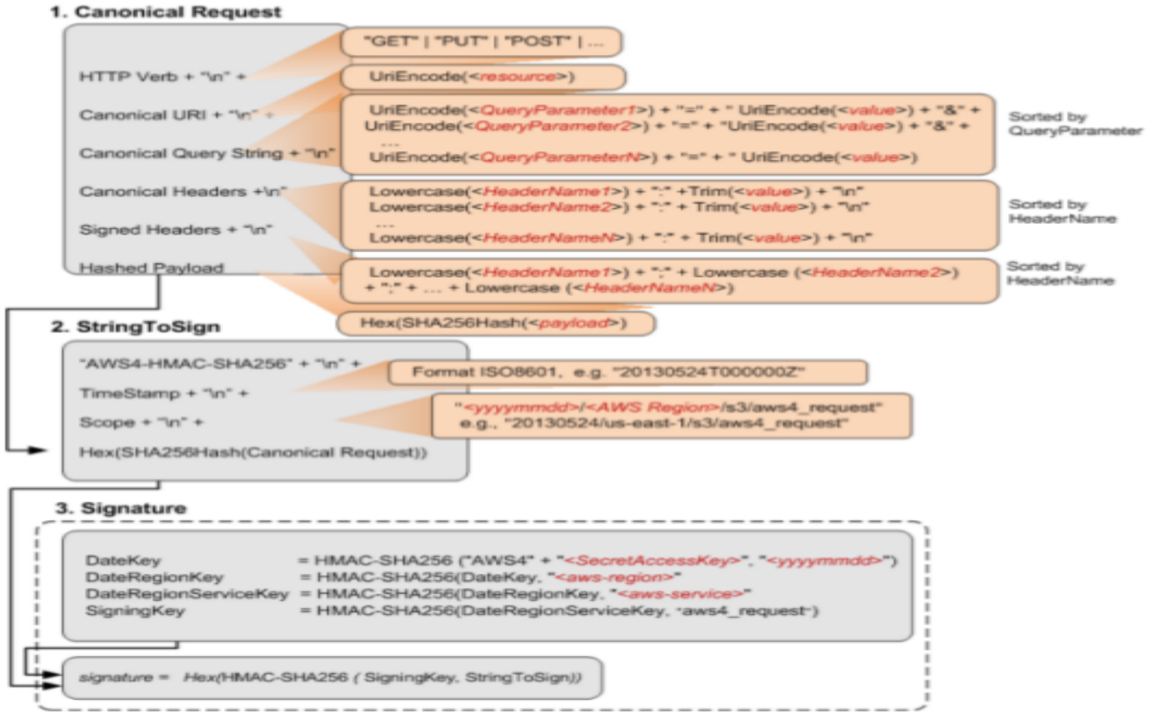
- 鉴权
- PUT
- MULTI PART

鉴权

Calculating a Signature

To calculate a signature, you first need a string to sign. You then calculate a `HMAC-SHA256` hash of the string to sign by using a signing key. The following diagram illustrates the process, including the various components of the string that you create for signing

When Amazon S3 receives an authenticated request, it computes the signature and then compares it with the signature that you provided in the request. For that reason, you must compute the signature by using the same method that is used by Amazon S3. The process of putting a request in an agreed-upon form for signing is called canonicalization.



1. Canonical Request

HTTP Verb + "\n" +
Canonical URI + "\n" +
Canonical Query String + "\n"
Canonical Headers +\n"
Signed Headers + "\n"
Hashed Payload

"GET" | "PUT" | "POST" | ...

UriEncode(<resource>)

UriEncode(<QueryParameter1>) + "=" + " UriEncode(<value>) + "&" +
UriEncode(<QueryParameter2>) + "=" + "UriEncode(<value>) + "&" +
...
UriEncode(<QueryParameterN>) + "=" + " UriEncode(<value>)

Sorted by QueryParameter

Lowercase(<HeaderName1>) + ":" +Trim(<value>) + "\n"
Lowercase(<HeaderName2>) + ":" + Trim(<value>) + "\n"
...
Lowercase(<HeaderNameN>) + ":" + Trim(<value>) + "\n"

Sorted by HeaderName

Lowercase(<HeaderName1>) + ";" + Lowercase (<HeaderName2>)
+ ";" + ... + Lowercase (<HeaderNameN>)

Sorted by HeaderName

Hex(SHA256Hash(<payload>)

2. StringToSign

"AWS4-HMAC-SHA256" + "\n" +
TimeStamp + "\n" +
Scope + "\n" +
Hex(SHA256Hash(Canonical Request))

Format ISO8601, e.g. "20130524T000000Z"

"<yyyymmdd>/<AWS Region>/s3/aws4_request"
e.g., "20130524/us-east-1/s3/aws4_request"

