

Cloud Object Storage

장애 처리

제품 문서



Tencent Cloud

Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

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

Trademark Notice



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

Service Statement

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

목록:

장애 처리

RequestId 가져오기

공용 네트워크로 COS에 파일 업로드 시 속도가 느린 문제

COS 액세스 시 403 에러 코드 반환

리소스 액세스 오류

파일 액세스 시 404 NoSuchKey 반환

비디오 액세스 시 정상적으로 재생되지 않는 문제

URL이 잘못된 파일로 연결되는 오류

CDN 도메인으로 COS 액세스 시 HTTP ERROR 403 반환

HTTPS를 통한 사용자 정의 원본 도메인 액세스 실패

POST Object 자주 발생하는 오류

장애 처리

RequestId 가져오기

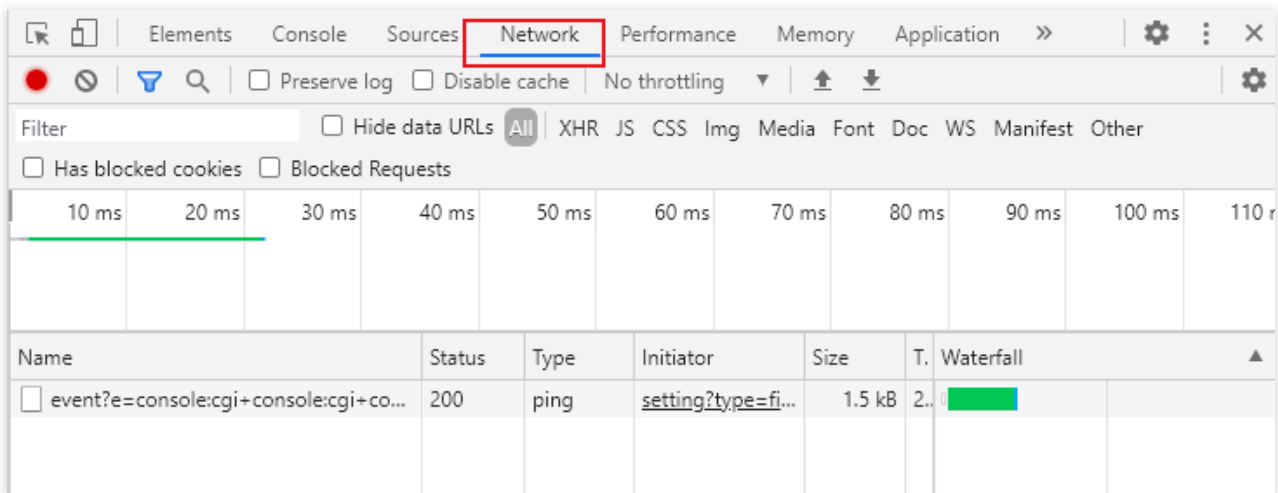
최종 업데이트 날짜: : 2024-06-25 10:42:28

개요

COS(Cloud Object Storage) 서비스에서 요청이 전송될 때마다, COS 서버는 해당 요청에 대한 ID, 즉 RequestId를 생성합니다. 본문은 각 시나리오에서 RequestId를 획득하는 방법에 대해 소개합니다.

콘솔에서 브라우저를 통해 획득

1. COS 콘솔에 로그인한 후, 왼쪽 사이드바에서 **버킷 리스트**를 클릭해 버킷 리스트 페이지로 이동합니다.
2. 액세스할 버킷을 클릭하여 이동합니다.
3. 'F12'를 눌러 브라우저의 개발자 툴 페이지로 이동합니다.
4. 개발자 툴 상단의 **Network**를 클릭합니다.



5. 다운로드할 파일명 우측의 **다운로드**를 클릭합니다. 개발자 툴 페이지에서 다운로드할 파일명 입력 및 필터링하여 파일을 선택하고, **Headers**를 클릭하여 **Response Headers** 섹션에서 RequestId 정보를 획득합니다.

파일 액세스 실패 시 획득

파일 액세스에 실패할 경우 페이지에서 표시한 반환된 XML 정보에서 **RequestId** 노드 정보를 획득합니다.

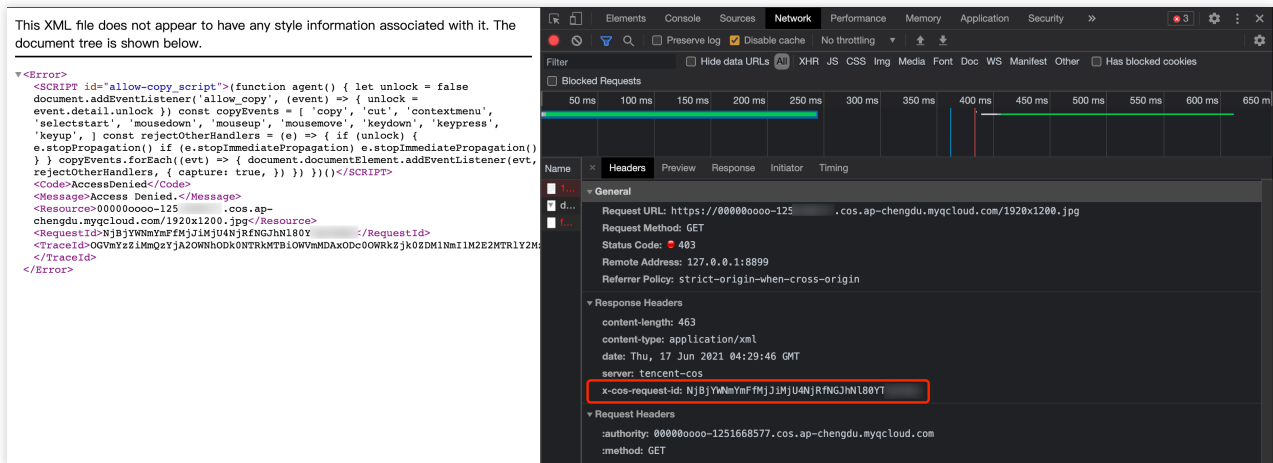
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<Error>
  <SCRIPT id="allow-copy_script">(function agent() { let unlock = false document.addEventListener('allow_copy',
  'contextmenu', 'selectstart', 'mousedown', 'mouseup', 'mousemove', 'keydown', 'keypress', 'keyup', ] const rej
  (e.stopImmediatePropagation) e.stopImmediatePropagation() } } copyEvents.forEach((evt) => { document.documentE
  <Code>AccessDenied</Code>
  <Message>Access Denied.</Message>
  <Resource>00000000-125...</Resource>
  <RequestId>NjBjYWNmYmFmJjJiMjU4NjRfNGJhN180YT</RequestId>
  <TraceId>OGVmYzZlMmQzYjAzOWNnODk0NTFRkMTB1OWVmMDAxODc0OWRkZjJkUzDM1NmI1M2E2MTRlY2MzZDhmNmI5MmI1OWI2NmQ0YjJkZWZ3N
  </Error>
  
```

아래의 작업을 통해서도 획득할 수 있습니다.

1. 'F12'를 눌러 브라우저의 개발자 툴 페이지로 이동합니다.
2. 페이지 상단의 **Network**를 클릭하여 All 유형을 선택하면 Response Headers에서 RequestId 필드 정보를 찾을 수 있습니다.



SDK를 통해 획득

SDK에는 너무 많은 인터페이스가 포함되어 있어 모든 인터페이스 예시를 하나씩 나열하는 것이 불가능하므로 모든 SDK는 파일 업로드를 예시로 현재 작업의 RequestId를 얻는 방법을 나타냅니다.

.NET SDK를 통해 획득

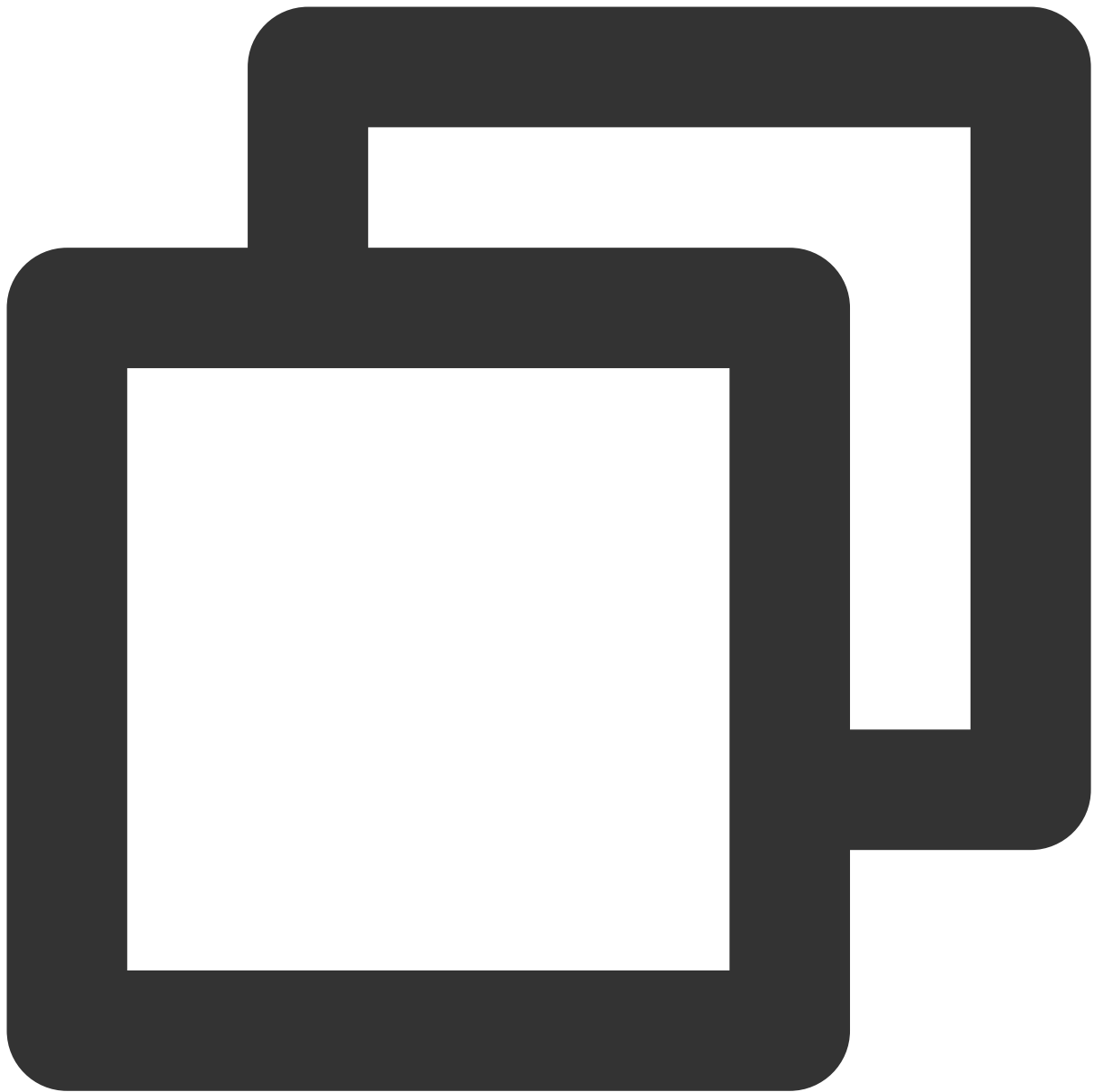


```
try
{
    String bucket = "examplebucket-1250000000"; //버킷, 형식: BucketName-APPID
    string cosPath = "test.cs"; // 객체 키
    byte[] data = System.Text.Encoding.Default.GetBytes("Hello COS"); // 이진법 데이터
    PutObjectRequest putObjectRequest = new PutObjectRequest(bucket, cosPath, data);

    PutObjectResult result = cosXml.PutObject(putObjectRequest);
    string requestId = result.responseHeaders.GetValueOrDefault("x-cos-request-id")[0]
    Console.WriteLine(requestId);
}
```

```
catch (COSXML.CosException.CosClientException clientEx)
{
    //요청 실패
    Console.WriteLine("CosClientException: " + clientEx);
}
catch (COSXML.CosException.CosServerException serverEx)
{
    //요청 실패
    Console.WriteLine("CosServerException: " + serverEx.GetInfo());
}
```

Go SDK를 통해 획득



```
package main

import (
    "context"
    "fmt"
    "net/http"
    "net/url"
    "strings"
    "github.com/tencentyun/cos-go-sdk-v5"
)
```



```
func main(){
    // examplebucket-1250000000과 COS_REGION을 실제 정보로 수정합니다.
    u, _ := url.Parse("https://examplebucket-1250000000.cos.COS_REGION.myqcloud.com")
    b := &cos.BaseURL{BucketURL: u}
    c := cos.NewClient(b, &http.Client{
        Transport: &cos.AuthorizationTransport{
            SecretID: "SECRETID",
            SecretKey: "SECRETKEY",
        },
    })
    // 객체 키(Key)는 버킷 내 객체의 고유 식별자입니다.
    // 예를 들어, 객체의 액세스 도메인 `examplebucket-1250000000.cos.COS_REGION.myqcloud
    name := "test.go"
    // 1. 문자열로 객체 업로드
    f := strings.NewReader("Hello COS")

    _, err := c.Object.Put(context.Background(), name, f, nil)
    if err != nil{
        // error 정보에는 RequestId 필드가 포함되어 있습니다.
        panic(err)
    }
    requestId := response.Header.Get("X-Cos-Request-Id")
    fmt.Println(requestId)
}
```

Java SDK를 통해 획득

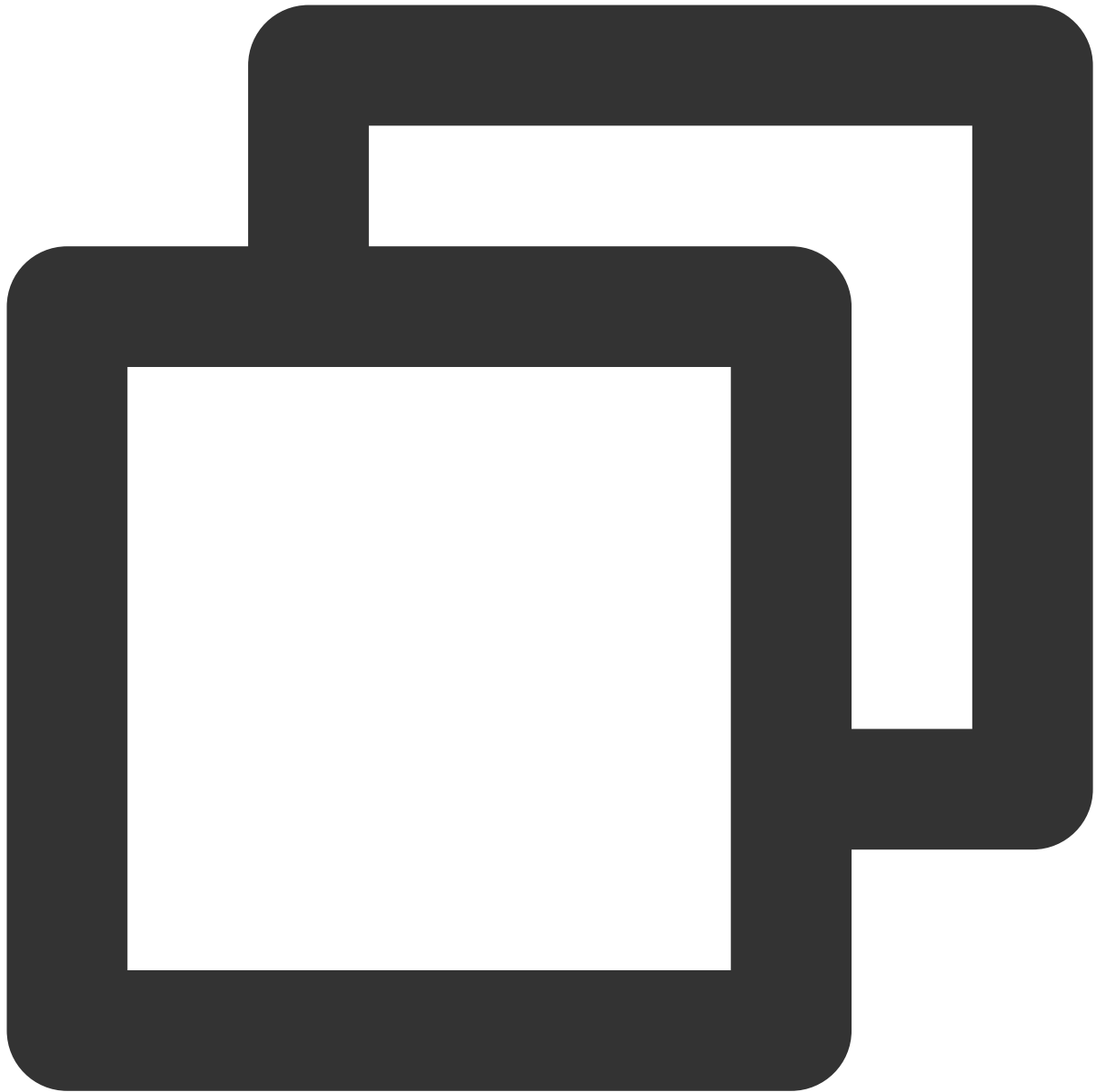


