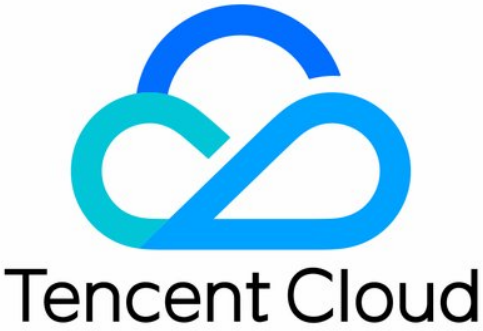
Batch Compute

Command Line Interface

Product Documentation



©2013-2022 Tencent Cloud. All rights reserved.

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

Tencent Cloud

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

Command Line Interface
- Preparation
- Quick Start
- Running Remote Package
- Mapping Remote Storage

©2013-2022 Tencent Cloud. All rights reserved.

Command Line Interface Preparation

Last updated：2024-01-13 11:19:29

Before installing TCCLI, make sure that your system has the Python environment installed. For details, see prerequisites instructed in Installing TCCLI.

Step 1: Installing TCCLI

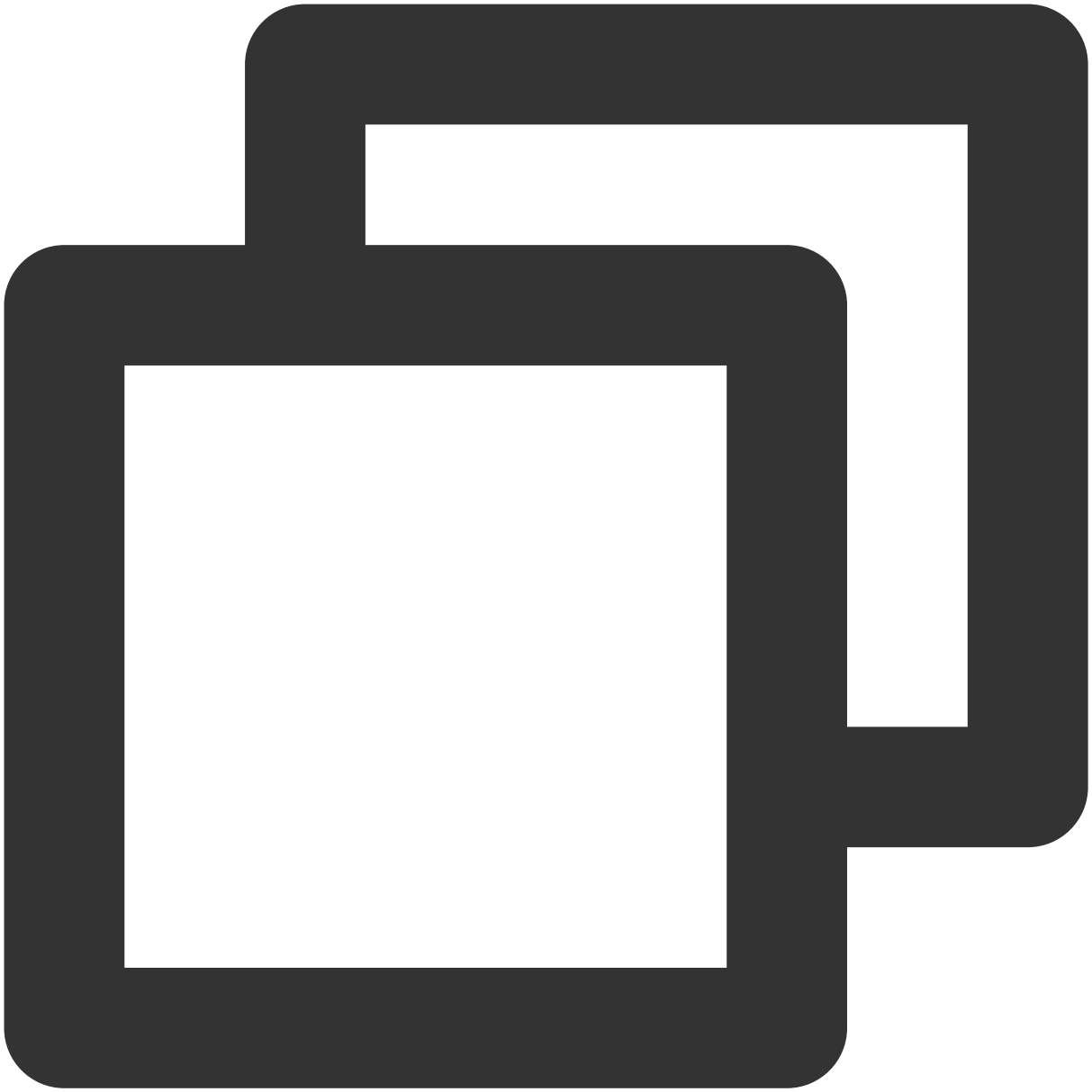
Installing TCCLI

Run commands based on the actual situation.

TCCLI not installed

Run the following command to quickly install TCCLI via pip. For details, please see Installing TCCLI.

©2013-2022 Tencent Cloud. All rights reserved.

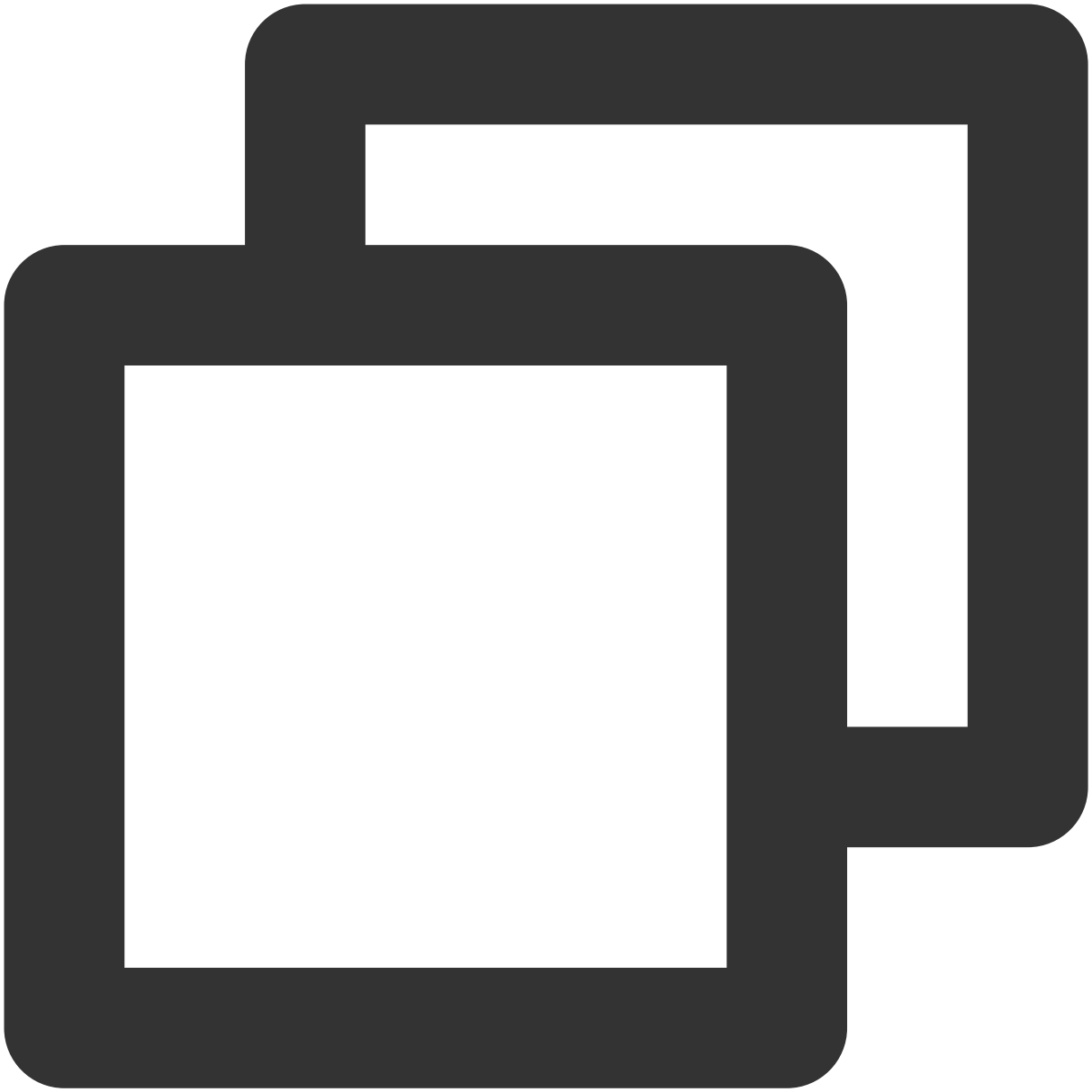


```
$ sudo pip install tccli
```

TCCLI installed

Run the following command to quickly upgrade TCCLI through pip:

©2013-2022 Tencent Cloud. All rights reserved.

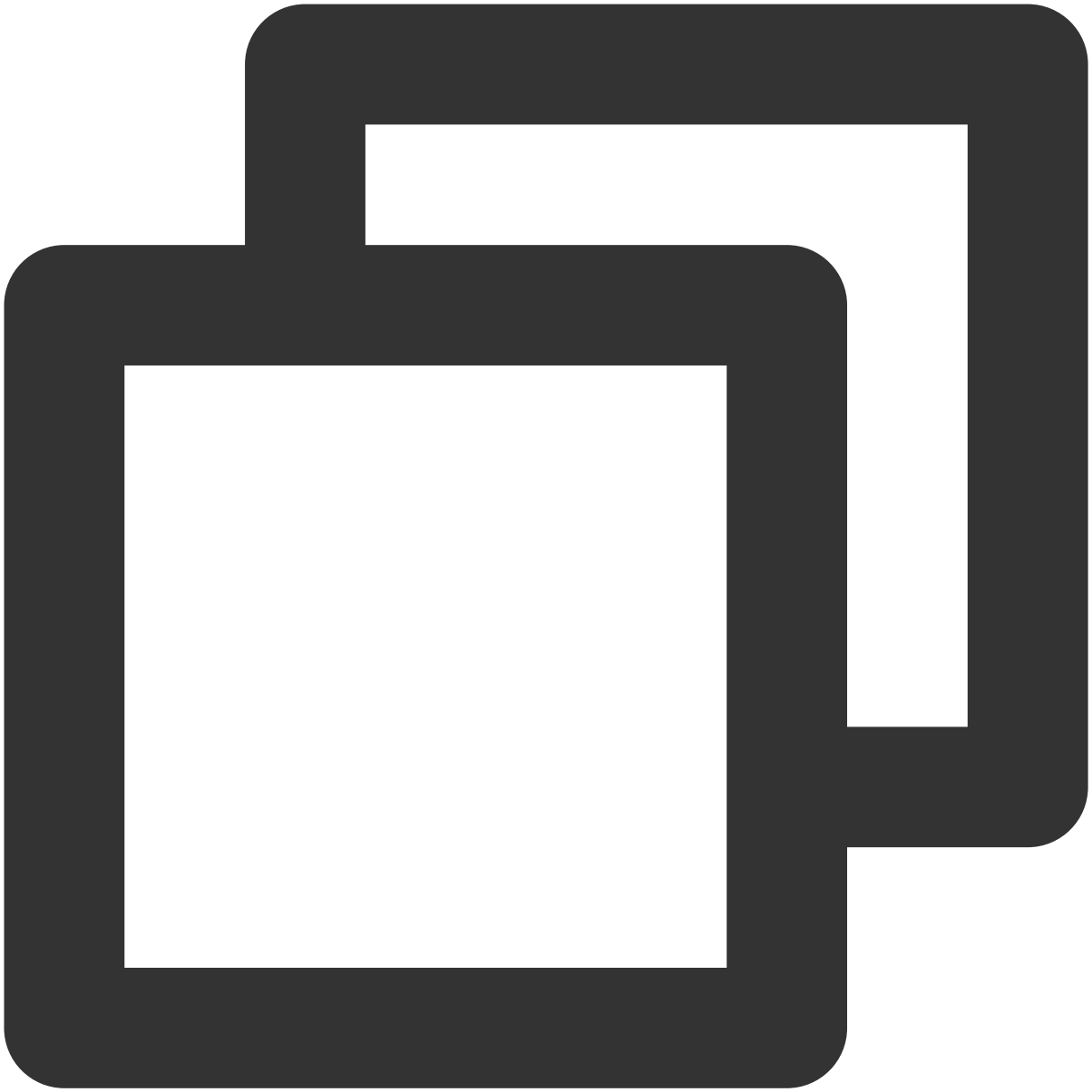


```
$ sudo pip install --upgrade tccli
```

Verifying installation

Run the following command to check whether TCCLI is successfully installed and has Batch-related capabilities:

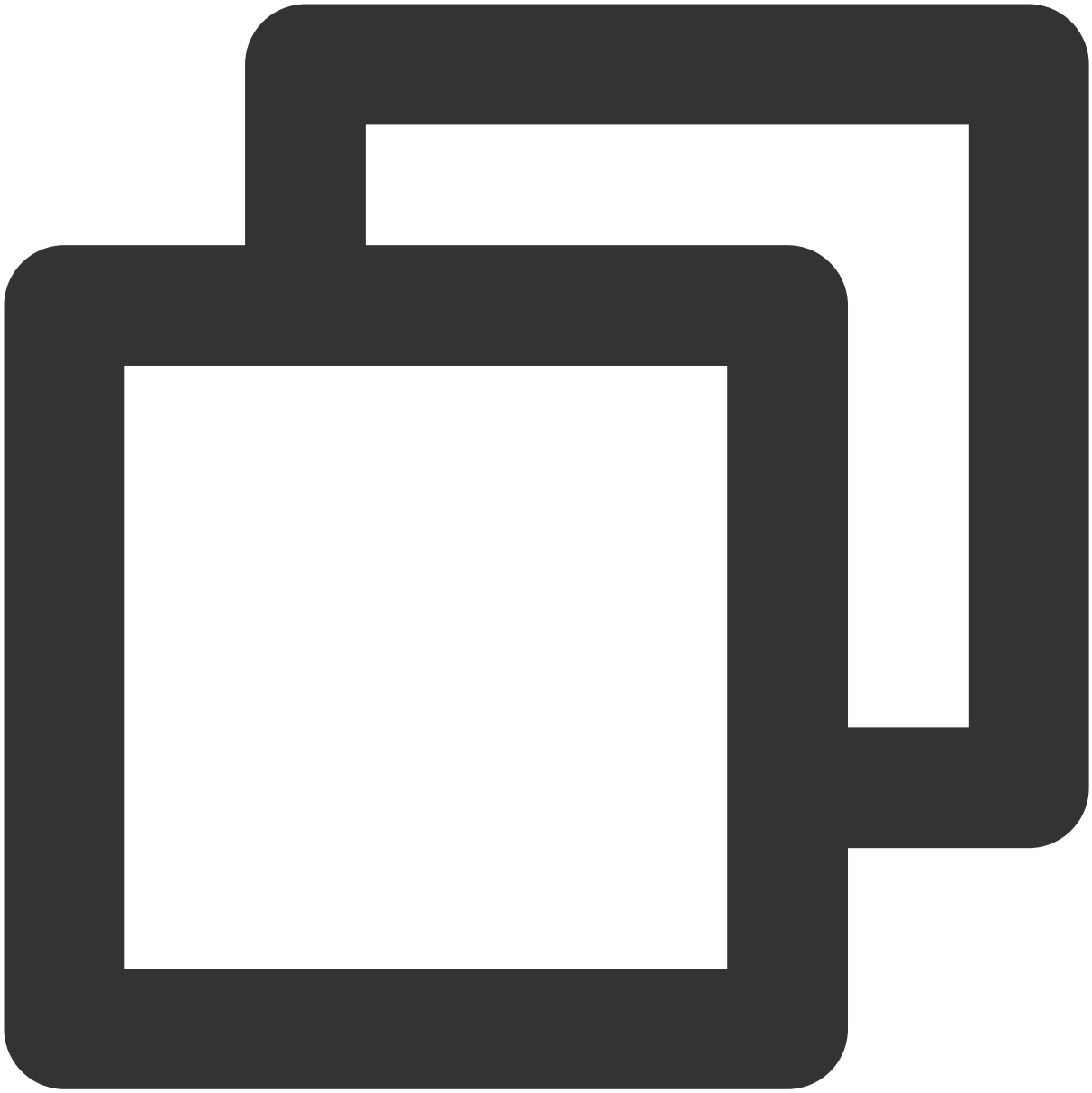
©2013-2022 Tencent Cloud. All rights reserved.



```
tccli batch help
```

The returned result is as follows, indicating that TCCLI is successfully installed:

©2013-2022 Tencent Cloud. All rights reserved.



