

Cloud Object Storage

Troubleshooting

Product Documentation



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.

Contents

Troubleshooting

- Obtaining RequestId

- Slow Upload over Public Network

- 403 Error for COS Access

- Resource Access Error

 - 404 NoSuchKey Is Returned

 - A Video Cannot Be Played

 - A URL Points to a Wrong File

 - “HTTP ERROR 403” Is Returned When I Access COS Using a CDN Domain

 - Accessing a Custom Origin Domain over HTTPS Failed

- POST Object Common Exceptions

Troubleshooting

Obtaining RequestId

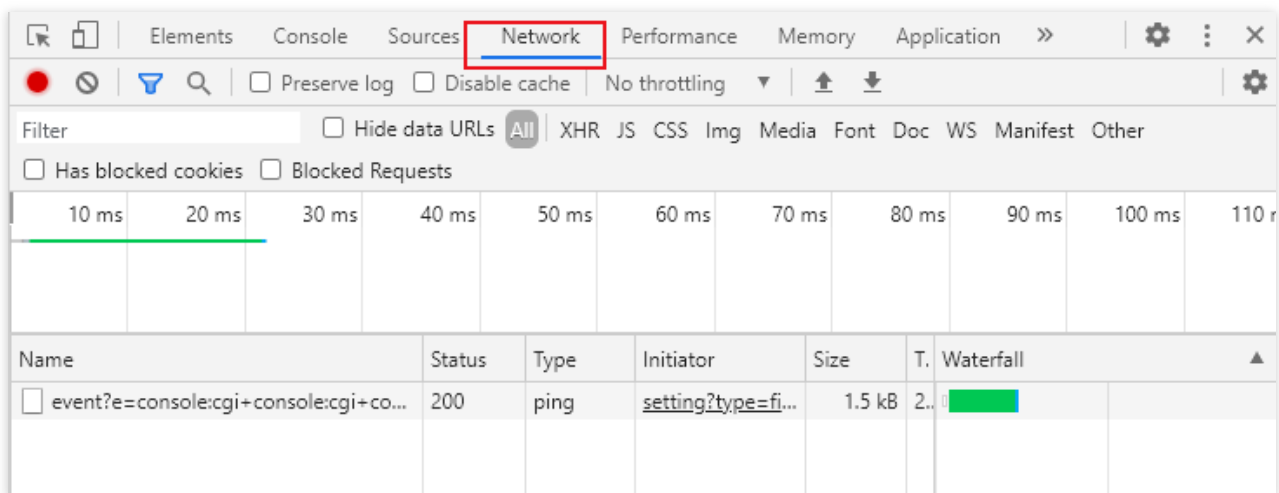
Last updated : 2024-03-25 18:01:58

Overview

The COS server will generate an ID (`RequestId`) for every request sent to COS. This document describes how to obtain `RequestId` in different scenarios.

Using the Console

1. Log in to the [COS console](#). Click **Bucket List** on the left sidebar to open the bucket list page.
2. Click the name of the target bucket.
3. Press F12 on the keyboard to open the developer tools of your browser.
4. Select the **Network** tab.



5. Click **Download** on the right of the target object. Then, in the developer tools, enter the filename in the **Filter** text box, select the file, and click **Headers**. You can then find `RequestId` from **Response Headers**.

From an Unsuccessful Access

When you fail to access an object, you can obtain the **RequestId** from the XML file returned.

```
<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>[REDACTED]</Resource>
<RequestId>NjBjYWNmYmFmJjJiMjU4NjRfNGJhN180YT.[REDACTED]</RequestId>
<TraceId>OGVmYzZiMmQyYjA5OWNmODkUNTRkMTBlOWVmMDAxODcUOWRkZjJkUzDMlNmI1M2EzMTRlY2MzZDhmNmI5MWI1OWI2NmQ0YjJkZWE3N
</Error>
```

1. Press F12 on the keyboard to open the developer tools of your browser.
2. Select the **Network** tab and select **All**. You can then find `RequestId` from **Response Headers**.



Using the .NET SDK



```
try
{
    string bucket = "examplebucket-1250000000"; // Bucket name in the format of Bucket
    string cosPath = "test.cs"; // Object key
    byte[] data = System.Text.Encoding.Default.GetBytes("Hello COS"); // Binary data
    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)
{
    // Request failed
    Console.WriteLine("CosClientException: " + clientEx);
}
catch (COSXML.CosException.CosServerException serverEx)
{
    // Request failed
    Console.WriteLine("CosServerException: " + serverEx.GetInfo());
}
```

Using the Go SDK



```
package main

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



```
func main(){
    // Replace `examplebucket-1250000000` and `COS_REGION` with the actual informati
    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",
        },
    })
    // An object key is the unique identifier of an object in a bucket
    // For example, in the access domain name `examplebucket-1250000000.cos.COS_REGI
    name := "test.go"
    // 1. Upload the object with a string.
    f := strings.NewReader("Hello COS")

    response, err := c.Object.Put(context.Background(), name, f, nil)
    if err != nil{
        // The error message contains the RequestId field.
        panic(err)
    }
    requestId := response.Header.Get("X-Cos-Request-Id")
    fmt.Println(requestId)
}
```

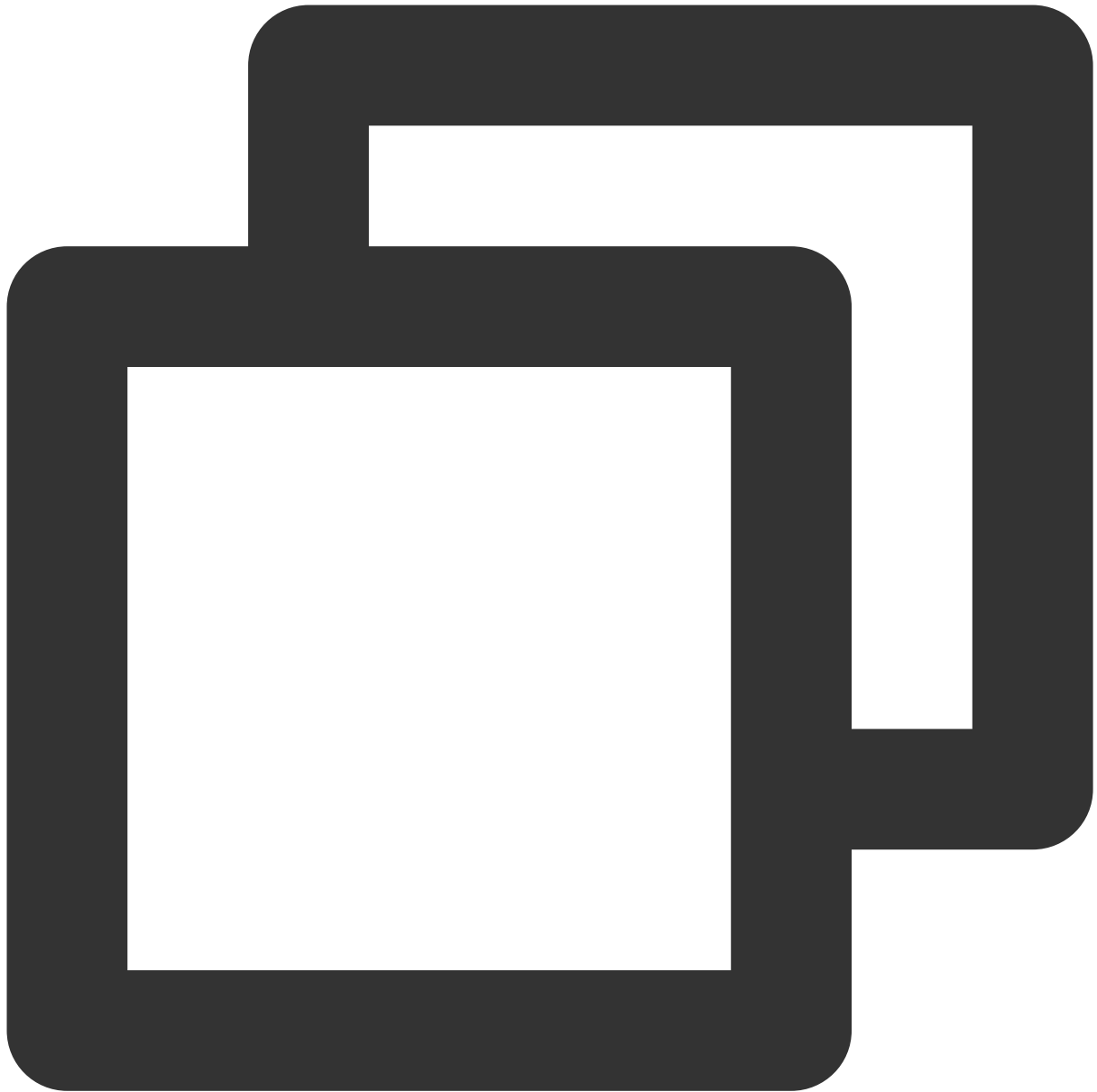
Using the Java SDK



```
// 1. Initialize the user credentials (secretId, secretKey).
String secretId = "SECRETID";
String secretKey = "SECRETKEY";
COSCredentials cred = new BasicCOSCredentials(secretId, secretKey);
// 2. Set the bucket region. For abbreviations of COS regions, please visit https://
// `clientConfig` contains the set methods to set region, HTTPS (HTTP by default),
Region region = new Region("COS_REGION");
ClientConfig clientConfig = new ClientConfig(region);
// The HTTPS protocol is recommended.
clientConfig.setHttpProtocol(HttpProtocol.https);
// 3. Generate a COS client.
```

```
COSClient cosClient = new COSClient(cred, clientConfig);  
// Enter the bucket name in the format of `BucketName-APPID`.  
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);
```

Using the Python SDK



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

# In most cases, set the log level to INFO. If you need to debug, you can set it to
logging.basicConfig(level=logging.INFO, stream=sys.stdout)

# 1. Set user attributes such as secret_id, secret_key, and region. Appid has been
secret_id = os.environ['COS_SECRET_ID']      # User `SecretId`. We recommend that yo
```

```
secret_key = os.environ['COS_SECRET_KEY']    # User `SecretKey`. We recommend that y
region = 'ap-beijing'                        # Replace it with the actual region, which can be viewed
                                              # For the list of regions supported by COS, see https://
token = None                                # Token is required for temporary keys but not permanent
scheme = 'https'                            # Specify whether to use HTTP or HTTPS protocol to acces

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

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

    # The request is successful. You can view `request-id` in the response
    if 'x-cos-request-id' in response:
        print(response['x-cos-request-id'])

# The request failed. You can view `request-id` in the exception information
except CosServiceError as e:
    print(e.get_request_id())
```

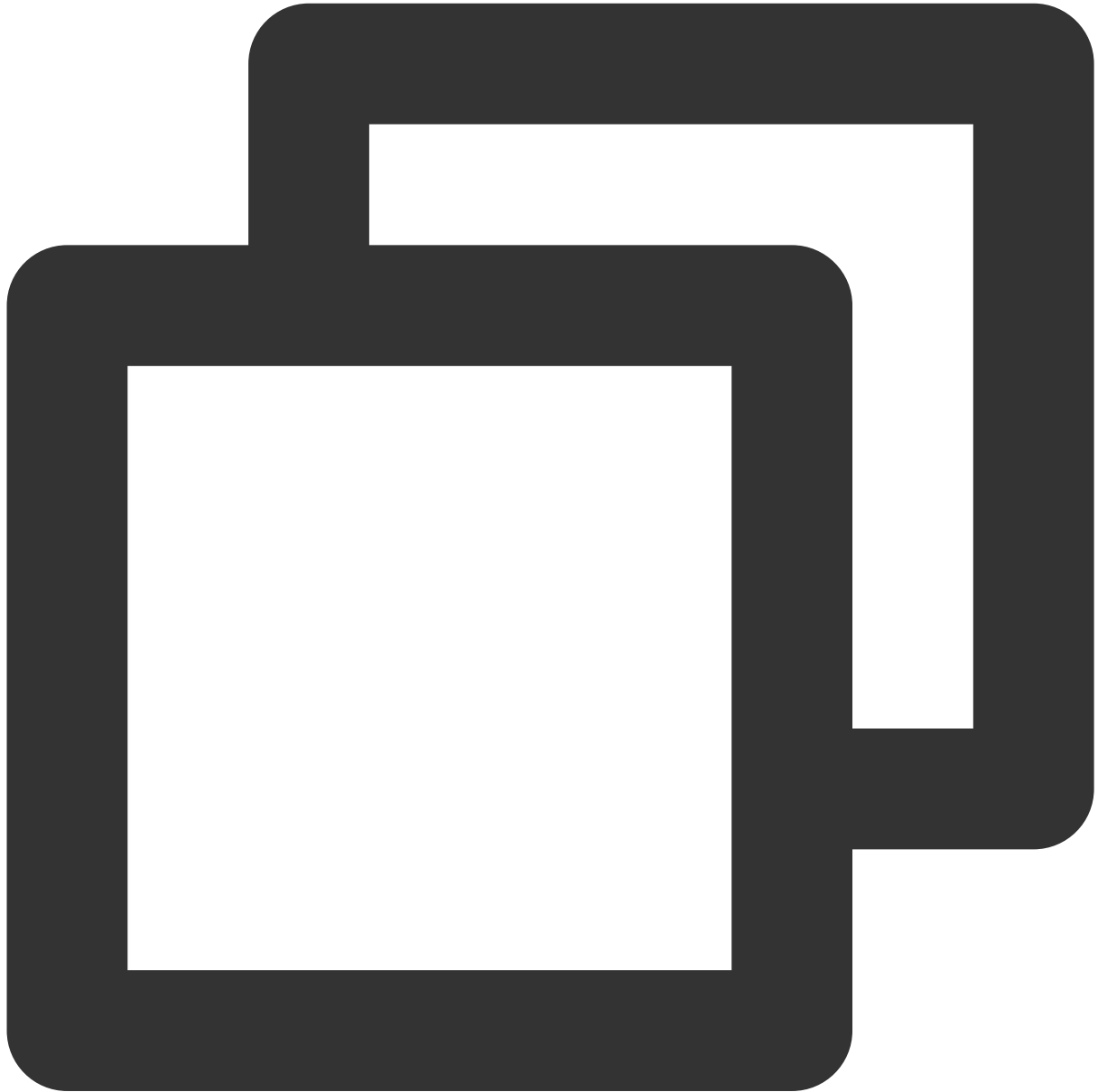
Using the JavaScript SDK



```
cos.putObject({
  Bucket: 'examplebucket-1250000000', /* Required */
  Region: 'COS_REGION', /* Required */
  Key: 'test.js', /* Required */
  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 );  
});
```

Using the Node.js SDK



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

```
cos.putObject({
  Bucket: 'examplebucket-1250000000', /* Required */
  Region: 'COS_REGION', /* Required */
  Key: 'test.nodejs', /* Required */
  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 );
});
```