```
// 1 사용자의 개인 정보(secretId, secretKey)를 초기화합니다.  
String secretId = "SECRETID";  
String secretKey = "SECRETKEY";  
COSCredentials cred = new BasicCOSCredentials(secretId, secretKey);  
// 2 bucket의 리전 설정. COS 리전의 약칭은 https://cloud.tencent.com/document/product/4  
// clientConfig에 region, https(기본값: http), 타임아웃, 프록시 등을 설정하는 set 메소드가  
Region region = new Region("COS_REGION");  
ClientConfig clientConfig = new ClientConfig(region);  
// https 프로토콜 설정 및 사용 권장  
clientConfig.setHttpProtocol(HttpProtocol.https);  
// 3 cos 클라이언트 생성
```

```
COSClient cosClient = new COSClient(cred, clientConfig);
// Bucket 이름 생성 형식은 BucketName-APPID이며, 입력하는 버킷 이름은 반드시 해당 형식을 따라
String bucketName = "examplebucket-1250000000";

String content = "Hello COS";
String key = "test.java";
PutObjectResult putObjectResult = cosClient.putObject(bucketName, key, content);
String requestId = putObjectResult.getRequestId();
System.out.println(requestId);
```

Python SDK를 통해 획득



```
# -*- coding=utf-8
from qcloud_cos import CosConfig
from qcloud_cos import CosS3Client
import sys
import os
import logging

# 일반적으로 로그 레벨은 INFO이며 위치 지정이 필요한 경우 DEBUG로 수정할 수 있으며 이 때 SDK는
logging.basicConfig(level=logging.INFO, stream=sys.stdout)

# 1. secret_id, secret_key, region 등을 포함한 사용자 속성을 설정합니다. Appid는 CosConf
```

```
secret_id = os.environ['COS_SECRET_ID']      # 사용자 SecretId. 리스크를 줄이기 위해 서브
secret_key = os.environ['COS_SECRET_KEY']    # 사용자 SecretKey. 리스크를 줄이기 위해 서.
region = 'ap-beijing'                       # 사용자의 region으로 대체합니다. 생성된 버킷이 속한 region은
                                             # COS에서 지원되는 모든 region 목록은 다음을 참고하십시오. http
token = None                                 # 영구 키를 사용하는 경우 token을 입력할 필요가 없으나, 임시 키를
scheme = 'https'                             # http/https 프로토콜로 COS에 액세스 지정. 기본값: https, 입

config = CosConfig(Region=region, SecretId=secret_id, SecretKey=secret_key, Token=token)
client = CosS3Client(config)

try:
    response = client.put_object(
        Bucket='examplebucket-1250000000',
        Key='exampleobject',
        Body=b'abcdefg'
    )

    # 요청 정상 반환 시 response를 통해 request-id 조회
    if 'x-cos-request-id' in response:
        print(response['x-cos-request-id'])

# 요청 실패 시 이상 경고를 통해 request-id 조회
except CosServiceError as e:
    print(e.get_request_id())
```

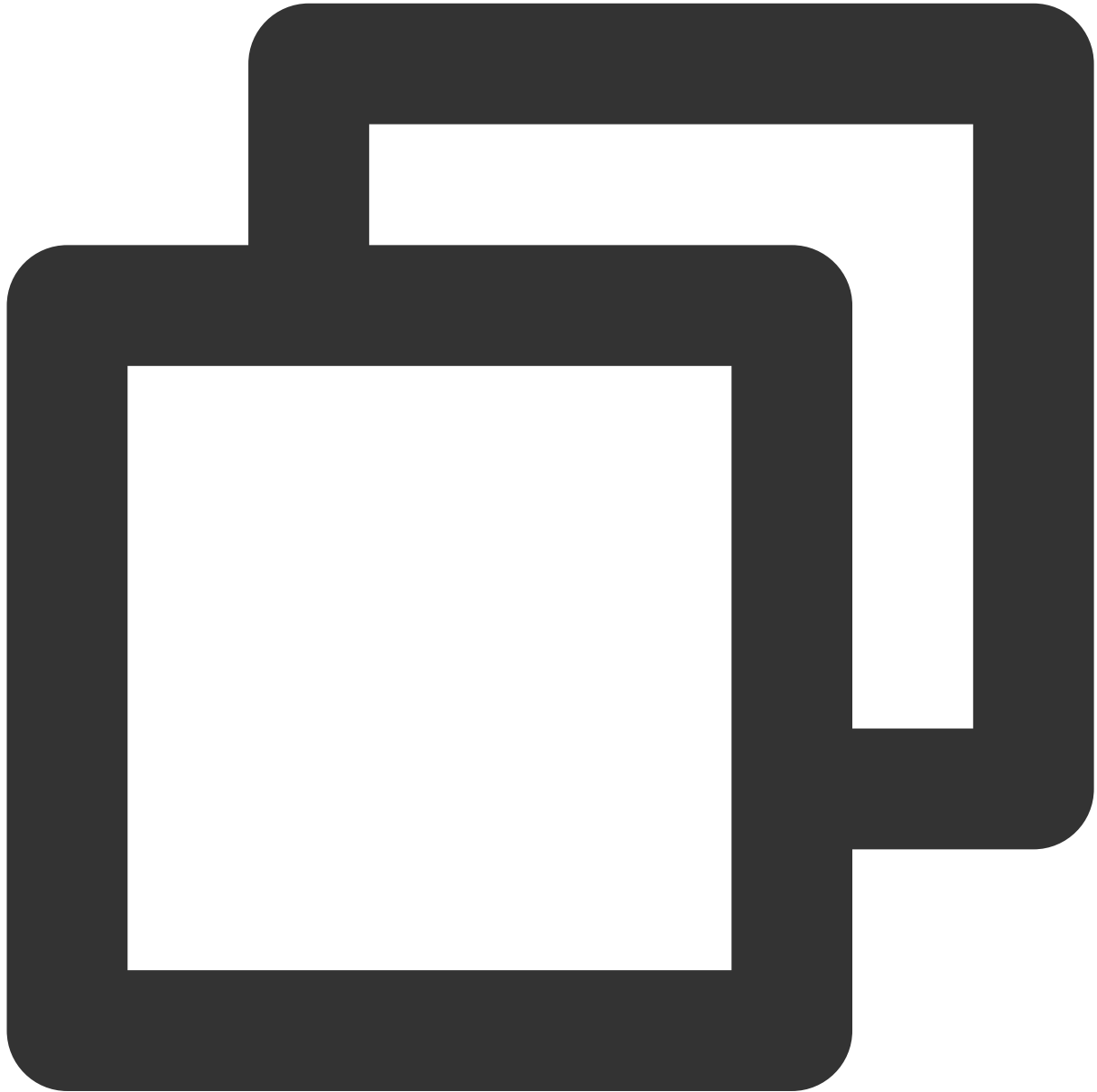
JavaScript SDK를 통해 획득



```
cos.putObject({
  Bucket: 'examplebucket-1250000000', /* 필수 */
  Region: 'COS_REGION',      /* 필수 */
  Key: 'test.js',            /* 필수 */
  StorageClass: 'STANDARD',
  Body: 'Hello COS',
  onProgress: function(progressData) {
    console.log(JSON.stringify(progressData));
  }
}, function(err, data) {
  var requestId = (err || data).headers['x-cos-request-id'];
});
```

```
console.log(requestId );  
});
```

Node.js SDK를 통해 획득



```
var COS = require('cos-nodejs-sdk-v5');  
var cos = new COS({  
  SecretId: 'SECRETID',  
  SecretKey: 'SECRETKEY'  
});
```

```
cos.putObject({
  Bucket: 'examplebucket-1250000000', /* 필수 */
  Region: 'COS_REGION', /* 필수 */
  Key: 'test.nodejs', /* 필수 */
  StorageClass: 'STANDARD',
  Body: Buffer.from('Hello COS'),
  onProgress: function(progressData) {
    console.log(JSON.stringify(progressData));
  }
}, function(err, data) {
  var requestId = (err || data).headers['x-cos-request-id'];
  console.log(requestId );
});
```

WeChat 미니프로그램 SDK를 통해 획득

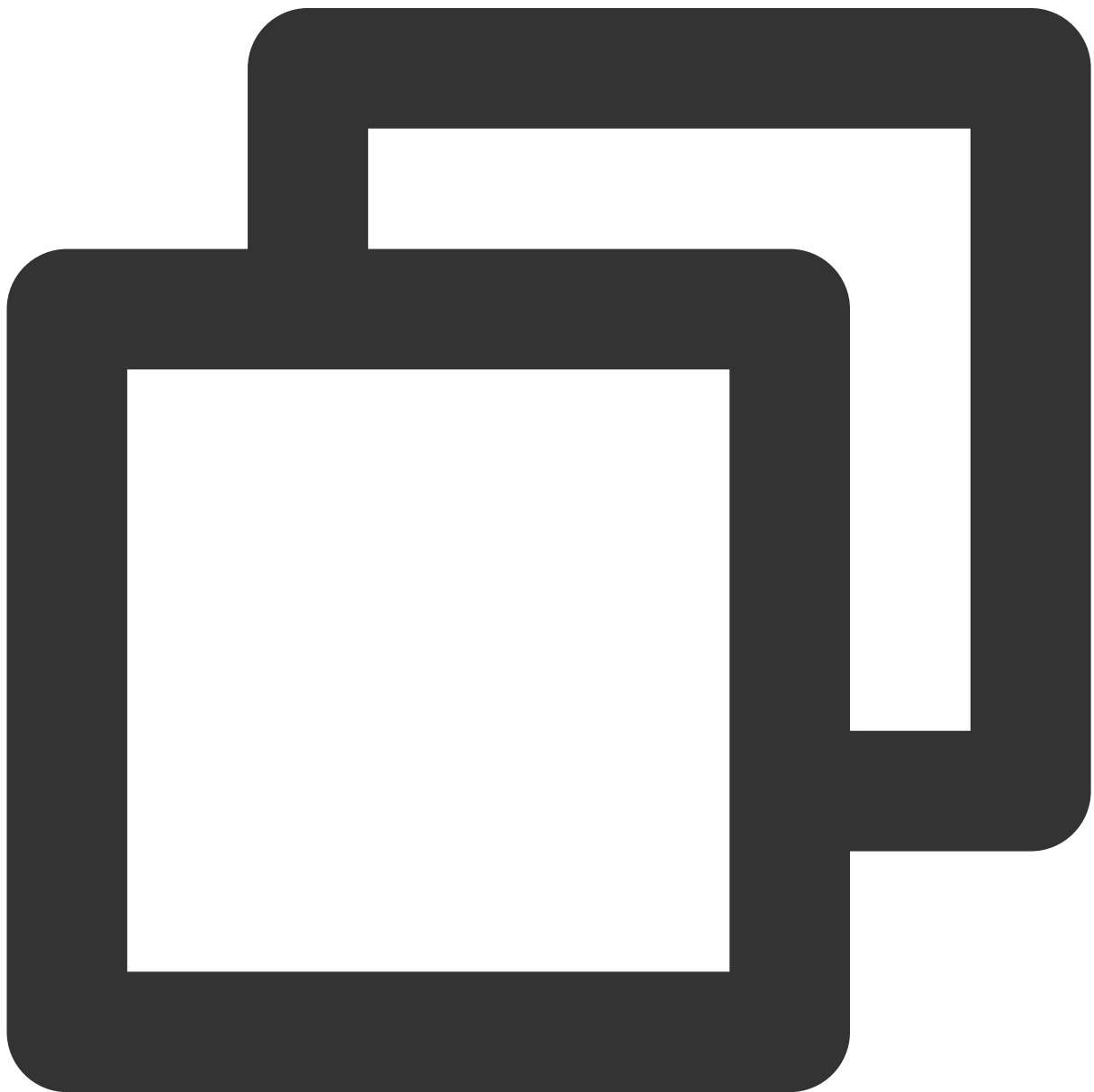


```
var COS = require('cos-wx-sdk-v5');
var cos = new COS({
  SecretId: 'SECRETID',
  SecretKey: 'SECRETKEY'
});

cos.putObject({
  Bucket: 'examplebucket-1250000000', /* 필수 */
  Region: 'COS_REGION', /* 필수 */
  Key: 'test.js', /* 필수 */
  StorageClass: 'STANDARD',
```

```
Body: 'Hello COS',
onProgress: function(progressData) {
    console.log(JSON.stringify(progressData));
}
}, function(err, data) {
    var requestId = (err || data).headers['x-cos-request-id'];
    console.log(requestId);
});
```

PHP SDK를 통해 획득



```
$secretId = "SECRETID"; //Tencent Cloud API 키 SecretId";
$secretKey = "SECRETKEY"; //Tencent Cloud API 키 SecretKey";
$region = "COS_REGION"; //기본 버킷 리전 설정
$cosClient = new Qcloud\Cos\Client(
    array(
        'region' => $region,
        'schema' => 'https', //프로토콜 헤더, 기본값: http
        'credentials'=> array(
            'secretId' => $secretId ,
            'secretKey' => $secretKey));
# 파일 업로드
## putObject(업로드 인터페이스, 최대 5G 파일까지 업로드 가능)
### 메모리에 있는 문자열 업로드
try {
    $bucket = "examplebucket-1250000000"; //버킷 이름. 형식: BucketName-APPID
    $key = "test.php"; //여기서 key는 객체 키로, 버킷 내 객체의 고유 식별자입니다.
    $result = $cosClient->putObject(array(
        'Bucket' => $bucket,
        'Key' => $key,
        'Body' => 'Hello COS'));
    $requestId = $result['RequestId'];
    print_r($requestId);
} catch (\Exception $e) {
    echo "$e\n";
}
```

iOS SDK를 통해 획득



```
QCloudCOSXMLUploadObjectRequest* put = [QCloudCOSXMLUploadObjectRequest new];  
/** 로컬 파일 경로. URL이 file:// 로 시작되는지 확인하십시오. 형식은 다음과 같습니다.  
1. [NSURL URLWithString:@"file:///var/mobile/Containers/Data/Application/DBPF7490-  
2. [NSURL fileURLWithPath:@"/var/mobile/Containers/Data/Application/DBPF7490-D5U8-4  
*/  
NSURL* url = [NSURL fileURLWithPath:@"파일의 URL"];  
// BucketName-Appid로 구성된 버킷 이름. COS 콘솔에서 조회할 수 있습니다. https://console.t  
put.bucket = @"examplebucket-1250000000";  
// 객체 키, 객체의 COS 상의 전체 경로로, 디렉터리가 있을 경우 형식은 "video/xxx/movie.mp4"입  
put.object = @"exampleobject";  
// 업로드가 필요한 객체 콘텐츠입니다. NSData* 또는 NSURL* 유형의 변수를 전달할 수 있습니다.
```

```
put.body = url;
// 업로드 진행률 수신
[put setSendProcessBlock:^(int64_t bytesSent,
                           int64_t totalBytesSent,
                           int64_t totalBytesExpectedToSend) {
    //     bytesSent           발송할 바이트 수 (대용량 파일은 여러 번으로 나누어 발
    //     totalBytesSent       발송한 바이트 수
    //     totalBytesExpectedToSend 이번 업로드에서 발송할 총 바이트 수 (파일 1개의 크기)
}];
// 업로드 결과 수신
[put setFinishBlock:^(QCloudUploadObjectResult *result, NSError *error) {
    // requestid 가져오기
    [result.__originHTTPURLResponse__.allHeaderFields objectForKey:@"x-cos-request-id"];
}];
[put setInitMultipleUploadFinishBlock:^(QCloudInitiateMultipartUploadResult *
                                         multipleUploadInitResult,
                                         QCloudCOSXMLUploadObjectResumeData resumeData) {
    // 멀티파트 업로드 초기화 완료 후 해당 block 콜백, 이곳에서 resumeData, uploadid를 획득
    NSString* uploadId = multipleUploadInitResult.uploadId;
}];
[[QCloudCOSTransferMangerService defaultManager] UploadObject:put];
```

Android SDK를 통해 가져오기



```
// 1. TransferService 초기화. 동일 설정의 경우 동일한 TransferService를 재사용해야 합니다.
TransferConfig transferConfig = new TransferConfig.Builder()
    .build();
CosXmlServiceConfig cosXmlServiceConfig = new CosXmlServiceConfig.Builder()
    .setRegion(COS_REGION)
    .builder();
CosXmlService cosXmlService = new CosXmlService(context, cosXmlServiceConfig, creden
TransferService transferService = new TransferService(cosXmlService, transferConfig

// 2. PutObjectRequest 초기화
String bucket = "examplebucket-1250000000"; //버킷. 형식: BucketName-APPID
```

```
String cosPath = "exampleobject"; //버킷 내 객체 위치 식별자, 즉 객체 키
String srcPath = "examplefilepath"; //로컬 파일의 절대 경로
PutObjectRequest putObjectRequest = new PutObjectRequest(bucket,
    cosPath, srcPath);

// 3. upload 메소드를 호출하여 파일 업로드
final COSUploadTask uploadTask = transferService.upload(putObjectRequest);
uploadTask.setCosXmlResultListener(new CosXmlResultListener() {
    @Override
    public void onSuccess(CosXmlRequest request, CosXmlResult result) {
        // 업로드 성공, 여기에서 requestId를 얻을 수 있습니다.
        String requestId = result.getHeader("x-cos-request-id");
    }