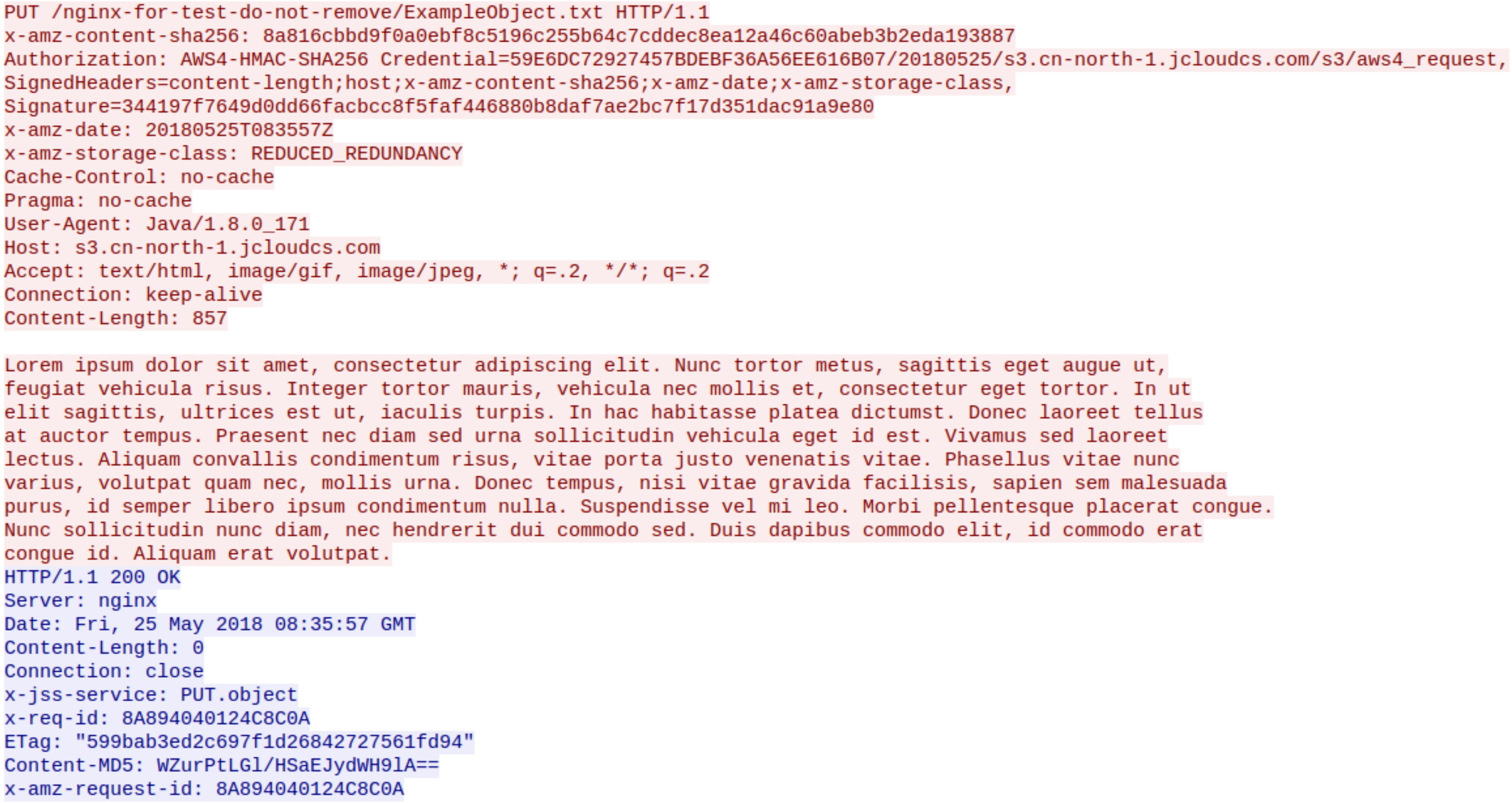
3. Signature

```
DateKey                = HMAC-SHA256 ("AWS4" + "<SecretAccessKey>", "<yyyymmdd>")
DateRegionKey          = HMAC-SHA256(DateKey, "<aws-region>"
DateRegionServiceKey   = HMAC-SHA256(DateRegionKey, "<aws-service>"
SigningKey             = HMAC-SHA256(DateRegionServiceKey, "aws4_request")
```

*signature = Hex(*HMAC-SHA256 (SigningKey, StringToSign))