Using the Weixin Mini Program SDK

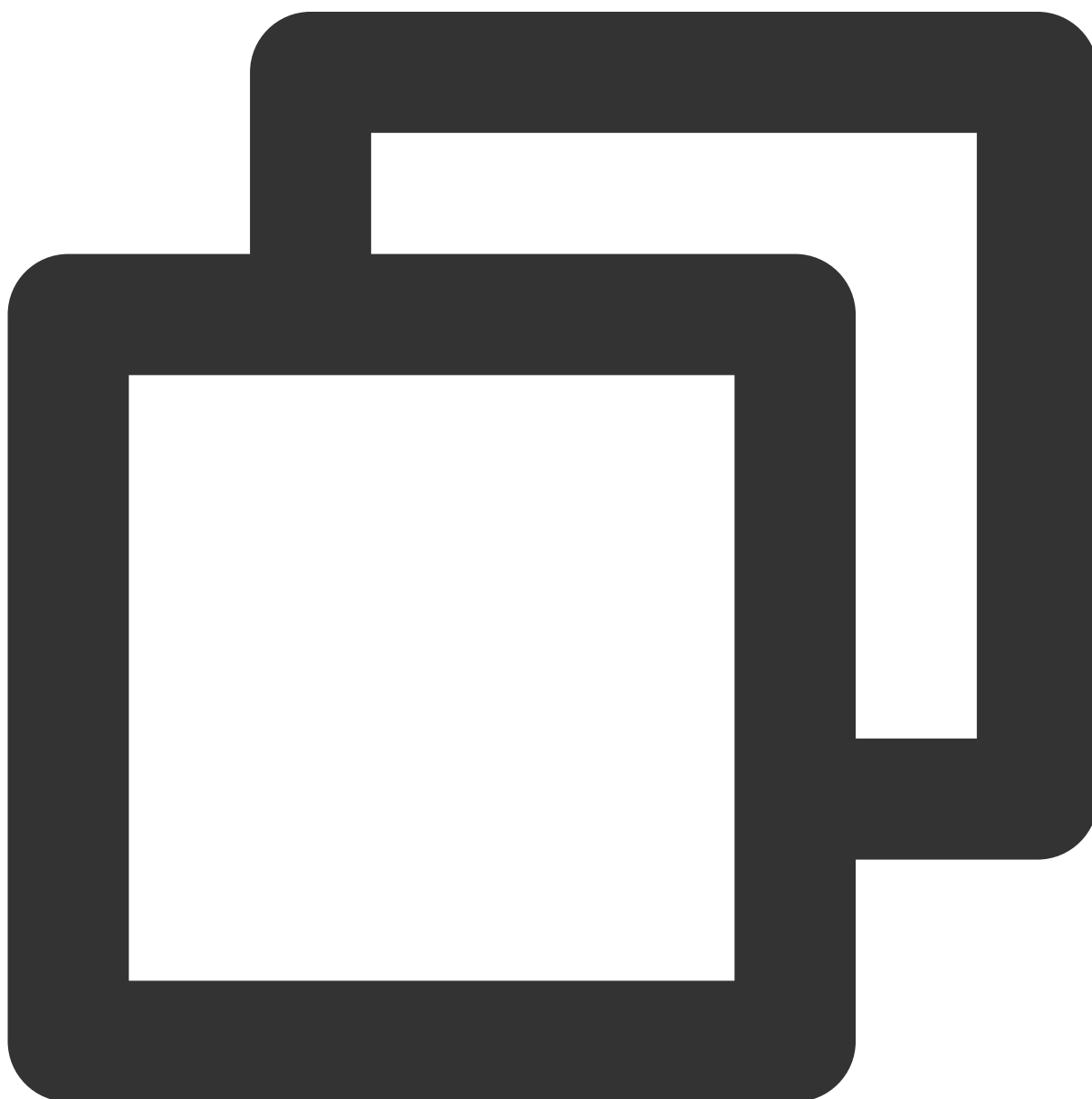


```
var COS = require('cos-wx-sdk-v5');
var cos = new COS({
  SecretId: 'SECRETID',
  SecretKey: 'SECRETKEY'
});

cos.putObject({
  Bucket: 'examplebucket-1250000000', /* Required */
  Region: 'COS_REGION', /* Required */
  Key: 'test.js', /* Required */
  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 );
});
```

Using the PHP SDK



```
$secretId = "SECRETID"; // "SecretId of your Tencent Cloud API key";
$secretKey = "SECRETKEY"; // "SecretKey of your Tencent Cloud API key";
$region = "COS_REGION"; // Set the default bucket region
$cosClient = new Qcloud\Cos\Client(
    array(
        'region' => $region,
        'schema' => 'https', // Protocol, which is `http` by default
        'credentials'=> array(
            'secretId' => $secretId ,
            'secretKey' => $secretKey));
# Upload a file
## putObject (an API that can upload files of up to 5 GB)
### Uploading strings in memory
try {
    $bucket = "examplebucket-1250000000"; // Bucket name in the format of BucketName
    $key = "test.php"; // Object key, which is the unique identifier of an object in
    $result = $cosClient->putObject(array(
        'Bucket' => $bucket,
        'Key' => $key,
        'Body' => 'Hello COS'));
    $requestId = $result['RequestId'];
    print_r($requestId);
} catch (\Exception $e) {
    echo "$e\n";
}
```

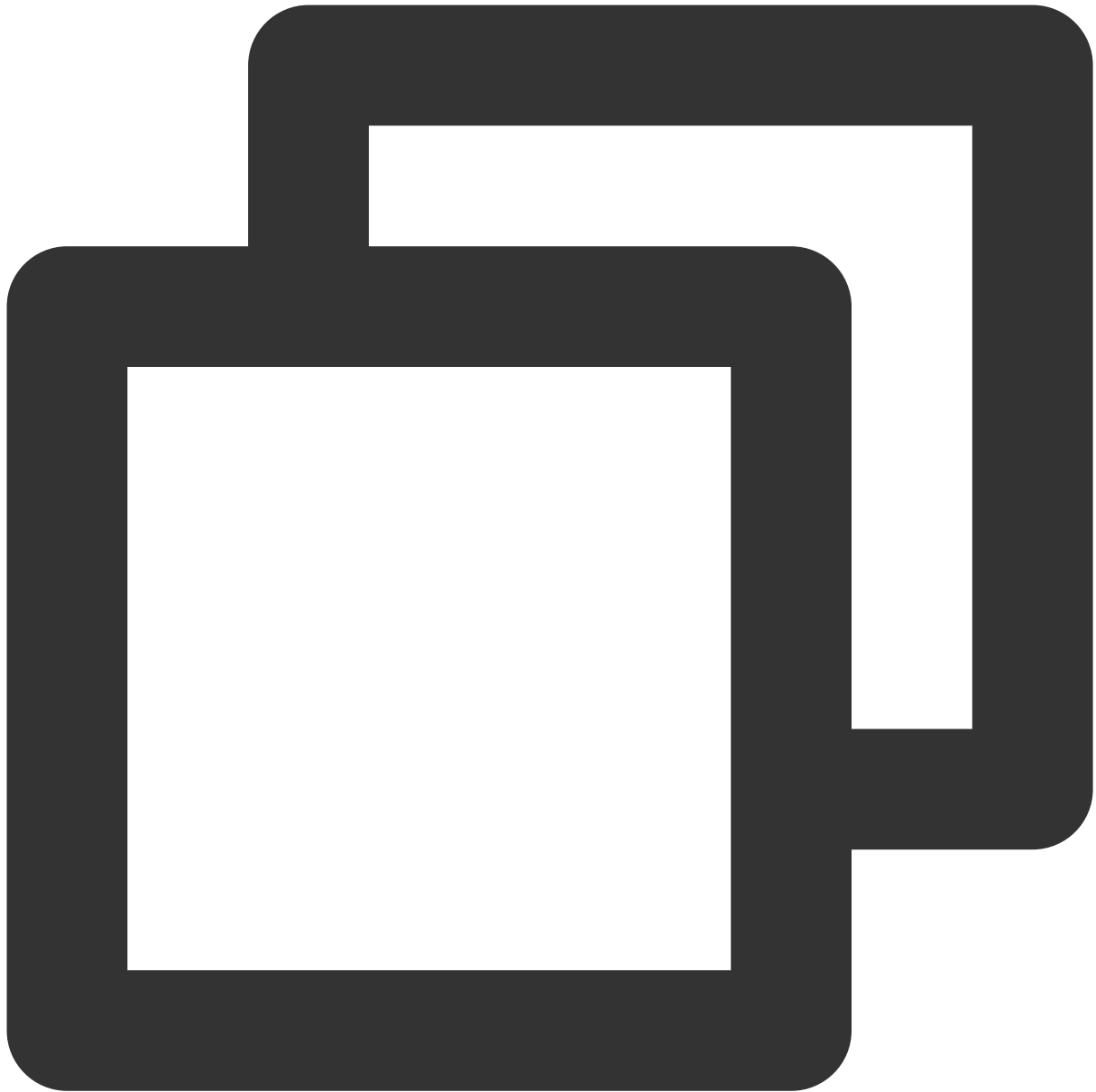
Using the iOS SDK



```
QCloudCOSXMLUploadObjectRequest* put = [QCloudCOSXMLUploadObjectRequest new];  
/** Path of the local file. Ensure that the URL starts with "file://" in the follow  
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:@"file URL"];  
// Bucket name in the format of BucketName-Appid, which can be viewed in the COS co  
put.bucket = @"examplebucket-1250000000";  
// Object key, i.e., the full path of a COS object. If the object is in a directory  
put.object = @"exampleobject";  
// Content of the object to be uploaded. You can pass variables of the `NSData*` or
```

```
put.body = url;
// Monitor the upload progress
[put setSendProcessBlock:^(int64_t bytesSent,
                           int64_t totalBytesSent,
                           int64_t totalBytesExpectedToSend) {
    //      bytesSent          Number of bytes to send in this request (a la
    //      totalBytesSent      Total number of bytes sent so far
    //      totalBytesExpectedToSend  Total number of bytes expected to send, i.e.
}];
// Monitor the upload result
[put setFinishBlock:^(QCloudUploadObjectResult *result, NSError *error) {
    // Obtain requestId
    [result.__originHTTPURLResponse__.allHeaderFields objectForKey:@"x-cos-request-id"];
}];
[put setInitMultipleUploadFinishBlock:^(QCloudInitiateMultipartUploadResult *
                                       multipleUploadInitResult,
                                       QCloudCOSXMLUploadObjectResumeData resumeData) {
    // This block will be called back after the Initiate Multipart Upload operation
    NSString* uploadId = multipleUploadInitResult.uploadId;
}];
[[QCloudCOSTransferMangerService defaultManager] UploadObject:put];
```

Using the Android SDK



```
// 1. Initialize TransferService. You should use the same TransferService for the s
TransferConfig transferConfig = new TransferConfig.Builder()
    .build();
CosXmlServiceConfig cosXmlServiceConfig = new CosXmlServiceConfig.Builder()
    .setRegion(COS_REGION)
    .builder();
CosXmlService cosXmlService = new CosXmlService(context, cosXmlServiceConfig, crede
TransferService transferService = new TransferService(cosXmlService, transferConfig

// 2. Initialize PutObjectRequest
String bucket = "examplebucket-1250000000"; // Bucket name in the format of BucketN
```

```
String cosPath = "exampleobject"; // Location identifier of the object in the bucket
String srcPath = "examplefilepath"; // Absolute path to the local file
PutObjectRequest putObjectRequest = new PutObjectRequest(bucket,
    cosPath, srcPath);

// 3. Call the upload method to upload the file
final COSUploadTask uploadTask = transferService.upload(putObjectRequest);
uploadTask.setCosXmlResultListener(new CosXmlResultListener() {
    @Override
    public void onSuccess(CosXmlRequest request, CosXmlResult result) {
        // Upload succeeded. You can get `requestId` here.
        String requestId = result.getHeader("x-cos-request-id");
    }

    @Override
    public void onFail(CosXmlRequest request,
        CosXmlClientException clientException,
        CosXmlServiceException serviceException) {
        // `requestId` exists only for `CosXmlServiceException`
        if (serviceException != null) {
            String requestId = serviceException.getRequestId();
        }
    }
});
```

Slow Upload over Public Network

Last updated : 2024-03-25 18:01:58

Symptom

Symptom 1

:

The upload speed is low (below 8 Mbps) over a home network, while it is normal over a corporate network.

The upload speed is low (below 8 Mbps) over a corporate network, while it is normal over a 4G network.

Symptom 2

: The upload is slow at a custom domain name.

Possible Causes

Symptom 1

1.1 The speed is low due to your ISP and network conditions.

1.2 CORS is used and thus the speed is low.

Symptom 2: CNAME resolves the custom domain name to other products (e.g., Content Delivery Network (CDN), Cloud Virtual Machine (CVM), or Anti-DDoS) before resolving to COS.

Solutions

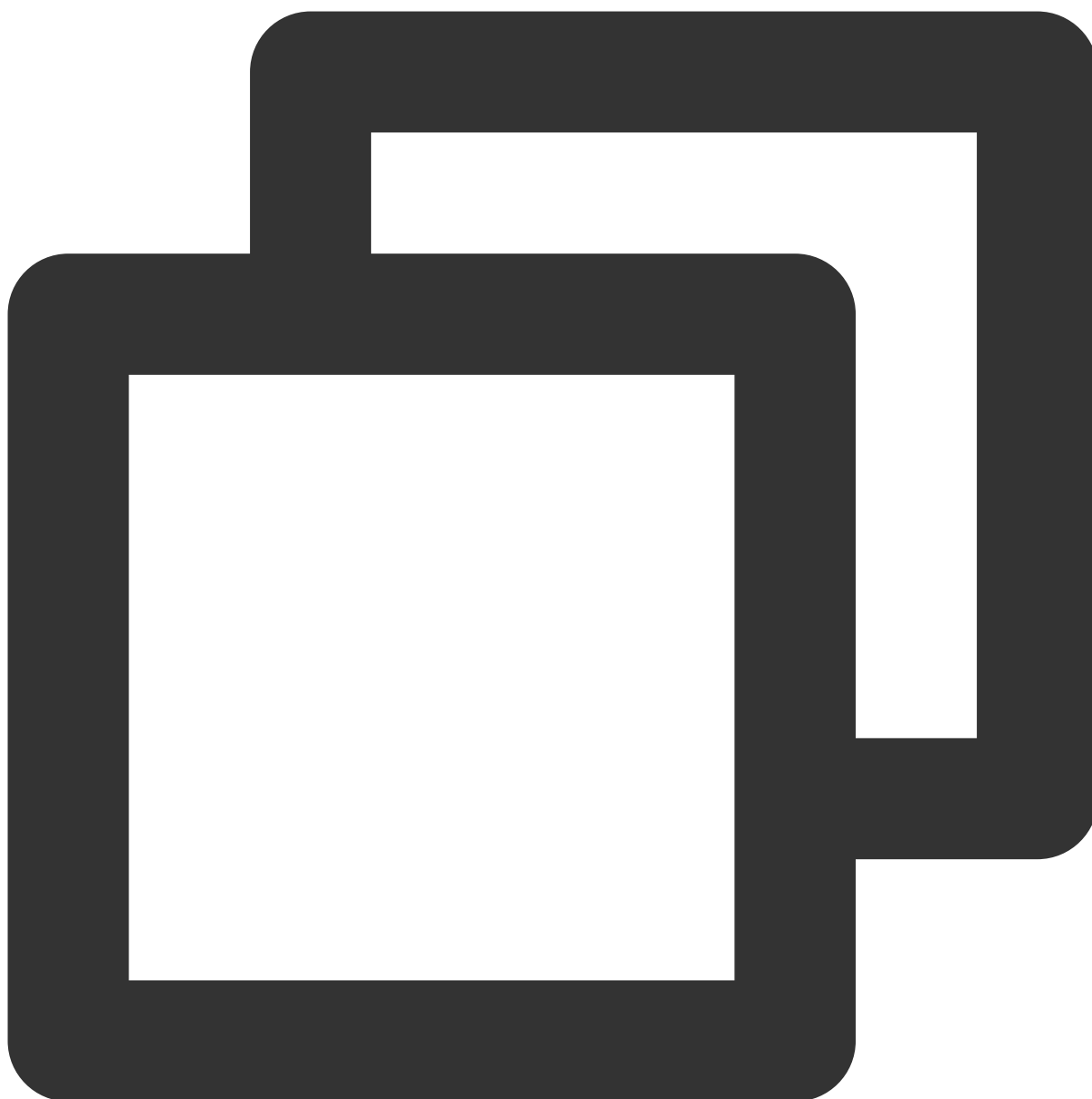
Symptom 1: Check the network conditions of your client. For detailed directions, see [Troubleshooting client network issues](#).

Symptom 2: Reduce the intermediate links by modifying the DNS record for your custom domain name to improve the transfer efficiency. For detailed directions, see [Modifying custom domain name's DNS record](#).