    @Override
    public void onFail(CosXmlRequest request,
        CosXmlClientException clientException,
        CosXmlServiceException serviceException) {
        // CosXmlServiceException 예외 발생 시에만 requestId가 있습니다.
        if (serviceException != null) {
            String requestId = serviceException.getRequestId();
        }
    }
});
```

공용 네트워크로 COS에 파일 업로드 시 속도가 느린 문제

최종 업데이트 날짜: : 2024-06-25 10:42:28

현상 설명

현상1:

회사 네트워크를 사용해 업로드 시 정상적으로 전송되나, 홈 네트워크를 사용해 업로드 시 전송이 느림(8Mbps 이하)
4G 네트워크를 사용해 업로드 시 정상적으로 전송되나, 회사 네트워크를 사용해 업로드 시 전송이 느림(8Mbps 이하)

현상2: 사용자 정의 도메인을 사용해 업로드 시 전송이 느림

가능한 원인

현상1:

1.1 서로 다른 네트워크 환경에서 COS 액세스 속도가 다른 경우, 해당 네트워크 ISP 및 네트워크 환경과 관련된 문제일 수 있습니다.

1.2 서로 다른 네트워크 환경에서 COS 액세스 속도가 다른 경우, 지역 간 액세스로 인한 문제일 수 있습니다.

현상2: Content Delivery Network(CDN), Cloud Virtual Machine(CVM), 높은 보안 제품과 같이 사용자 정의 도메인 CNAME에서 다른 제품을 거쳐 COS로 전송되는 경우일 수 있습니다.

해결 방법

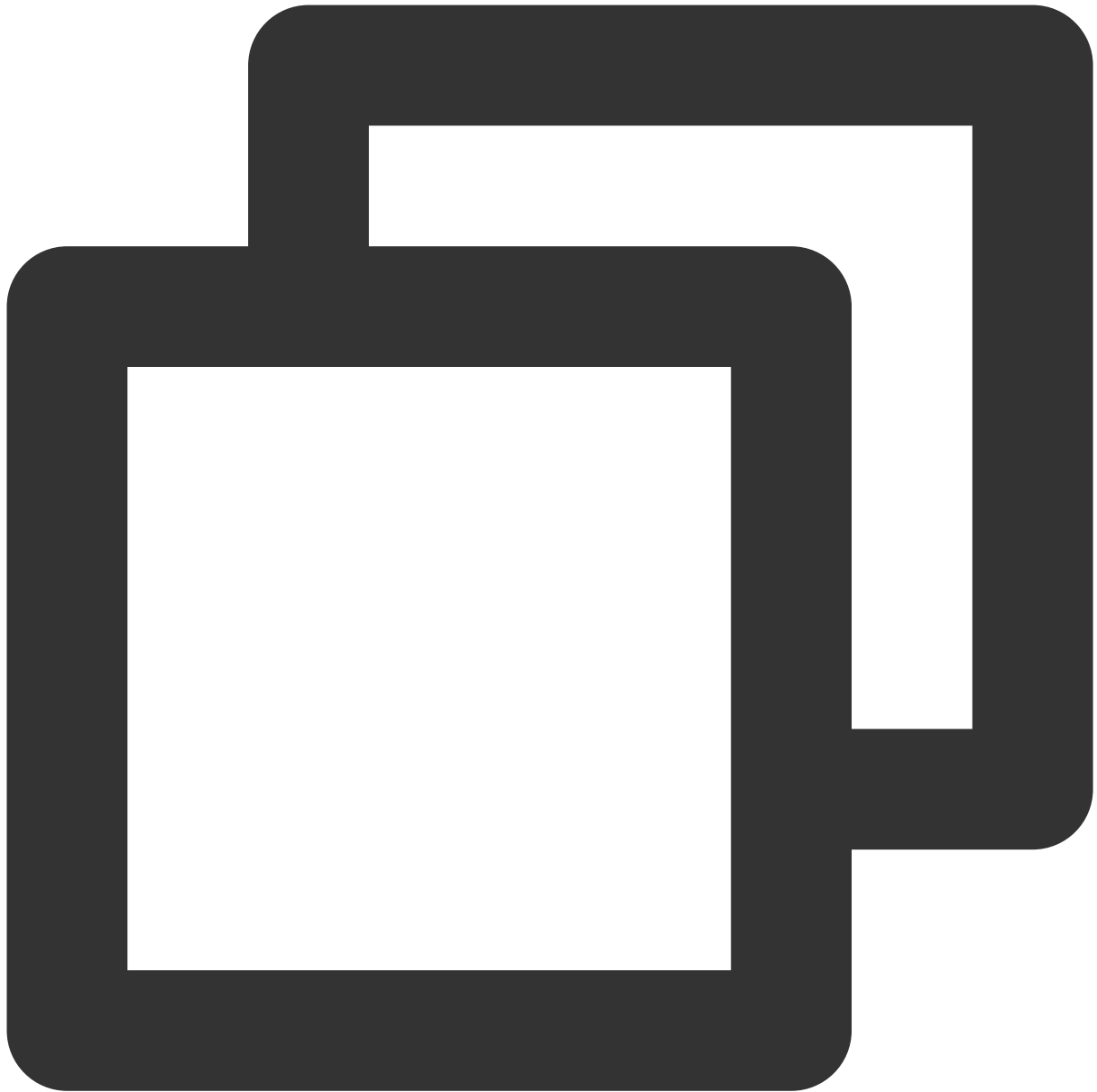
현상1의 상황이 발생하는 경우 클라이언트 네트워크 환경을 점검하여 자체적으로 처리할 수 있습니다. 자세한 작업 방법은 [클라이언트 네트워크 진단](#)을 참고하십시오.

현상2의 상황이 발생하는 경우 사용자 정의 도메인 리졸브 방식을 변경해 중간에 경유하는 링크를 줄임으로써 전송 효율을 향상시킬 수 있습니다. 자세한 작업 방법은 [사용자 정의 도메인 리졸브 변경](#)을 참고하십시오.

처리 단계

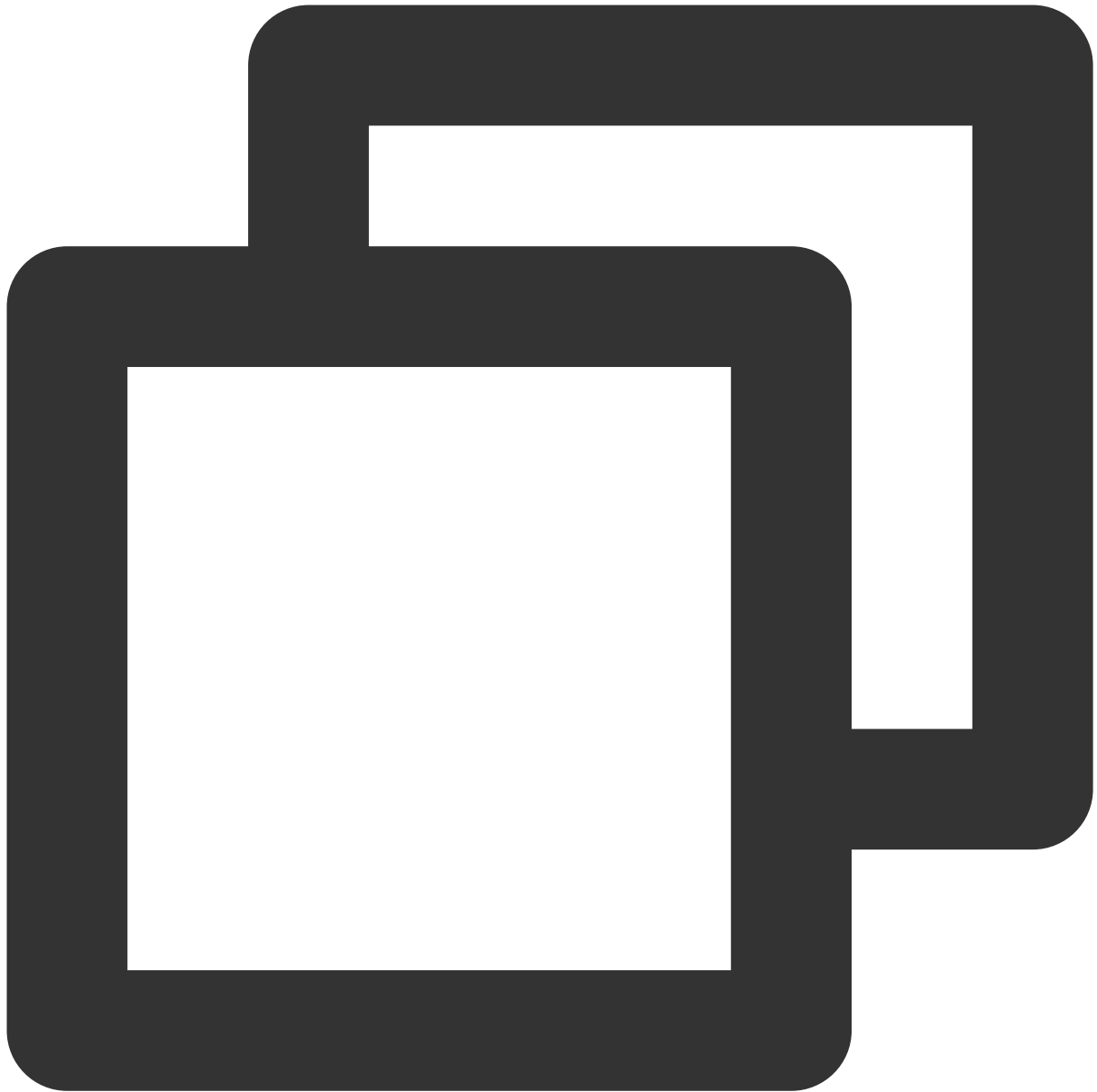
클라이언트 네트워크 진단

1. 다음 명령어를 실행하여 IP 주소 ISP와 클라이언트 네트워크 ISP가 동일한지 확인합니다.



ping COS의 액세스 도메인

예시:




```
ping examplebucket-1250000000.cos.ap-beijing.mqcloud.com
```

예. 3단계를 실행합니다.

아니요. 2단계를 실행합니다.

2. 브라우저에 프록시를 설정했는지 확인합니다. 다음은 Chrome 브라우저 예시입니다.

2.1 Chrome 브라우저를 열고 오른쪽 상단의

>  설정을 클릭해 설정 페이지를 엽니다.

2.2 고급을 클릭하고 '시스템'에서 **컴퓨터 프록시 설정**을 클릭하면 운영 체제의 설정창이 열립니다.

프록시를 설정했는지 확인합니다.

예. 프록시를 끕니다.

아니요. **3단계**를 실행합니다.

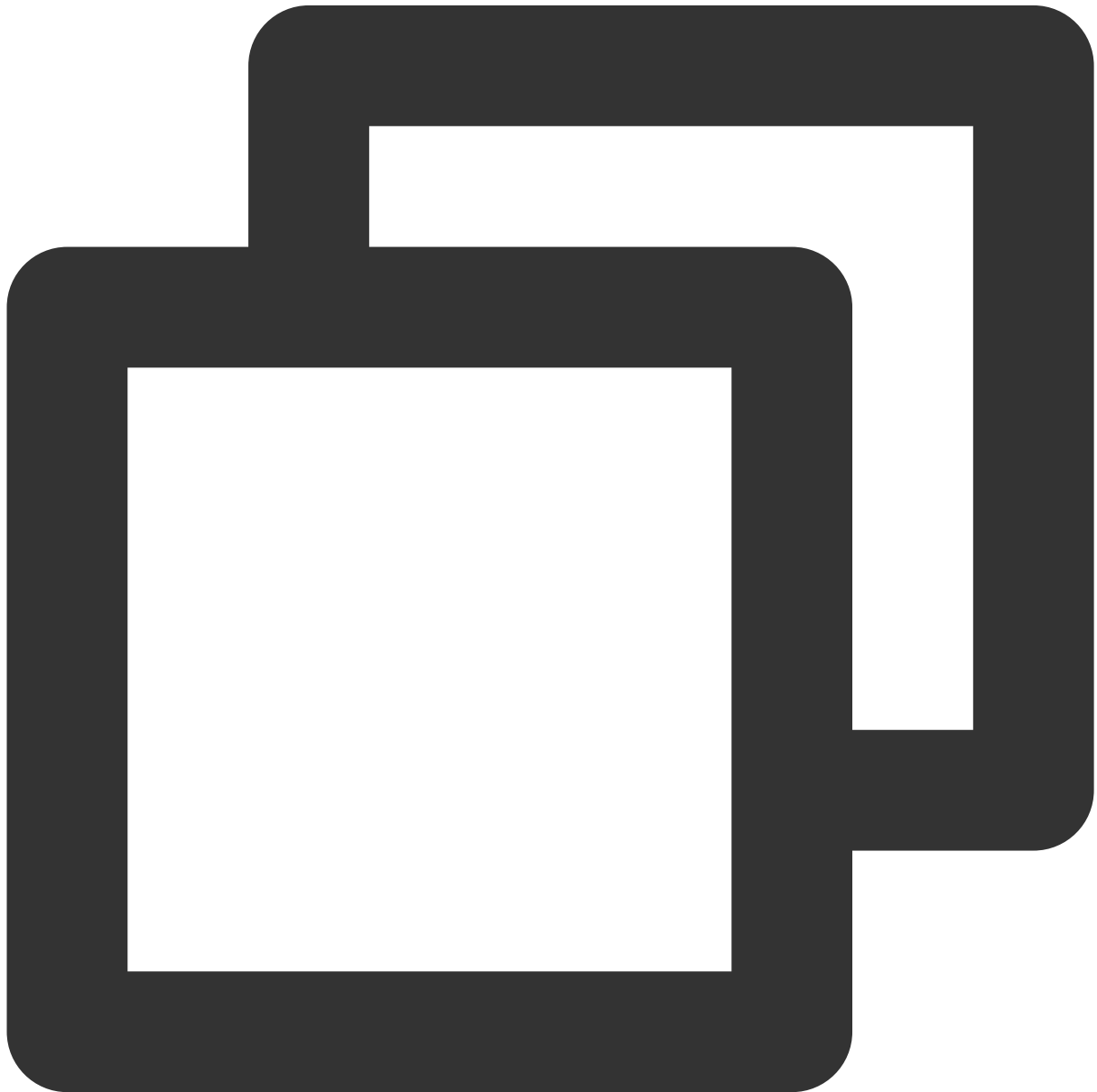
3. 사용하는 Wi-Fi 라우터에 속도 제한이 존재하는지 확인합니다.

예. 실제 필요에 따라 개방합니다.

아니요. **4단계**를 실행합니다.

4. 현재 네트워크의 COS 업로드 전송 성능을 확인합니다.

20MB 객체로 업로드 및 다운로드 성능을 테스트합니다. 다음은 COS의 COSCMD 툴 예시입니다.



```
coscmd probe -n 1 -s 20
```

평균 속도(Average), 최저 속도(Min), 최고 속도(Max)로 나뉘어 다음과 유사한 결과를 반환합니다.