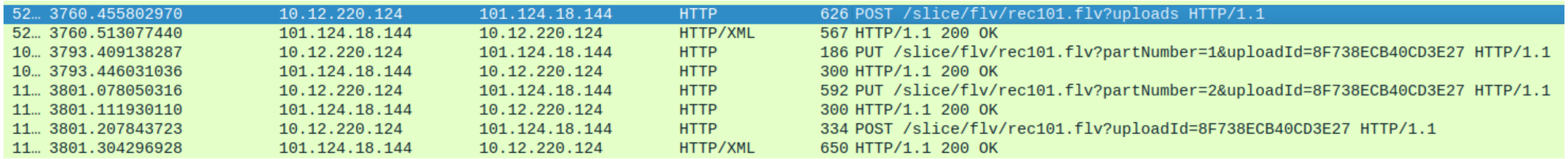
ngx_http_secure_link_module

PUT



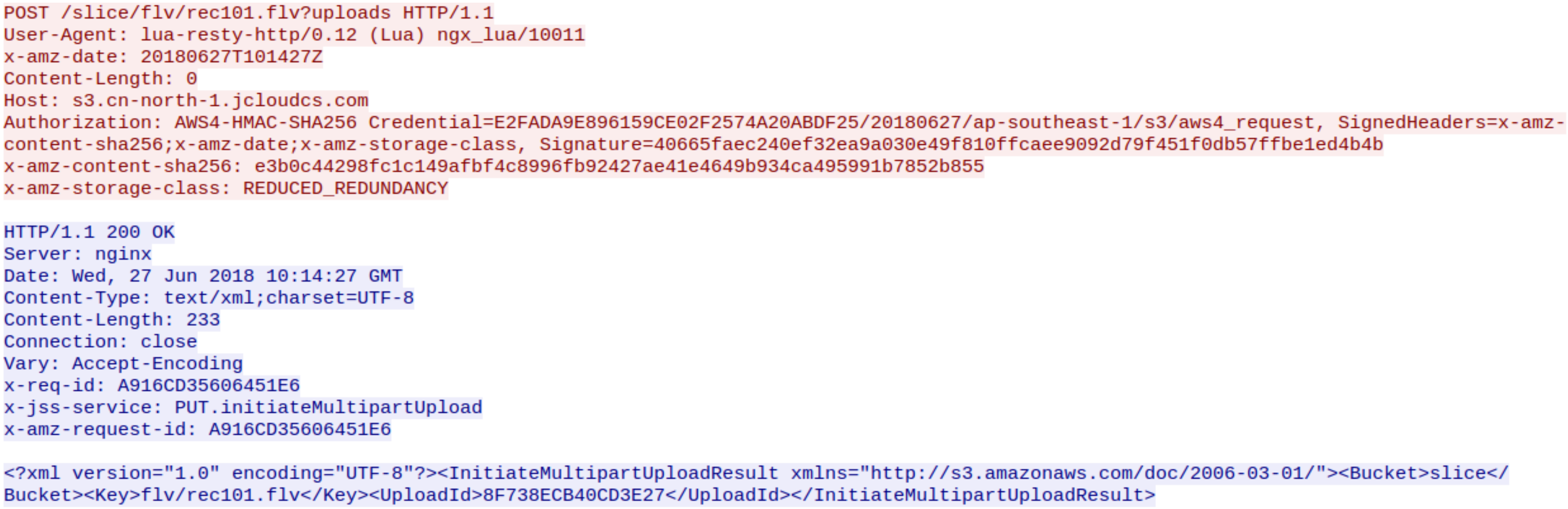
```
PUT /nginx-for-test-do-not-remove/ExampleObject.txt HTTP/1.1
x-amz-content-sha256: 8a816cbbd9f0a0ebf8c5196c255b64c7cddec8ea12a46c60abeb3b2eda193887
Authorization: AWS4-HMAC-SHA256 Credential=59E6DC72927457BDEBF36A56EE616B07/20180525/s3.cn-north-1.jcloudcs.com/s3/aws4_request,
SignedHeaders=content-length;host;x-amz-content-sha256;x-amz-date;x-amz-storage-class,
Signature=344197f7649d0dd66facbcc8f5faf446880b8daf7ae2bc7f17d351dac91a9e80
x-amz-date: 20180525T083557Z
x-amz-storage-class: REDUCED_REDUNDANCY
Cache-Control: no-cache
Pragma: no-cache
User-Agent: Java/1.8.0_171
Host: s3.cn-north-1.jcloudcs.com
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
Content-Length: 857

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc tortor metus, sagittis eget augue ut,
feugiat vehicula risus. Integer tortor mauris, vehicula nec mollis et, consectetur eget tortor. In ut
elit sagittis, ultrices est ut, iaculis turpis. In hac habitasse platea dictumst. Donec laoreet tellus
at auctor tempus. Praesent nec diam sed urna sollicitudin vehicula eget id est. Vivamus sed laoreet
lectus. Aliquam convallis condimentum risus, vitae porta justo venenatis vitae. Phasellus vitae nunc
varius, volutpat quam nec, mollis urna. Donec tempus, nisi vitae gravida facilisis, sapien sem malesuada
purus, id semper libero ipsum condimentum nulla. Suspendisse vel mi leo. Morbi pellentesque placerat congue.
Nunc sollicitudin nunc diam, nec hendrerit dui commodo sed. Duis dapibus commodo elit, id commodo erat
congue id. Aliquam erat volutpat.
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 25 May 2018 08:35:57 GMT
Content-Length: 0
Connection: close
x-jss-service: PUT.object
x-req-id: 8A894040124C8C0A
ETag: "599bab3ed2c697f1d26842727561fd94"
Content-MD5: WZurPtLGl/HSaEJydWH9lA==
x-amz-request-id: 8A894040124C8C0A
```