Troubleshooting

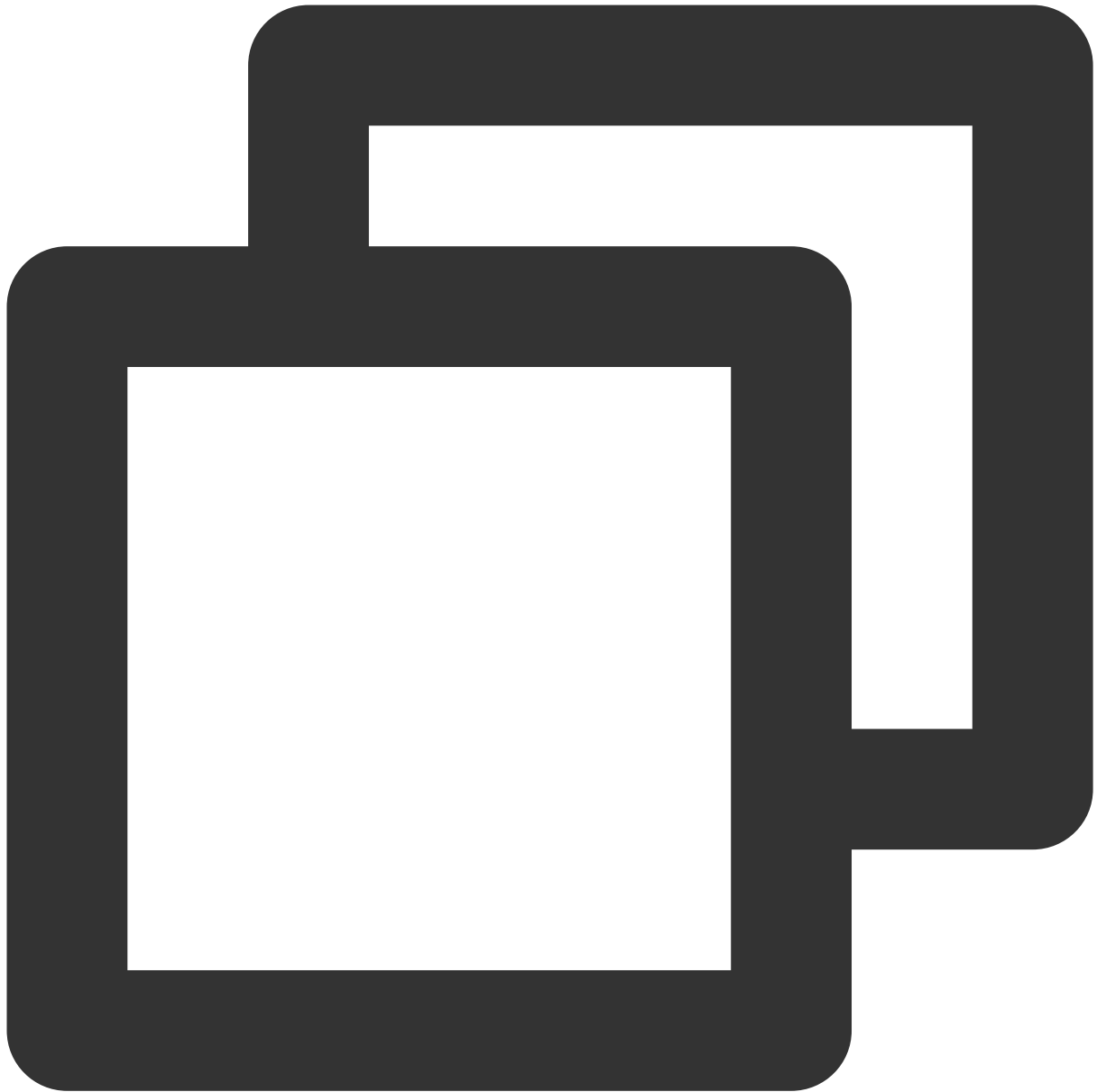
Troubleshooting client network issues

1. Run the following command to check whether the ISP of the IP address is the same as that of the client network:



```
ping COS domain name
```

For example:



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

If so, proceed to [step 3](#).

If not, proceed to [step 2](#).

2.

Check whether a proxy is enabled in the browser (with Chrome as an example)

.

2.1 Open Chrome and click



> **Settings** in the top-right corner.

2.2 Click **Advanced**. On the **System** page, click **Open your computer's proxy settings** to open the configuration window of your OS.

Check whether a proxy is enabled.

If so, disable the proxy.

If not, proceed to [step 3](#).

3.

Check whether your Wi-Fi router limits the speed.

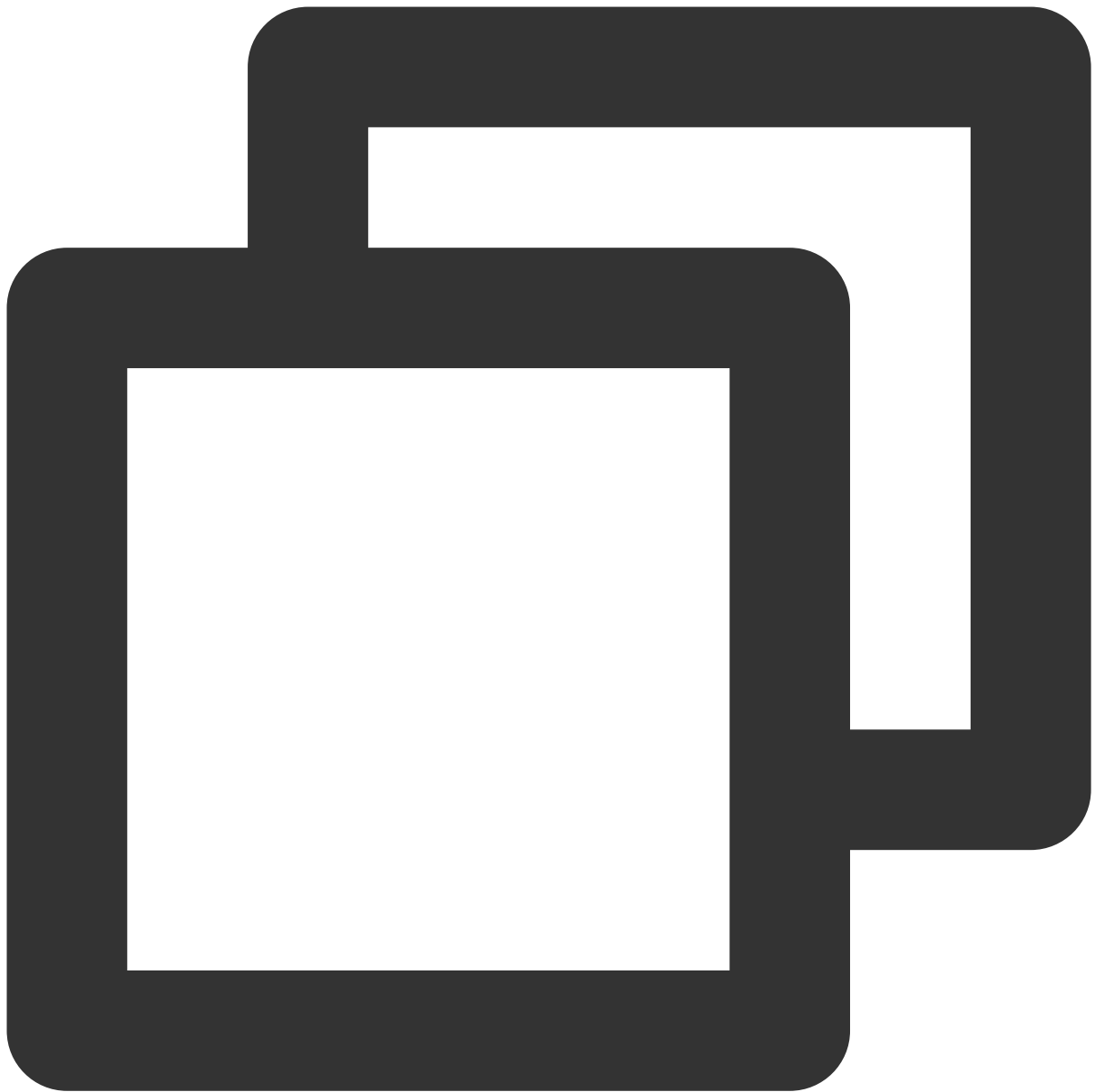
If so, allocate the bandwidth as needed.

If not, proceed to [step 4](#).

4.

Test the performance of the upload to COS over the current network.

The following example uses a 20 MB object and COSCMD to test the upload and download speeds:



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

A message similar to the following will be displayed, from which you can get the average, min, and max speeds:

```
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. Visit [Speed Test](#) to test the speed. Then, compare the tested speed with the data obtained in [step 4](#) to determine whether the client has used the maximum bandwidth.

If the speeds obtained in step 4 are below the client bandwidth, [contact us](#).

If the speeds obtained in step 4 equal to the client bandwidth but don't reach the speed promised by your ISP, contact your ISP.

If the speeds obtained in step 4 equal to the client bandwidth and reach the bandwidth promised by your ISP, proceed to [step 6](#).

6.
Check whether the client is accessing a bucket residing in another country/region.

If so, we recommend you enable COS' global acceleration feature.

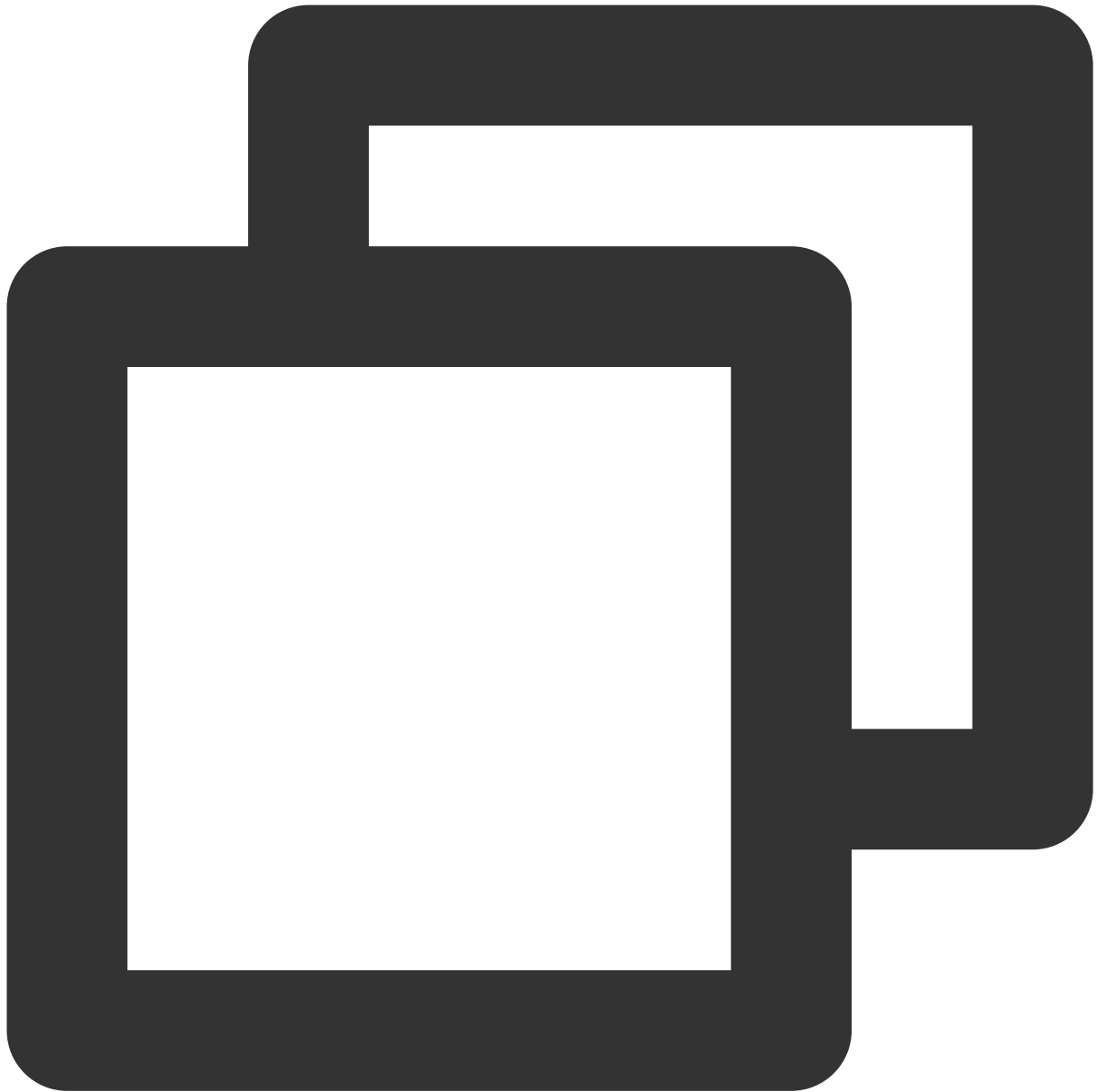
If not, [contact us](#).

Modifying custom domain name's DNS record

1. Check whether the custom domain name is resolved to a COS domain name.

If so, [contact us](#).

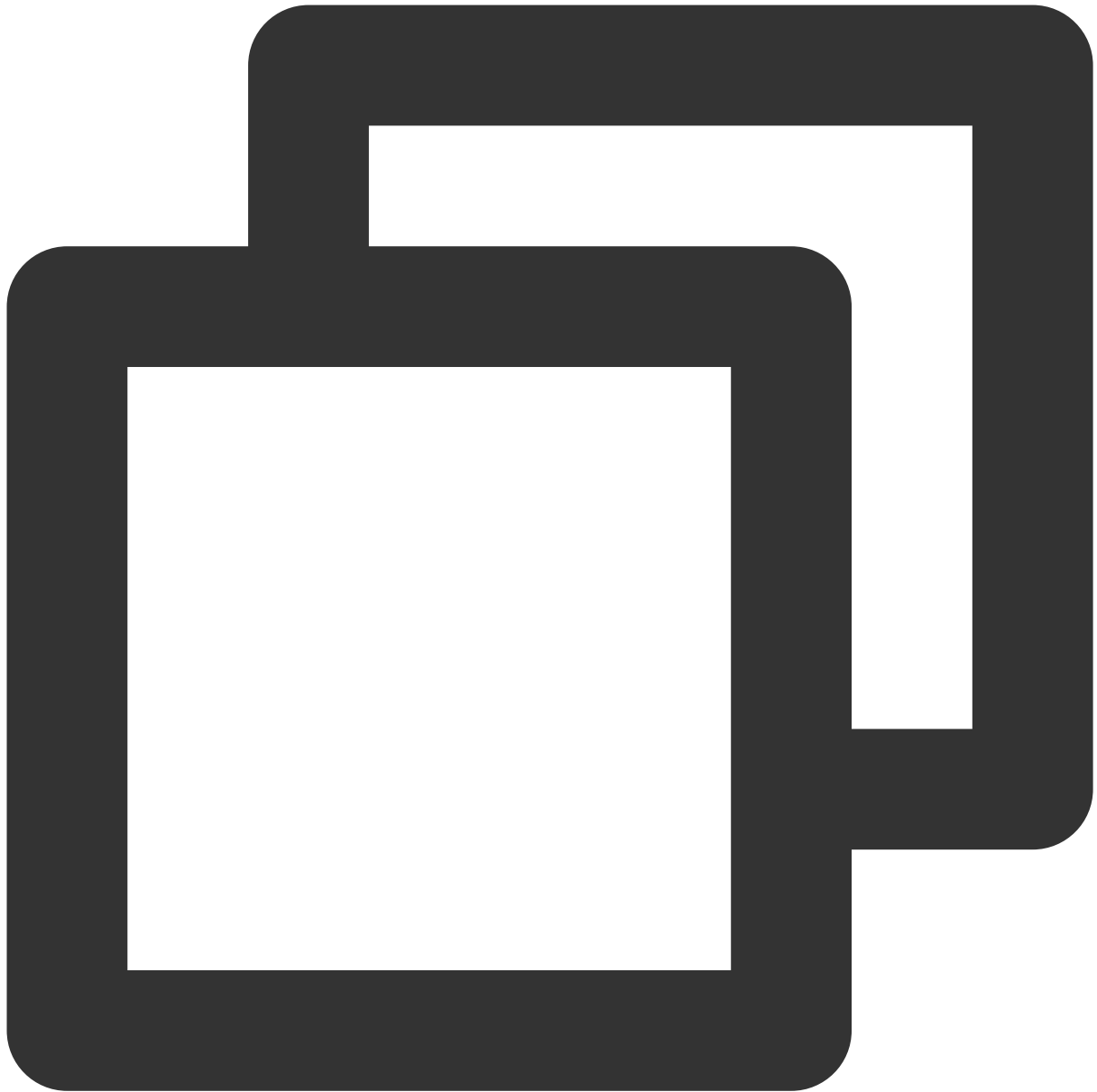
Common COS domain names are as follows:



```
XXX.cos.ap-beijing.myqcloud.com (default COS domain name)
XXX.cos.accelerate.myqcloud.com (COS global acceleration domain name)
XXX.cos-website.ap-beijing.myqcloud.com (COS static website domain name)
XXX.picbj.myqcloud.com (COS' default CI domain name)
```

If not, proceed to [step 2](#).

Common non-COS domain names are as follows:



```
XXX.file.myqcloud.com or XXX.cdn.dnsv1.com (Tencent Cloud CDN's default domain name  
XXX.w.kunlungr.com (Alibaba Cloud CDN's default domain name)
```

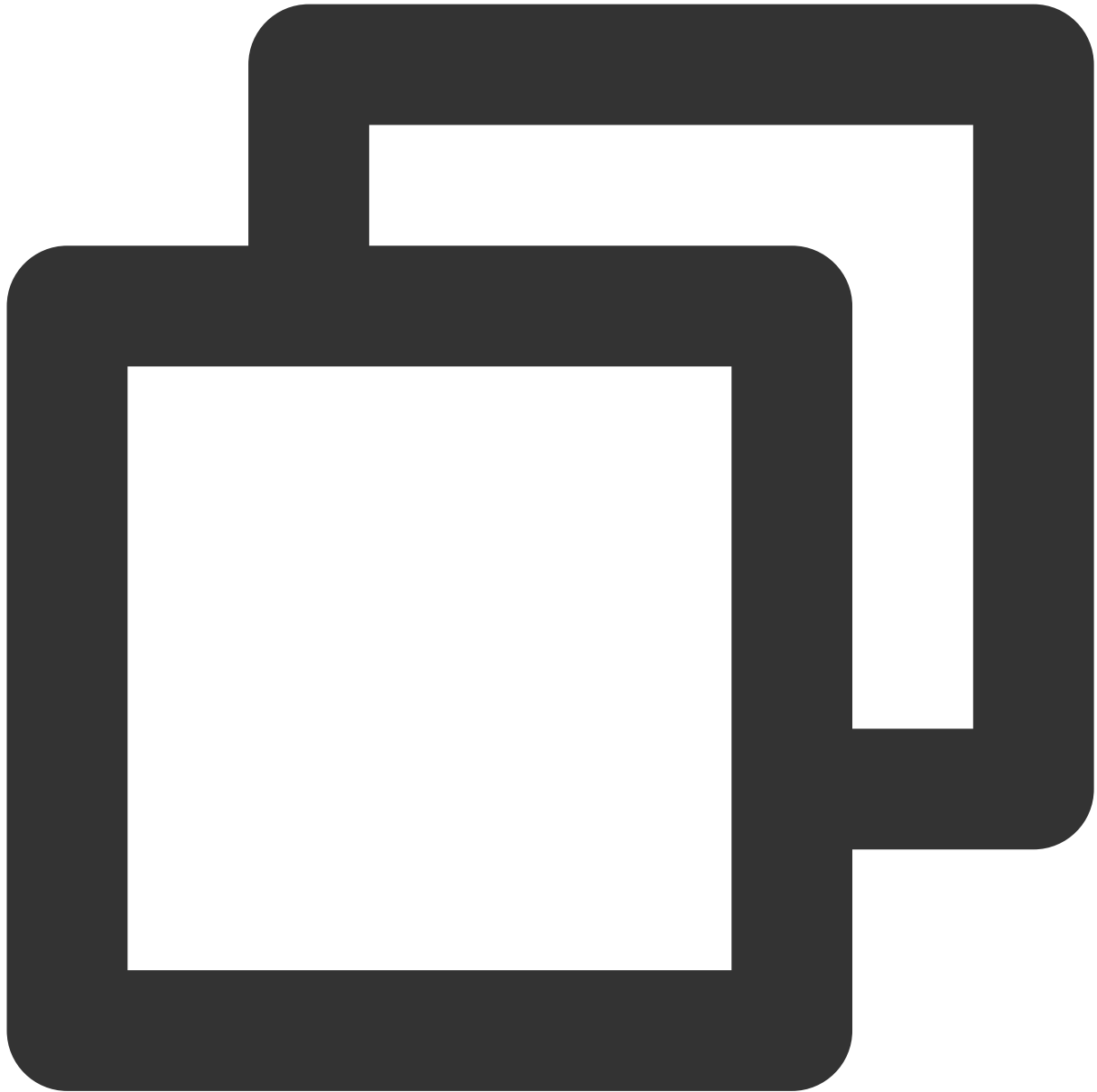
2.

Resolve the custom domain name to the desired COS domain name via the CNAME record and upload data.

For example, `upload.mydomain.com cname XXX.cos.ap-beijing.myqcloud.com`. For detailed directions, see [Enabling Custom Origin Domains](#).

3. Modify the default COS domain name of the client.

The following uses C# as an example:



```
CosXmlConfig config = new CosXmlConfig.Builder()
    .SetConnectionTimeoutMs(60000) // Set the connection timeout period in milliseconds
    .SetReadWriteTimeoutMs(40000) // Set the read/write timeout period in milliseconds,
    .IsHttps(true) // Set HTTPS as the default request method.
    .SetAppid(appid) // Set the APPID of your Tencent Cloud account
    .SetRegion(region) // Set the default bucket region.
    .SetHost("XXXXXX.com") // Enter the custom domain name.
    .SetDebugLog(true) .Build(); // Create the CosXmlConfig object.
```

For more information on how to call other SDKs, see [SDK Overview](#).

403 Error for COS Access

Last updated : 2024-03-25 18:01:58

Problem

Error code 403 is returned when you use the APIs or SDKs of COS to upload/download files.

Error code 403 is returned when you access COS resources by using a sub-account or temporary key.

Error code 403 is returned when you modify the bucket configuration.

Analysis

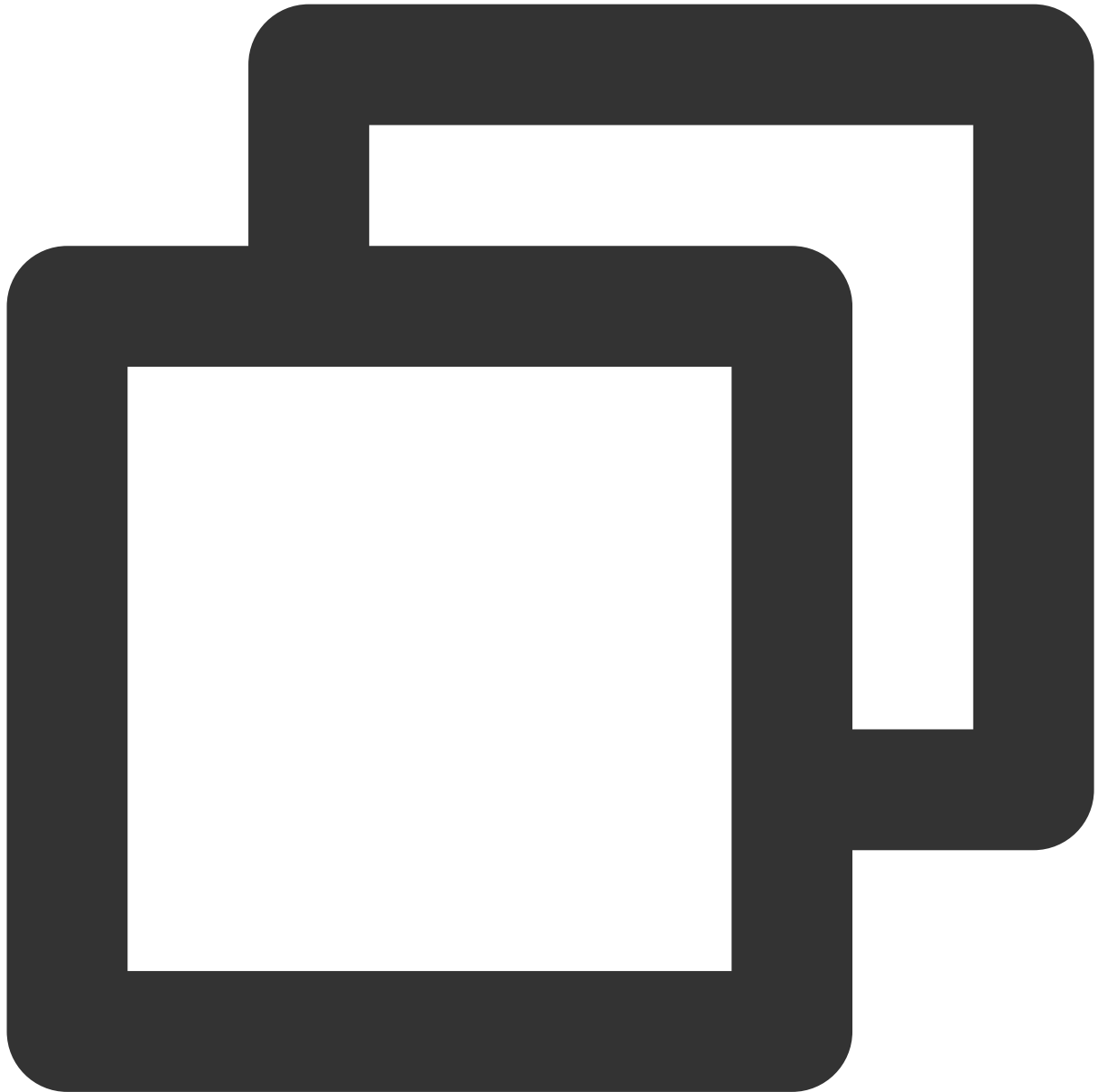
When the error code 403 appears in the COS request, you can refer to the following procedures to analyze the cause of the problem:

1. Check whether the request is a CORS request. For failed cross-origin requests, COS will return ["AccessForbidden"](#).
2. Check whether the request hits the bucket's referer hotlink protection configuration. For the error code 403 caused by the hotlink protection rule, COS will return ["You are denied by bucket referer rule"](#).
3. Check whether the request is anonymous. For unsigned requests to a non-publicly readable object, COS will return "Access Denied." To set a bucket or object publicly readable, see [Setting Access Permission](#) or [Setting Object Access Permission](#).
4. Check whether the request key and signature are correct.
 - 4.1 If the signature uses an incorrect `SecretId`, COS will return ["InvalidAccessKeyId"](#).
 - 4.2 If the local system time is inaccurate, or the request time is after the validity period of the signature, COS will return ["RequestTimeTooSkewed"](#) or ["Request has expired"](#).
 - 4.3 If there is a problem with the calculation method of the signature, COS will return ["SignatureDoesNotMatch"](#).
5. Check whether the sub-account or temporary key that initiated the request has been granted corresponding access permission.
 - 5.1 Check the access permission of the sub-account. For requests from sub-accounts without resource access permissions, COS will return ["Access Denied."](#).
 - 5.2 For requests initiated with a temporary key, the policy entered when the temporary key was applied for limits the scope of accessible resources. For more information, see [Generating and Using Temporary Keys](#).
6. Check whether the requested object is stored in ARCHIVE or DEEP ARCHIVE storage class. For such requests, COS will return ["InvalidObjectState"](#).

Troubleshooting

Message: "Access Denied."

If the following message is displayed when you access COS:



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

You can:

1. Log in to the [COS console](#).
2. Click **Bucket List** on the left sidebar.
3. Click the name of the target bucket to enter the configuration page.

4. Click **Permission Management > Bucket ACL (Access Control List)** on the left sidebar.
5. In the **Bucket ACL (Access Control List)** section, check whether the COS account is granted access permission.
If so, proceed to the next step.
If not, click **Add User** and grant the account the needed permission.
6. Verify whether the account has the granted permission.
If so, proceed to the next step.
If not, click **Edit** to configure again.
7. In the **Permission Policy Settings** section, check whether a policy has been configured for the account.

Note:

If the bucket is set to **Private Read/Write** but anonymous access is set in the policy, the permission set in the policy takes effect.

If an action is set to **Allow** and **Deny** in different policies associated with the same sub-account, the action will be denied.

Policies set for **Everyone** have lower priorities than those set for a **Specified user**.

If so, proceed to the next step.

If not, click **Add Policy** and add the needed permission to the sub-account that initiates the signed request.

8. Verify whether the account has the granted permission.

If so, proceed to the next step.

If not, click **Edit** to configure again.

9. Check whether the value of `q-ak` (case-sensitive) is the `secretID` and `secretKey` of the root account of the bucket.

If so, proceed to the next step.

If not, change the value of `q-ak` to the `secretID` and `secretKey` of the root account.

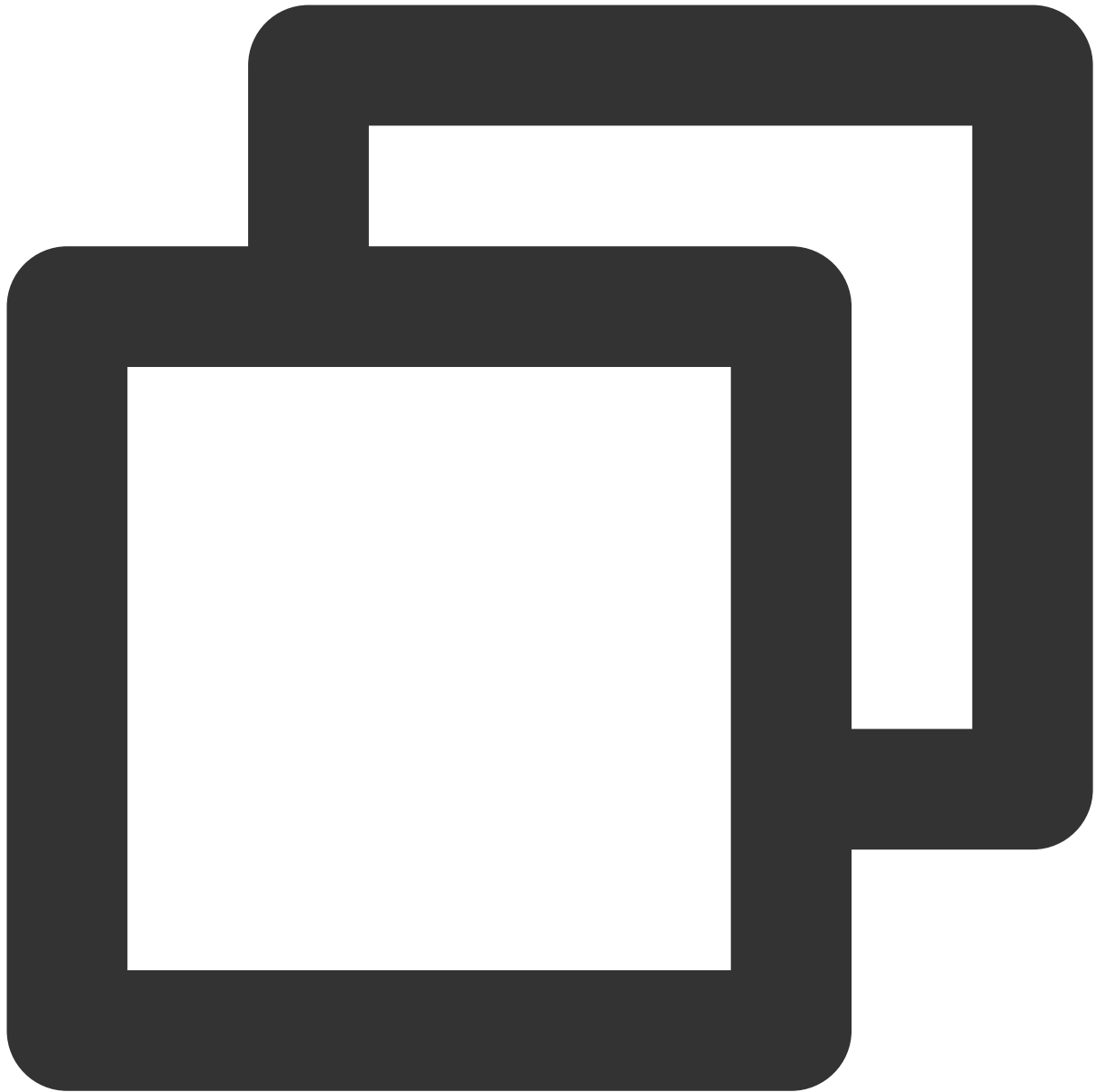
10. Check whether the resource is accessed cross accounts.

If so, grant the account the needed permission as instructed in [Authorizing Cross-Account's Sub-account Read/Write Access to Specified File](#).

If not, [contact us](#).

Message: "AccessForbidden"

If the following message is displayed when you access COS:



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

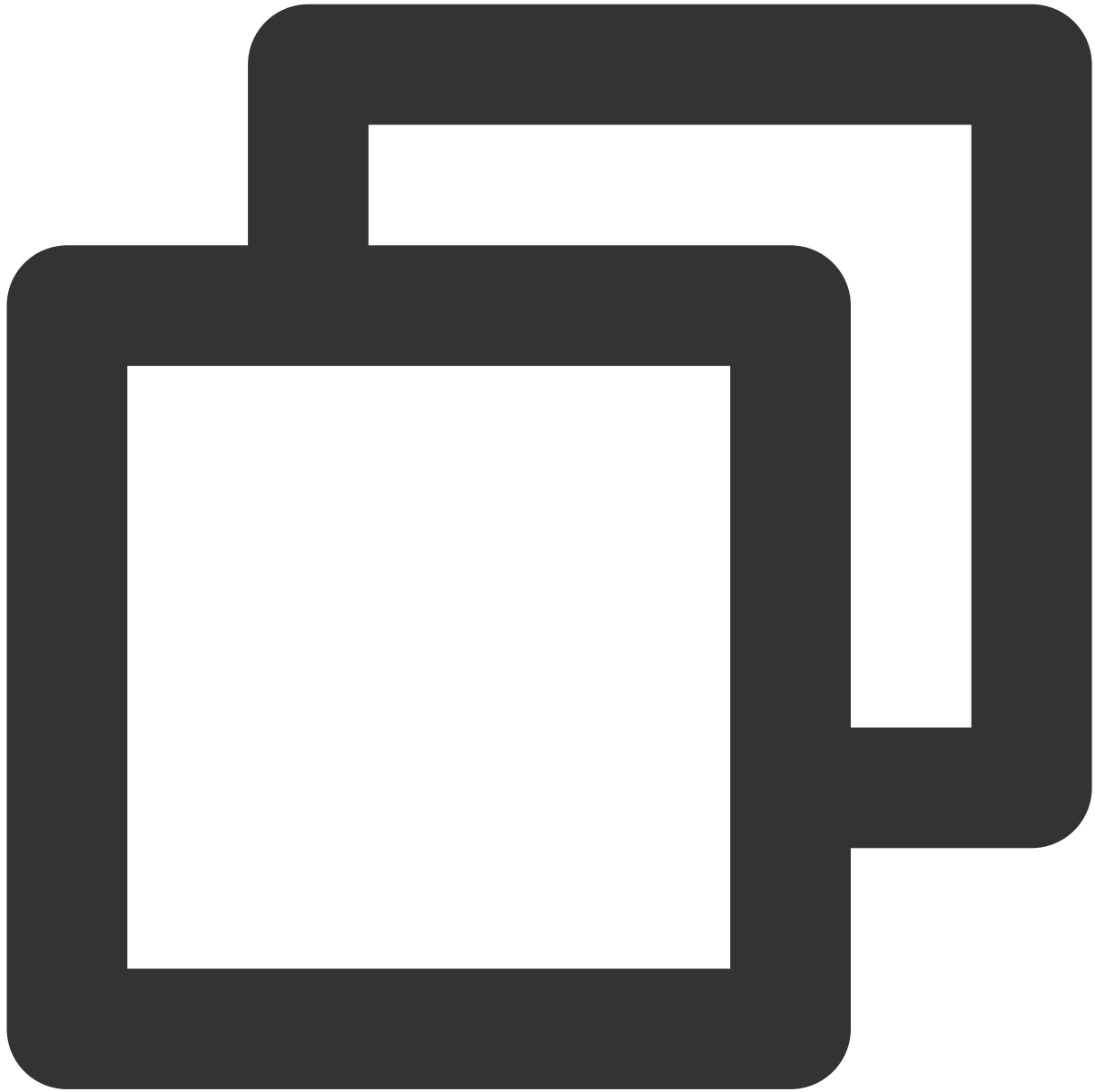
You can:

1. Log in to the [COS console](#).
2. Click **Bucket List** on the left sidebar.
3. Click the name of the target bucket to enter the configuration page.
4. Click **Security Management > CORS (Cross-Origin Resource Sharing)** on the left sidebar.
5. In the **CORS (Cross-Origin Resource Sharing)** section, check whether the request is across origins.

If so, proceed to the next step.

If not, modify the rule.

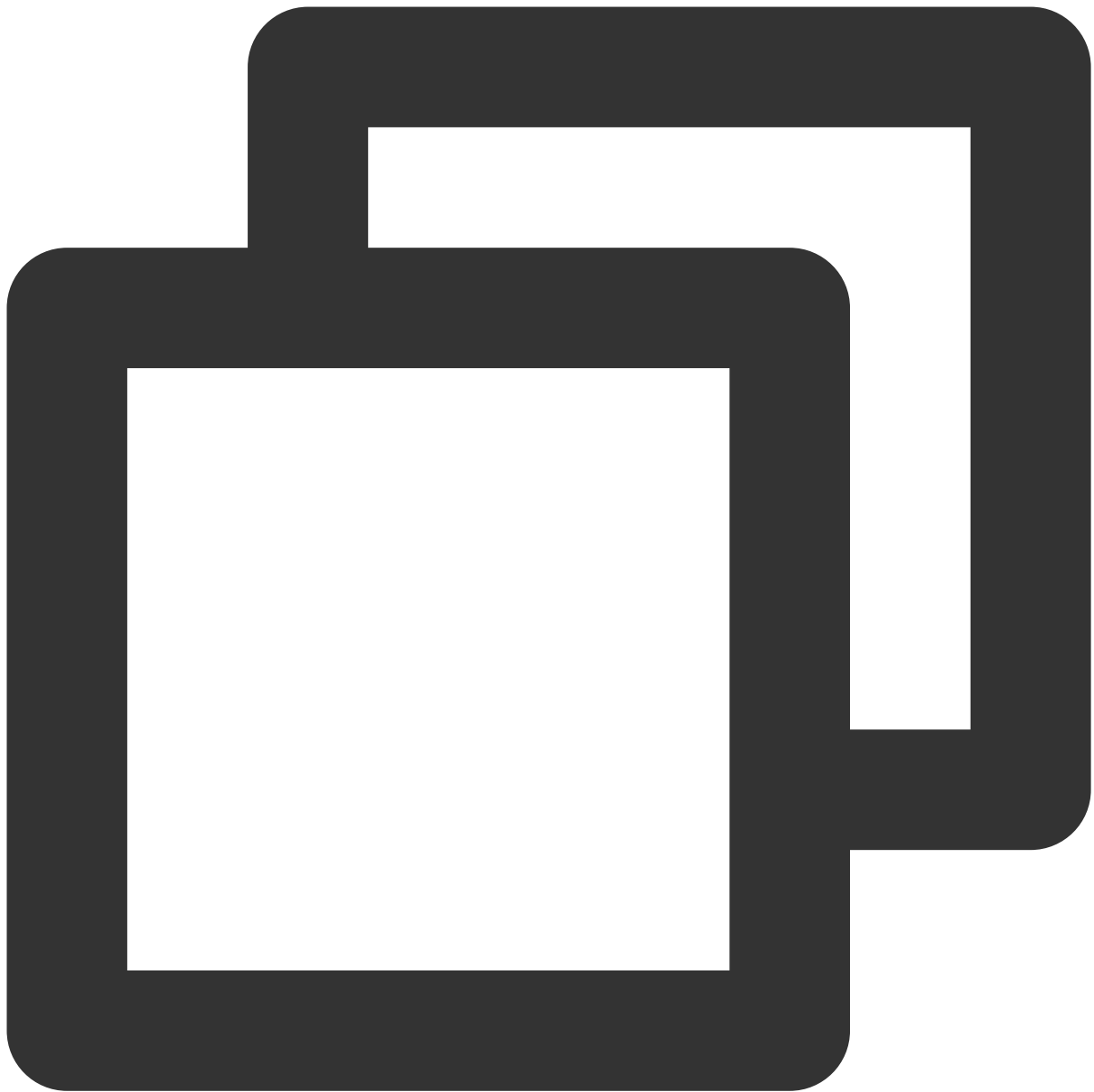
6. Run the following command to check whether the cross-origin request is correctly configured:



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

If the following message is displayed, the configuration is correct:

Page 39 of 64



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

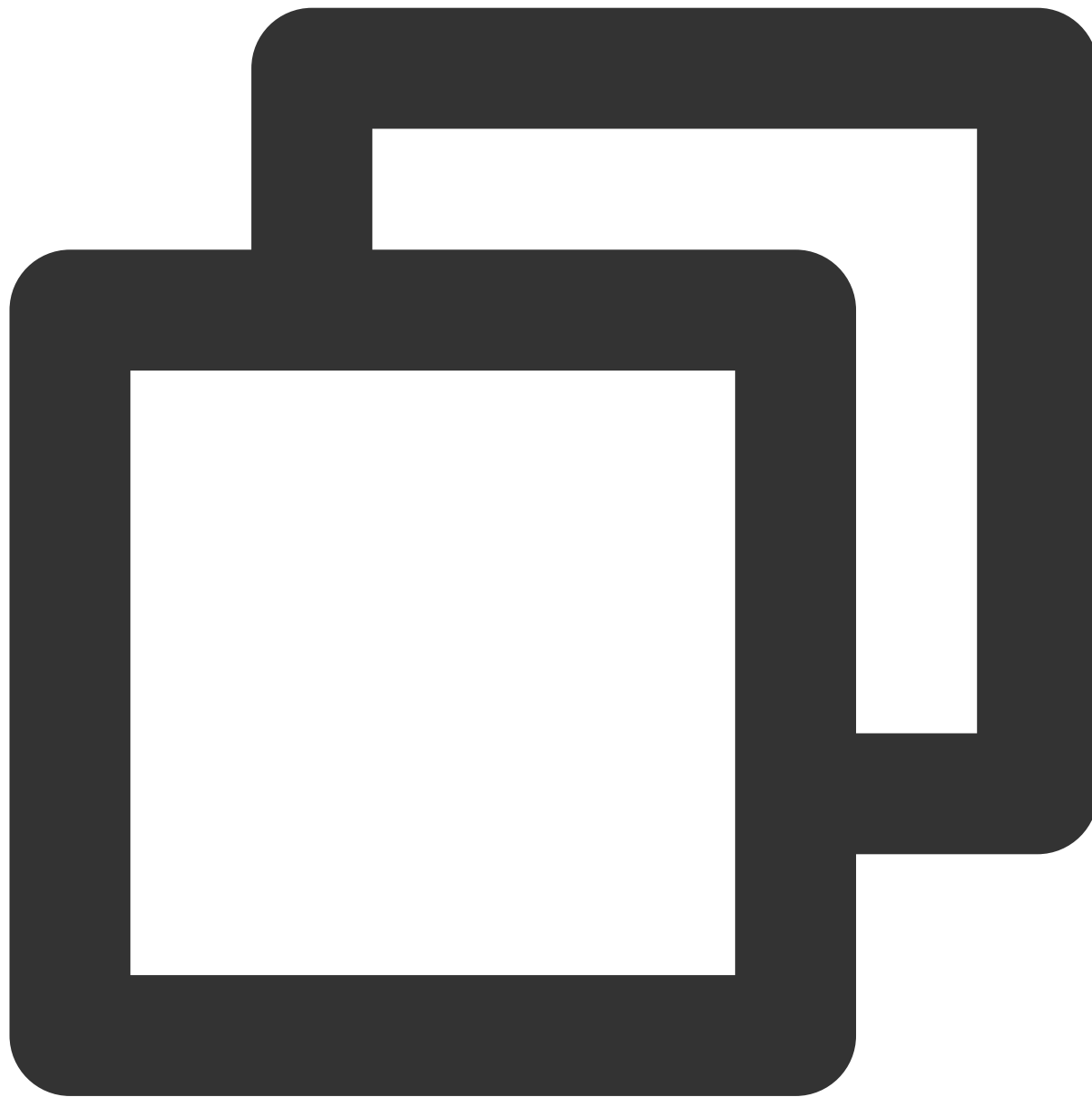
You can:

1. Log in to the [COS console](#).
2. Click **Bucket List** on the left sidebar.
3. Click the name of the target bucket to enter the configuration page.
4. Click **Security Management** > **Hotlink Protection** on the left sidebar.
5. In the **Hotlink Protection** section, check whether hotlink protection is configured.

If so, proceed to the next step.

If not, [contact us](#).

6. Run the following command to check whether hotlink protection is correctly configured:



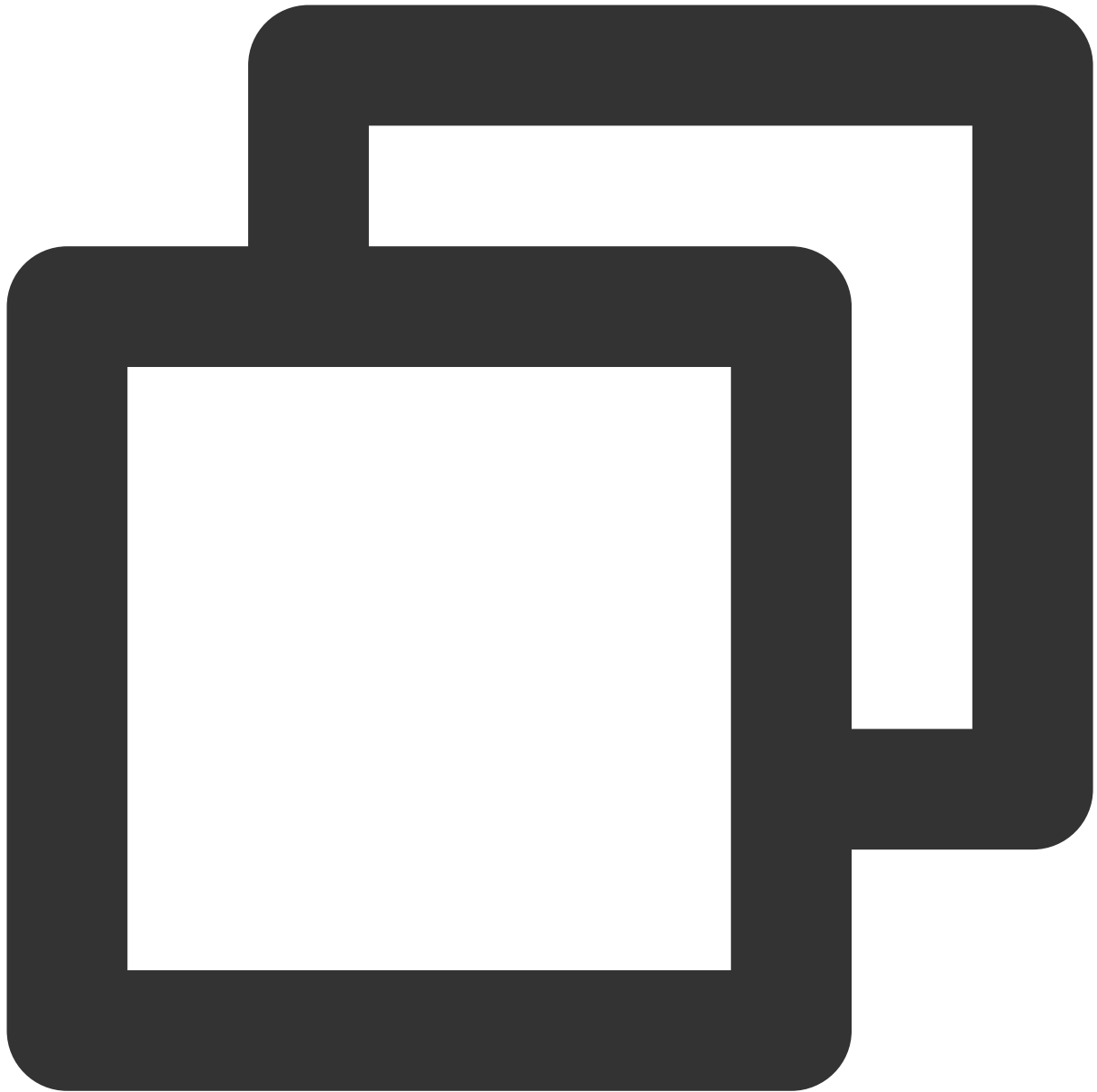
```
curl 'http://bucket-appid.cos.ap-guangzhou.myqcloud.com/object' -voa /dev/null -H '
```

If the following message is displayed, the configuration is correct:

```
(base) [root@VM_32_5_centos ~]# curl 'http://[redacted].cos.ap-guangzhou.myqcloud.com/1.gif' -v --no /dev/null -H 'referer: http://[redacted]'
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
* Connected to [redacted].cos.ap-guangzhou.myqcloud.com ([redacted]) port 80 (#0)
> GET /1.gif HTTP/1.1
> Host: [redacted].cos.ap-guangzhou.myqcloud.com
> User-Agent: curl/7.29.0
> Accept: */*
> referer: http://[redacted]
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Content-Type: image/gif
< Content-Length: 307293
< Connection: keep-alive
< Accept-Ranges: bytes
< Date: Tue, 30 Mar 2021 03:15:40 GMT
< ETag: "2c1f37c1cd1eb5da8-[redacted]"
< Last-Modified: Tue, 16 Mar 2021 14:03:51 GMT
< Server: tencent-cos
< x-cos-hash-crc64ecma: 639785857755905391
< x-cos-replication-status: Replica
< x-cos-request-id: NjA2Mjk3ZGNfYzhhMzNi[redacted]
< x-cos-version-id: MTg0NDUxMjgxOTkw[redacted]
<
{ [data not shown]
100 300k 100 300k 0 0 520k 0 --:--:-- --:--:-- --:--:-- 519k
* Connection #0 to host [redacted].cos.ap-guangzhou.myqcloud.com left intact
* Closing connection -1
```

Message: "InvalidAccessKeyId"

If the following message is displayed when you access COS:



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

You can:

1. Check whether the `q-ak` value of `Authorization` in the request signature is correct.

If so, proceed to the next step.

If not, modify the value of `q-ak` (case-sensitive), which should be the same as `SecretId` of the key.

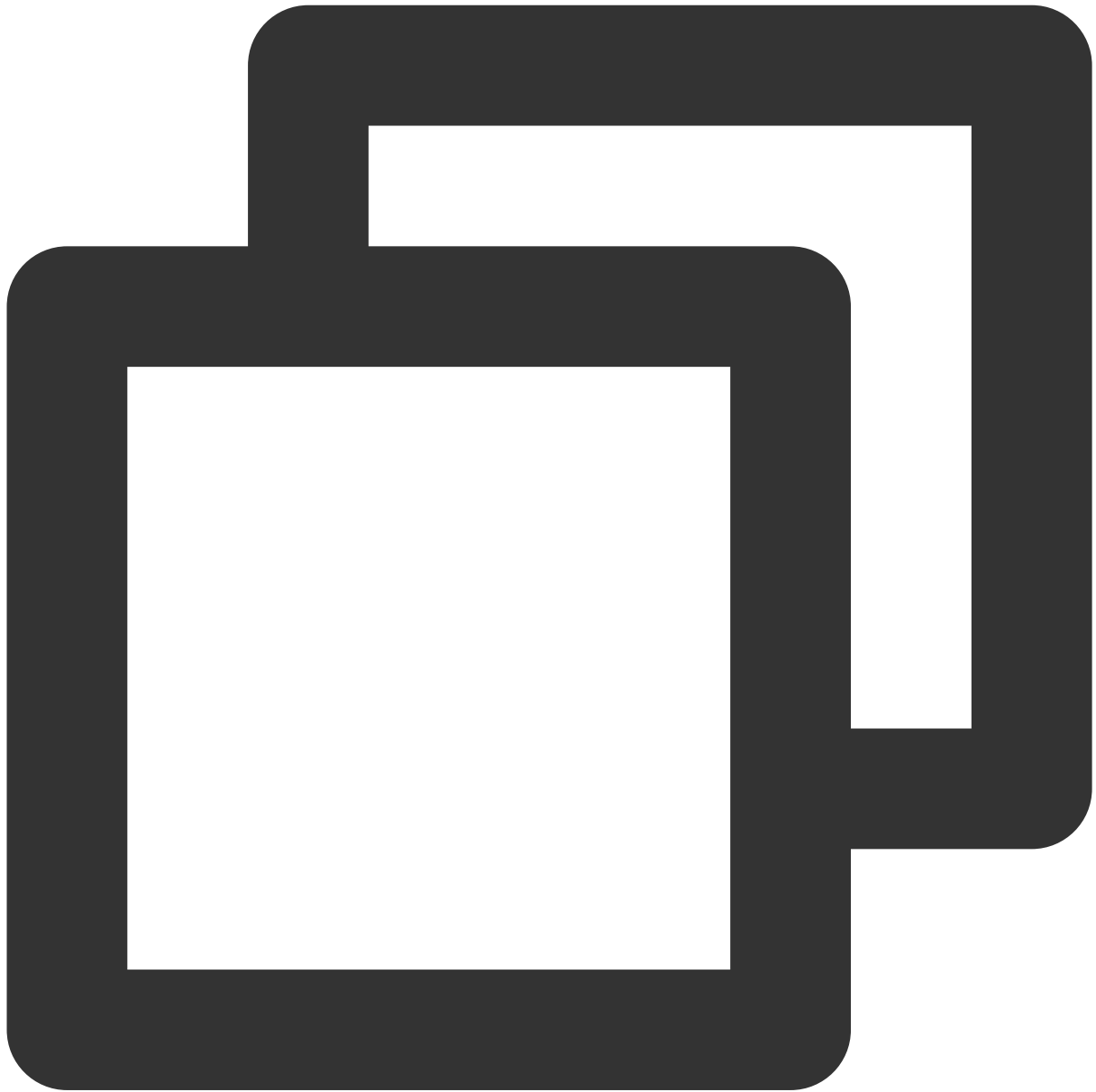
2. Go to the [Manage API Key](#) page to check whether the API key is enabled.

If so, [contact us](#).

If not, enable the API key.

Message: "InvalidObjectState"

If the following message is displayed when you access COS:



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

You can:

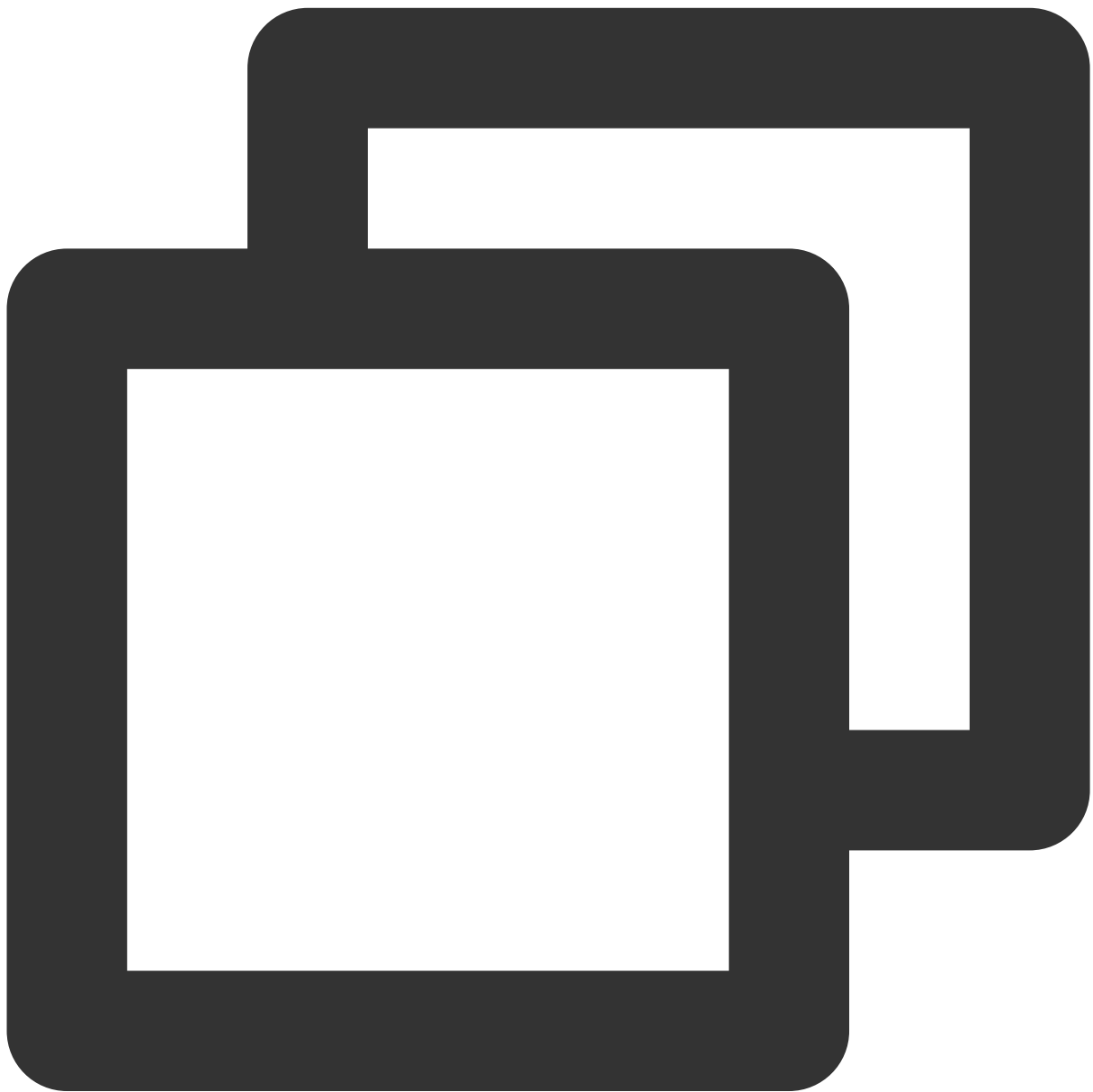
Check whether the requested object is stored in ARCHIVE or DEEP ARCHIVE storage class.

If so, restore the object first. For more information, see [POST Object restore](#).

If not, [contact us](#).

Message: "RequestTimeTooSkewed"

If the following message is displayed when you access COS:



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

You can:

1. Check the client-side time based on the OS as follows:

Windows (with Windows Server 2012 as an example): Click



> **Control Panel > Clock, Language, and Region > Set the time and date.**

Linux: Run the `date -R` command.

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

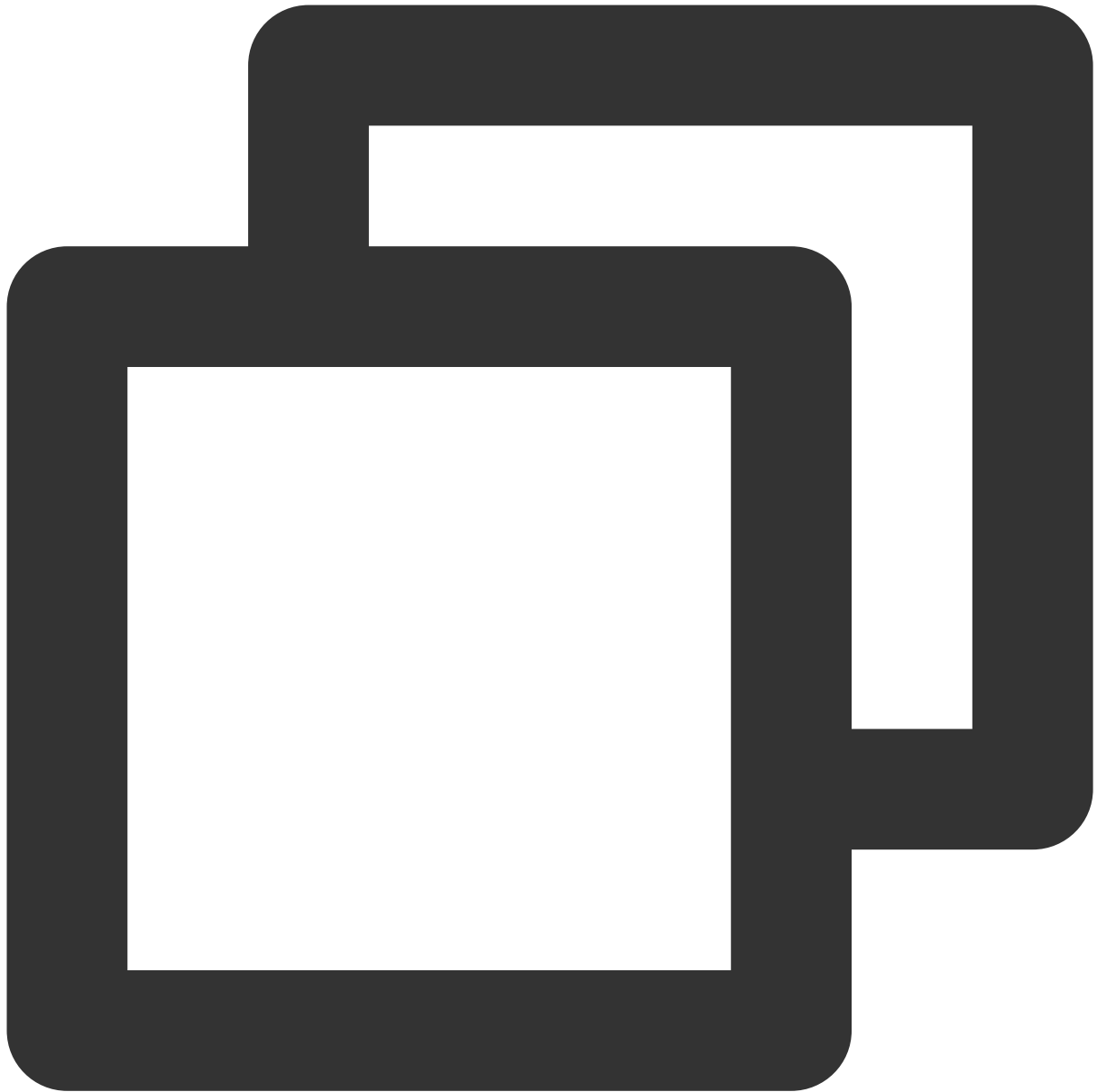
2. Check whether the clock skew between the client and server is larger than 15 minutes.

If so, sync the clock.

If not, [contact us](#).

Message: "Request has expired"

If the following message is displayed when you access COS:



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

Possible causes are as follows:

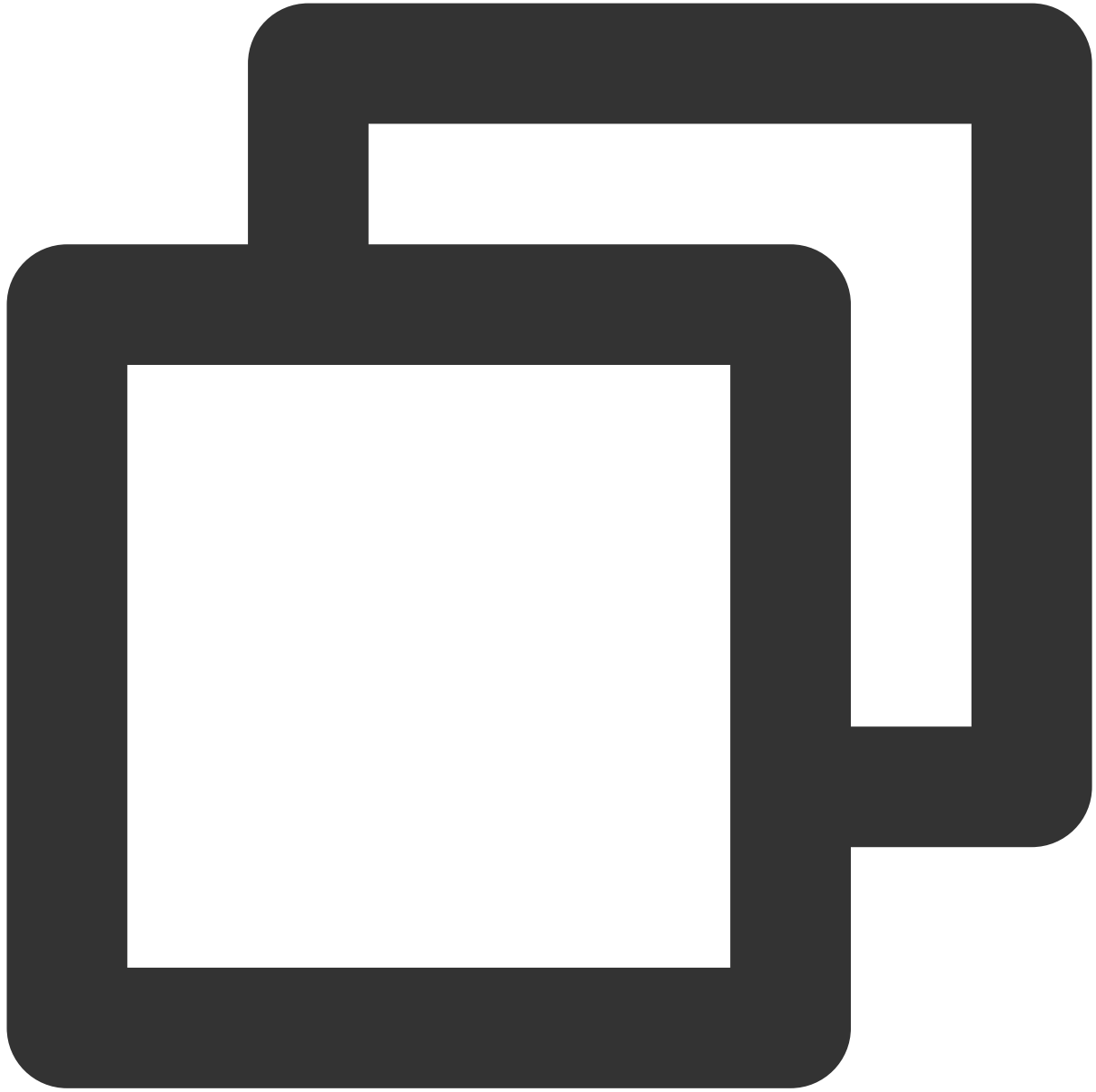
The signature has expired when you initiate the request.

Your local system time is out of sync with the local time in your time zone.

You need to set the effective time of your signature again or sync the local system time. If the problem persists, [contact us](#).

Message: "SignatureDoesNotMatch"

If the following message is displayed when you access COS:



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

You can:

Check whether the signature calculated on the client is the same as that calculated on the server.

If so, [contact us](#).

If not, see [Request Signature](#) and use COS' signature tool to check how the signature is generated.

A Video Cannot Be Played

Last updated : 2024-03-25 18:01:58

Symptom

The uploaded video cannot be played normally (for example, the video has sound only, the video has no sound, or the video cannot be played at all).