```
D:\work>coscmd probe -n 1 -s 20
Upload tmp_test_20M => cos://[redacted]/tmp_
[success]
Download cos://[redacted]/tmp_test_20M => tm
[success]
Success Rate: [1/1]
  20M TEST      Average      Min          Max
  Upload       12.6MB/s    12.6MB/s    12.6MB/s
  Download     10.2MB/s    10.2MB/s    10.2MB/s
```

5. 브라우저를 통해 [속도 테스트 웹사이트](#)에 액세스하고 [4단계](#)와 결합하여 클라이언트의 네트워크 대역폭 점유율이 한계에 도달하는지 확인합니다.

4단계의 속도가 클라이언트 대역폭 속도보다 낮은 경우 [고객센터](#)로 문의하십시오.

4단계의 속도가 클라이언트 대역폭 속도와 동일하고 ISP가 약정한 대역폭에 도달하지 않는 경우 ISP 고객센터로 문의하십시오.

4단계의 속도가 클라이언트 대역폭 속도와 동일하고 ISP가 약정한 대역폭에 도달하는 경우 [6단계](#)를 실행하십시오.

6. 중국 내 클라이언트가 해외 노드 bucket에 액세스했는지, 해외 클라이언트가 중국 내 노드 bucket에 액세스했는지 확인합니다.

예. COS의 글로벌 가속 기능 사용을 권장합니다.

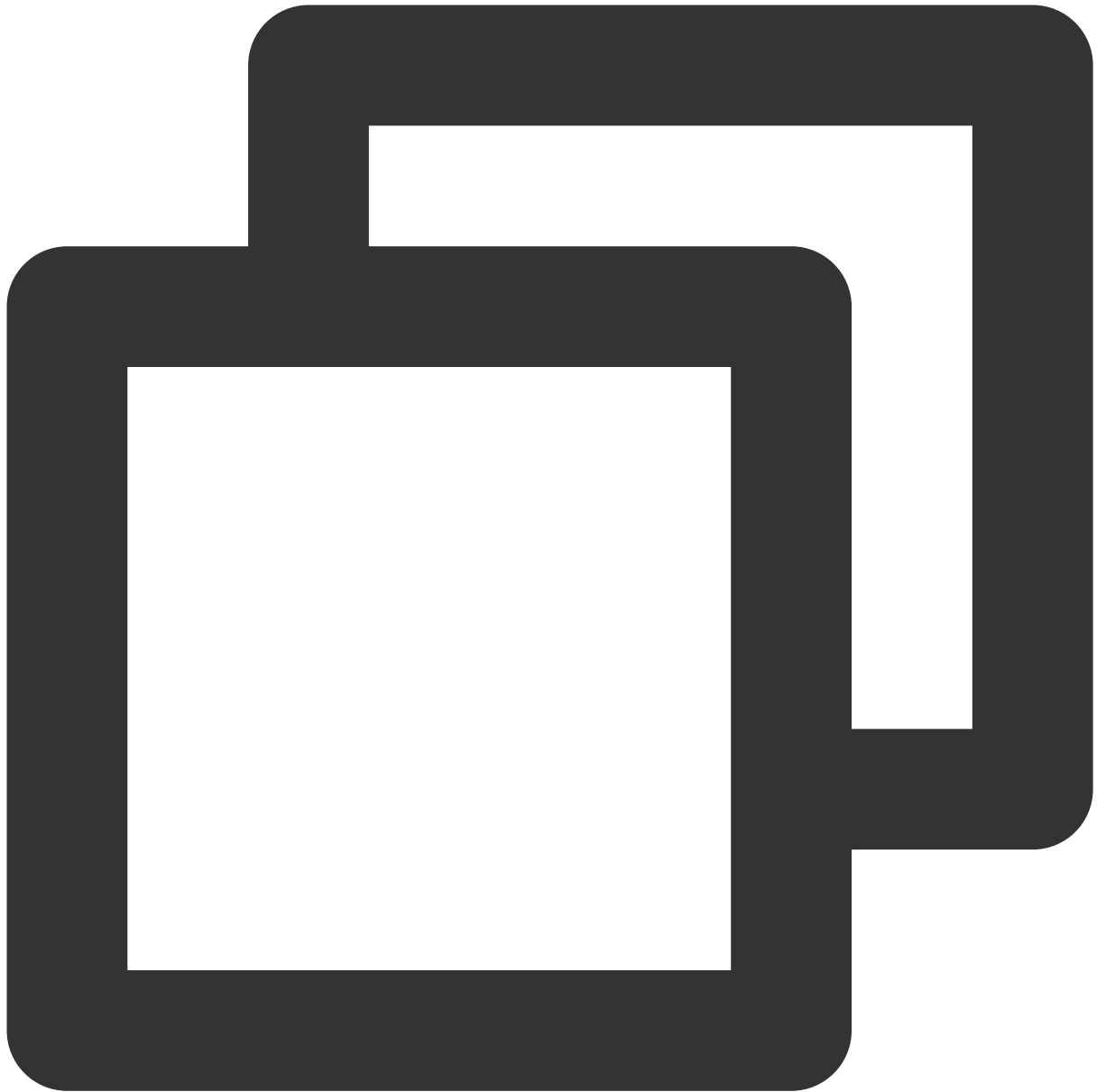
해당 오류가 아닌 경우: [고객센터](#)에 문의하십시오.

사용자 정의 도메인 리졸브 변경

1. 사용자 정의 도메인 리졸브가 COS 도메인인지 확인합니다.

해당 오류일 경우: [고객센터](#)에 문의하십시오.

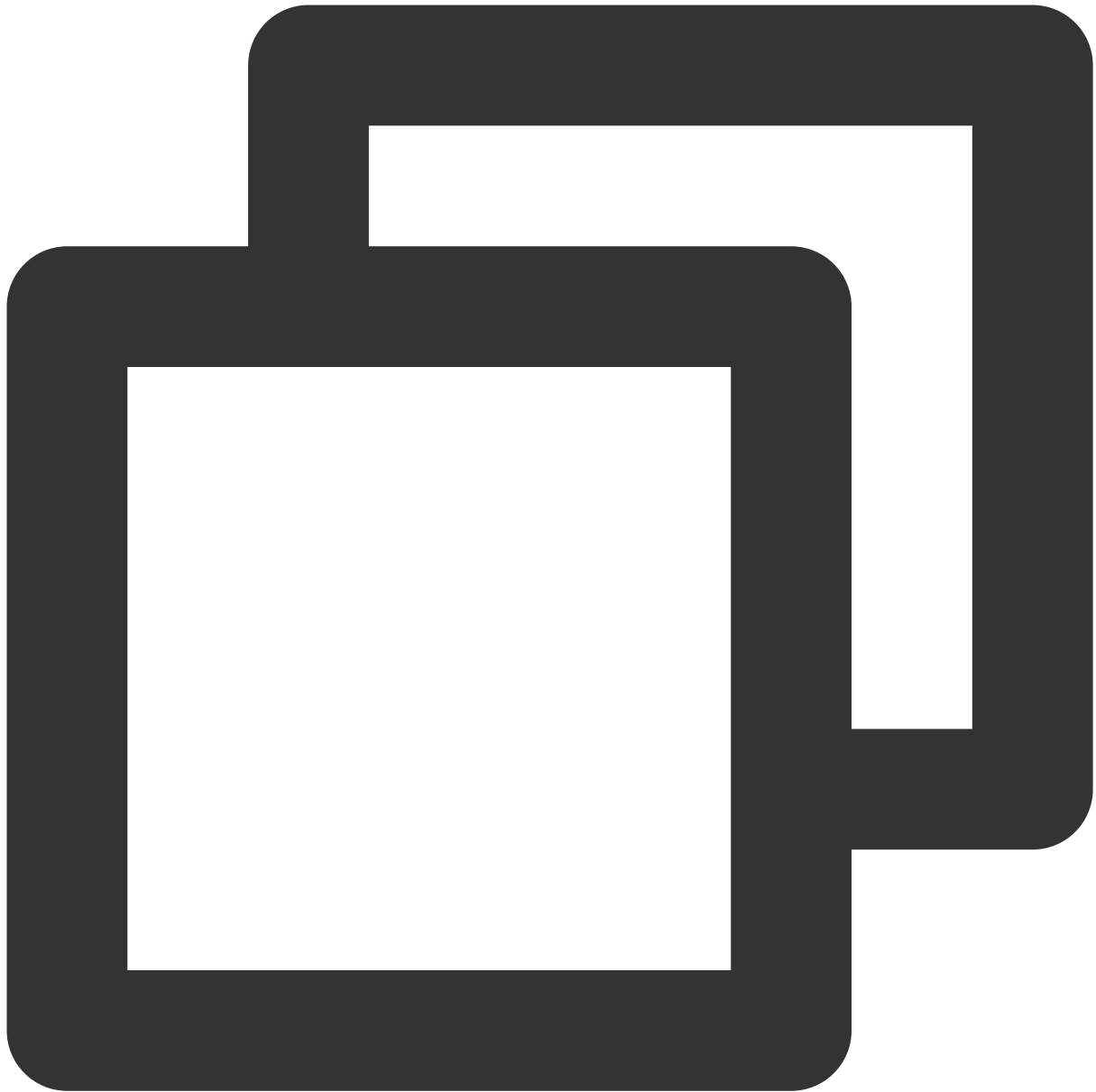
자주 볼 수 있는 COS 도메인으로는 다음이 있습니다.



```
XXX.cos.ap-beijing.myqcloud.com (COS 기본 도메인)  
XXX.cos.accelerate.myqcloud.com (COS 글로벌 가속 도메인)  
XXX.cos-website.ap-beijing.myqcloud.com (COS 정적 페이지 도메인)  
XXX.picbj.myqcloud.com (COS CI 기본 도메인)
```

아니요. [2단계](#)를 실행합니다.

자주 볼 수 있는 COS가 아닌 도메인으로는 다음이 있습니다.



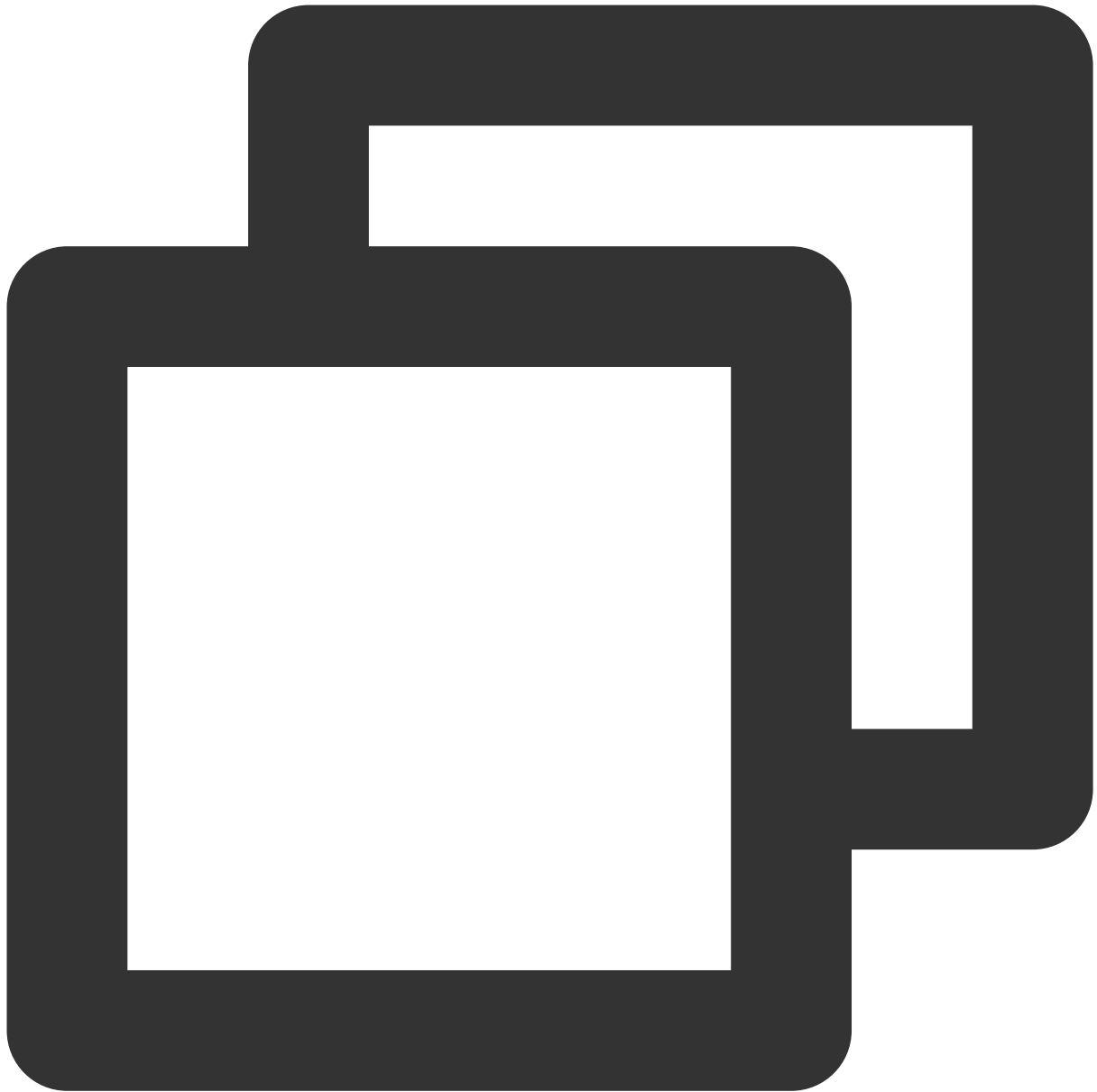
XXX.file.myqcloud.com 또는 XXX.cdn.dnsv1.com(Tencent Cloud CDN 기본 도메인)
XXX.w.kunlungr.com(aliyunCDN 기본 도메인)

2. 사용자 정의 도메인의 CNAME을 필요한 COS 도메인으로 리졸브하고 데이터 업로드를 진행합니다.

예: upload.mydomain.com cname XXX.cos.ap-beijing.myqcloud.com . 자세한 작업 방법은 [사용자 정의 일본 서버 도메인 활성화](#)를 참고하십시오.

3. 클라이언트의 기본 COS 도메인을 변경합니다.

C# 코드 예시:



```
CosXmlConfig config = new CosXmlConfig.Builder()
    .SetConnectionTimeoutMs(60000) //연결 타임아웃 시간 설정. 단위: 밀리초, 기본값: 45000ms
    .SetReadWriteTimeoutMs(40000) //읽기 및 쓰기 타임아웃 시간 설정. 단위: 밀리초, 기본값: 450
    .IsHttps(true) //기본 https 요청 설정
    .SetAppid(appid) //Tencent Cloud 계정의 계정 식별 APPID 설정
    .SetRegion(region) //기본 버킷 리전 설정
    .SetHost("XXXXXX.com") //사용자 정의 도메인 입력
    .SetDebugLog(true) .Build(); //CosXmlConfig 객체 생성
```

기타 SDK 호출은 [SDK 개요](#)를 참고하십시오.

COS 액세스 시 403 에러 코드 반환

최종 업데이트 날짜: : 2024-06-25 10:42:28

장애 현상

COS API, SDK로 리소스 업로드 및 다운로드 시 403 에러 코드 반환
임시 키 또는 서브 계정으로 COS 리소스 액세스 시 403 에러 코드 반환
COS bucket 설정 수정 시 403 에러 코드 반환

장애 분석 방식

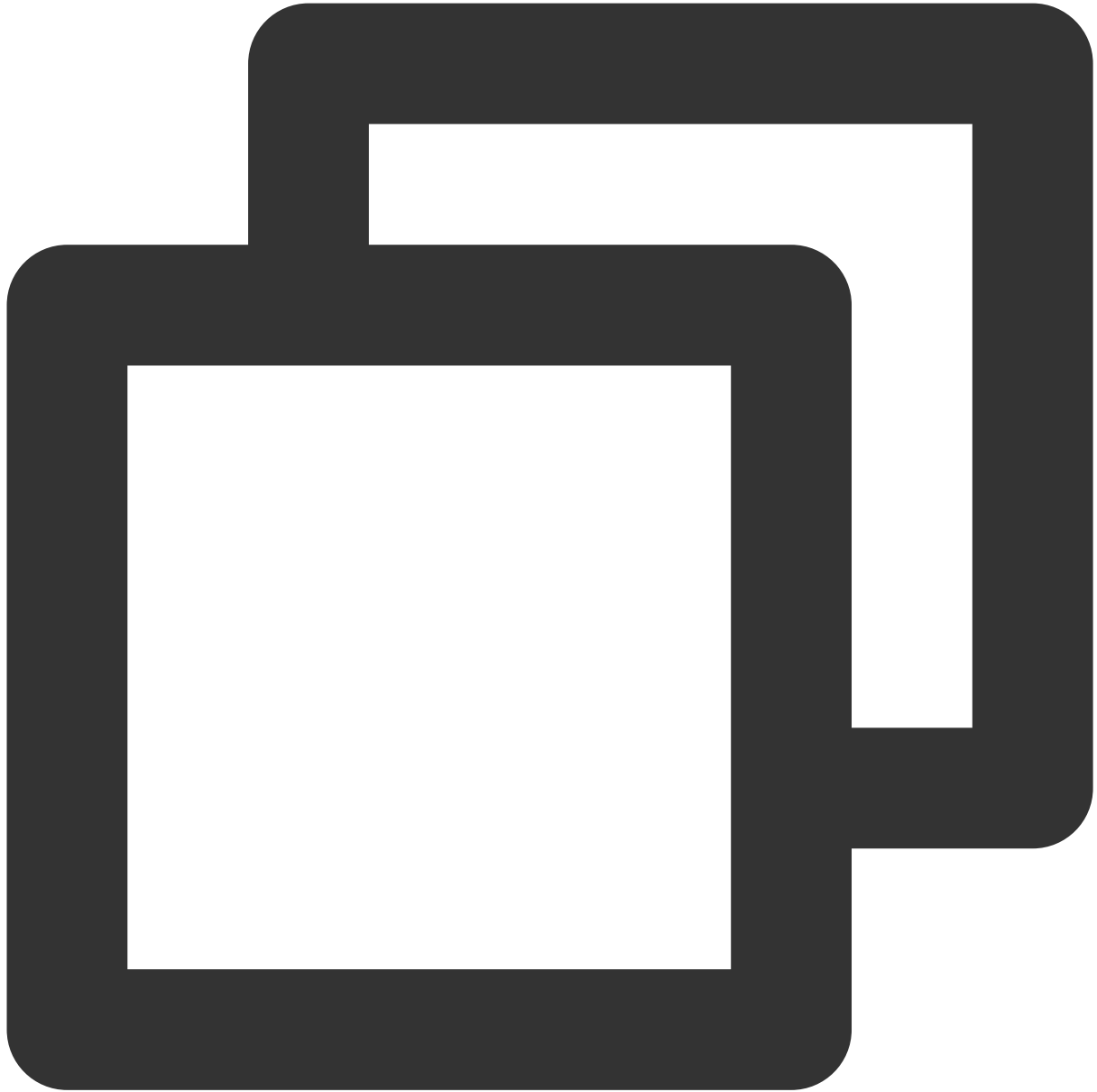
COS 요청에 403 에러 코드가 나타나면 다음 절차를 참고하여 문제의 원인을 진단할 수 있습니다.

- 요청이 CORS 크로스 도메인 액세스 요청인지 확인하십시오. 실패한 크로스 도메인 요청은 "[AccessForbidden](#)"을 반환합니다.
- 요청이 버킷 링크 도용 방지 설정에 히트되었는지 확인하십시오. 링크 도용 방지 규칙으로 인한 403 에러 코드에 대해 COS는 "[You are denied by bucket referer rule](#)"을 반환합니다.
- 요청이 익명 요청인지 확인하고, 공개 읽기가 아닌 객체에 서명이 없는 요청을 보내면 "Access Denied"를 반환합니다. 버킷 또는 객체에 대한 공개 읽기 설정은 [액세스 권한 설정](#) 또는 [객체의 액세스 권한 설정](#)을 참고하십시오.
- 요청 키와 요청 서명이 올바른지 확인합니다.
 - SecretId 사용 시 서명이 정확하지 않으면 COS는 ["[InvalidAccessKeyId](#)"]를 반환합니다.
 - 해당 기기의 시간이 정확하지 않거나 요청 시간이 서명의 유효 기간을 초과한 경우 COS는 "[RequestTimeTooSkewed](#)" 또는 "[Request has expired](#)"를 반환합니다.
 - COS는 서명 생성의 계산 방식에 문제가 있는 경우 "[SignatureDoesNotMatch](#)"를 반환합니다.
- 요청을 제안한 서브 계정 또는 임시 키에 해당 액세스 권한이 부여되었는지 확인합니다.
 - 서브 계정의 액세스 권한을 확인하고, 리소스 액세스 권한이 없는 서브 계정 요청은 "[Access Denied.](#)"가 반환됩니다.
 - 임시 키로 제안된 요청의 경우 임시 키 신청 시 작성하는 정책에 따라 임시 키 요청의 리소스 범위가 제한되며, 자세한 내용은 [임시 키 생성 및 사용 가이드](#)를 참고하십시오.
- 요청 객체가 ARCHIVE 유형인지 또는 DEEP ARCHIVE 유형인지 확인합니다. 이러한 요청 COS는 "[InvalidObjectState](#)"를 반환합니다.

장애 진단 및 처리

Message가 'Access Denied.'인 경우

COS 액세스 시 아래 정보가 출력되는 경우,