Multipart Upload



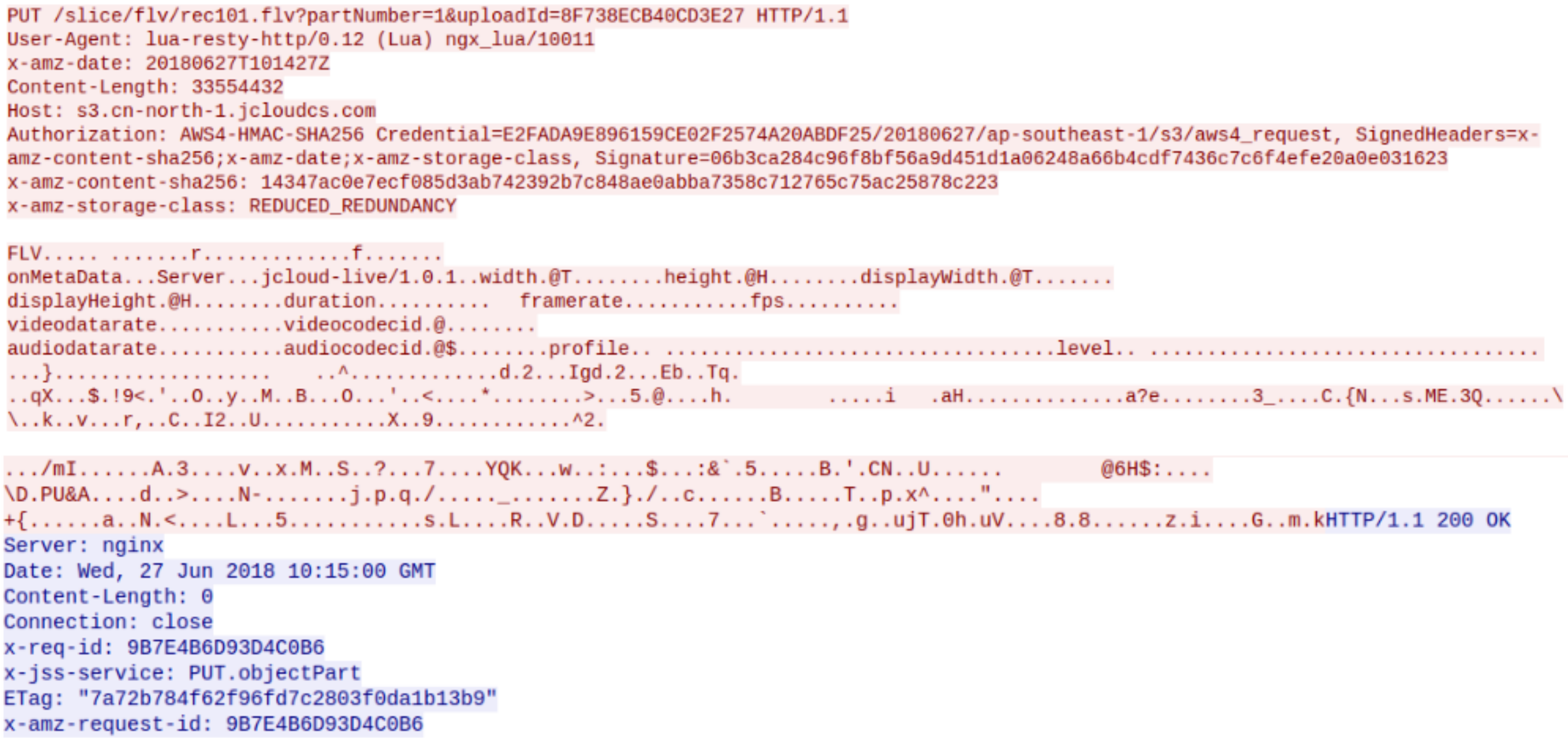
```
52… 3760.455802970   10.12.220.124    101.124.18.144   HTTP       626 POST /slice/flv/rec101.flv?uploads HTTP/1.1
52… 3760.513077440   101.124.18.144   10.12.220.124    HTTP/XML   567 HTTP/1.1 200 OK
10… 3793.409138287   10.12.220.124    101.124.18.144   HTTP       186 PUT /slice/flv/rec101.flv?partNumber=1&uploadId=8F738ECB40CD3E27 HTTP/1.1
10… 3793.446031036   101.124.18.144   10.12.220.124    HTTP       300 HTTP/1.1 200 OK
11… 3801.078050316   10.12.220.124    101.124.18.144   HTTP       592 PUT /slice/flv/rec101.flv?partNumber=2&uploadId=8F738ECB40CD3E27 HTTP/1.1
11… 3801.111930110   101.124.18.144   10.12.220.124    HTTP       300 HTTP/1.1 200 OK
11… 3801.207843723   10.12.220.124    101.124.18.144   HTTP       334 POST /slice/flv/rec101.flv?uploadId=8F738ECB40CD3E27 HTTP/1.1
11… 3801.304296928   101.124.18.144   10.12.220.124    HTTP/XML   650 HTTP/1.1 200 OK
```