Possible Causes

The codec of the original video is unsupported.

The original video is corrupted. As a result, the uploaded video cannot be played normally.

Troubleshooting Procedure

1. Check whether the codec of the original video is supported.

If yes, proceed with the next step.

If not, upload a video with a supported codec. For more information, see [Specifications and Limits](#).

2. Check whether the original video can be played normally using VLC or other professional players.

If yes, [contact us](#).

If not, you are advised to try using Tencent Cloud's transcoding service to fix the video. For more information, see [Video Transcoding](#).

A URL Points to a Wrong File

Last updated : 2024-03-25 18:01:58

Problems

Problem 1: The same URL points to different files.

Problem 2: After a file is updated, its old version is still accessed.

Possible Causes

The CDN cache has not expired.

The browser cache is not disabled.

The accessed file is hijacked, so the accessed file content is not as expected.

Troubleshooting

Check whether the CDN cache has expired

Check whether the CDN cache has expired as instructed in [Cache Configuration FAQs](#).

If so, [check whether the browser cache is disabled](#).

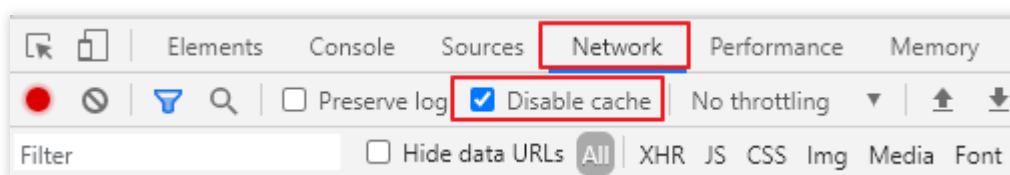
If not, purge CDN URLs or directories as instructed in [Purge Cache](#).

Check whether the browser cache is disabled

Note:

The following uses Chrome as an example.

1. Open Chrome.
2. Press **F12** to open the Developer tools.
3. Select the **Network** tab and check whether **Disable cache** is selected.



If so, [check whether the file is hijacked](#).

If not, select **Disable cache** and restart the browser.

Check whether the file is hijacked

If the file accessed is not as expected (for example, the `content-length` or response headers are not as expected), the file has been hijacked. In this case, we recommend you access it over HTTPS.

“HTTP ERROR 403” Is Returned When I Access COS Using a CDN Domain

Last updated : 2024-03-25 18:01:58

Symptom

When I use a Content Delivery Network (CDN) domain to access Cloud Object Storage (COS), the error code “HTTP ERROR 403” is returned.

Possible Causes

The CDN acceleration domain is disabled.

Troubleshooting Procedure

1. Log in to the [COS console](#).
2. Click **Bucket List** in the left sidebar.
3. Click the name of the desired bucket to go to the configuration page.
4. Select **Domains and Transfer > Default CDN Acceleration Domain**.
5. In the **Default CDN Acceleration Domain** area, check whether the **Status** is **Enabled**.
If not, [enable default acceleration domain](#).
If yes, proceed with the next step.
6. In the **Custom CDN Acceleration Domain** area, check whether the **Status** is **Online**.
If yes, [contact us](#).
If not, [enable custom acceleration domain](#).

Accessing a Custom Origin Domain over HTTPS Failed

Last updated : 2024-03-25 18:01:58

Symptom

When I access a custom origin domain over HTTPS, an error is reported.

Possible Causes

The certificate configuration is incorrect or the custom origin domain is not configured.

Troubleshooting

Using a CDN certificate

1. Log in to the [CDN console](#).
 2. On the left sidebar, click **Domain Management**.
 3. Click the domain to configure. Then, select the **HTTPS Configuration** tab.
 4. In the **HTTPS Configuration** area, click **Configure Certificate**.
- For more information, see [HTTPS Configuration Guide](#).
5. Wait for about 5 minutes. When the CDN domain is deployed again, you can use HTTPS.

Using a COS certificate

1. Log in to the [COS console](#).
2. Click **Bucket List** on the left sidebar.
3. Click the name of the target bucket to go to the configuration page.