```
<Code>AccessDenied</Code>  
<Message>Access Denied.</Message>
```

다음 작업을 진행해야 합니다.

1. [COS 콘솔](#)에 로그인합니다.
2. 왼쪽 사이드바에서 **버킷 리스트**를 선택해 버킷 관리 페이지로 이동합니다.
3. 작업할 버킷을 찾아 버킷 이름을 클릭하여 버킷 설정 페이지로 이동합니다.
4. 왼쪽 사이드바에서 **권한 관리 > 버킷 액세스 권한**을 선택해 버킷 액세스 관리 페이지로 이동합니다.

5. '버킷 액세스 권한'에서 COS에 액세스하는 계정에 액세스 권한을 설정했는지 확인합니다.

예. 다음 단계를 실행합니다.

아니오. **사용자 추가**를 클릭하여 COS에 액세스하는 계정에 필요한 권한을 설정합니다.

6. 액세스 권한이 설정된 계정에 필요한 권한을 부여했는지 확인합니다.

예. 다음 단계를 실행합니다.

아니오. **편집**을 클릭하여 다시 설정하십시오.

7. 'Policy 권한 설정'에서 COS에 액세스하는 계정에 policy 라이선스 정책이 설정되어 있는지 확인합니다.

주의 :

버킷 액세스 권한이 개인 읽기이고 Policy 권한이 익명 액세스인 경우, Policy 권한의 우선순위가 버킷 액세스 권한보다 높습니다.

Policy 라이선스 정책에서 동일한 서브 계정에 허용 및 금지 정책을 동시에 설정한 경우, 금지 정책의 우선순위가 허용 정책보다 높습니다.

Policy 라이선스 정책에서 '모든 사용자' 정책의 우선순위가 '지정 사용자' 정책보다 낮습니다.

예. 다음 단계를 실행합니다.

아니오. **정책 추가**를 클릭하고 실제 서명 액세스 시 계정에 필요한 권한에 따라 설정합니다.

8. Policy 권한을 설정한 계정에 필요한 권한을 부여했는지 확인합니다.

예. 다음 단계를 실행합니다.

아니오. **편집**을 클릭하여 다시 설정하십시오.

9. COS 리소스 액세스 시 사용하는 q-ak 매개변수가 타깃 버킷에 소속된 계정(대소문자 구분)인지 확인합니다.

예. 다음 단계를 실행합니다.

아니오. q-ak 매개변수를 해당 타깃 버킷의 소속 계정으로 수정합니다.

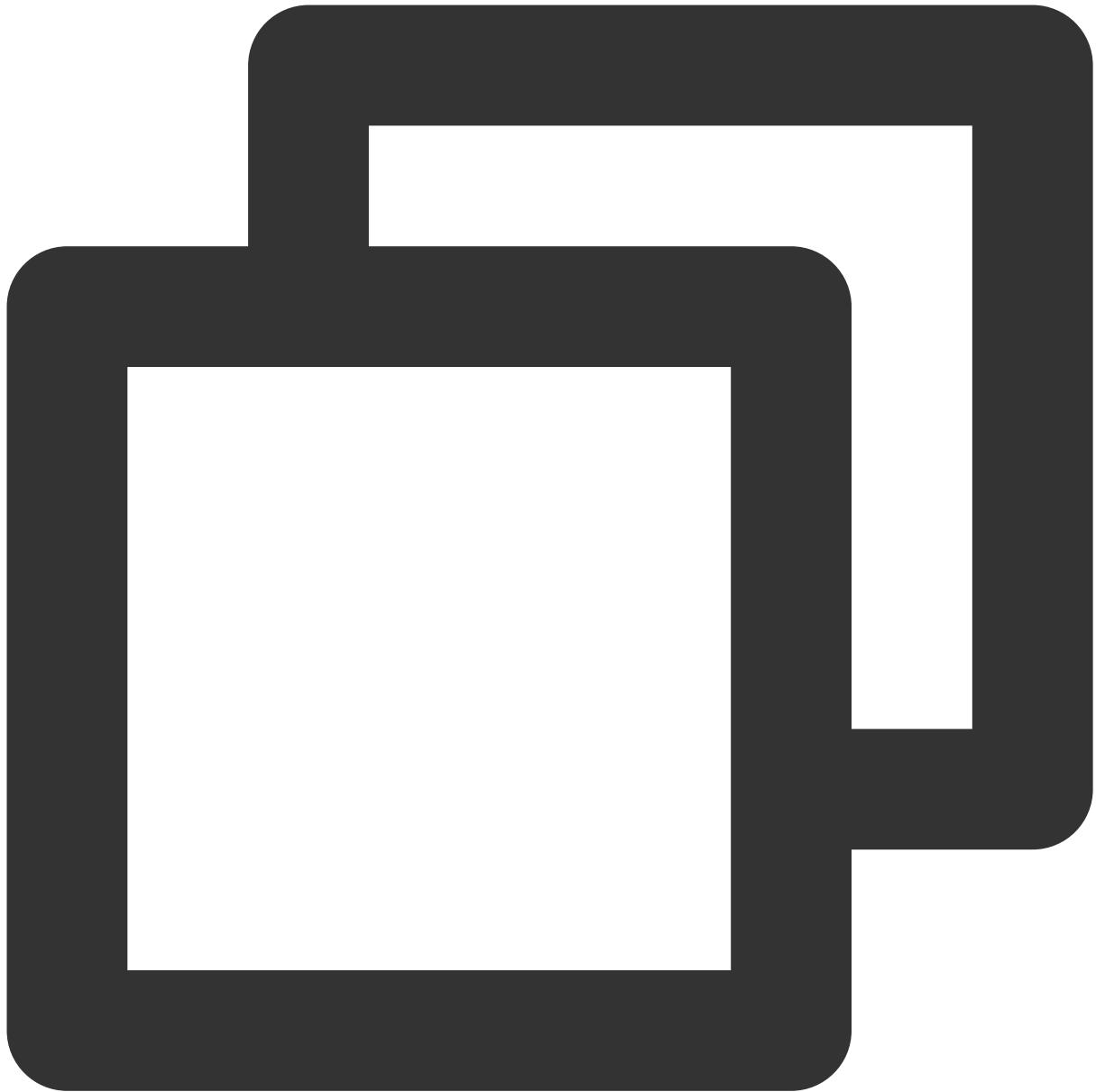
10. COS 리소스 액세스 시 교차 계정으로 액세스했는지 확인합니다.

예. 계정에 대한 교차 계정 인증을 진행하십시오. 자세한 내용은 교차 계정의 서브 계정에 지정된 파일에 대한 읽기/쓰기 권한 인증을 참고하십시오.

해당 오류가 아닌 경우: [고객센터](#)에 문의하십시오.

Message가 'AccessForbidden'인 경우

COS 액세스 시 아래 정보가 출력되는 경우,



```
<Code>AccessDenied</Code>  
<Message>AccessForbidden</Message>
```

다음 작업을 진행해야 합니다.

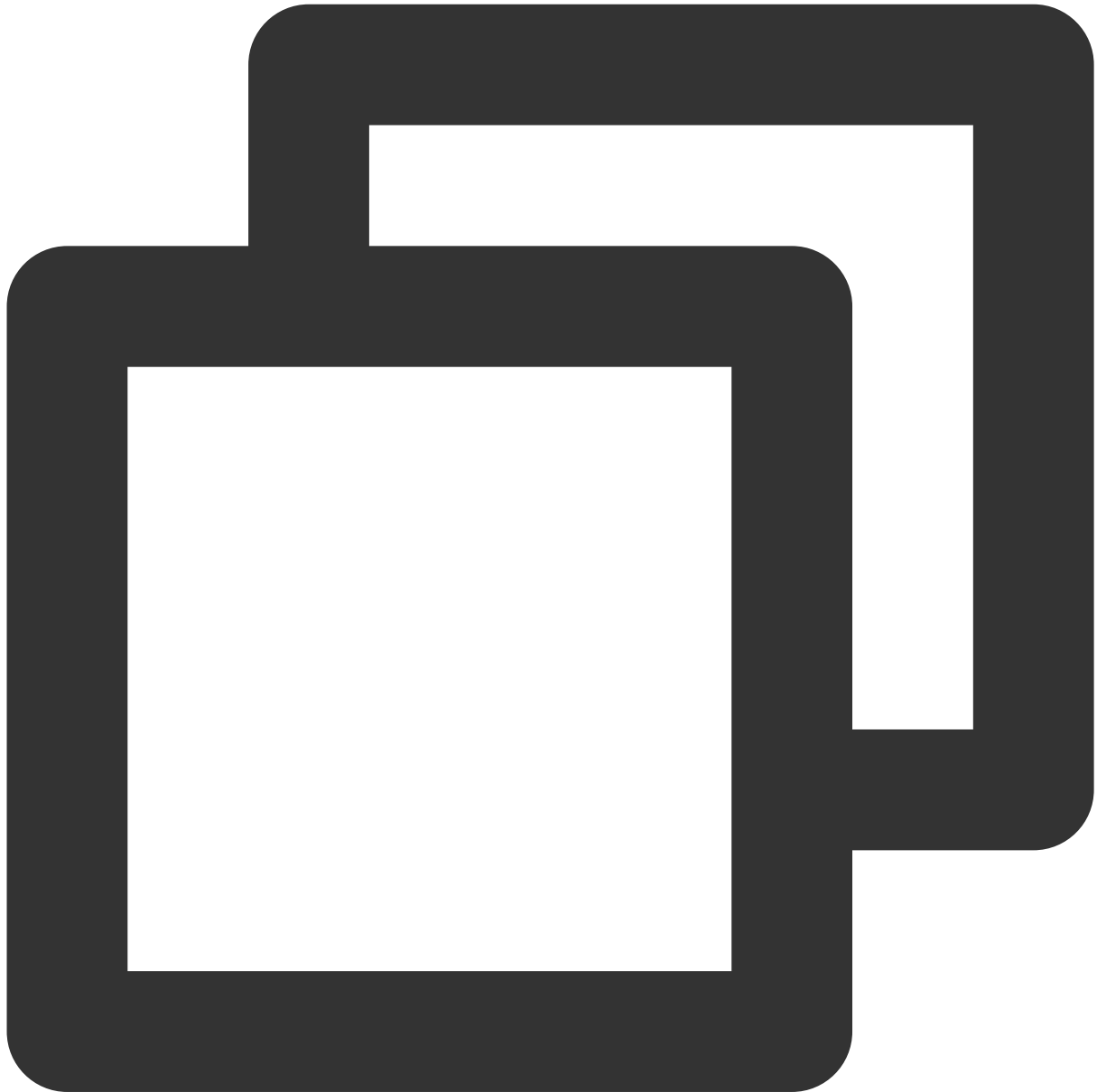
1. [COS 콘솔](#)에 로그인합니다.
2. 왼쪽 사이드바에서 **버킷 리스트**를 선택해 버킷 관리 페이지로 이동합니다.
3. 작업할 버킷을 찾아 버킷 이름을 클릭하여 버킷 설정 페이지로 이동합니다.
4. 왼쪽 사이드바에서 **보안 관리 > 크로스 도메인 액세스(CORS) 설정**을 선택해 크로스 도메인 액세스 CORS 설정 페이지로 이동합니다.

5. '크로스 도메인 액세스(CORS) 설정'에서 크로스 도메인 요청 여부를 확인합니다.

예. 다음 단계를 실행합니다.

아니요. 규칙을 수정합니다.

6. 다음 명령어를 실행하여 크로스 도메인 요청이 정확하게 설정되어 있는지 확인합니다.



```
curl 'http://bucket-appid.cos.ap-guangzhou.myqcloud.com/object' -voa /dev/null -H '
```

다음 정보가 반환되면 정확하게 설정되었다는 의미입니다.