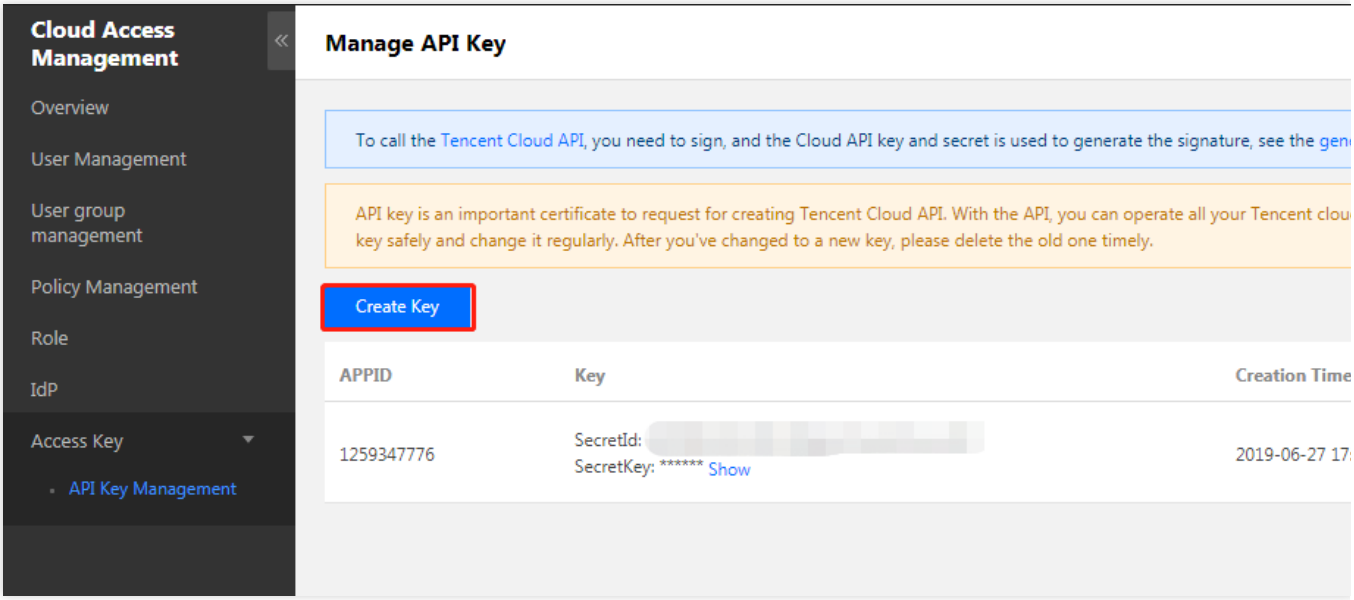
```
NAME
        batch
DESCRIPTION
        batch-2017-03-12
USEAGE
        tccli batch <action> [--param...]
OPTIONS
        help
        show the tccli batch help info
        --version
        specify a batch api version
```

©2013-2022 Tencent Cloud. All rights reserved.

```
AVAILABLE ACTION
        DescribeComputeEnv
        Used to query details of the computing environment
        CreateTaskTemplate
        Used to create a task template
```

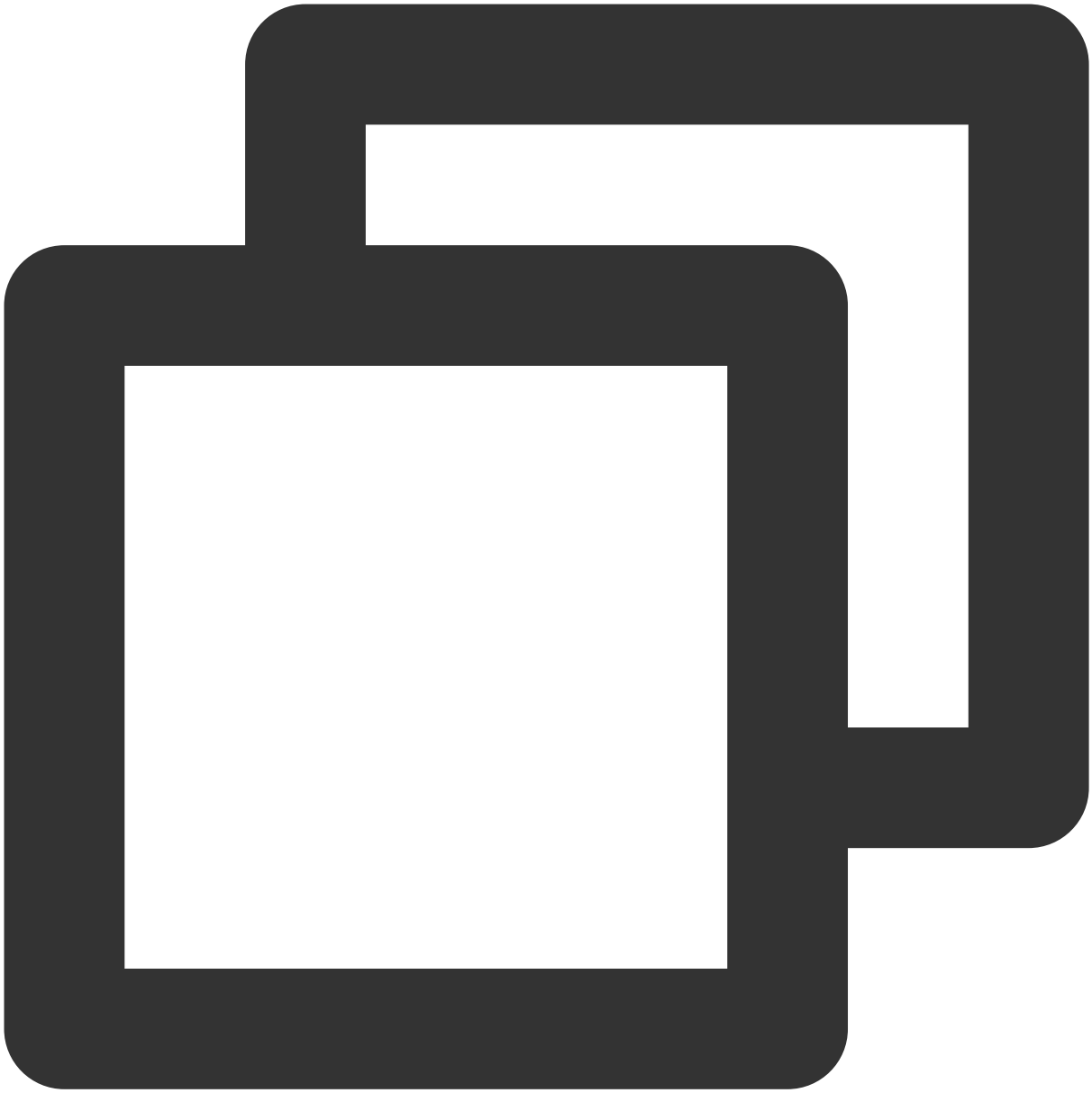
Step 2: Configuring TCCLI

1. Log in to the API Key Management.
2. Click **Create Key** or use an available key to record `SecretID` and `SecretKey` . See the figure below:



3. Run the command `tccli configure` and enter the TCCLI configuration information. For details, please see Configuring TCCLI.

©2013-2022 Tencent Cloud. All rights reserved.



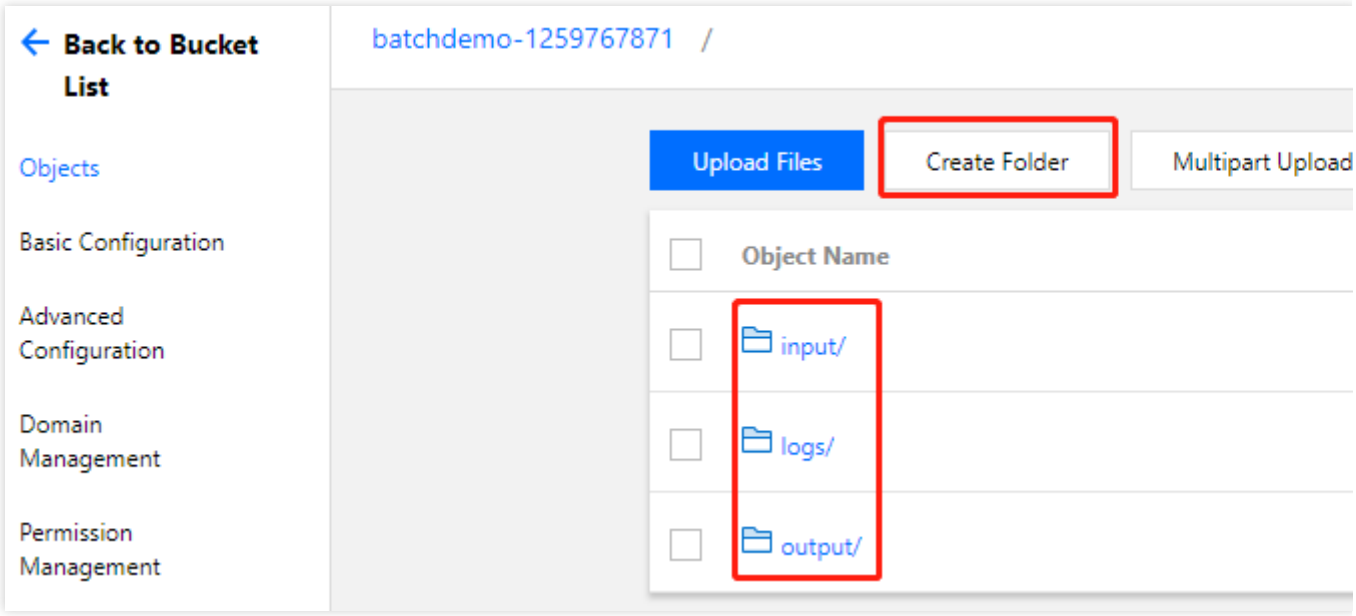
```
$ tccli configure
TencentCloud API secretId[None]:
TencentCloud API secretKey[None]:
region[None]:
output[json]:
```

Step 3: Preparing the COS Directory

©2013-2022 Tencent Cloud. All rights reserved.

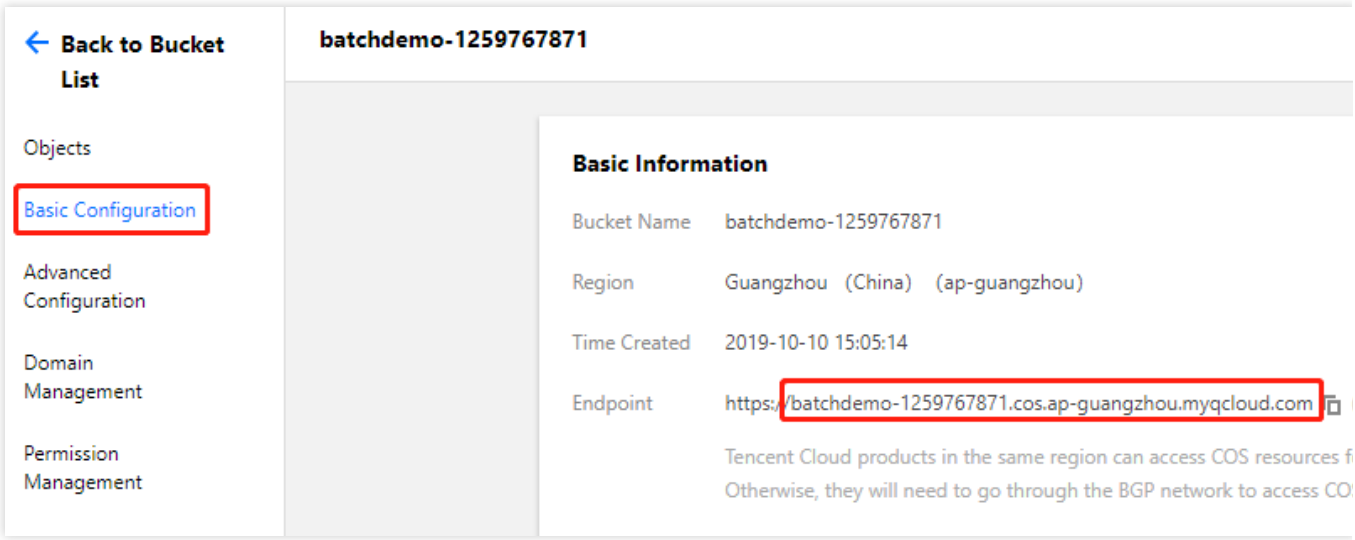
Creating a bucket and sub-folders

1. Log in to the COS console and choose **Bucket List** in the left sidebar.
2. Create a bucket and create 3 folders in the bucket. See the figure below:



Acquiring COS-related endpoints

1. Click **Basic Configuration** on the left to view the endpoint in **Basic Information**. See the figure below:



2. Acquire the endpoints of subfolders in the COS bucket.

Note:

Acquire COS-related endpoints based on the actual situation.

The acquired COS bucket endpoint is `https://batchdemo-xxxxxxxxx.cos.ap-guangzhou.myqcloud.com`. The endpoints of the three folders created in Creating a Bucket and Sub-folders can

©2013-2022 Tencent Cloud. All rights reserved.

be acquired by combining `domain and sub-folder names` as follows:

```
cos://batchdemo-xxxxxxxxx.cos.ap-guangzhou.myqcloud.com/logs/
cos://batchdemo-xxxxxxxxx.cos.ap-guangzhou.myqcloud.com/input/
cos://batchdemo-xxxxxxxxx.cos.ap-guangzhou.myqcloud.com/output/
```

Step 4: Downloading the Demo File

Access the Batch demo, download the test package, and decompress it.