Multipart Upload Init



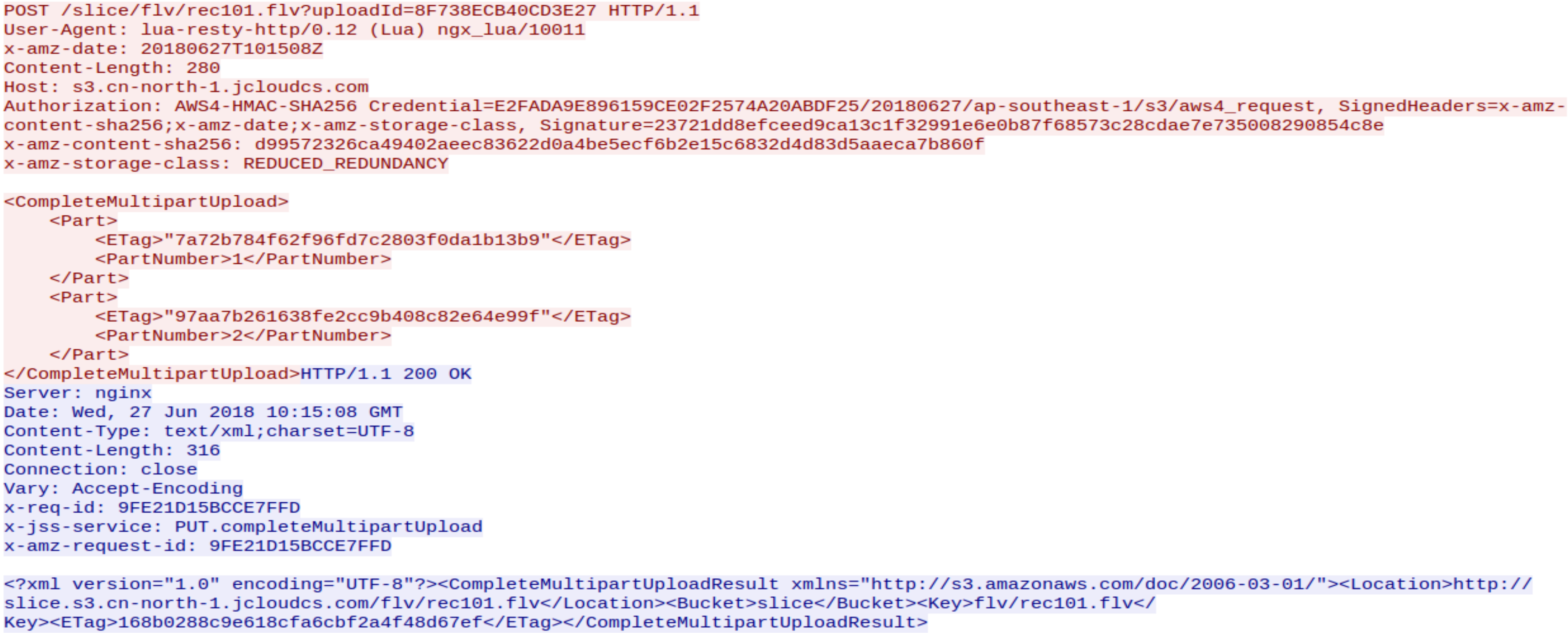
```
POST /slice/flv/rec101.flv?uploads HTTP/1.1
User-Agent: lua-resty-http/0.12 (Lua) ngx_lua/10011
x-amz-date: 20180627T101427Z
Content-Length: 0
Host: s3.cn-north-1.jcloudcs.com
Authorization: AWS4-HMAC-SHA256 Credential=E2FADA9E896159CE02F2574A20ABDF25/20180627/ap-southeast-1/s3/aws4_request, SignedHeaders=x-amz-
content-sha256;x-amz-date;x-amz-storage-class, Signature=40665faec240ef32ea9a030e49f810ffcaee9092d79f451f0db57ffbe1ed4b4b
x-amz-content-sha256: e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-storage-class: REDUCED_REDUNDANCY

HTTP/1.1 200 OK
Server: nginx
Date: Wed, 27 Jun 2018 10:14:27 GMT
Content-Type: text/xml;charset=UTF-8
Content-Length: 233
Connection: close
Vary: Accept-Encoding
x-req-id: A916CD35606451E6
x-jss-service: PUT.initiateMultipartUpload
x-amz-request-id: A916CD35606451E6

<?xml version="1.0" encoding="UTF-8"?><InitiateMultipartUploadResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/"><Bucket>slice</
Bucket><Key>flv/rec101.flv</Key><UploadId>8F738ECB40CD3E27</UploadId></InitiateMultipartUploadResult>
```