```
<Code>AccessDenied</Code>  
<Message>You are denied by bucket referer rule</Message>
```

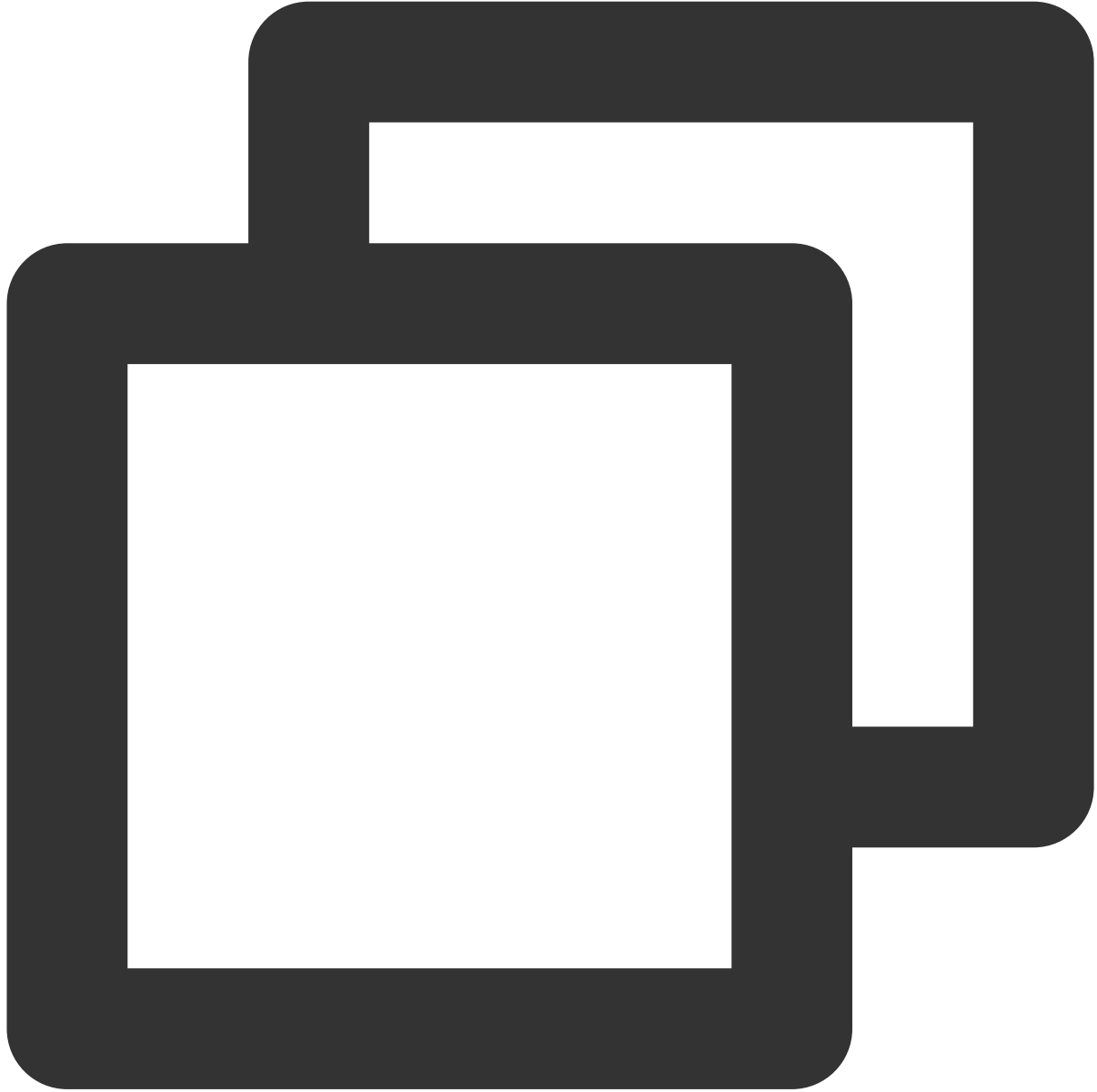
다음 작업을 진행해야 합니다.

1. [COS 콘솔](#)에 로그인합니다.
2. 왼쪽 사이드바에서 **버킷 리스트**를 선택해 버킷 관리 페이지로 이동합니다.
3. 작업할 버킷을 찾아 버킷 이름을 클릭하여 버킷 설정 페이지로 이동합니다.
4. 왼쪽 사이드바에서 **보안 관리 > 링크 도용 방지 설정**을 선택해 링크 도용 방지 설정 페이지로 이동합니다.
5. '링크 도용 방지 설정'에서 링크 도용 방지를 설정했는지 확인합니다.

예. 다음 단계를 실행합니다.

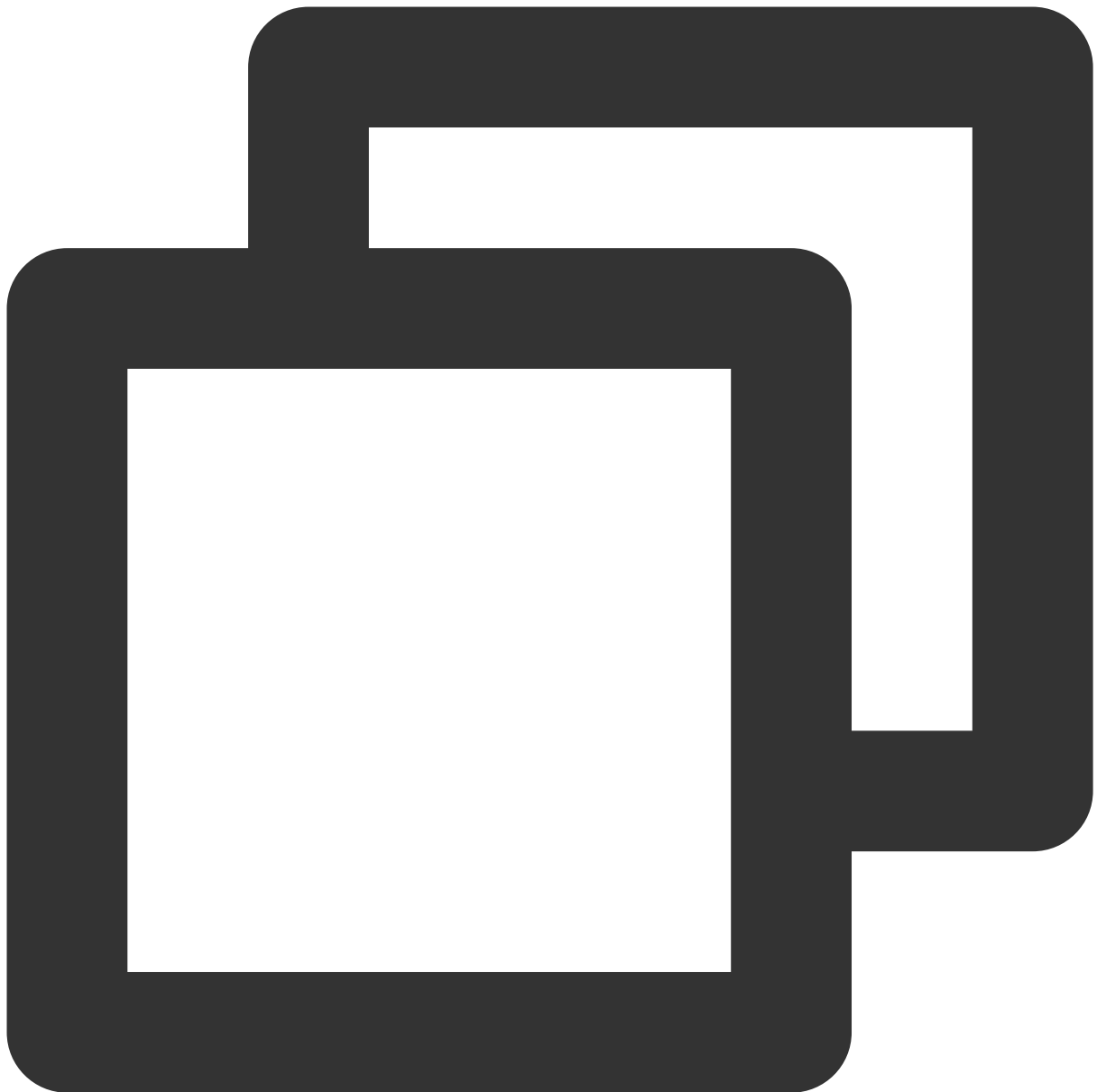
해당 오류가 아닌 경우: [고객센터](#)에 문의하십시오.

6. 다음 명령어를 실행하여 링크 도용 방지가 정확하게 설정되어 있는지 확인합니다.



```
curl 'http://bucket-appid.cos.ap-guangzhou.myqcloud.com/object' -voa /dev/null -H '
```

다음 정보가 반환되면 정확하게 설정되었다는 의미입니다.



```
<Code>AccessDenied</Code>  
<Message>InvalidAccessKeyId</Message>
```

다음 작업을 진행해야 합니다.

1. 요청 서명 **Authorization**의 **q-ak** 매개변수를 정확하게 입력했는지 확인합니다.

예. 다음 단계를 실행합니다.

아니요. **q-ak** 매개변수를 수정합니다. 키의 **SecretId**는 **q-ak** 매개변수와 동일해야 하며 대소문자를 구분합니다.

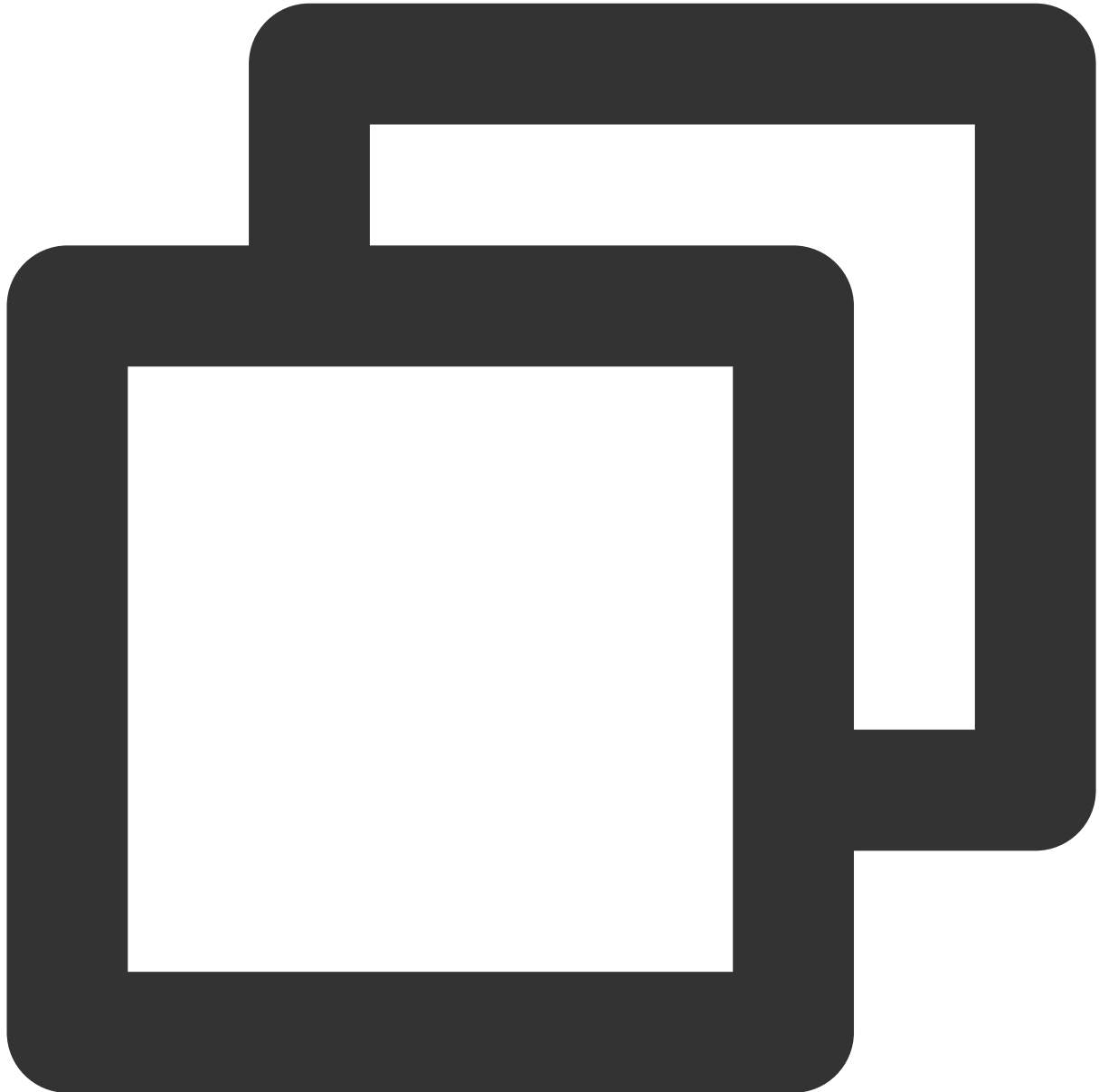
2. [API 키 관리](#)로 이동해 API 키가 활성화되어 있는지 확인합니다.

해당 오류일 경우: [고객센터](#)에 문의하십시오.

아니요. 해당 API 키를 활성화합니다.

Message가 'InvalidObjectState'인 경우

COS 액세스 시 아래 정보가 출력되는 경우,



```
<Code>AccessDenied</Code>  
<Message>InvalidObjectState</Message>
```

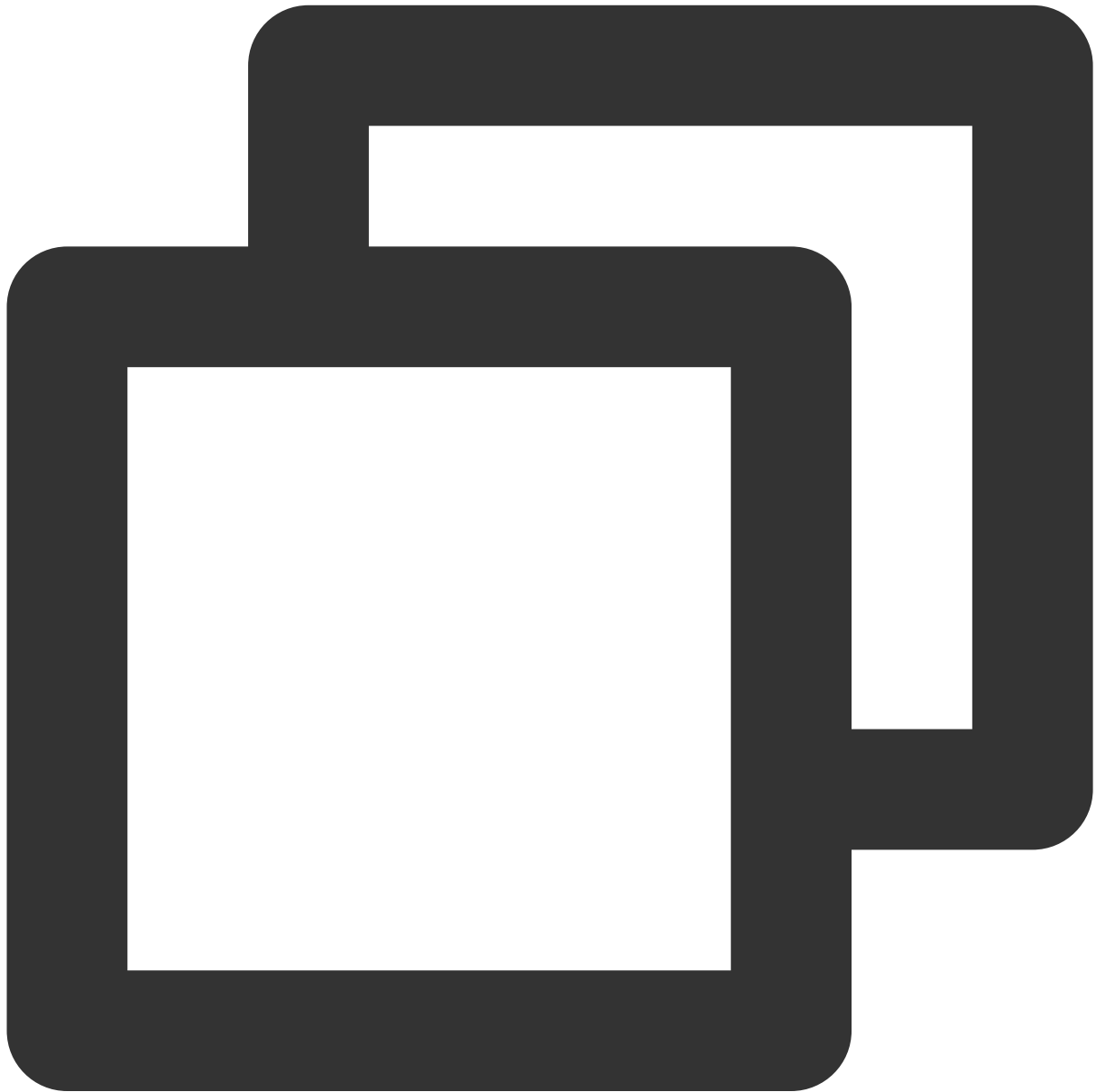
다음 작업을 진행해야 합니다.

요청하는 객체가 CAS 또는 DEEP ARCHIVE 유형인지 확인합니다.

예. 객체 복구 후 다시 액세스합니다. 자세한 작업 방법은 [POST Object restore](#)를 참조하십시오.
해당 오류가 아닌 경우: [고객센터](#)에 문의하십시오.

Message가 'RequestTimeTooSkewed'인 경우

COS 액세스 시 아래 정보가 출력되는 경우,



```
<Code>AccessDenied</Code>  
<Message>RequestTimeTooSkewed</Message>
```

다음 작업을 진행해야 합니다.

1. 운영 체제 유형에 따라 클라이언트의 현재 시간을 확인합니다.

Windows 시스템(Windows Server 2012 기준):



> 제어판 > 시간, 언어 및 지역 > 날짜 및 시간.

Linux 시스템: `date -R` 명령어 실행

```
(base) [root@VM_32_5_centos ~]# date -R
Tue, 30 Mar 2021 11:44:00 +0800
```

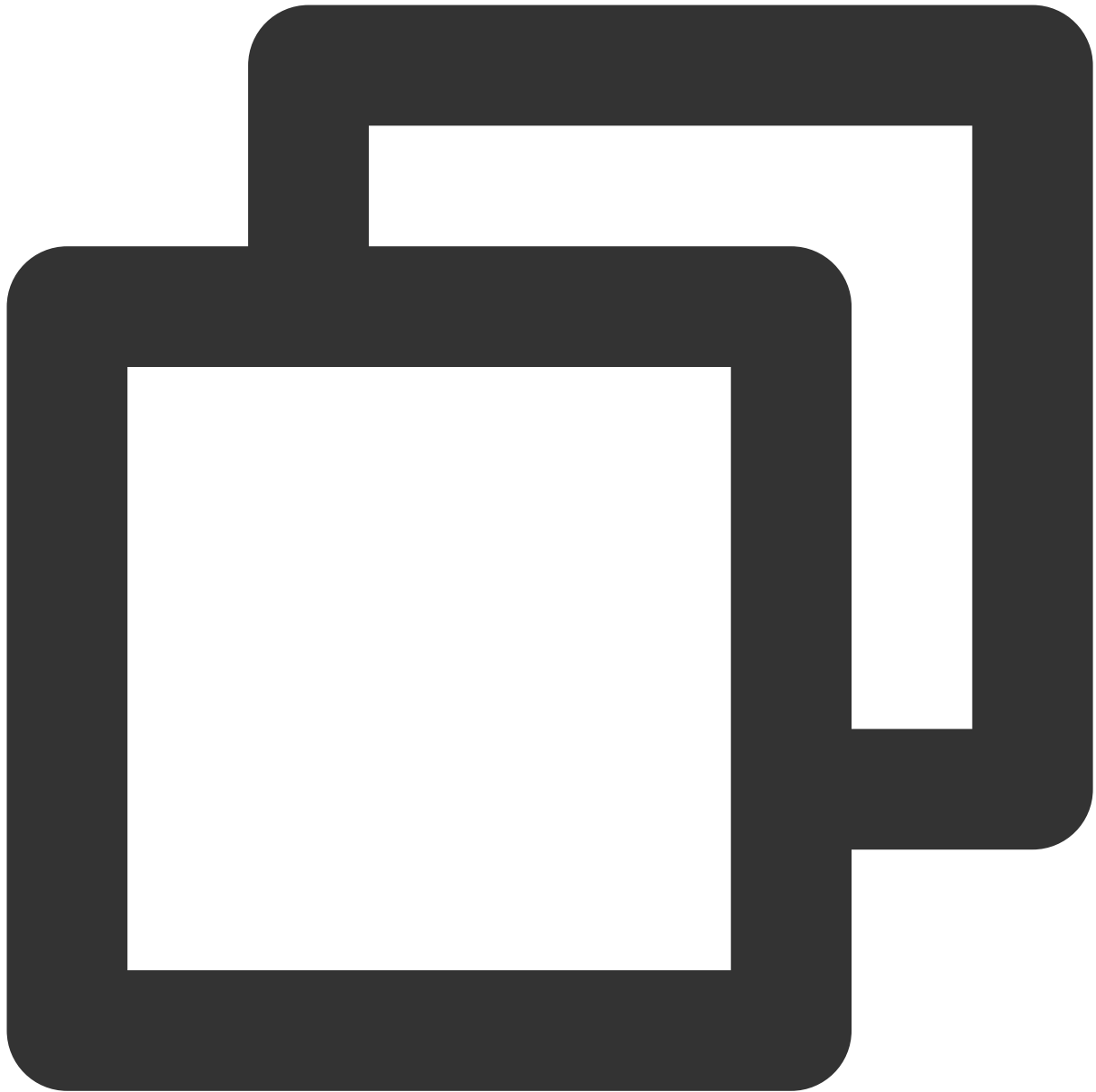
2. 클라이언트의 현재 시간과 서버 시간에 오차가 있는지 오차 시간이 15분을 초과하는지 확인합니다.

예. 시간을 동기화합니다.

해당 오류가 아닌 경우: [고객센터](#)에 문의하십시오.

Message가 'Request has expired'인 경우

COS 액세스 시 아래 정보가 출력되는 경우,