Note:

The demo is provided in the format of the Python+Batch CLI. Since BatchCompute has many capabilities and configuration items, you can work with it more conveniently by using Python scripts.

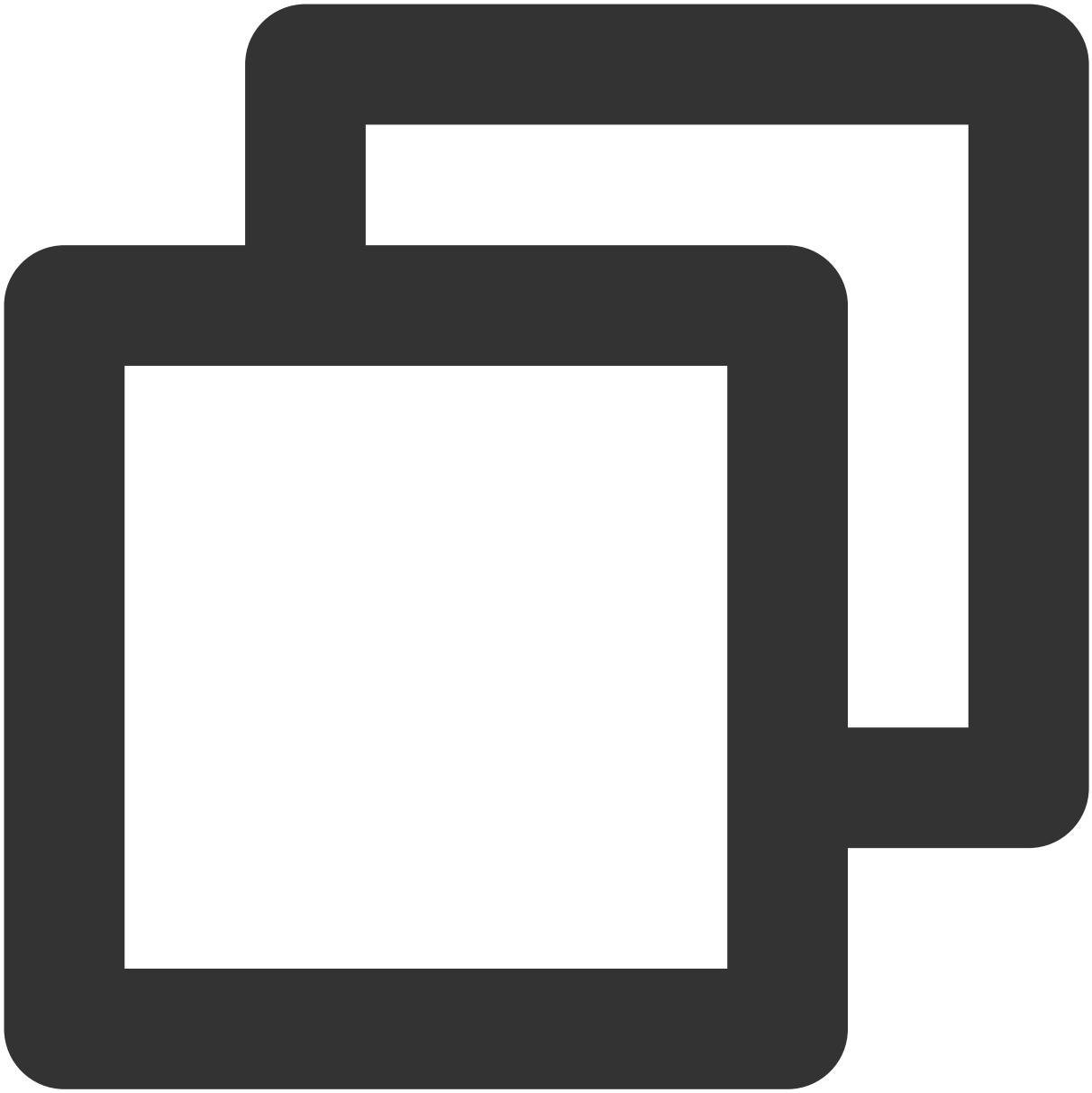
Step 5: Modifying Demo Custom Information

Note:

Change the general part of the custom information about the Batch demo. Modify all files in Demo as follows:

Use the following custom information in `1_SimpleStart.py` as an example:

©2013-2022 Tencent Cloud. All rights reserved.

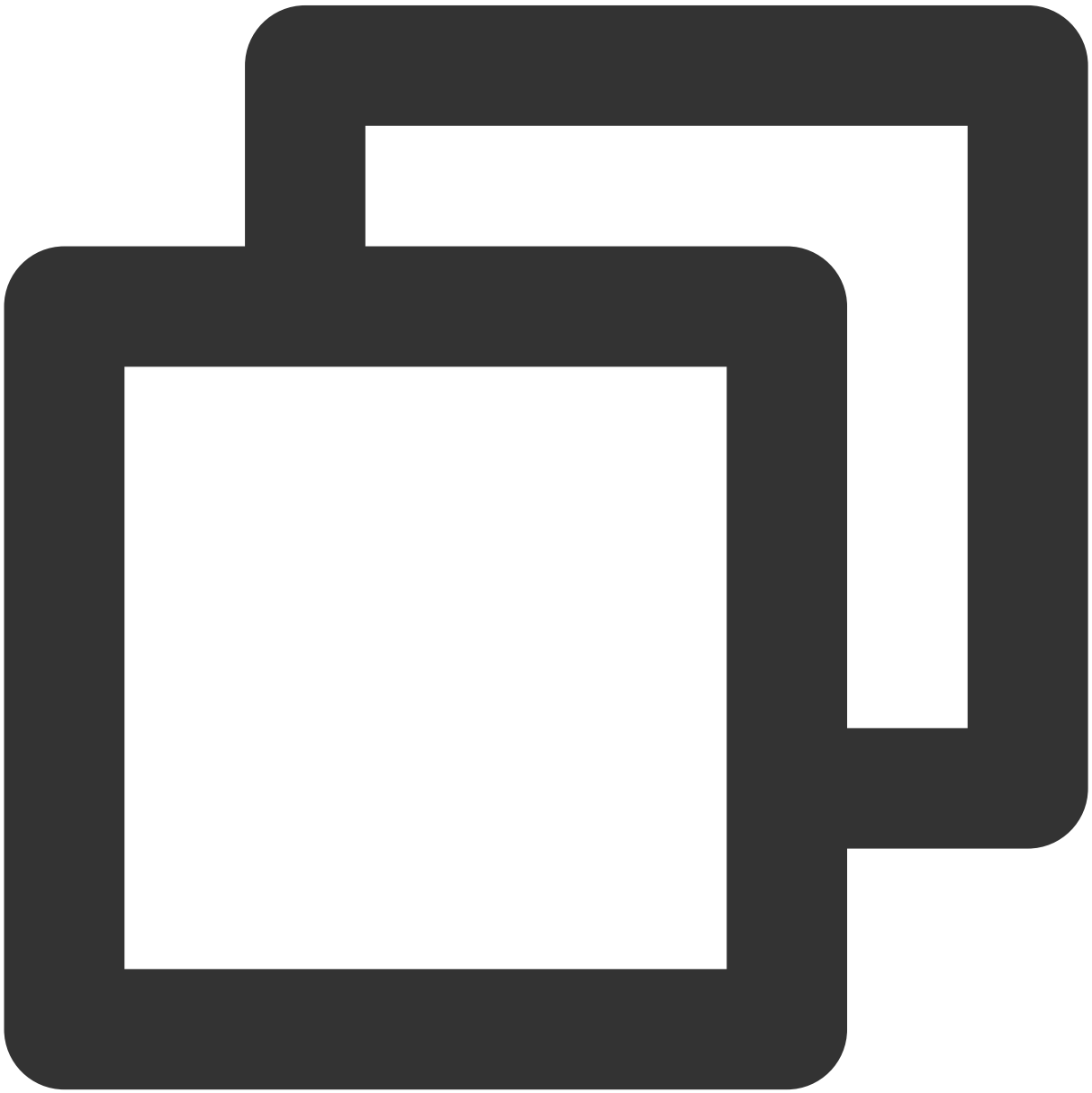


```
# custom (Change to your info)
imageId = "img-m4q71qnf"
Application = {
    "DeliveryForm": "LOCAL",
    "Command": " python -c \\"fib=lambda n:1 if n<=2 else fib(n-1)+fib(n-2); print(
}
StdoutRedirectPath = "cos://batchdemo-xxxxxxxxxx.cos.ap-guangzhou.myqcloud.com/logs
StderrRedirectPath = "cos://batchdemo-xxxxxxxxxx.cos.ap-guangzhou.myqcloud.com/logs
```

The following table lists the information to be modified.

©2013-2022 Tencent Cloud. All rights reserved.

Configuration Item	Description
imageId	Custom images are created based on this image. For more information, see Windows Custom Image.
StdoutRedirectPath	Enter the complete log folder endpoint acquired in Acquiring COS-related endpoints.
StderrRedirectPath	
Application	The startup command. Use the default setting.



©2013-2022 Tencent Cloud. All rights reserved.

```
cmd = "tccli batch SubmitJob \\
    --version 2017-03-12 \\
    --Placement '{\\"Zone\\": \\"ap-guangzhou-2\\"}' \\
    --Job ' %s ' "%(json.dumps(testJob))
```

The demo specifies Guangzhou Zone 2 for resource application. You can select the corresponding availability zone to apply for resources based on the default region configured in TCCLI.

For more information about regions and availability zones, see Regions and AZs.

Step 6: Testing

Experience the Batch usage methods and computing capability in the following sequence according to the reference course.

1. 1_SimpleStart.py: Quick Start
2. 2_RemoteCodePkg.py: Running Remote Package
3. 3_StoreMapping.py: Mapping Remote Storage

©2013-2022 Tencent Cloud. All rights reserved.

Quick Start

Last updated：2024-01-13 11:19:29

Overview

This document describes the usage methods and computing capability of Batch.

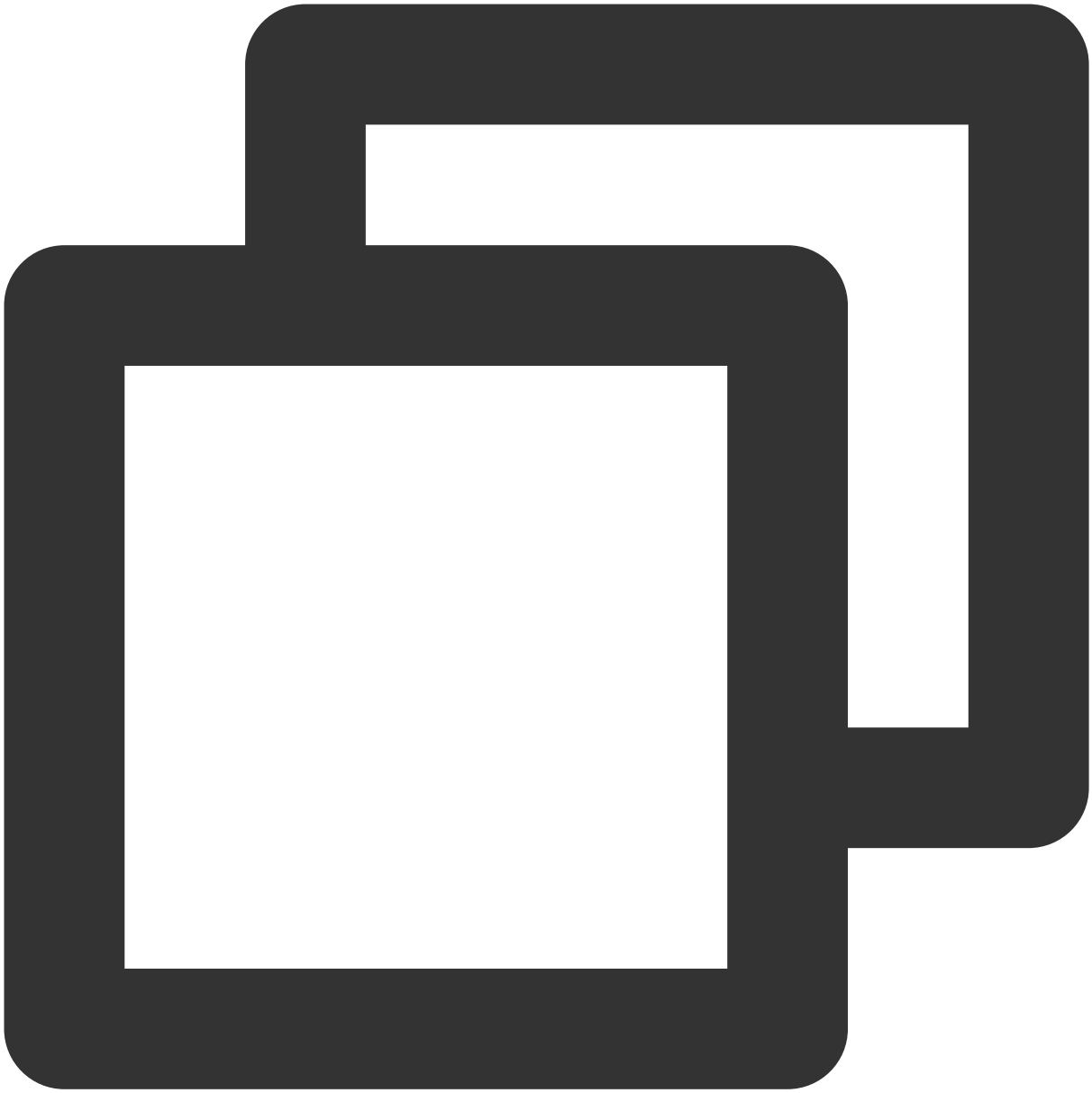
Prerequisites

Complete preparations based on the instructions in Preparation, and learn how to configure the general part of the custom information.

Viewing the Demo

Note:
Modify the general part of the custom information in `1_SimpleStart.py` based on the instructions in Preparation.
Open `1_SimpleStart.py` in an editor.

©2013-2022 Tencent Cloud. All rights reserved.



```
# custom (Change to your info)
imageId = "img-m4q71qnf"
Application = {
    "DeliveryForm": "LOCAL",
    "Command": " python -c \\"fib=lambda n:1 if n<=2 else fib(n-1)+fib(n-2); print(
}
StdoutRedirectPath = "your cos path"
StderrRedirectPath = "your cos path"
```

©2013-2022 Tencent Cloud. All rights reserved.

All information except `Application` in the general part is described in Preparation. Set the parameters of `Application` based on the following table.