Multipart Upload part



```
PUT /slice/flv/rec101.flv?partNumber=1&uploadId=8F738ECB40CD3E27 HTTP/1.1
User-Agent: lua-resty-http/0.12 (Lua) ngx_lua/10011
x-amz-date: 20180627T101427Z
Content-Length: 33554432
Host: s3.cn-north-1.jcloudcs.com
Authorization: AWS4-HMAC-SHA256 Credential=E2FADA9E896159CE02F2574A20ABDF25/20180627/ap-southeast-1/s3/aws4_request, SignedHeaders=x-
amz-content-sha256;x-amz-date;x-amz-storage-class, Signature=06b3ca284c96f8bf56a9d451d1a06248a66b4cdf7436c7c6f4efe20a0e031623
x-amz-content-sha256: 14347ac0e7ecf085d3ab742392b7c848ae0abba7358c712765c75ac25878c223
x-amz-storage-class: REDUCED_REDUNDANCY

FLV..... .......r.............f.......
onMetaData...Server...jcloud-live/1.0.1..width.@T........height.@H........displayWidth.@T.......
displayHeight.@H........duration..........   framerate...........fps..........
videodatarate...........videocodecid.@........
audiodatarate...........audiocodecid.@$........profile.. ...................................level.. .................................
...}..................    ..^.............d.2...Igd.2...Eb..Tq.
..qX...$.!9<.'..O..y..M..B...O...'..<....*........>...5.@....h.        .....i   .aH..............a?e........3_....C.{N...s.ME.3Q......\
\..k..v...r,..C..I2..U...........X..9............^2.

.../mI......A.3....v..x.M..S..?...7....YQK...w..:...$...:&`.5.....B.'.CN..U......        @6H$:....
\D.PU&A....d..>....N-........j.p.q./....._........Z.}./..c......B.....T..p.x^...."....
+{......a..N.<....L...5............s.L....R..V.D.....S....7...`.....,.g..ujT.0h.uV....8.8......z.i....G..m.kHTTP/1.1 200 OK
Server: nginx
Date: Wed, 27 Jun 2018 10:15:00 GMT
Content-Length: 0
Connection: close
x-req-id: 9B7E4B6D93D4C0B6
x-jss-service: PUT.objectPart
ETag: "7a72b784f62f96fd7c2803f0da1b13b9"
x-amz-request-id: 9B7E4B6D93D4C0B6
```