```
<Code>AccessDenied</Code>  
<Message>Request has expired</Message>
```

예상 원인은 다음과 같습니다.

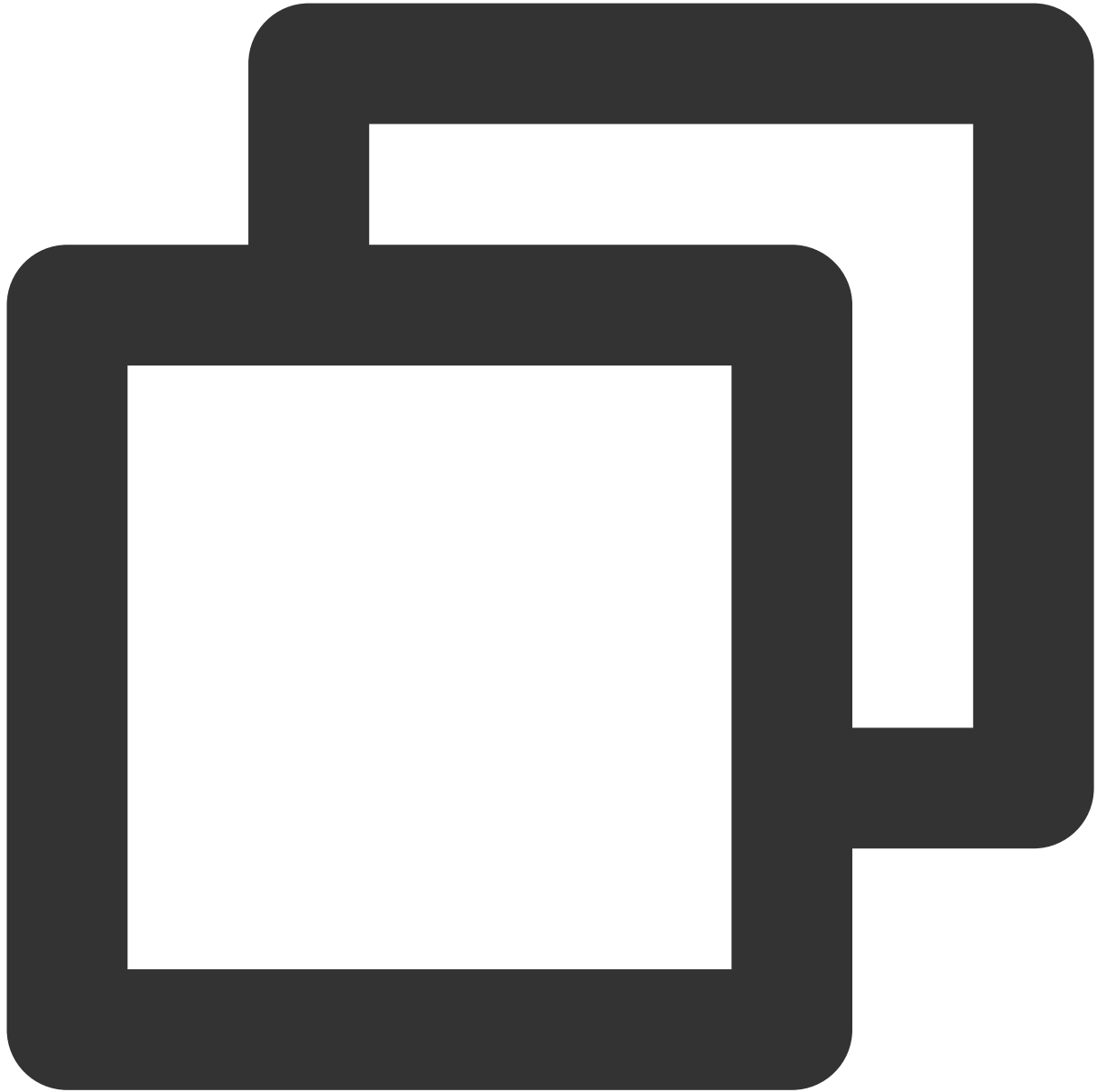
요청 시작 시간이 서명 유효 시간을 초과한 경우

로컬 시스템 시간과 해당 지역 시간이 일치하지 않는 경우

서명 유효시간을 다시 설정하거나 로컬 시스템 시간을 동기화해야 합니다. 해당 방법으로도 해결되지 않는 경우 [고객 센터](#)로 문의하십시오.

Message가 'SignatureDoesNotMatch'인 경우

COS 액세스 시 아래 정보가 출력되는 경우,



```
<Code>AccessDenied</Code>  
<Message>SignatureDoesNotMatch</Message>
```

다음 작업을 진행해야 합니다.

클라이언트가 계산한 서명과 서버가 계산한 서명이 일치하는지 확인합니다.

해당 오류일 경우: [고객센터](#)에 문의하십시오.

아니요. [Request Signature](#) 문서를 참조하여 COS 서명 툴을 사용해 직접 구현되는 서명 과정을 확인하십시오.

리소스 액세스 오류

파일 액세스 시 404 NoSuchKey 반환

최종 업데이트 날짜: : 2024-06-25 10:42:28

현상 설명

파일 액세스 시 파일을 찾지 못하거나 파일이 표시되지 않고 404 NoSuchKey 에러 코드를 반환합니다.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Error>
  <Code>NoSuchKey</Code>
  <Message>The specified key does not exist.</Message>
  <Resource>cos.ap-guangzhou.myqcloud.com/3.gif</Resource>
  <RequestId>NjA4MjM4NjRlMjM0OTRkMTBiOThmMDAxODc0OWRkZjk0ZDMlNmI1M2E2MTRlY2ZlZDhmNmI5</RequestId>
  <TraceId>OGVmYzZiMmQzYjA2OWNhODk0NTRkMTBiOThmMDAxODc0OWRkZjk0ZDMlNmI1M2E2MTRlY2ZlZDhmNmI5</TraceId>
</Error>
```

예상 원인

액세스하는 파일 경로 입력 오류

액세스하는 파일 경로의 대소문자 불일치

처리 순서

액세스하는 파일 경로 및 대소문자가 정확한지 확인합니다.

예. [고객센터](#)로 문의하십시오.

아니요. 파일 경로를 수정하십시오. 객체(Object)의 이름 생성 규칙은 [객체 개요](#)를 참조하십시오.

비디오 액세스 시 정상적으로 재생되지 않는 문제

최종 업데이트 날짜: : 2024-06-25 10:42:28

현상 설명

비디오 재생 시 정상적으로 재생되지 않습니다. 예를 들어, 오디오나 화면 둘 중 하나만 나오거나 아예 재생되지 않을 때도 있습니다.

예상 원인

원본 비디오의 인코딩 포맷에 문제가 있는 경우

원본 비디오가 손상되어 비디오 업로드 후 재생 시 오디오 또는 화면만 나오거나 재생되지 않는 경우

처리 순서

1. 원본 비디오의 인코딩 포맷이 올바른지 확인합니다.

예. 다음 단계를 실행합니다.

아니요. 올바른 인코딩 포맷의 비디오를 업로드합니다. 자세한 요구 사항은 [규격 및 제한](#)을 참조하십시오.

2. VLC 등 전문 플레이어를 사용해 원본 비디오가 정상적으로 재생되는지 확인합니다.

예. [고객센터](#)로 문의하십시오.

아니요. Tencent Cloud의 트랜스 코딩 제품을 사용해 원본 비디오를 복구해 보시기 바랍니다. 자세한 내용은 [비디오 트랜스 코딩](#)을 참조하십시오.

URL이 잘못된 파일로 연결되는 오류

최종 업데이트 날짜: : 2024-06-25 10:42:28

현상

현상1: 동일한 링크에 액세스하였는데 다른 파일에 연결됩니다.

현상2: 파일을 업데이트했는데 이전 파일로 연결됩니다.

가능한 원인

Content Delivery Network(CDN) 캐시가 만료되지 않았습니다.

브라우저에서 로컬 캐시 사용을 비활성화하지 않았습니다.

액세스하는 파일이 하이재킹되어 액세스하는 리소스 콘텐츠와 설정된 콘텐츠가 일치하지 않습니다.

처리 단계

CDN 캐시 만료 여부 확인

[CDN 캐시 설정 FAQ](#)를 참고하여 CDN 캐시 만료 여부를 확인합니다.

만료된 경우, [브라우저에서 로컬 캐시 사용 비활성화 여부](#)를 확인하십시오.

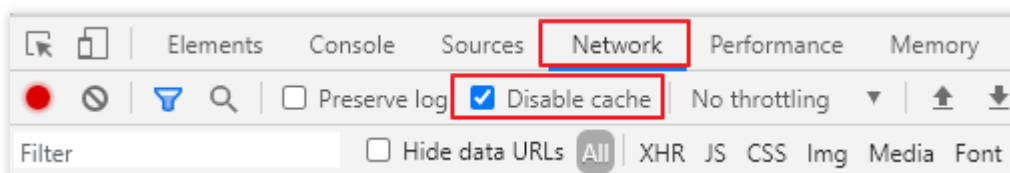
만료되지 않은 경우, [캐시 퍼지](#) 문서를 참조하여 CDN URL 또는 CDN 디렉터리를 퍼지하십시오.

브라우저에서 로컬 캐시 사용 비활성화 여부 확인

설명 :

다음은 Google 브라우저 작업 예시입니다.

1. Google 브라우저를 실행합니다.
2. **F12** 키를 눌러 디버깅 창을 엽니다.
3. **Network** 탭에서 **Disable cache**가 선택되었는지 확인합니다.



선택된 경우, [액세스하는 파일이 하이재킹되었는지](#) 확인하십시오.

선택되지 않은 경우, **Disable cache**를 선택하고 브라우저를 재시작 하십시오.

액세스하는 파일이 하이재킹되었는지 확인

액세스하는 리소스 콘텐츠가 설정과 일치하지 않는 경우(예: 파일의 `content-length` 불일치, 응답 `header` 불일치 등) 하이재킹된 것입니다. HTTPS 프로토콜을 사용하여 해당 파일에 액세스할 것을 권장합니다.

CDN 도메인으로 COS 액세스 시 HTTP ERROR 403 반환

최종 업데이트 날짜: : 2024-06-25 10:42:28

현상 설명

Content Delivery Network(CDN) 도메인을 사용해 Cloud Object Storage(COS)에 액세스하는 경우 HTTP ERROR 403 에러 코드 반환

예상 원인

CDN 가속 도메인이 비활성화 상태인 경우

처리 순서

1. [COS 콘솔](#)에 로그인합니다.
2. 왼쪽 메뉴에서 [버킷 리스트]를 선택해 버킷 관리 페이지로 이동합니다.
3. 작업할 버킷을 찾아 버킷 이름을 클릭하여 버킷 설정 페이지로 이동합니다.
4. 왼쪽 메뉴에서 [도메인 및 전송 관리]>[기본 CDN 가속 도메인]을 선택해 기본 CDN 가속 도메인 페이지로 이동합니다.
5. '기본 CDN 가속 도메인'에서 현재 상태가 비활성화 상태인지 확인합니다.
예. [기본 CDN 가속 도메인 활성화](#)를 참조하십시오.
아니요. 다음 단계를 실행합니다.
6. '사용자 정의 CDN 가속 도메인'에서 상태가 런칭으로 되어 있는지 확인합니다.
예. [고객센터](#)로 문의하십시오.
아니요. [사용자 정의 CDN 가속 도메인 활성화](#)를 참조하십시오.

HTTPS를 통한 사용자 정의 원본 도메인 액세스 실패

최종 업데이트 날짜: : 2024-06-25 10:42:28

현상 설명

HTTPS 프로토콜을 사용하여 사용자 정의 원본 서버 도메인 액세스 시 오류

예상 원인

인증서 설정 오류 또는 사용자 정의 원본 서버 도메인을 설정하지 않은 경우

처리 단계

CDN 인증서 사용

1. [CDN 콘솔](#)에 로그인합니다.
 2. 왼쪽 사이드바에서 **도메인 관리**를 선택해 도메인 관리 페이지로 이동합니다.
 3. 설정할 도메인을 클릭하고 **HTTPS 설정** 탭을 선택해 HTTPS 설정 페이지로 이동합니다.
 4. 'HTTPS 설정'에서 **설정하기**를 클릭해 인증서 설정 페이지로 이동합니다.
- 자세한 내용은 CDN의 [인증서 설정](#) 문서를 참고하십시오.
5. CDN 도메인이 다시 배포를 완료할 때까지 5분 정도 기다리면 HTTPS 프로토콜 액세스 설정이 완료됩니다.

COS 인증서 사용

1. [COS 콘솔](#)에 로그인합니다.
 2. 왼쪽 사이드바에서 **버킷 리스트**를 선택해 버킷 관리 페이지로 이동합니다.
 3. 작업할 버킷을 찾아 버킷 이름을 클릭하여 버킷 설정 페이지로 이동합니다.
 4. 왼쪽 사이드바에서 **도메인 및 전송 관리 > 사용자 정의 원본 서버 도메인**을 선택해 사용자 정의 원본 서버 도메인 페이지로 이동합니다.
 5. 작업할 도메인을 선택하고 **인증서 바인딩**을 클릭해 인증서를 설정합니다.
 6. **확인**을 클릭하여 연결합니다.
- 'HTTPS 인증서'가 '업로드됨' 상태라면 HTTPS 프로토콜 액세스 설정이 완료된 것입니다.

CVM 리버스 프록시 인증서 사용

[사용자 정의 도메인 설정으로 HTTPS 액세스 지원](#)을 참고하십시오.

POST Object 자주 발생하는 오류

최종 업데이트 날짜: : 2024-06-25 10:42:28

장애 현상

COS API를 사용하여 POST 요청 시 다음 예외 오류 코드가 반환됩니다.

Condition key q-ak doesn't match the value XXXXXX

You post object request has been expired, expiration time: 1621188104 but the time now : 1621245817

The Signature you specified is invalid.

You must provide condition if you specify a policy in post object request.

Condition key bucket doesn't match the value [bucket-appid]

Condition key key doesn't match the value XXXXX

The body of your POST request is not well-formed multipart/form-data.

장애 진단 및 처리

Message: "Condition key q-ak doesn't match the value XXXXXX"

COS API를 사용하여 POST 요청 시 다음 메시지가 나타납니다.



```
<Code>AccessDenied</Code>  
<Message>Condition key q-ak doesn't match the value XXXXXX</Message>
```

예상 원인

q-ak 매개변수 입력 오류.