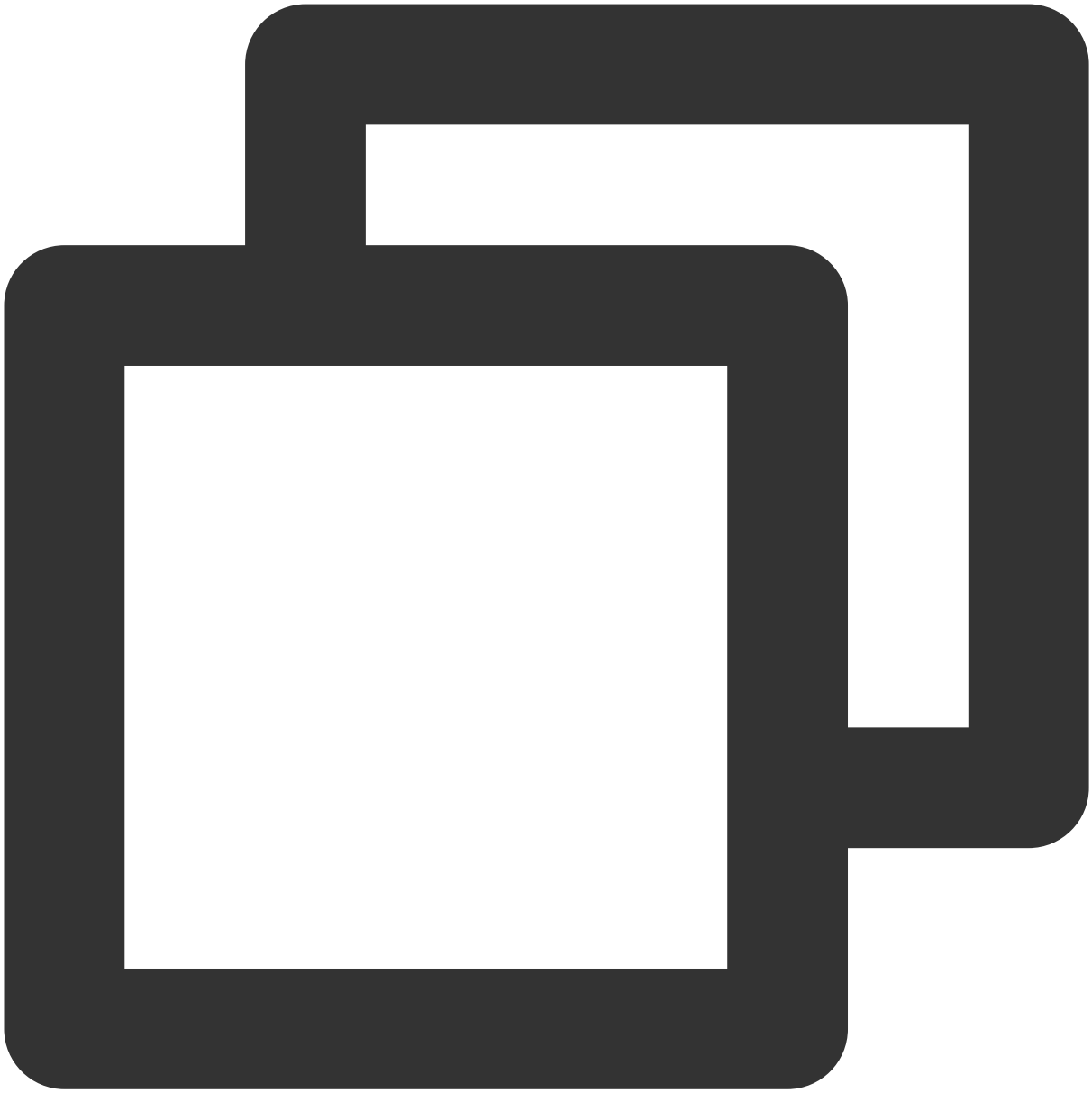
Configuration Item	Description
DeliveryForm	Three methods are available for delivering an application: software packaging, container image, and direct running within a CVM instance. LOCAL indicates direct running within a CVM instance.
Command:	The command for starting a job. In this example, a Python script is executed. Starting from 1, the script adds up the first 20 numbers of the Fibonacci sequence and outputs the result to StdOutput.

Submitting a Job

Run the following command to run the Python script.

The demo encapsulates the job submission process by using Python scripts and the Batch command line tool.

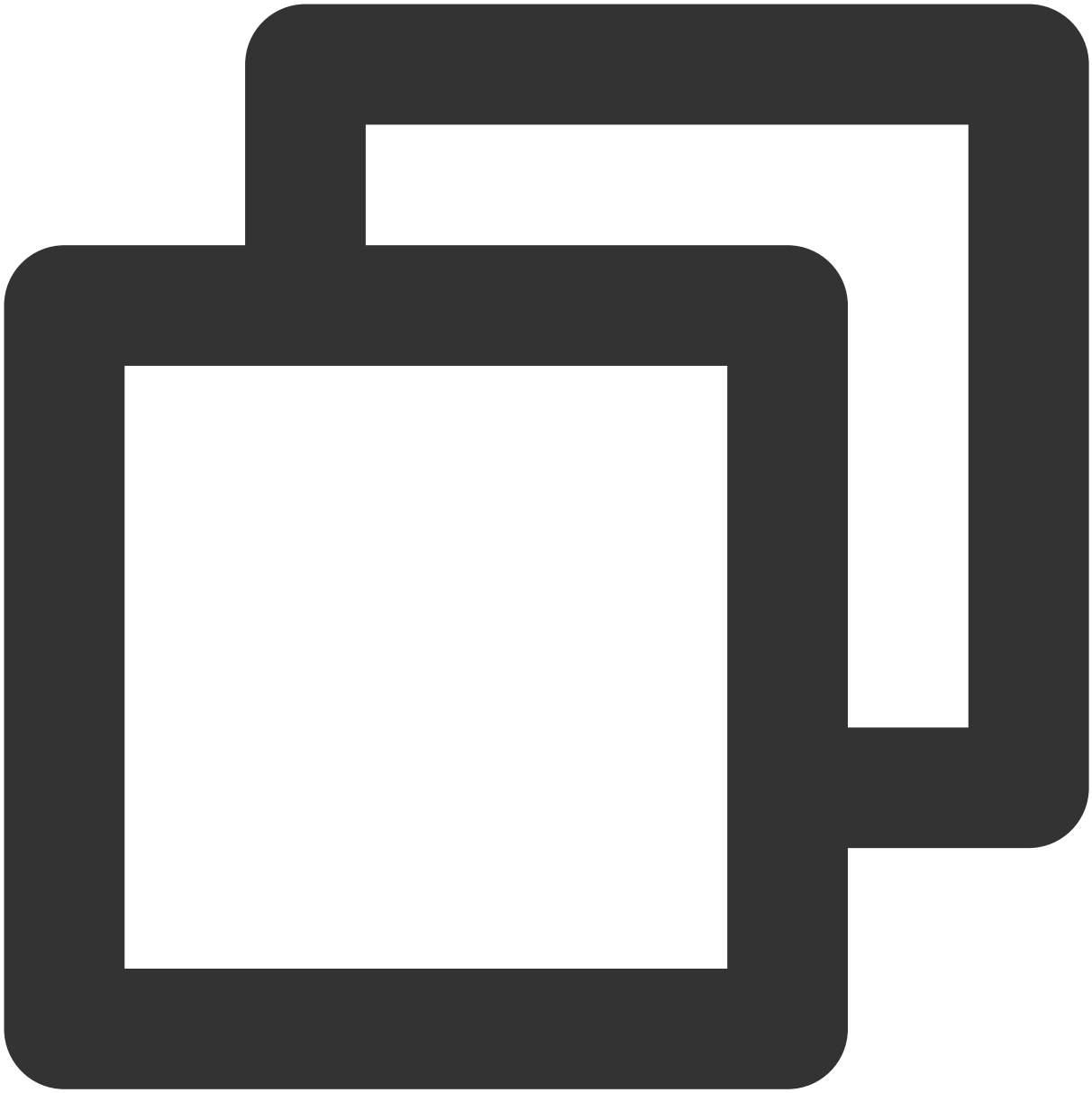
©2013-2022 Tencent Cloud. All rights reserved.



```
python 1_SimpleStart.py
```

The returned result is as follows, indicating that the job is successfully submitted:

©2013-2022 Tencent Cloud. All rights reserved.



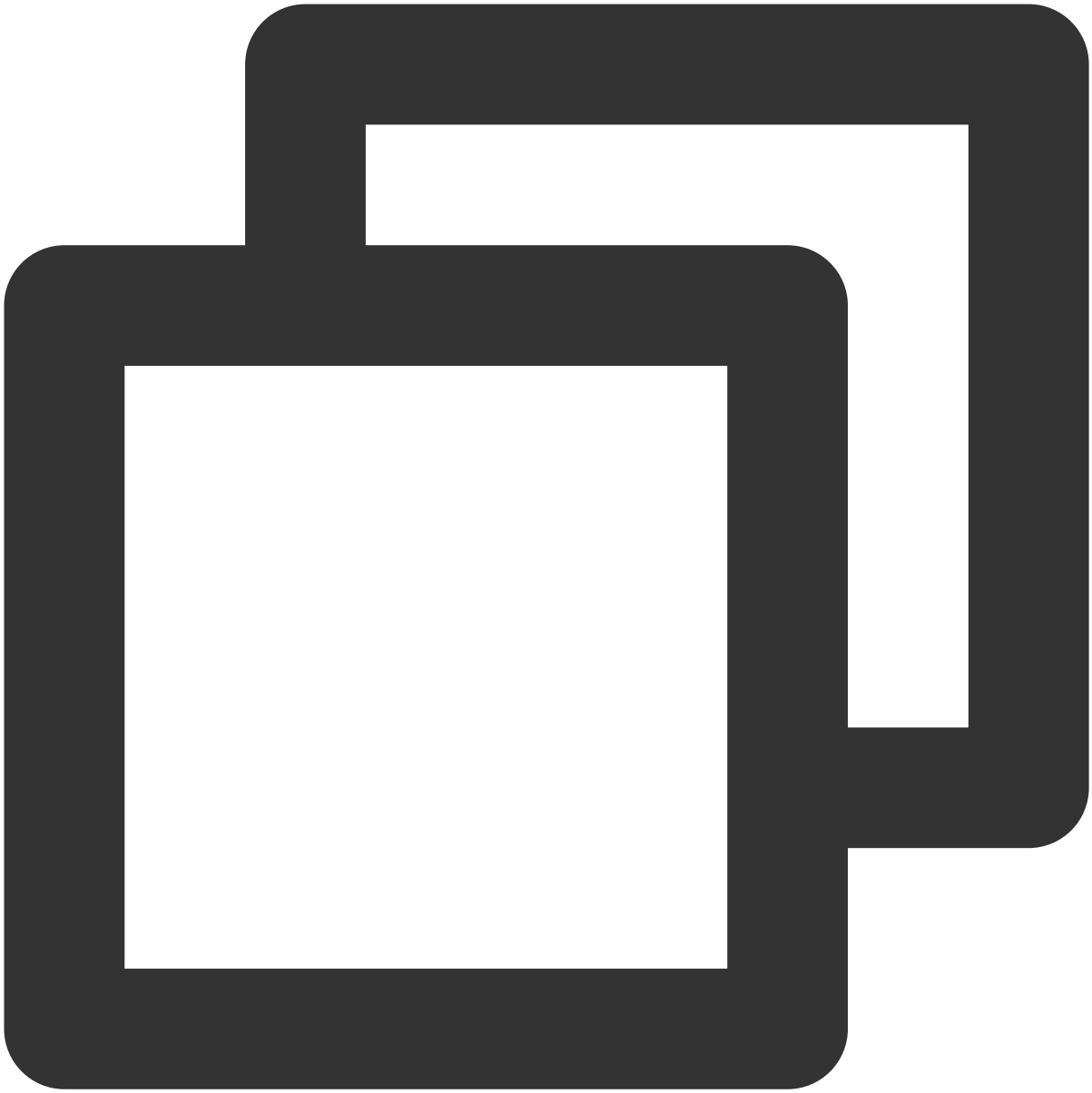
```
{
    "RequestId": "393292f4-5583-48ad-a9f5-f673138ea637",
    "JobId": "job-o0xxxxxq7"
}
```

If the submit operation fails, check the returned value, or contact us.

Viewing State

Run the following command to view the execution status through DescribeJob:

©2013-2022 Tencent Cloud. All rights reserved.



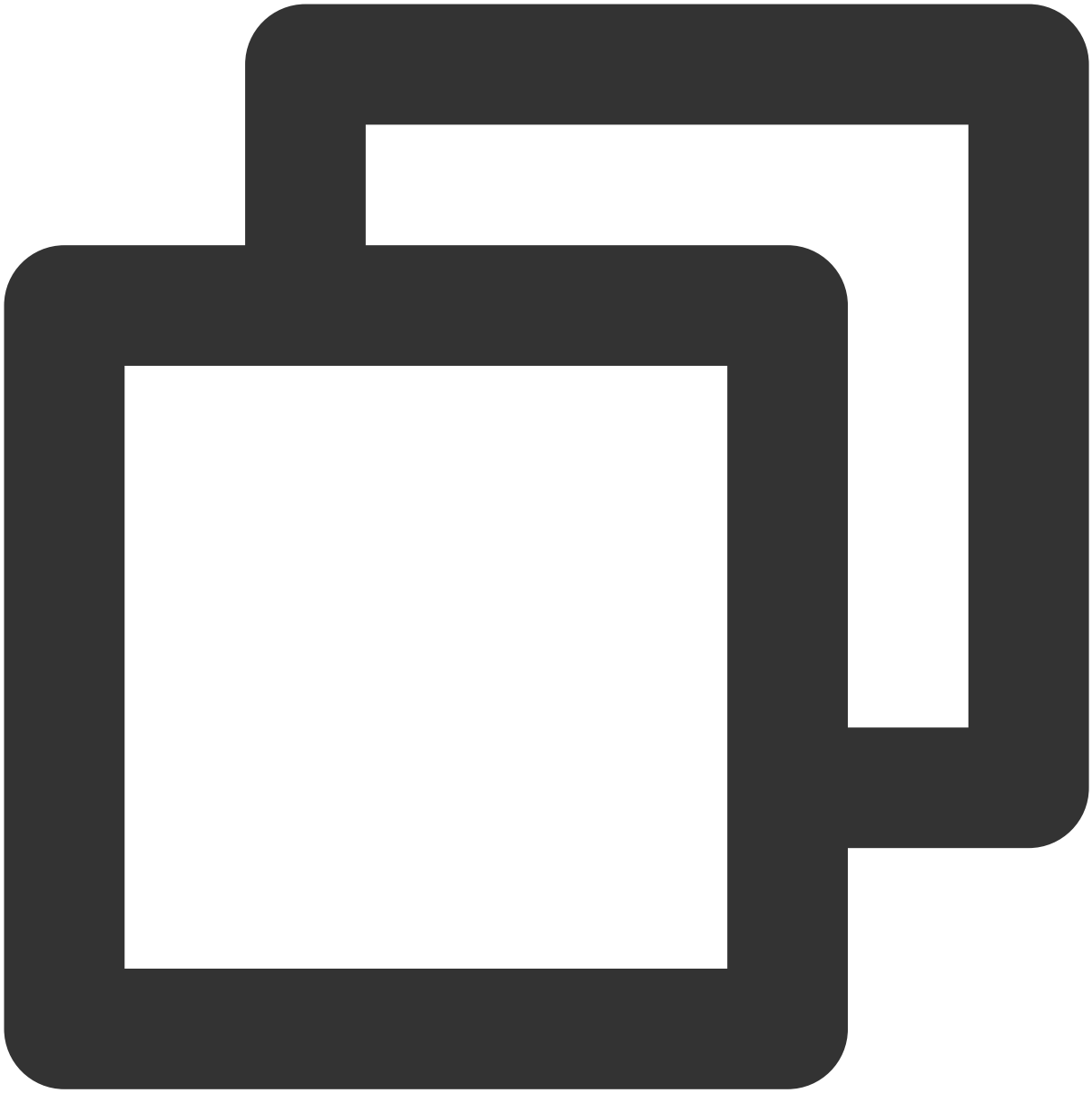
```
$ tccli batch DescribeJob --version 2017-03-12 --JobId job-xxx
```

Note:

--Replace job-xxx with the JobId that is returned after you submit the job.