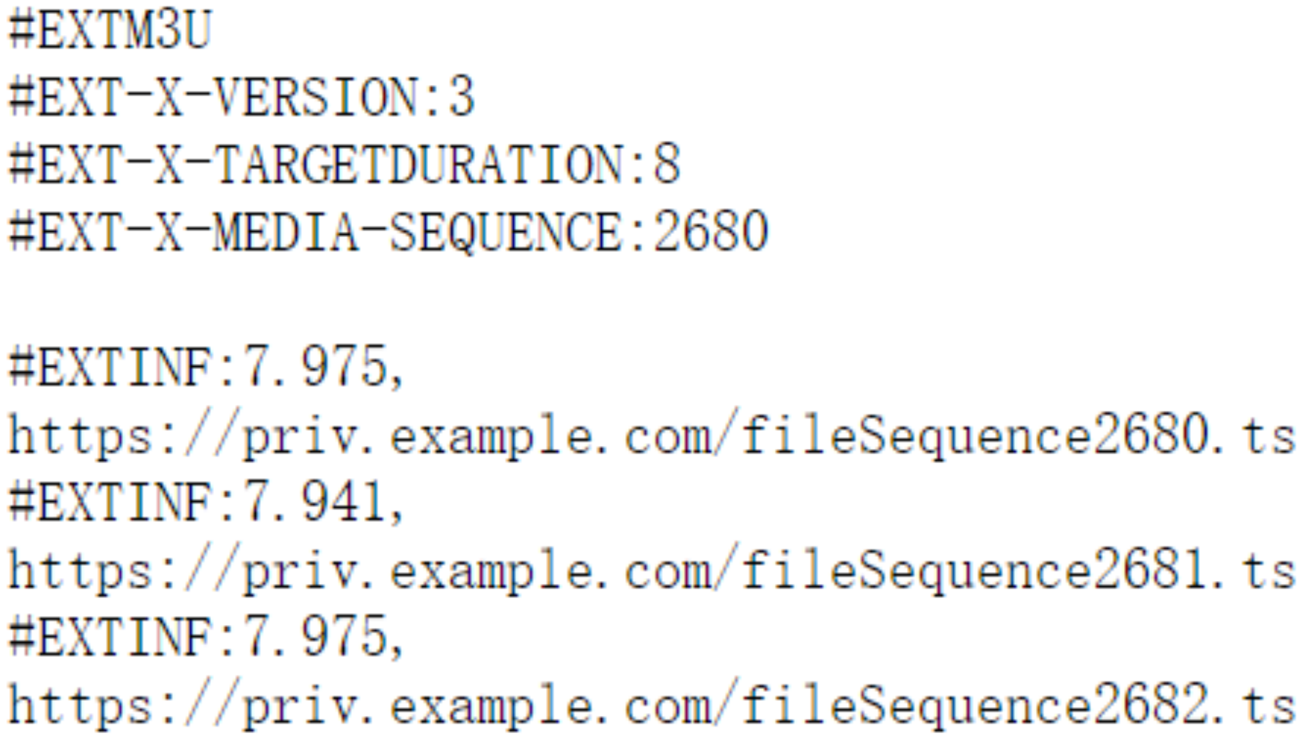
Multipart Upload Complete



```
POST /slice/flv/rec101.flv?uploadId=8F738ECB40CD3E27 HTTP/1.1
User-Agent: lua-resty-http/0.12 (Lua) ngx_lua/10011
x-amz-date: 20180627T101508Z
Content-Length: 280
Host: s3.cn-north-1.jcloudcs.com
Authorization: AWS4-HMAC-SHA256 Credential=E2FADA9E896159CE02F2574A20ABDF25/20180627/ap-southeast-1/s3/aws4_request, SignedHeaders=x-amz-
content-sha256;x-amz-date;x-amz-storage-class, Signature=23721dd8efceed9ca13c1f32991e6e0b87f68573c28cdae7e735008290854c8e
x-amz-content-sha256: d99572326ca49402aeec83622d0a4be5ecf6b2e15c6832d4d83d5aaeca7b860f
x-amz-storage-class: REDUCED_REDUNDANCY

<CompleteMultipartUpload>
    <Part>
        <ETag>"7a72b784f62f96fd7c2803f0da1b13b9"</ETag>
        <PartNumber>1</PartNumber>
    </Part>
    <Part>
        <ETag>"97aa7b261638fe2cc9b408c82e64e99f"</ETag>
        <PartNumber>2</PartNumber>
    </Part>
</CompleteMultipartUpload>HTTP/1.1 200 OK
Server: nginx
Date: Wed, 27 Jun 2018 10:15:08 GMT
Content-Type: text/xml;charset=UTF-8
Content-Length: 316
Connection: close
Vary: Accept-Encoding
x-req-id: 9FE21D15BCCE7FFD
x-jss-service: PUT.completeMultipartUpload
x-amz-request-id: 9FE21D15BCCE7FFD

<?xml version="1.0" encoding="UTF-8"?><CompleteMultipartUploadResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/"><Location>http://
slice.s3.cn-north-1.jcloudcs.com/flv/rec101.flv</Location><Bucket>slice</Bucket><Key>flv/rec101.flv</
Key><ETag>168b0288c9e618cfa6cbf2a4f48d67ef</ETag></CompleteMultipartUploadResult>
```

s3： https://github.com/jie123108/lua-resty-s3

HLS：HTTP Live Streaming协议



```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:8
#EXT-X-MEDIA-SEQUENCE:2680

#EXTINF:7.975,
https://priv.example.com/fileSequence2680.ts
#EXTINF:7.941,
https://priv.example.com/fileSequence2681.ts
#EXTINF:7.975,
https://priv.example.com/fileSequence2682.ts
```