해결 방법

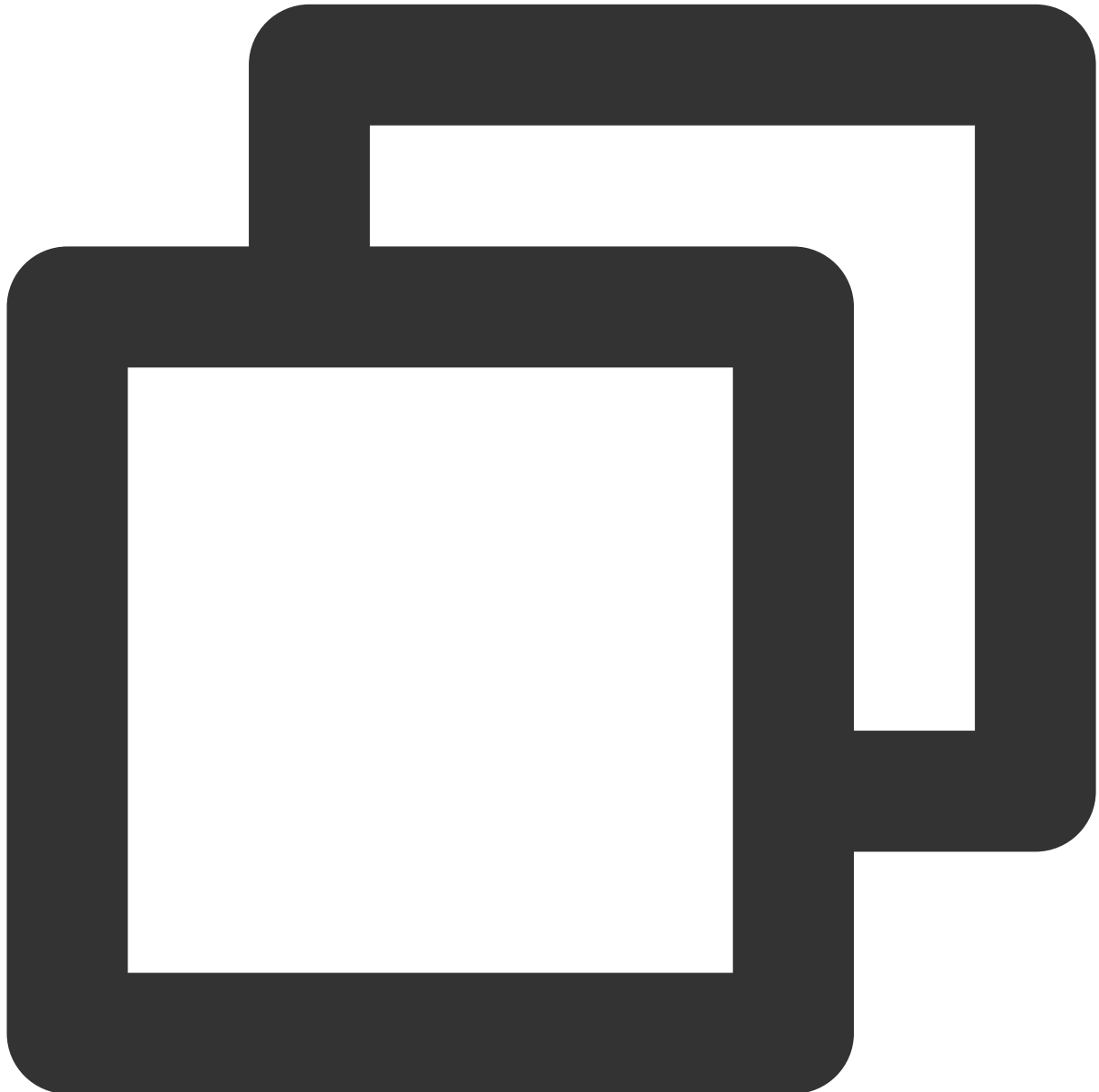
1. 액세스 관리 콘솔 로그인 후, [\[API 키 관리\]](#) 페이지로 이동하여 키 정보를 확인하십시오.
2. 조회된 키 정보에 따라 q-ak 매개변수가 잘못 입력되었는지 확인합니다.

해당 오류인 경우: q-ak 매개변수를 올바른 SecretId로 수정하십시오.

해당 오류가 아닌 경우: [고객센터](#)에 문의하십시오.

Message: “You post object request has been expired, expiration time: 1621188104 but the time now : 1621245817”

COS API를 사용하여 POST 요청 시 다음 메시지가 나타납니다.



```
<Code>AccessDenied</Code>
```

```
<Message>You post object request has been expired, expiration time: 1621188104 but
```

예상 원인

Policy의 expiration 값 만료.

해결 방법

Policy 중의 expiration 값을 수정하십시오.

주의 :

expiration 값은 현재 시간 이후여야 하며, 현재 시간 + 30분(UTC 시간)으로 설정하는 것을 권장합니다.

Message: “The Signature you specified is invalid.”

COS API를 사용하여 POST 요청 시 다음 메시지가 나타납니다.



```
<Code>SignatureDoesNotMatch</Code>  
<Message>The Signature you specified is invalid.</Message>
```

예상 원인

서명 계산 오류.

해결 방법

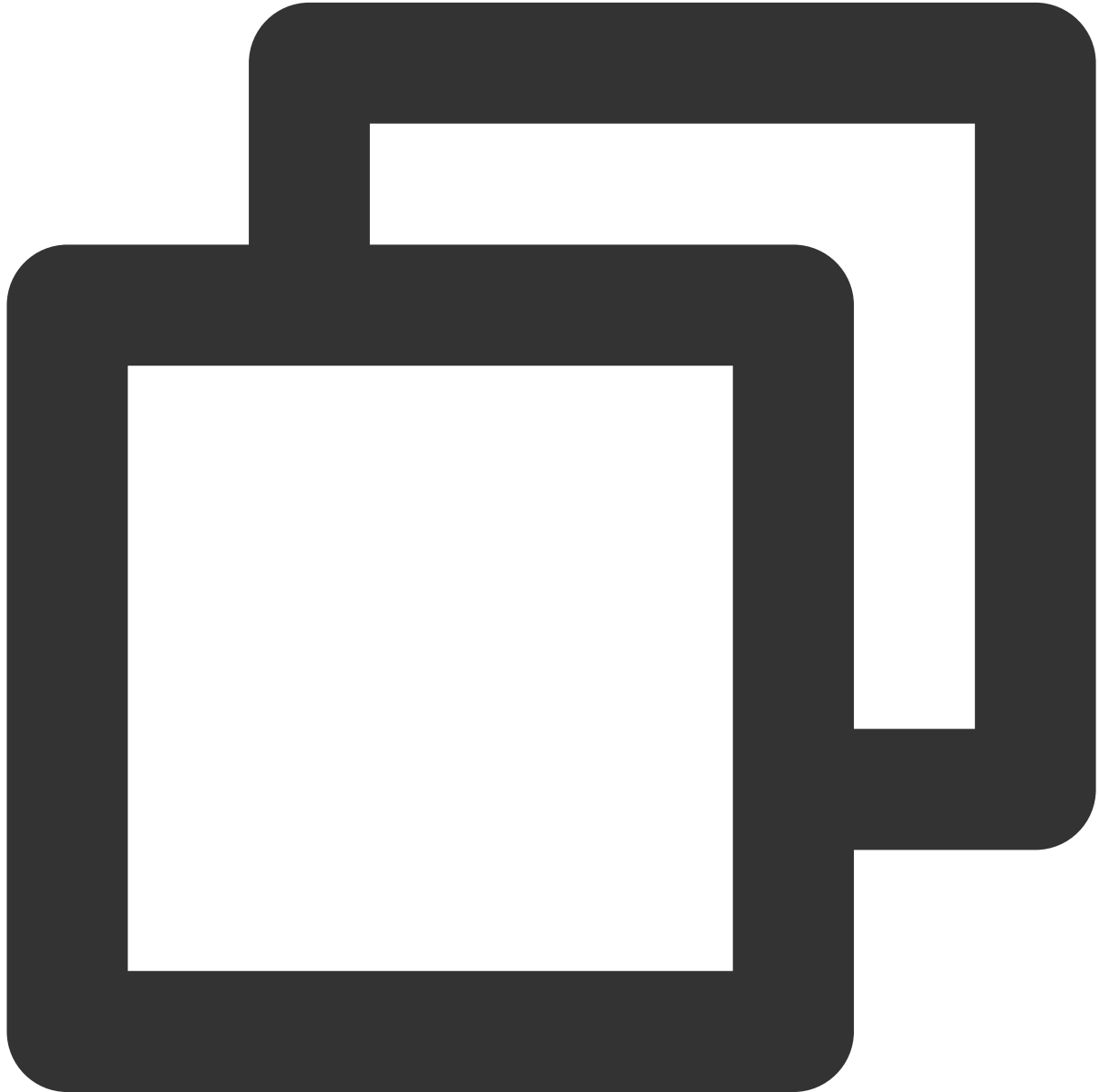
[서명 요청](#) 문서를 참조하여 POST 서명 문자열 생성 규칙이 올바른지 확인하십시오.

해당 오류일 경우: [고객센터](#)에 문의하십시오.

해당 오류가 아닐 경우: 온라인 보조 툴: COS 서명 툴을 사용하여 POST 요청 서명을 다시 계산하십시오.

Message: “You must provide condition if you specify a policy in post object request.”

COS API를 사용하여 POST 요청 시 다음 메시지가 나타납니다.



```
<Code>InvalidPolicyDocument</Code>
```

```
<Message>You must provide condition if you specify a policy in post object request.
```

예상 원인

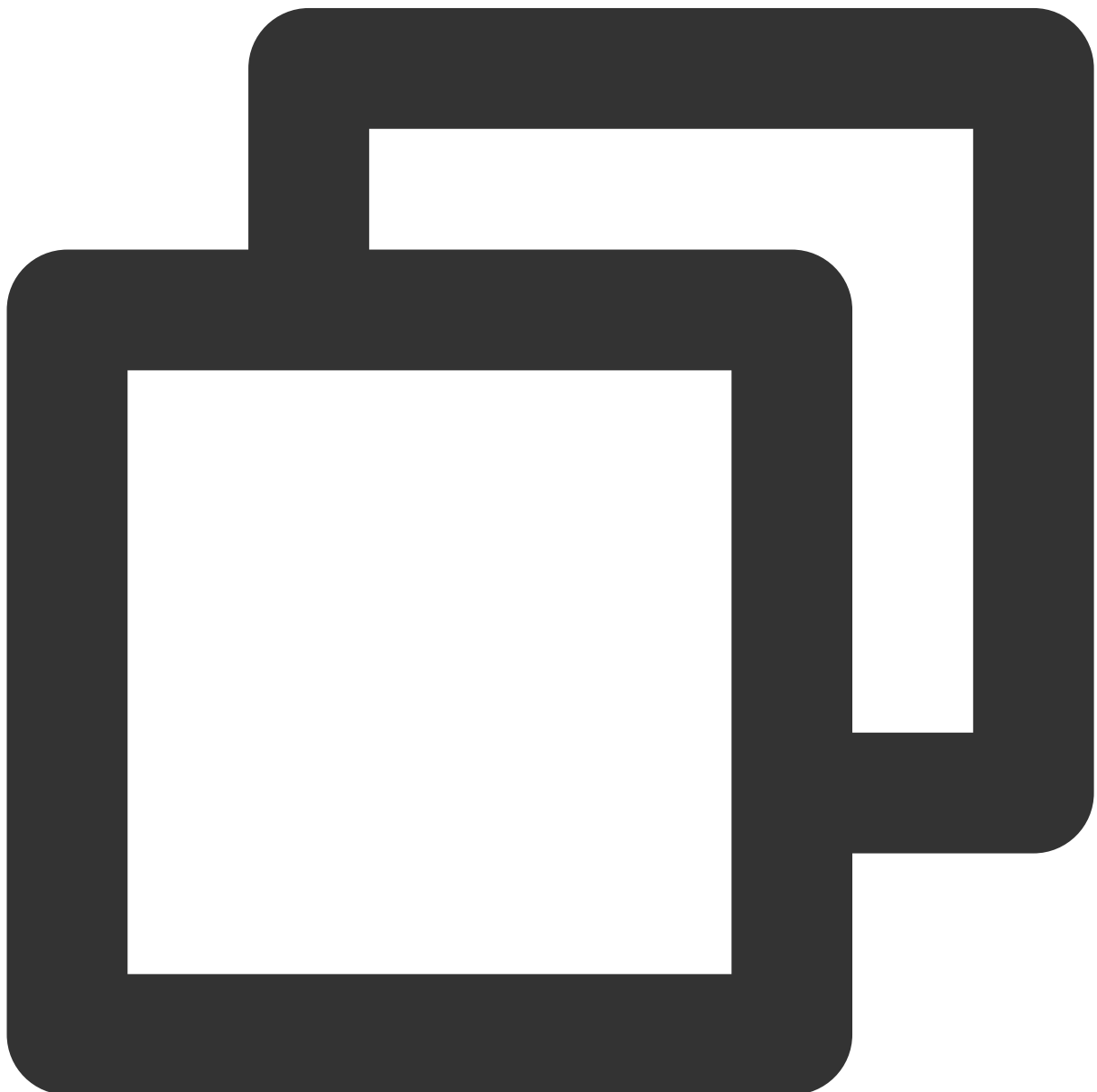
Policy 형식 오류.

해결 방법

[POST Object](#) 문서를 참고하여, Policy 형식을 표준 JSON형식으로 수정하십시오.

Message: “Condition key bucket doesn’t match the value [bucket-appid]”

COS API를 사용하여 POST 요청 시 다음 메시지가 나타납니다.



```
<Code>AccessDenied</Code>  
<Message>Condition key bucket doesn't match the value [bucket-appid]</Message>
```

예상 원인

Policy의 bucket이 요청 bucket과 불일치함.

해결 방법

Policy의 bucket을 사용하여 요청하십시오.

Message: “Condition key key doesn't match the value XXXXX”

COS API를 사용하여 POST 요청 시 다음 메시지가 나타납니다.



```
<Code>AccessDenied</Code>  
<Message>Condition key key doesn't match the value XXXXX</Message>
```

예상 원인

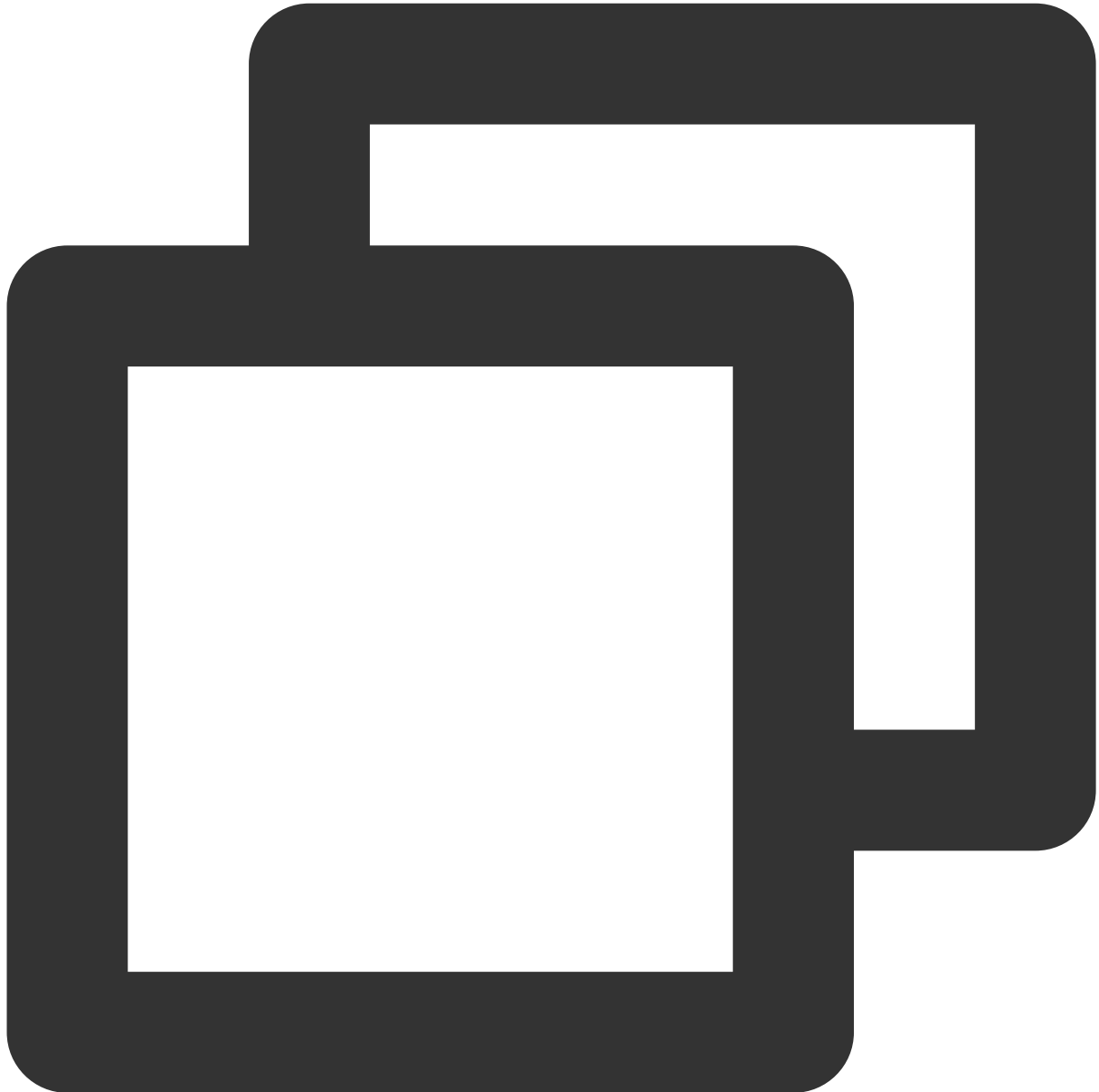
업로드한 콘텐츠가 policy 규칙에 부합하지 않음.

해결 방법

Policy의 Condition에 따라 해당 조건에 부합하는 콘텐츠를 업로드 합니다.

Message: “The body of your POST request is not well-formed multipart/form-data.”

COS API를 사용하여 POST 요청 시 다음 메시지가 나타납니다.



```
<Code>MalformedPOSTRequest</Code>
```

```
<Message>The body of your POST request is not well-formed multipart/form-data.</Mes
```

예상 원인

규범에 맞지 않는 POST body 형식.

해결 방법

[POST Object](#) 문서를 참고하여 body 형식을 최적화합니다.