The returned result is as follows (some information is omitted):

©2013-2022 Tencent Cloud. All rights reserved.




```
{
    "EndTime": "2019-10-08T04:06:58Z",
    "JobState": "SUCCEED",
    "TaskInstanceMetrics": {
        ...
    },
    "Zone": "ap-guangzhou-2",
    "TaskMetrics": {
        ...
    },
    "JobName": "TestJob",
```


©2013-2022 Tencent Cloud. All rights reserved.

```
    "Priority": 1,
    "RequestId": "7a5f4c94-1357-486c-9c48-8286ba01b5b2",
    "TaskSet": [
        ...
    ],
    "StateReason": null,
    "JobId": "job-o0xxxxxq7",
    "DependenceSet": [],
    "CreateTime": "2019-10-08T04:05:54Z"
}
```

The common values of **JobState** in the returned result are as follows:

STARTING: Launching

RUNNING: Running

SUCCEED: Running successfully

FAILED: Failed to run

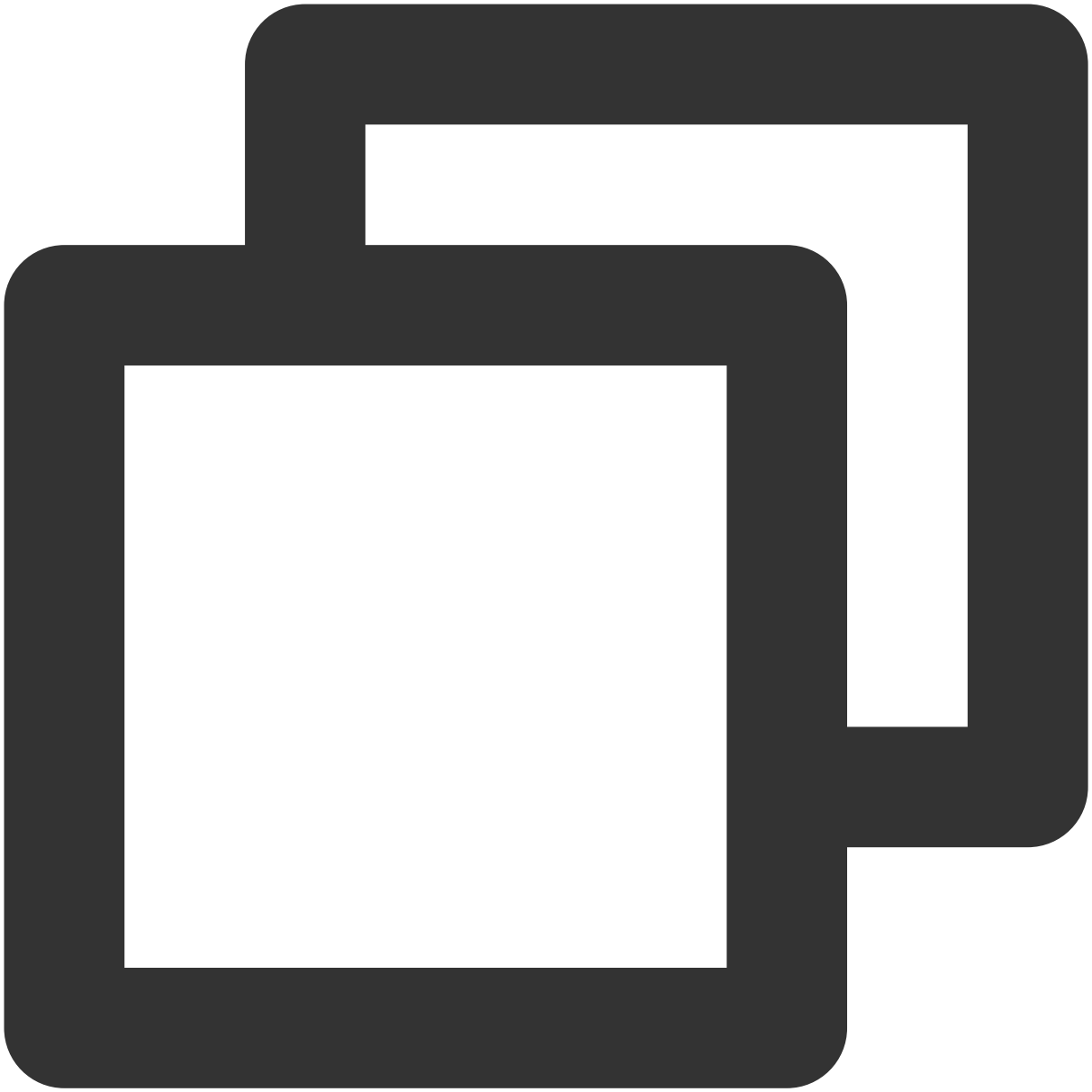
Viewing the Result

1. Log in to the COS console. In the left sidebar, click **Bucket List**.

2. Click the ID of the desired bucket, go to the object list page, and view the log files that store the execution results. See the figure below:

If the job succeeds, view the standard output in stdout.job-xxx.xxxx.0.log. The content is as follows:

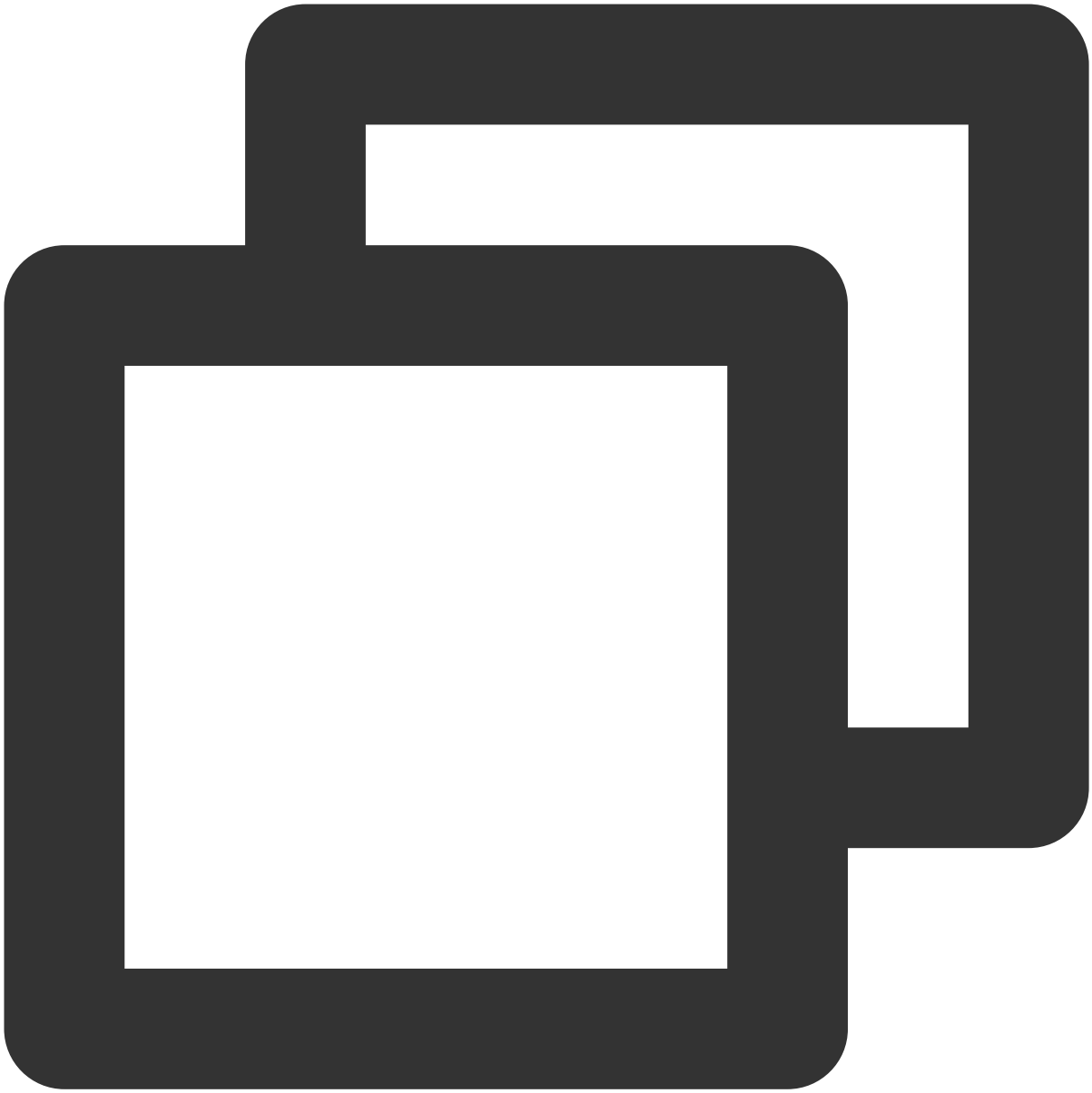
©2013-2022 Tencent Cloud. All rights reserved.



```
6765
```

If the job fails, view the standard error in stderr.job-xxx.xxxx.0.log. The content may be as follows:

©2013-2022 Tencent Cloud. All rights reserved.



```
/bin/sh: -c: line 0: syntax error near unexpected token `('
/bin/sh: -c: line 0: ` python -c \\"fib=lambda n:1 if n<=2 else fib(n-1)+fib(n-2);
```

©2013-2022 Tencent Cloud. All rights reserved.

Running Remote Package

Last updated：2024-01-13 11:19:29

Scenario

Batch allows you to acquire a code package from a .tgz file via HTTP. You can compress the code and upload it to COS. This helps you organize the code more conveniently than using LOCAL mode.

Prerequisites

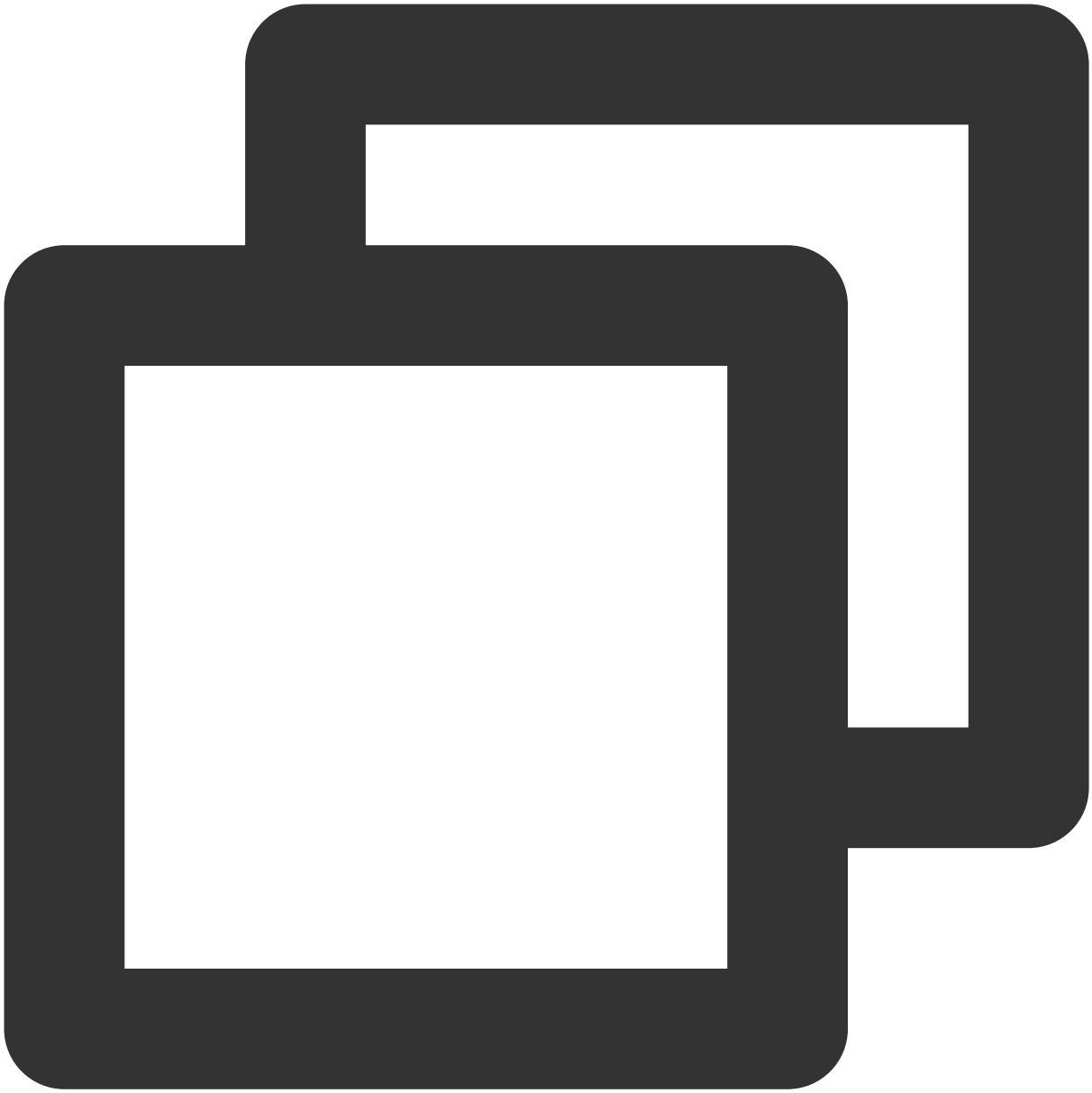
Complete preparations based on the instructions in Preparation, and learn how to configure the general part of the custom information.

Steps

Viewing the Demo

Note：
Modify the general part of the custom information in `2_RemoteCodePkg.py` based on the instructions in Preparation.
Open the `2_RemoteCodePkg.py` file in an editor.