4. Click **Domains and Transfer > Custom Endpoint**.
5. Select the target domain and click **Bind Certificate** to configure the certificate.
6. Click **OK**.

When the value of **HTTPS certificate** is **Uploaded**, you can use HTTPS.

Using a CVM reverse proxy certificate

For more information, see [Supporting HTTPS for Custom Endpoints](#).

POST Object Common Exceptions

Last updated : 2024-03-25 18:01:58

Symptoms

When POST requests are initiated via COS APIs, the following error messages are returned:

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]

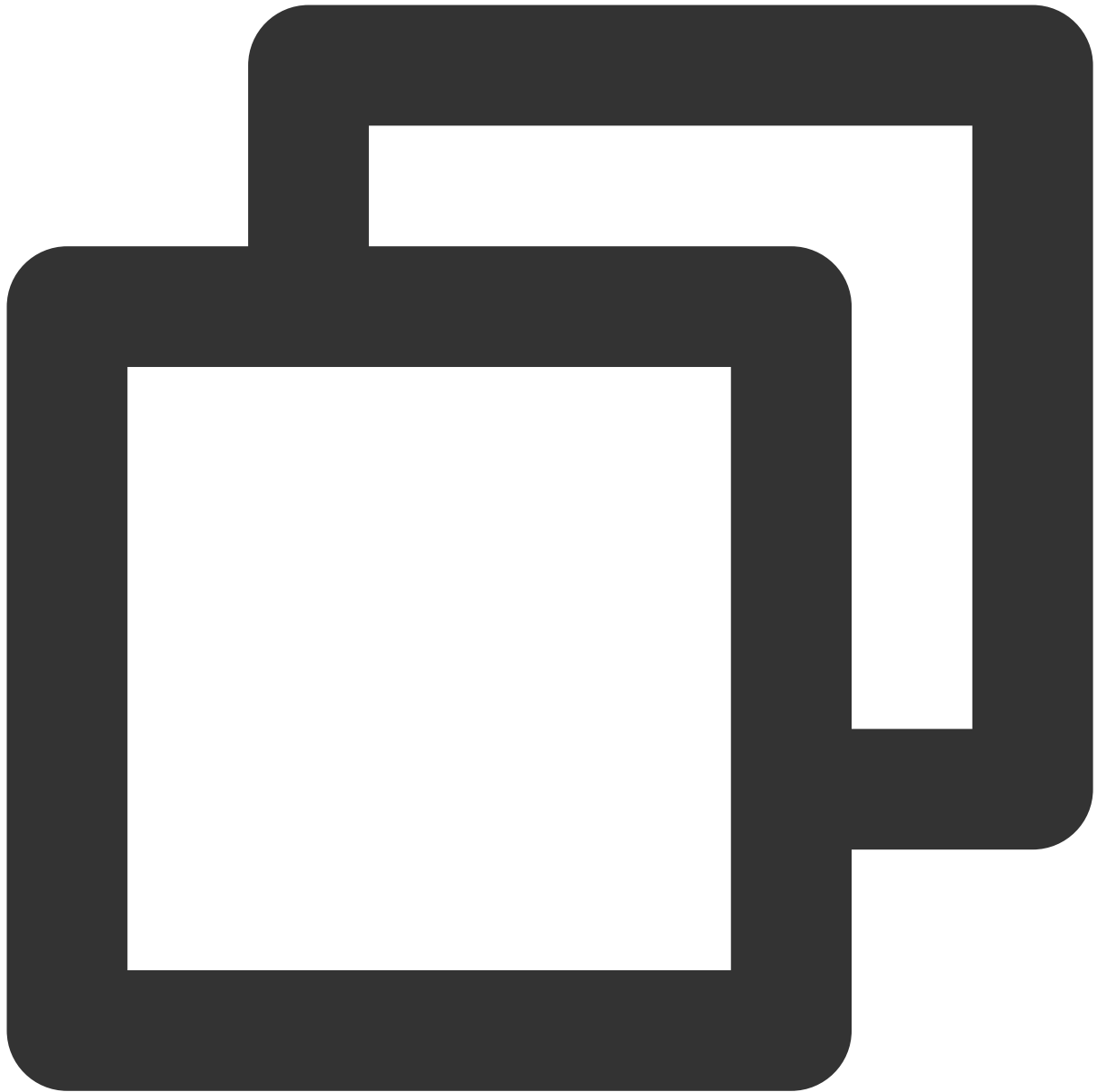
Condition key key doesn't match the value XXXXX

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

Troubleshooting

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

When POST requests are initiated via COS APIs, the following information is displayed:



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

Possible cause

`q-ak` parameter input error.

Solution

1. Log in to the CAM console and go to the [Manage API Key](#) page to view the key information.

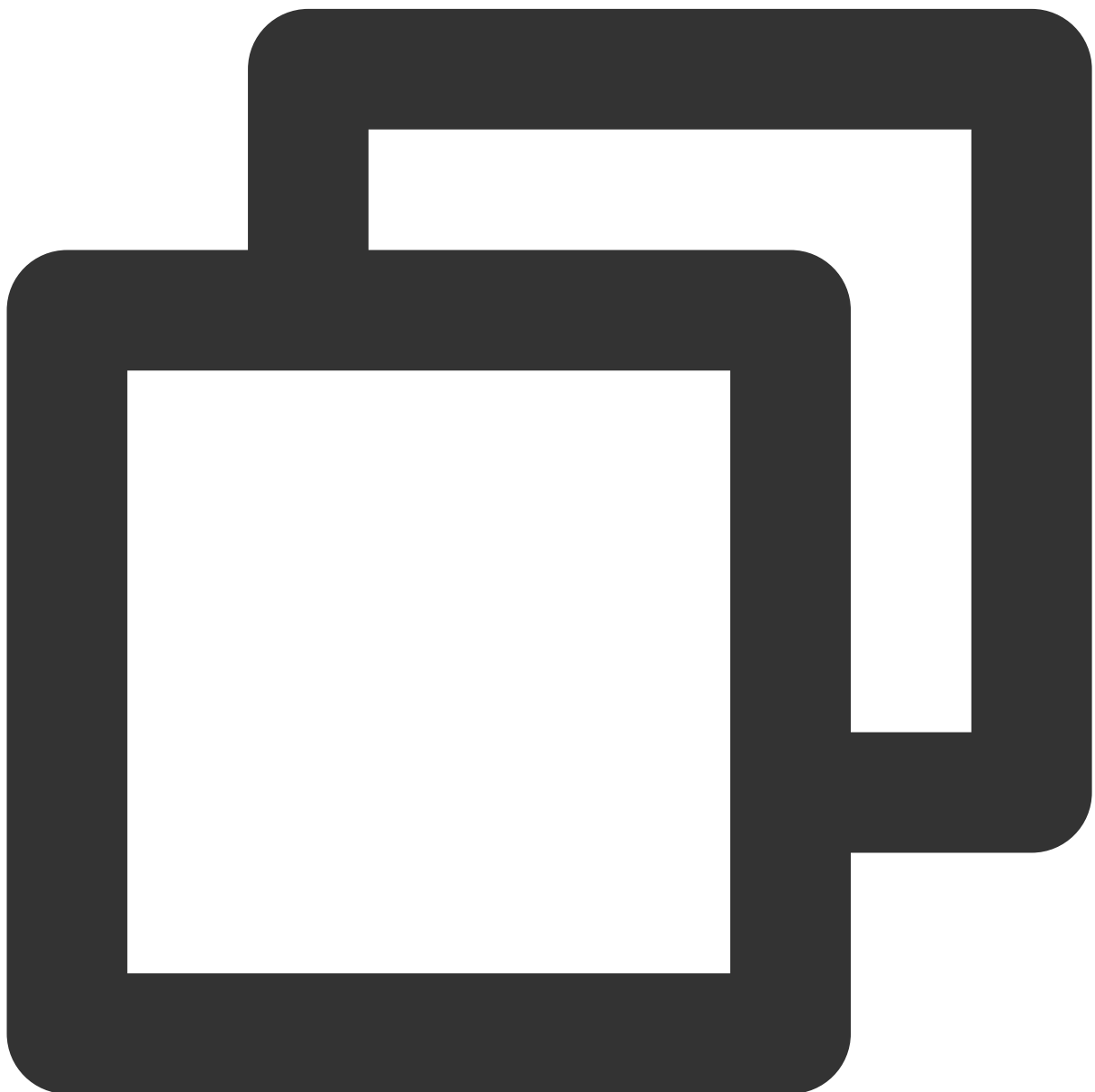
2. Check whether the `q-ak` parameter is incorrectly set based on the key information:

Yes: change the value of the `q-ak` parameter to the correct SecretId.

No: [contact us](#).

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

When POST requests are initiated via COS APIs, the following information is displayed:



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

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

Possible cause

The value of `expiration` in the policy has expired.

Solution

Change the value of `expiration` in the policy.

Note:

The value of `expiration` must be later than the current time. The recommended value is a time 30 minutes later than the current time (UTC time).

Error message "The Signature you specified is invalid."

When POST requests are initiated via COS APIs, the following information is displayed:



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

Possible cause

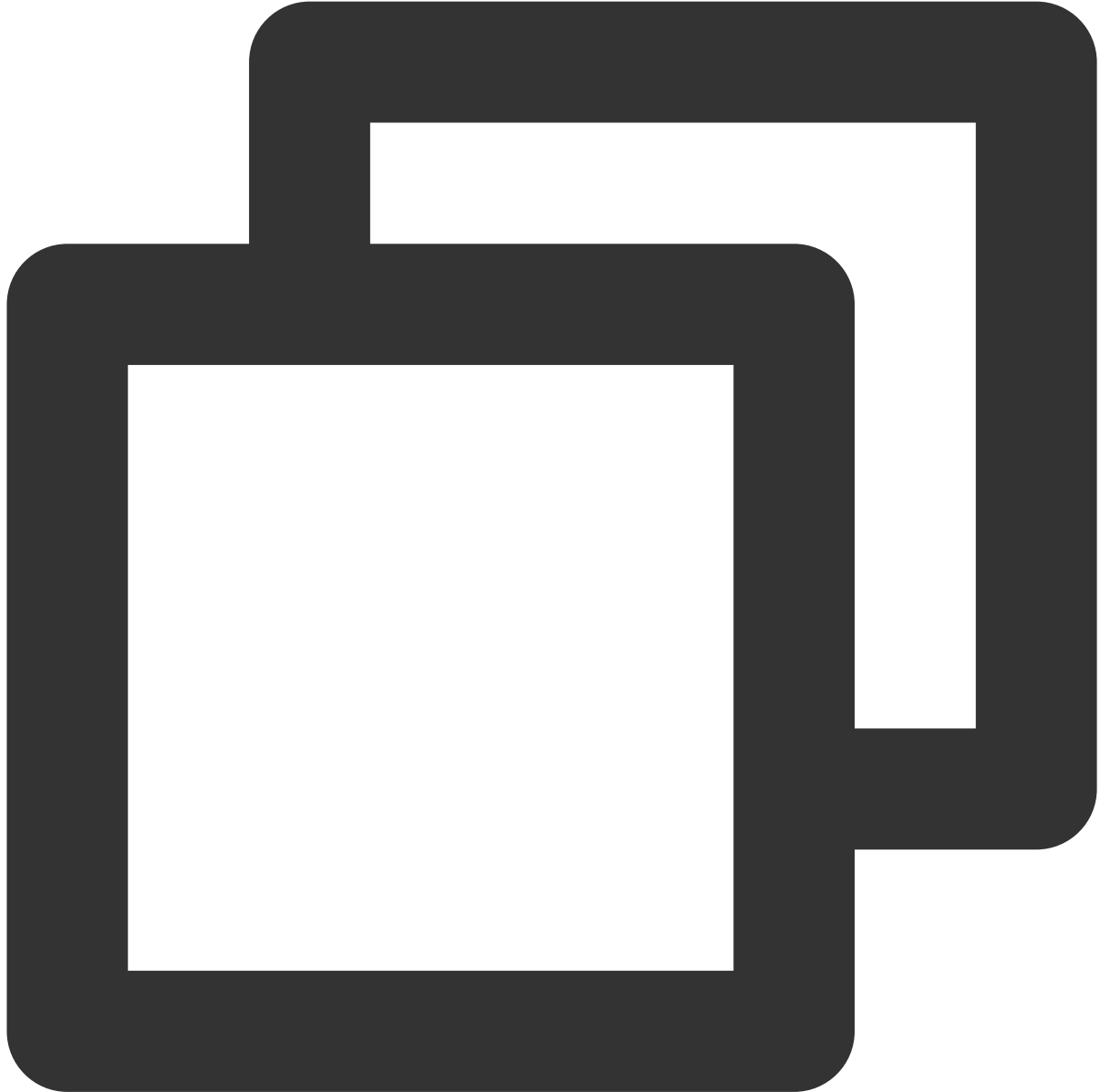
Signature calculation error.

Solution

Check whether the POST signature string generation rule is correct by referring to [Request Signature](#) :

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

When POST requests are initiated via COS APIs, the following information is displayed:



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

Possible cause

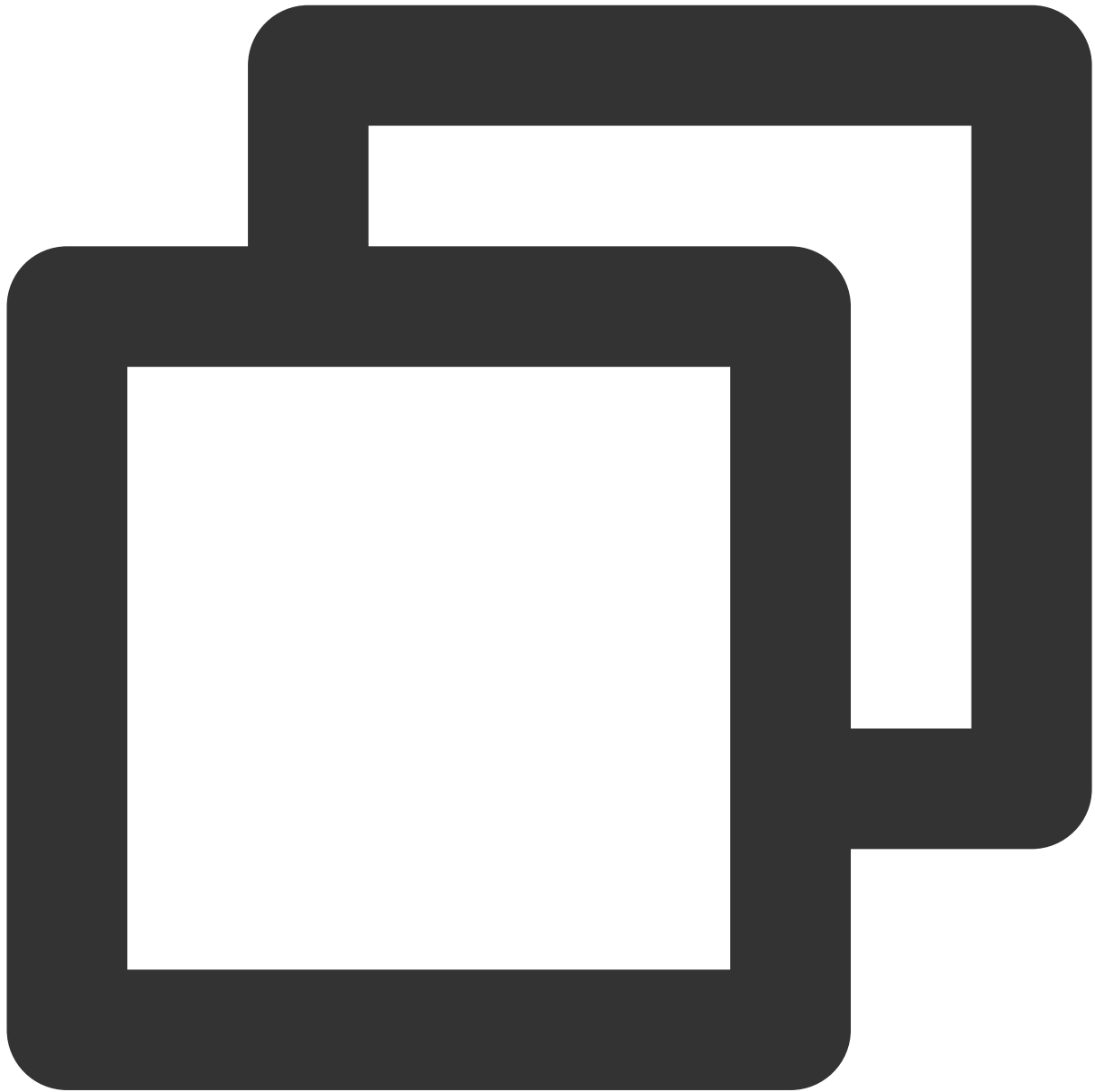
Policy format error.

Solution

Change the policy format to the standard JSON format by referring to [POST Object](#).

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

When POST requests are initiated via COS APIs, the following information is displayed:



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

Possible cause

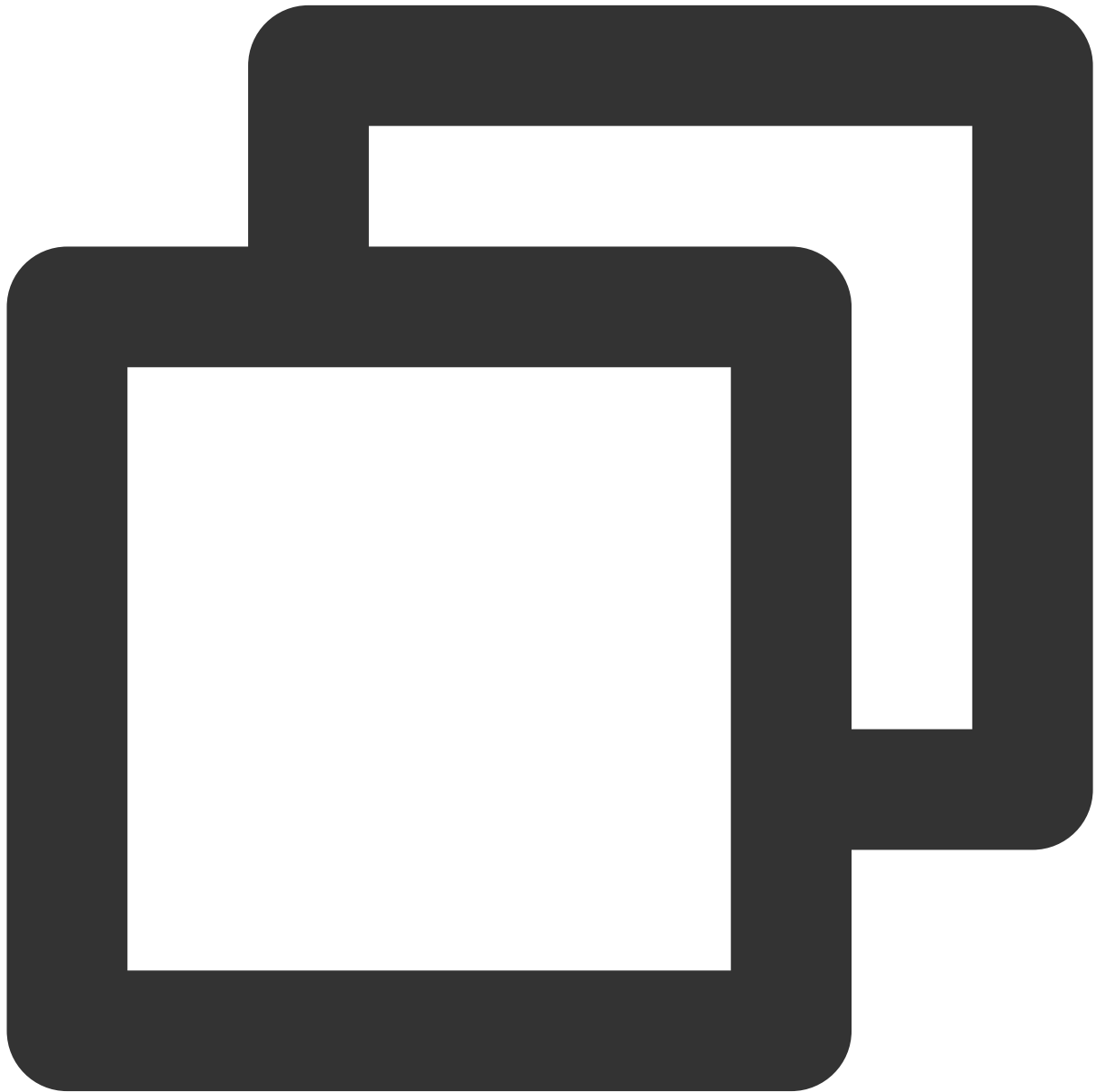
The bucket in the policy is inconsistent with that in the request.

Solution

Use the bucket specified in the policy to initiate the request.

Error message "Condition key key doesn't match the value XXXXX"

When POST requests are initiated via COS APIs, the following information is displayed:



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

Possible cause

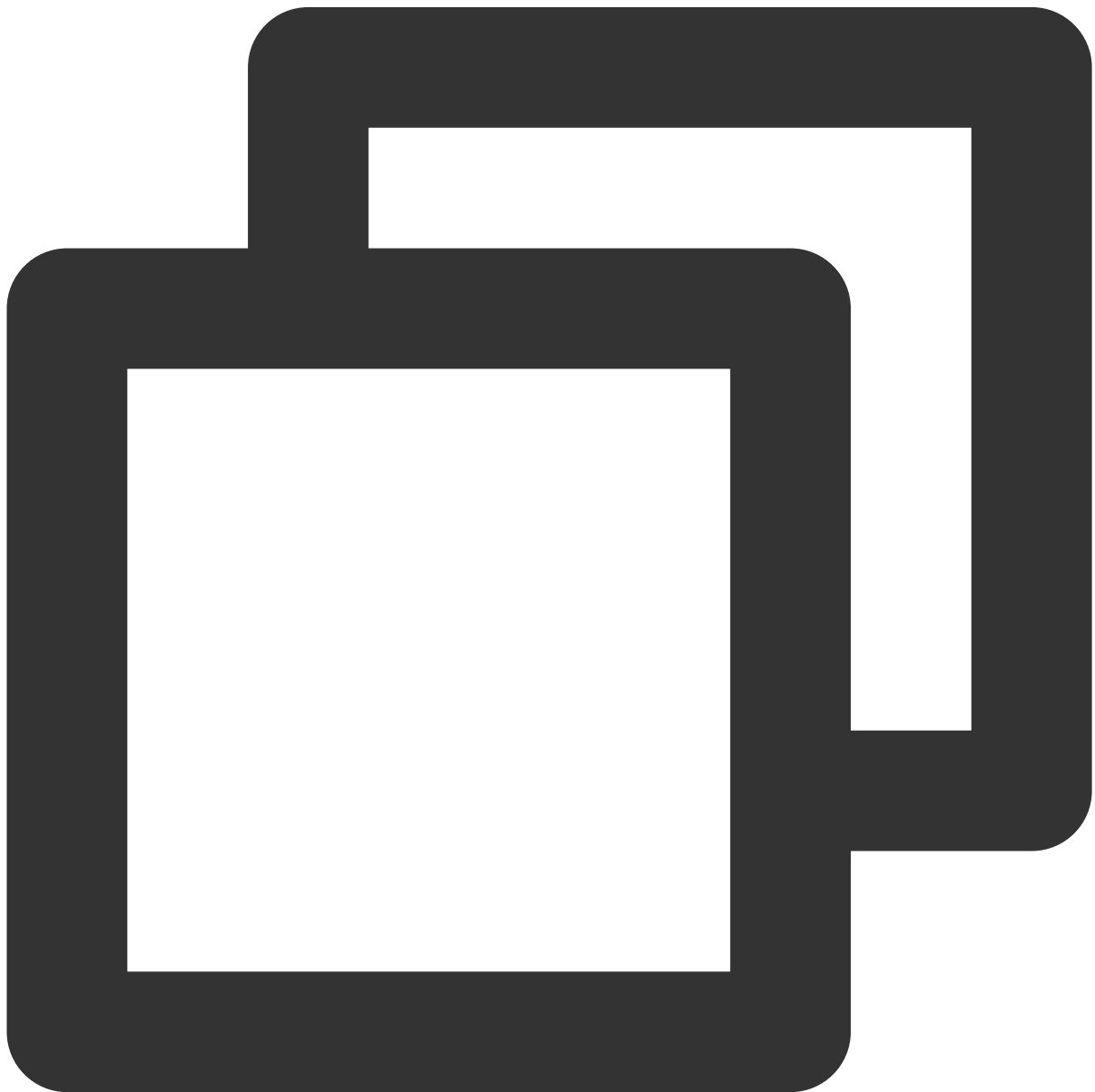
The uploaded content does not comply with the policy.

Solution

Upload content that meets the conditions specified in the policy.

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

When POST requests are initiated via COS APIs, the following information is displayed:



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

Possible cause

The POST body format is invalid.

Solution

Correct the body format by referring to [POST Object](#).