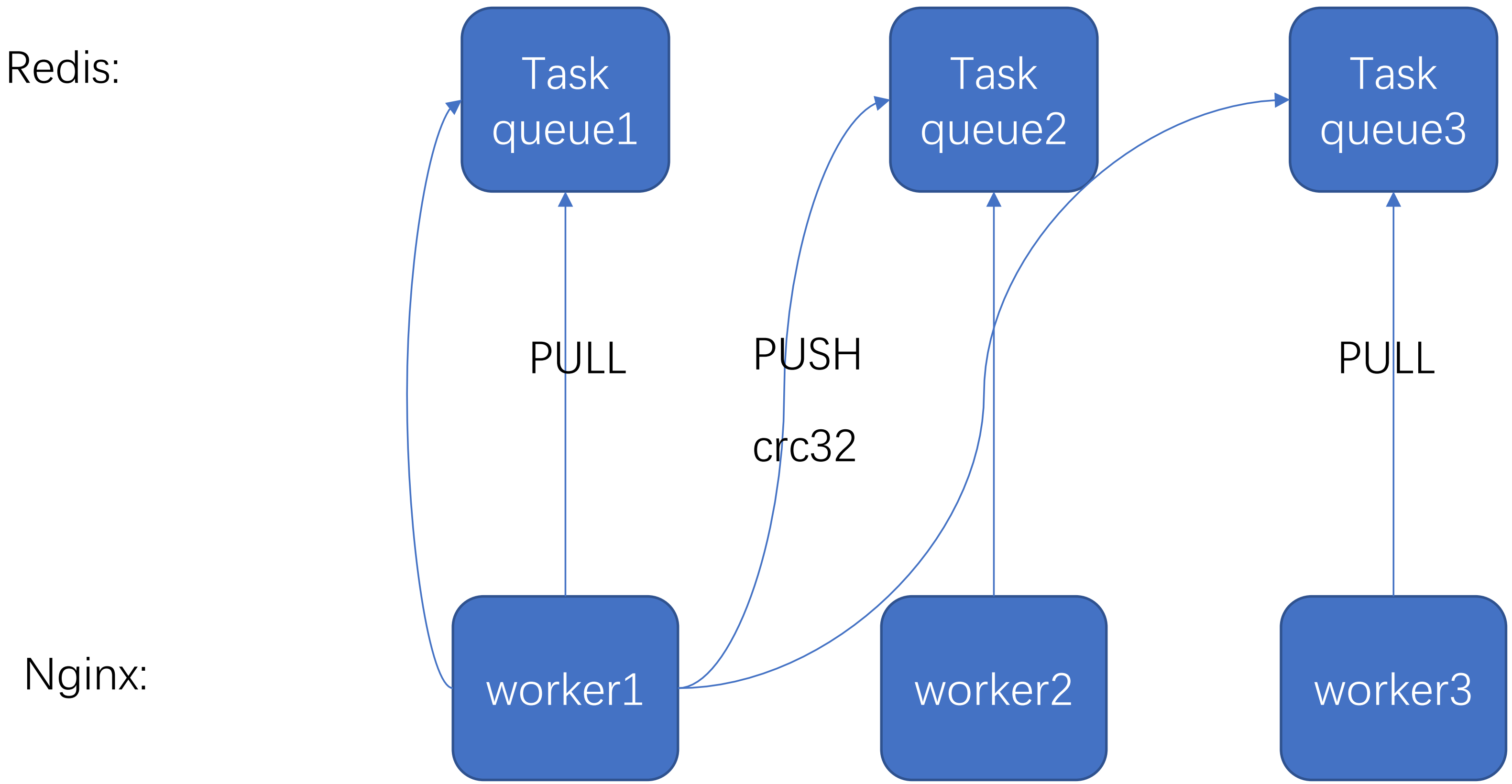
任务队列、均衡、流控

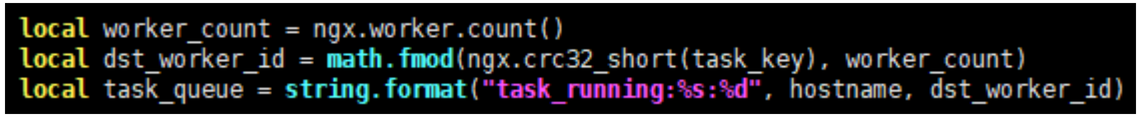
- nginx进程将任务基于crc32，分发至各进程的私有任务队列，存储至 redis zset。

Key: domain+app+stream+TS_name

- 可以接受突发大量请求，平滑处理
- 各nginx进程拉取任务、同步处理
- TS 片上传完成后生成、上传直播、点播M3U8
- 后端固定连接数，请求速率被动自适应
- 任务区分优先级
- 动态调节后端连接数，请求速率，流量

任务队列





```lua
local worker_count = ngx.worker.count()
local dst_worker_id = math.fmod(ngx.crc32_short(task_key), worker_count)
local task_queue = string.format("task_running:%s:%d", hostname, dst_worker_id)
```

失败重试、降级、高可靠

- 失败任务放入失败队列，指数退避重试
- 重试成功后完成后续任务：添加进点播m3u8；multi complete
- Redis 失败时数据存入本地内存（LRU cache, share dict），
恢复时同步数据至redis
- 支持Redis 集群，实现redis 高可用，突破内存限制
- 磁盘巡检，重新处理超时任务

Redis cluster:

https://github.com/cuiweixie/lua-resty-redis-cluster

https://github.com/steve0511/resty-redis-cluster

遇到问题、优化方案

- 单一任务队列，woker 获取任务不均衡，锁冲突严重，CPU消耗高
- 一个TS 更新一次M3U8，一次生成一个哈希表，CPU开销大
- Redis 内存消耗高

- 每个worker 一个任务队列，基于CRC_32分发
- 固定间隔生成点播M3U8
- 片数量过多时直播丢片，基于redis锁
- Redis 集群

直播时移回看服务

- 获取用户请求时间段的M3U8
- 基于mlcache 缓存点播m3u8 文件
- 回看：指定开始、结束时间段的M3U8
- 时移：当前时间向前的偏移的M3U8
- 时移每个请求创建session，记录上次返回M3U8的位置

mlcache：

https://github.com/thibaultcha/lua-resty-mlcache

Q&A

61717619