©2013-2022 Tencent Cloud. All rights reserved.



```
# custom (Change to your info)
imageId = "img-m4q71qnf"
Application = {
    "DeliveryForm": "PACKAGE",
    "Command": "python ./codepkg/fib.py",
    "PackagePath": "http://batchdemo-xxxxxxxxx.cos.ap-guangzhou.myqcloud.com/codepk
}
StdoutRedirectPath = "your cos path"
StderrRedirectPath = "your cos path"
```

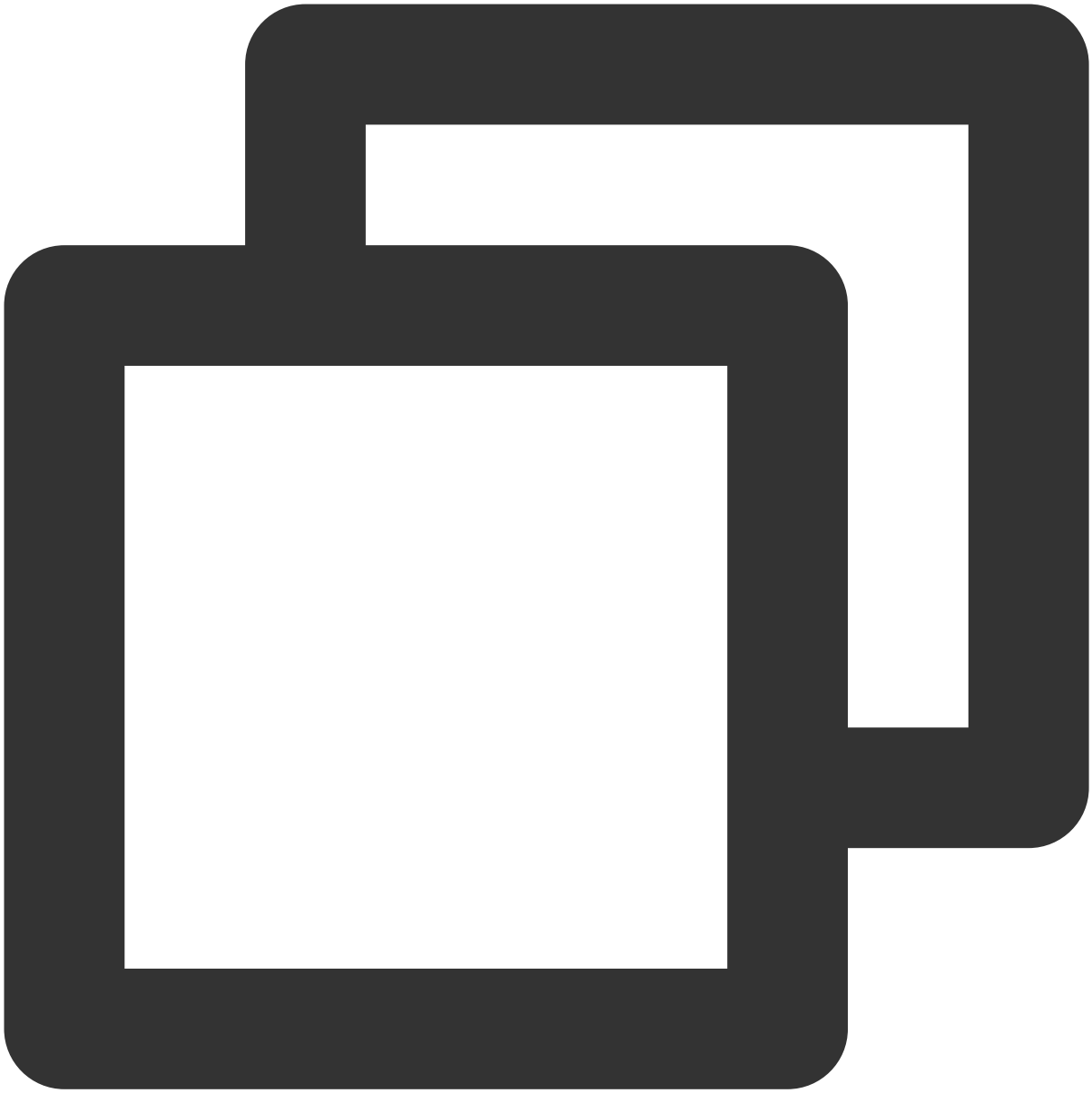
©2013-2022 Tencent Cloud. All rights reserved.

All information except `Application` in the general part is described in Preparation. Set the parameters of `Application` based on the following table.

Configuration Item	Description
DeliveryForm	Three delivery methods of applications are available: software packaging, container image, and direct running within the CVM. In this case, PACKAGE indicates software packaging.
PackagePath	Address of the .tgz software package in HTTP. Batch downloads the software package to a directory of the scheduled CVM and runs `Command` in the directory.
Command	Task startup command. In this case, a Python script file in the software package is directly called. You can download the package, and view the file structure and content in it.

`fib.py` is composed as below:

©2013-2022 Tencent Cloud. All rights reserved.



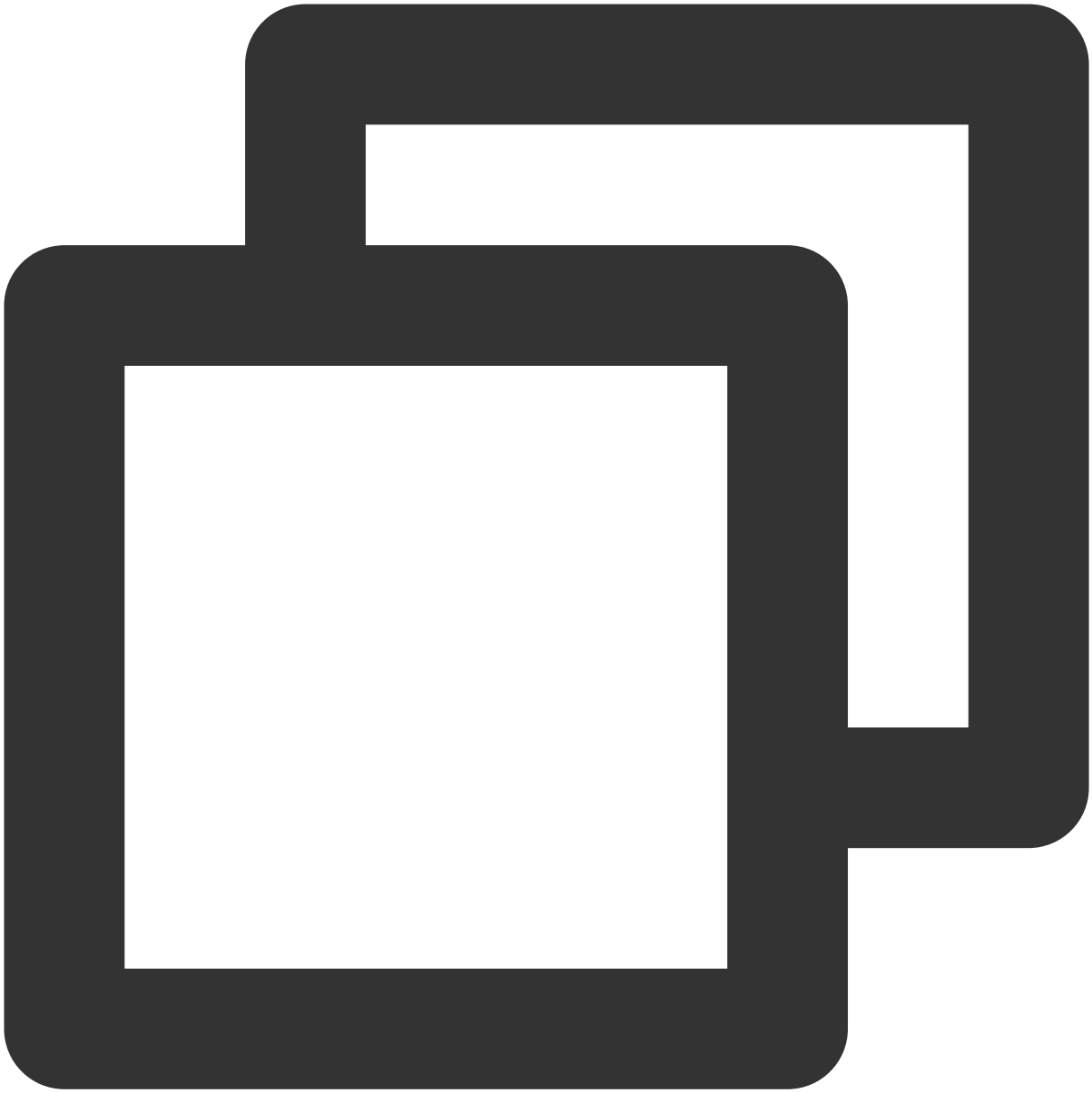
```
fib = lambda n:1 if n<=2 else fib(n-1)+fib(n-2)
print("Remote Code Package : %d"%(fib(20)))
```

Submitting a Job

Run the following command to run the Python script.

The demo encapsulates the job submission process by using Python scripts and the Batch command line tool.

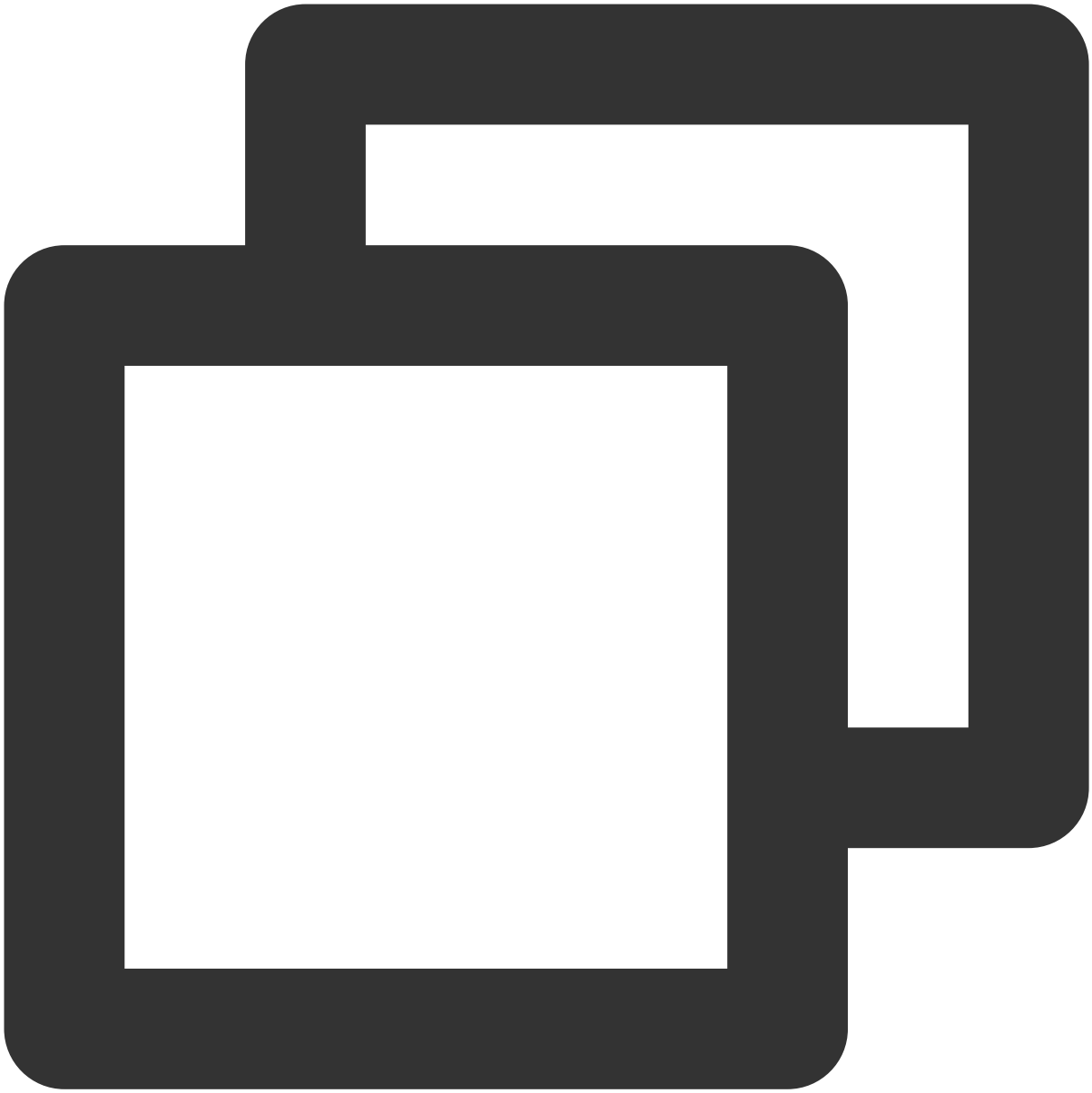
©2013-2022 Tencent Cloud. All rights reserved.



```
python 2_RemoteCodePkg.py
```

The returned result is as follows, indicating that the job is successfully submitted:

©2013-2022 Tencent Cloud. All rights reserved.



```
{
    "RequestId": "c09e9291-2661-xxxx-8783-72d36f91ec8a",
    "JobId": "job-7xxxx26l"
}
```

If the submit operation fails, check the returned value or Contact Us.

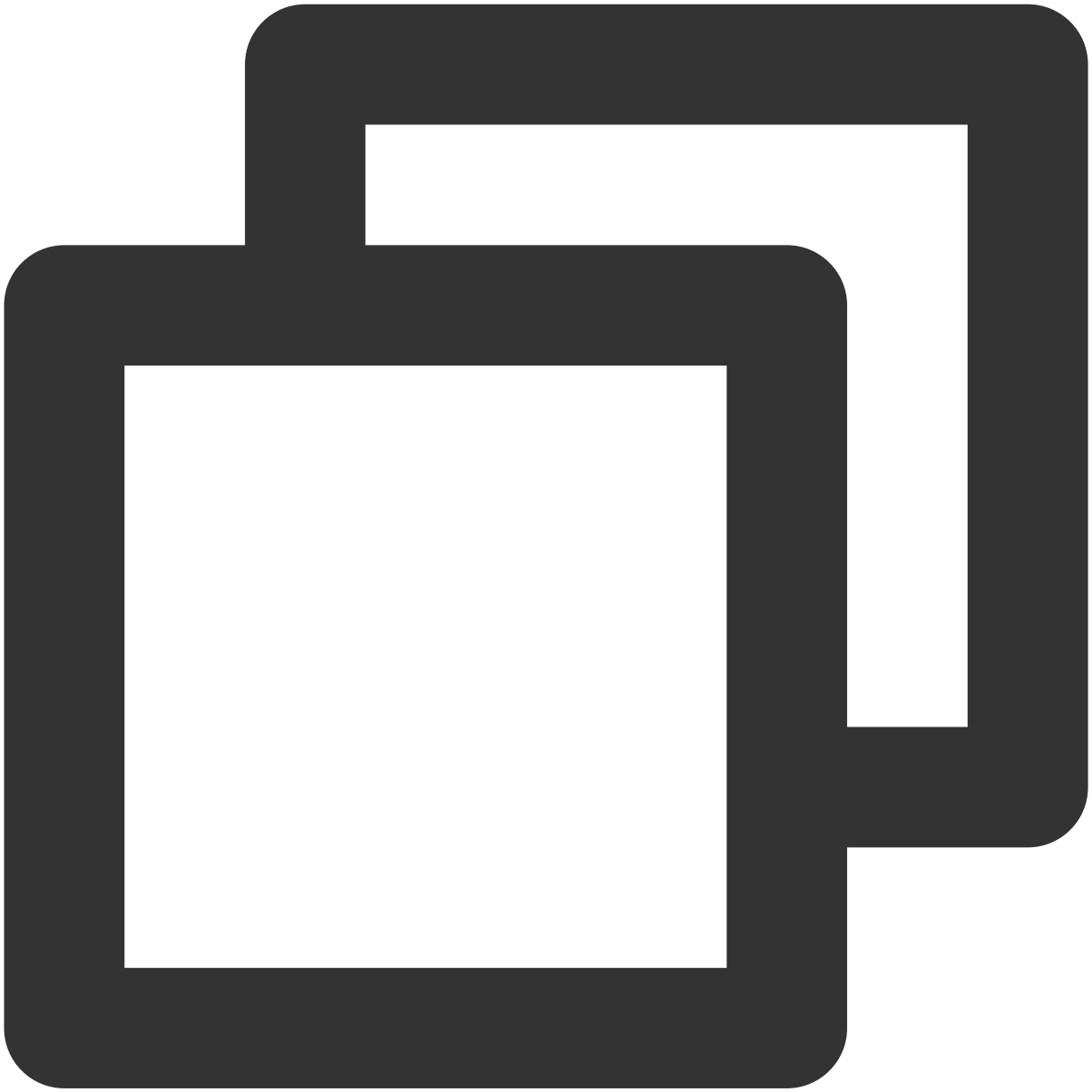
Viewing State

See Viewing State in Quick Start.

©2013-2022 Tencent Cloud. All rights reserved.

Viewing the Result

1. See Viewing the Result in Quick Start.
2. The execution result of `2_RemoteCodePkg.py` is as follows:



```
Remote Code Package : 6765
```

©2013-2022 Tencent Cloud. All rights reserved.

Mapping Remote Storage

Last updated：2024-01-13 11:19:29

Operation Scenarios

Remote mapping is used as an auxiliary feature by BatchCompute for storage services to map remote storage, such as COS and CFS, to a local folder.

Prerequisites

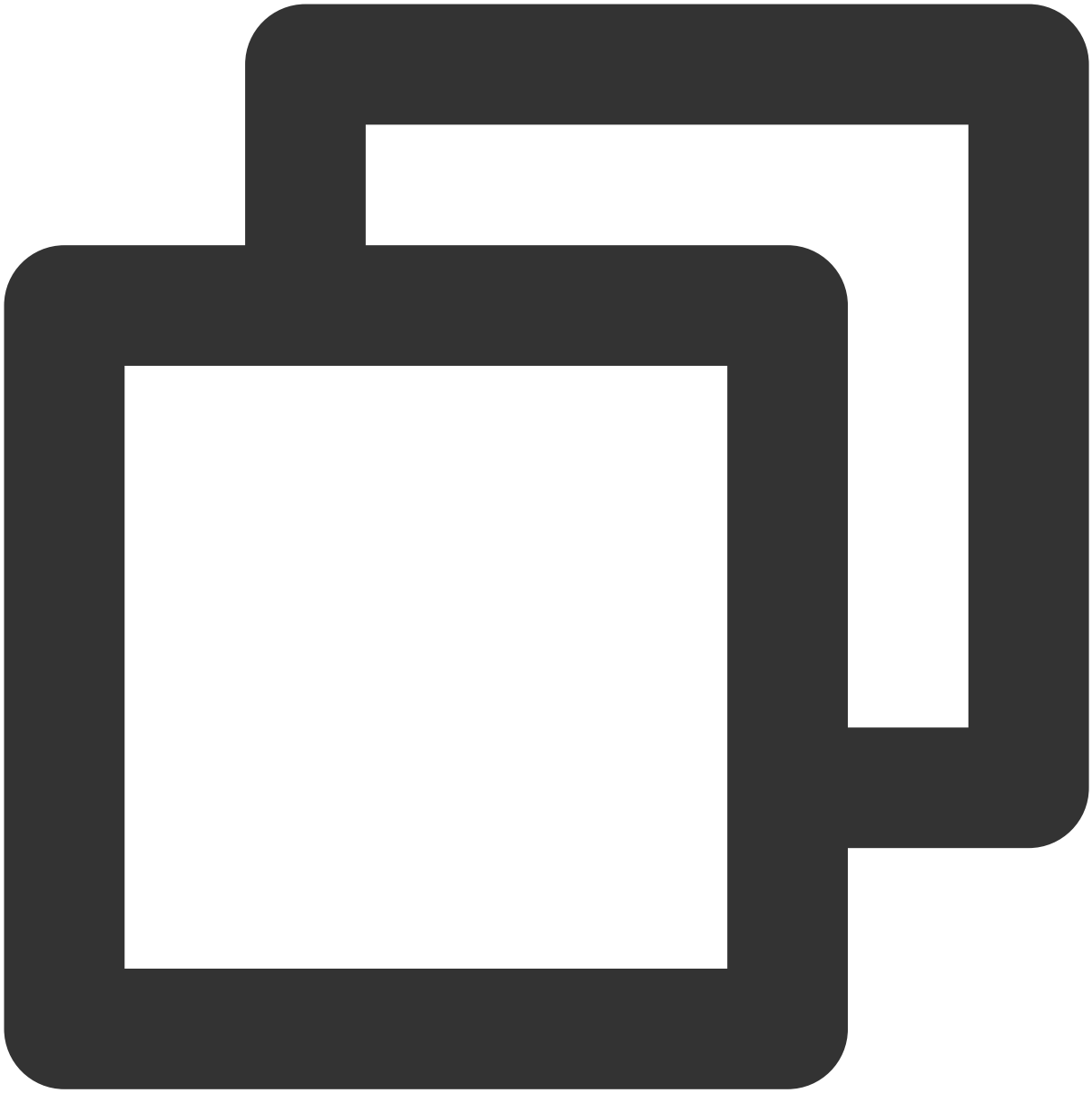
Complete the preparations as instructed in Preparations and learn how to configure the general part of the custom information.

Directions

Uploading input data file

1. Create `number.txt` with the following content:

©2013-2022 Tencent Cloud. All rights reserved.

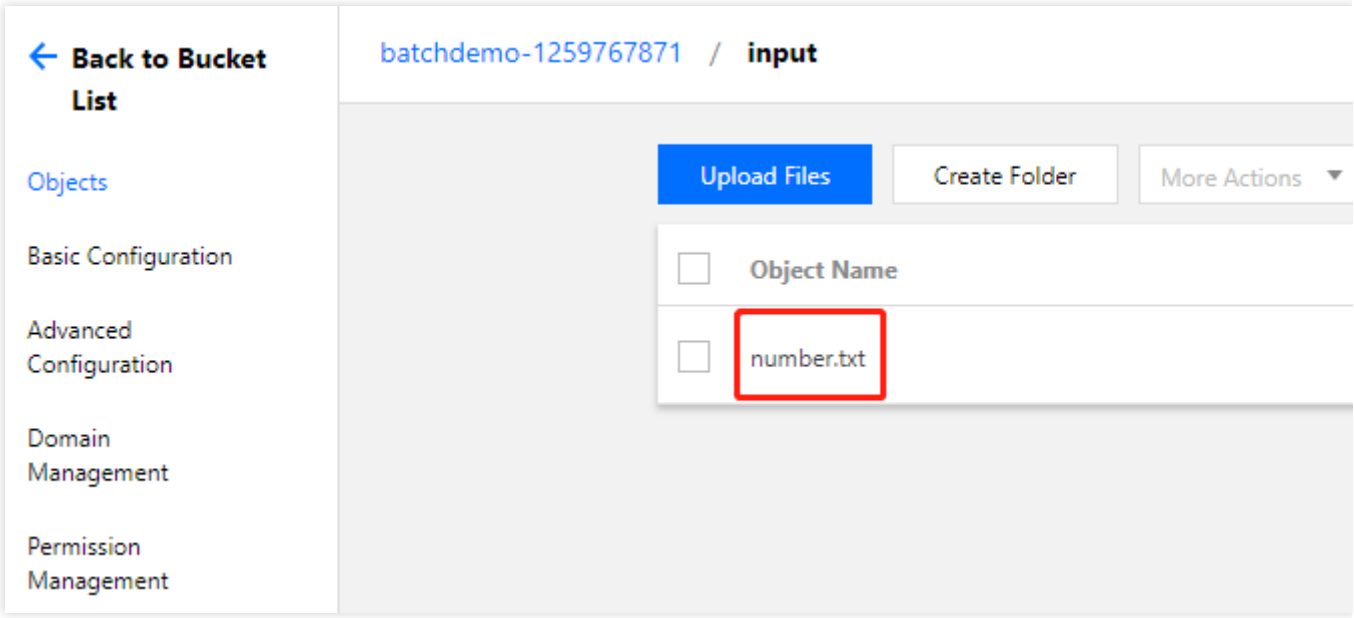


```
1
2
3
4
5
6
7
8
9
```

2. Log in to the COS Console and click **Bucket List** on the left sidebar.

©2013-2022 Tencent Cloud. All rights reserved.

3. Select the ID of the created bucket, click **Objects**, select **input**, and upload `number.txt` as shown below:



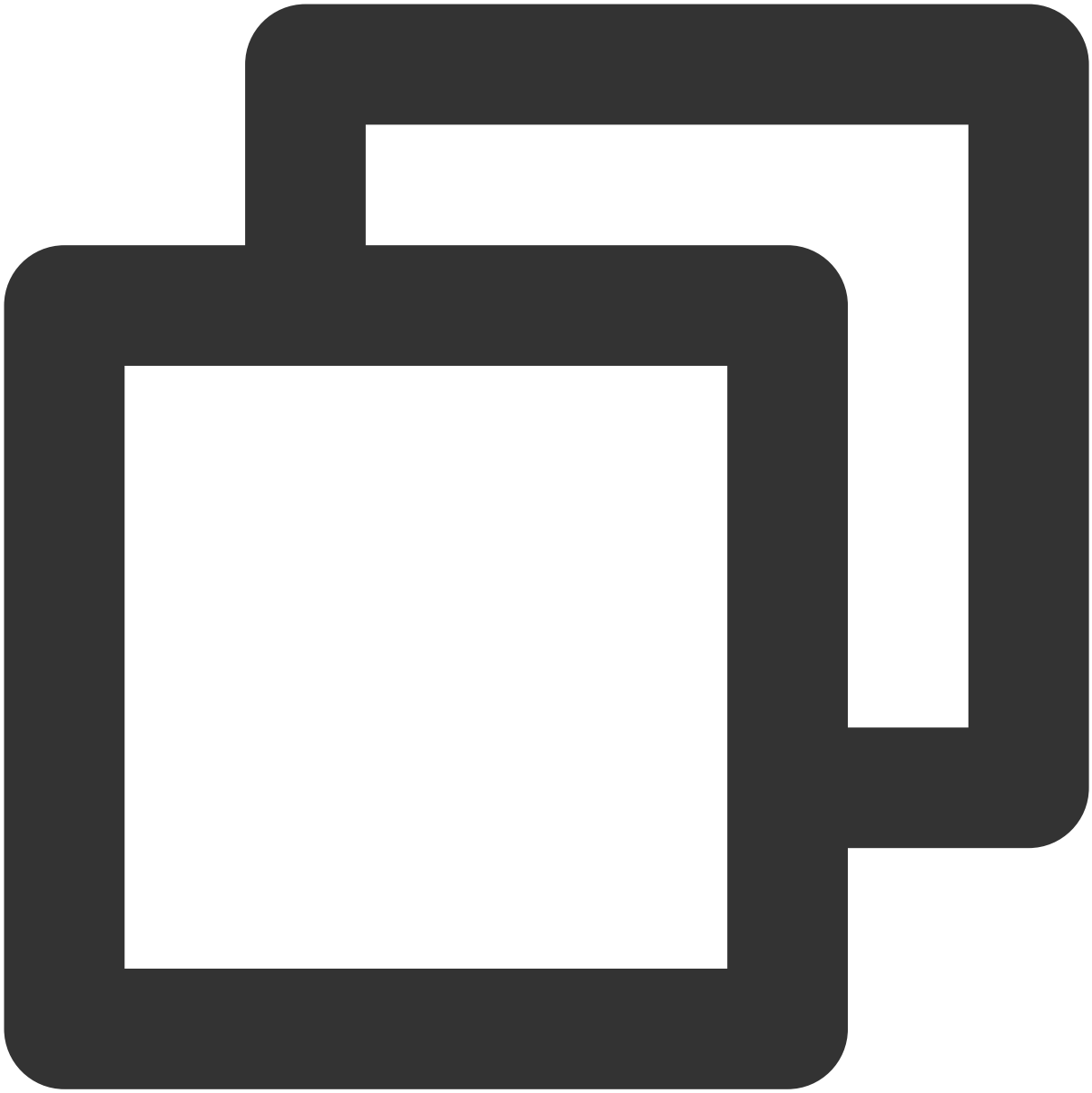
Viewing and modifying demo

Note:

Modify the general part of the custom information in `3_StoreMapping.py` as instructed in Preparations.

Open `3_StoreMapping.py` in an editor.

©2013-2022 Tencent Cloud. All rights reserved.



```
# custom (Change to your info)
imageId = "img-m4q71qnf"
Application = {
    "DeliveryForm": "PACKAGE",
    "Command": "python ./codepkg/sumnum.py",
    "PackagePath": "http://batchdemo-xxxxxxxxx.cos.ap-guangzhou.myqcloud.com/codepk
}
StdoutRedirectPath = "your cos path"
StderrRedirectPath = "your cos path"
InputMapping = {
    "SourcePath": "cos://batchdemo-xxxxxxxxx.cos.ap-guangzhou.myqcloud.com/input/",
```

©2013-2022 Tencent Cloud. All rights reserved.

```
    "DestinationPath": "/data/input/"
}
OutputMapping = {
    "SourcePath": "/data/output/",
    "DestinationPath": "your output remote path"
}
```

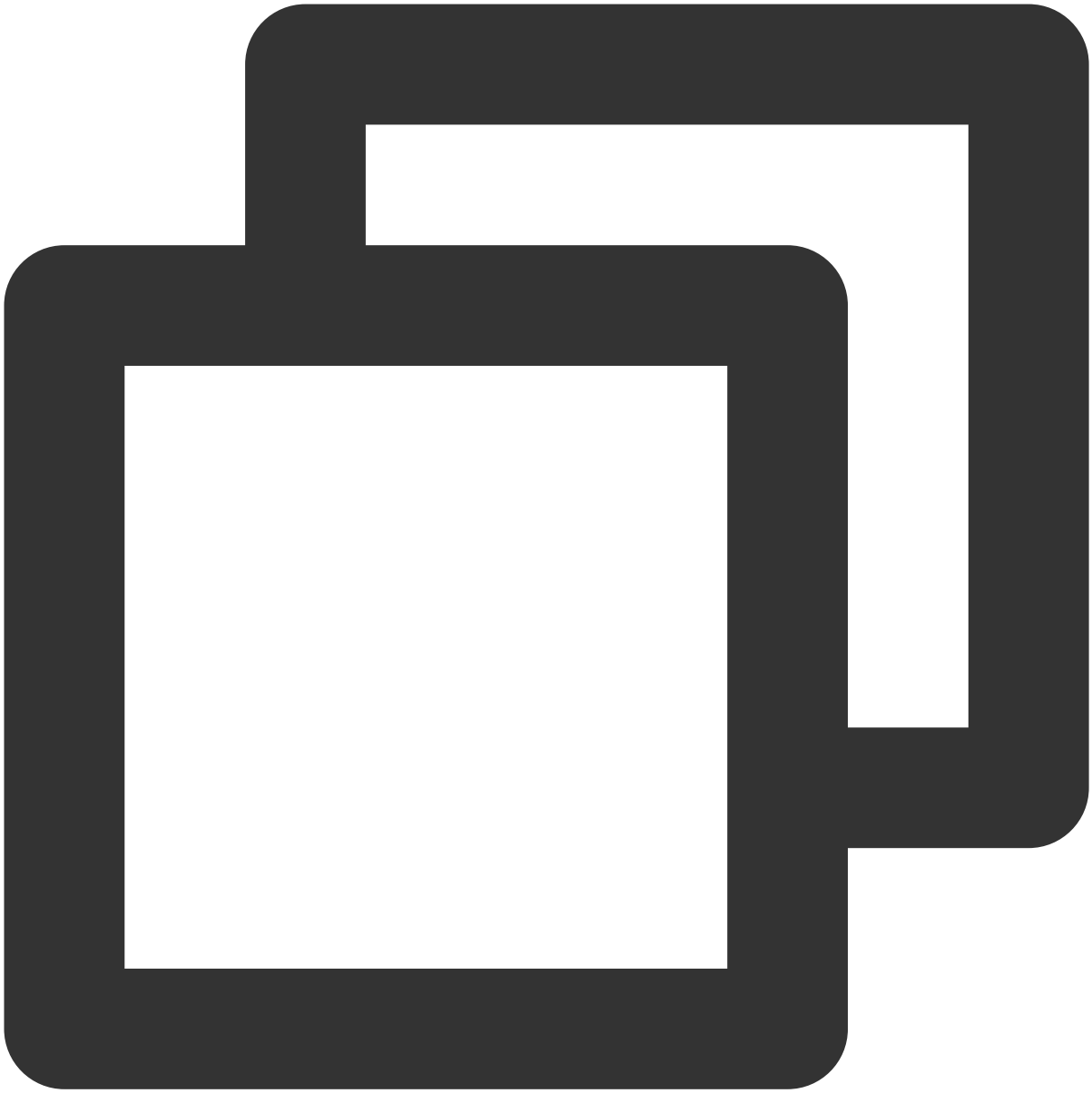
Compared to `2_RemoteCodePkg.py`, some of the parameters in the custom part are modified as follows:

Configuration Item	Description
Application	Modify `Command` to `sumnum.py`.
InputMapping	Input mapping. `SourcePath` remote storage address: modify this address to the path of the **input** folder in "Preparations". For more information, please see Getting COS Endpoints. `DestinationPath` local directory: retain the original setting for the time being.
OutputMapping	Output mapping. `SourcePath` local directory: retain the original setting for the moment. `DestinationPath` remote storage address: modify this address to the path of the **output** folder in "Preparations". For more information, please see Getting COS Endpoints.

sumnum.py is composed as below:

Open `input/number.txt`, add up all numbers in each row, and write the result into `output/result.txt`.

©2013-2022 Tencent Cloud. All rights reserved.



```
import os

inputfile = "/data/input/number.txt"
outputfile = "/data/output/result.txt"

def readFile(filename):
    total = 0
    fopen = open(filename, 'r')
    for eachLine in fopen:
        total += int(eachLine)
    fopen.close()
```

©2013-2022 Tencent Cloud. All rights reserved.

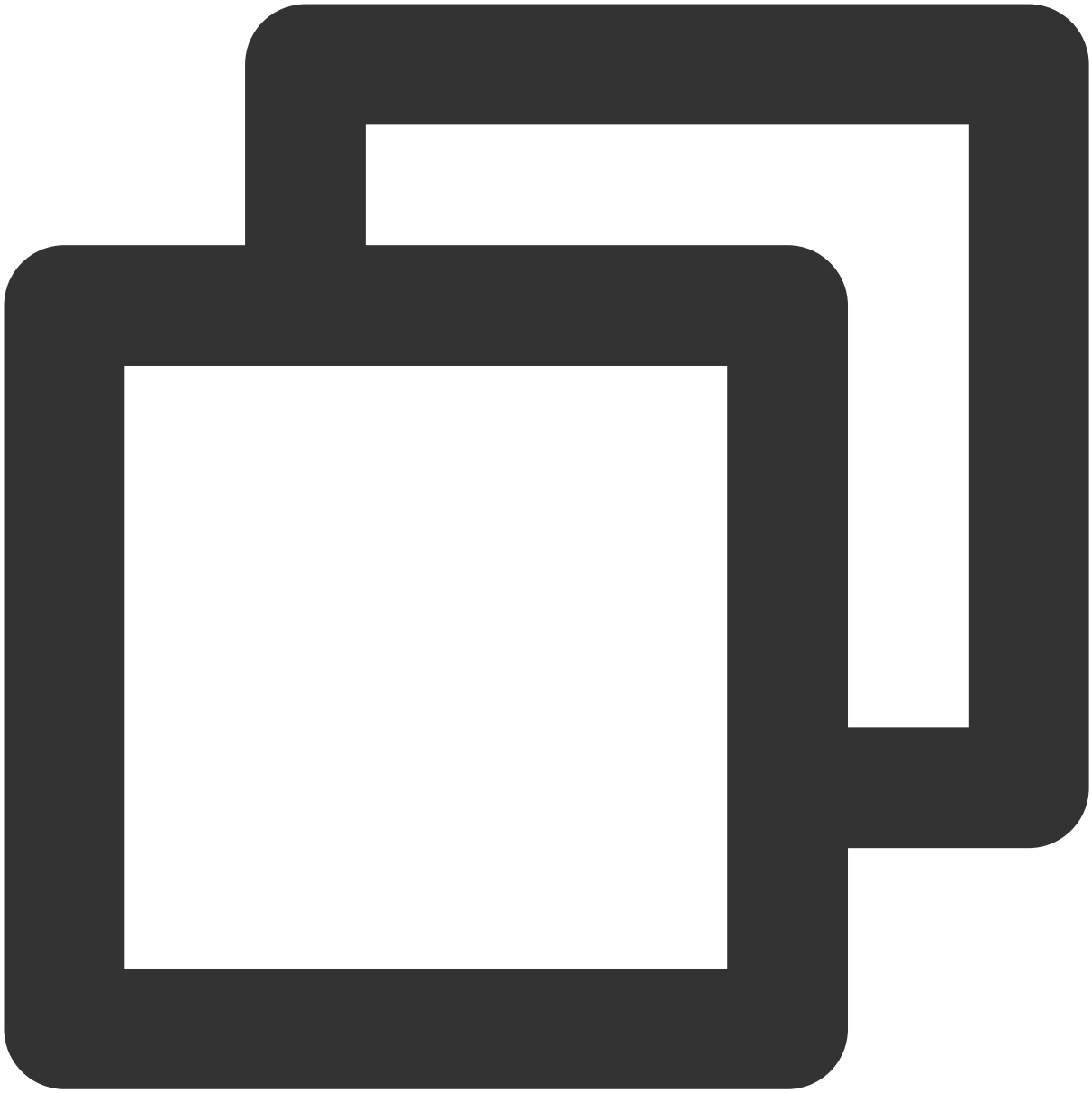
```
    print "total = ",total
    fwrite = open(outputfile, 'w')
    fwrite.write(str(total))
    fwrite.close()

print("Local input file is ",inputfile)
readFile(inputfile)
```

Submitting job

Run the following command to run the Python script.

The demo encapsulates the job submission process by using Python scripts and the BatchCompute command line tool.

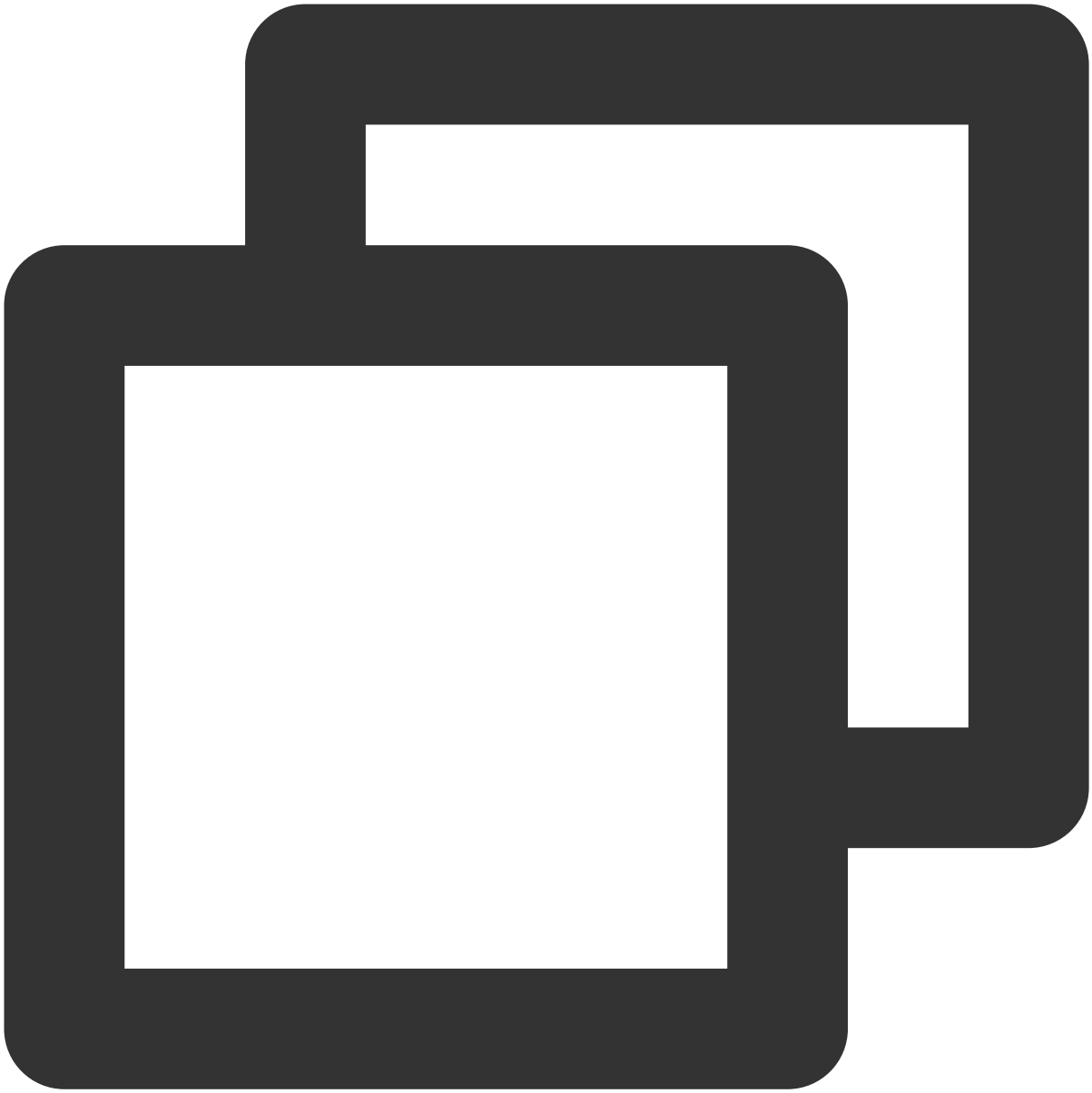
©2013-2022 Tencent Cloud. All rights reserved.



```
python 3_StoreMapping.py
```

The returned result is as follows, indicating that the job is successfully submitted:

©2013-2022 Tencent Cloud. All rights reserved.



```
{
    "RequestId": "8eaeb01e-94a6-41a1-b40f-95f15417c0b4",
    "JobId": "job-97smiptb"
}
```

If the submission operation fails, check the returned value or contact us for assistance.

Viewing status

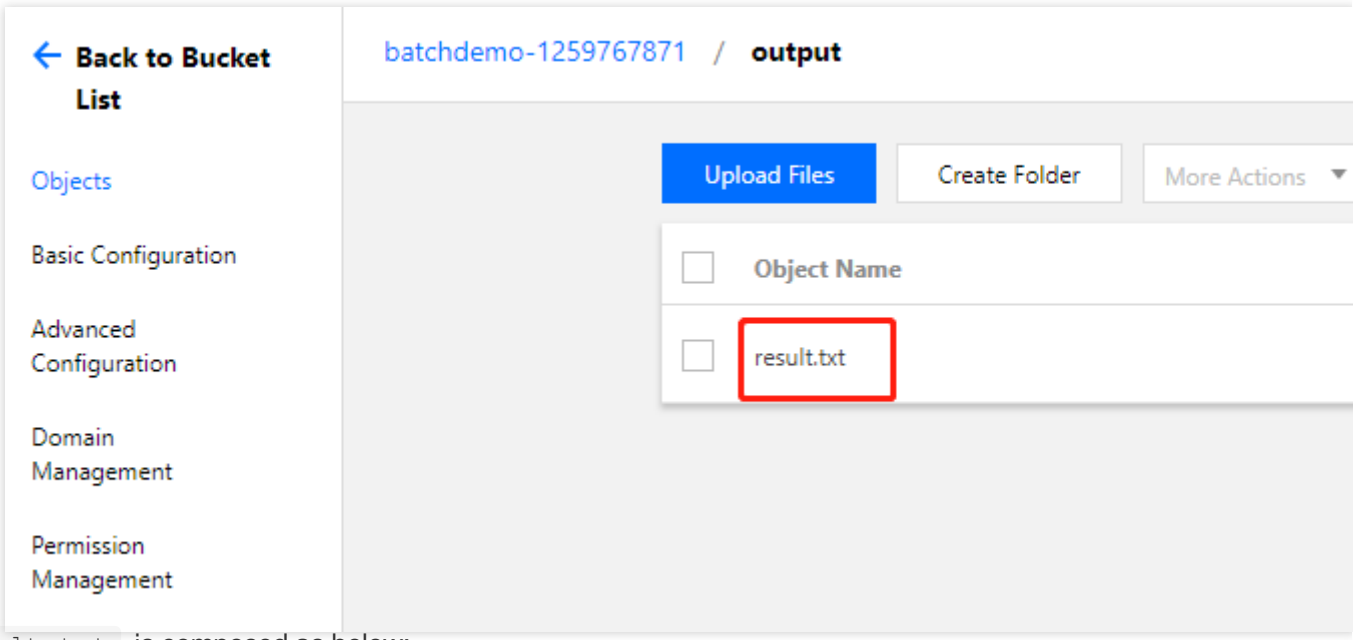
Please see Viewing Status in Quick Start.

©2013-2022 Tencent Cloud. All rights reserved.

Viewing result

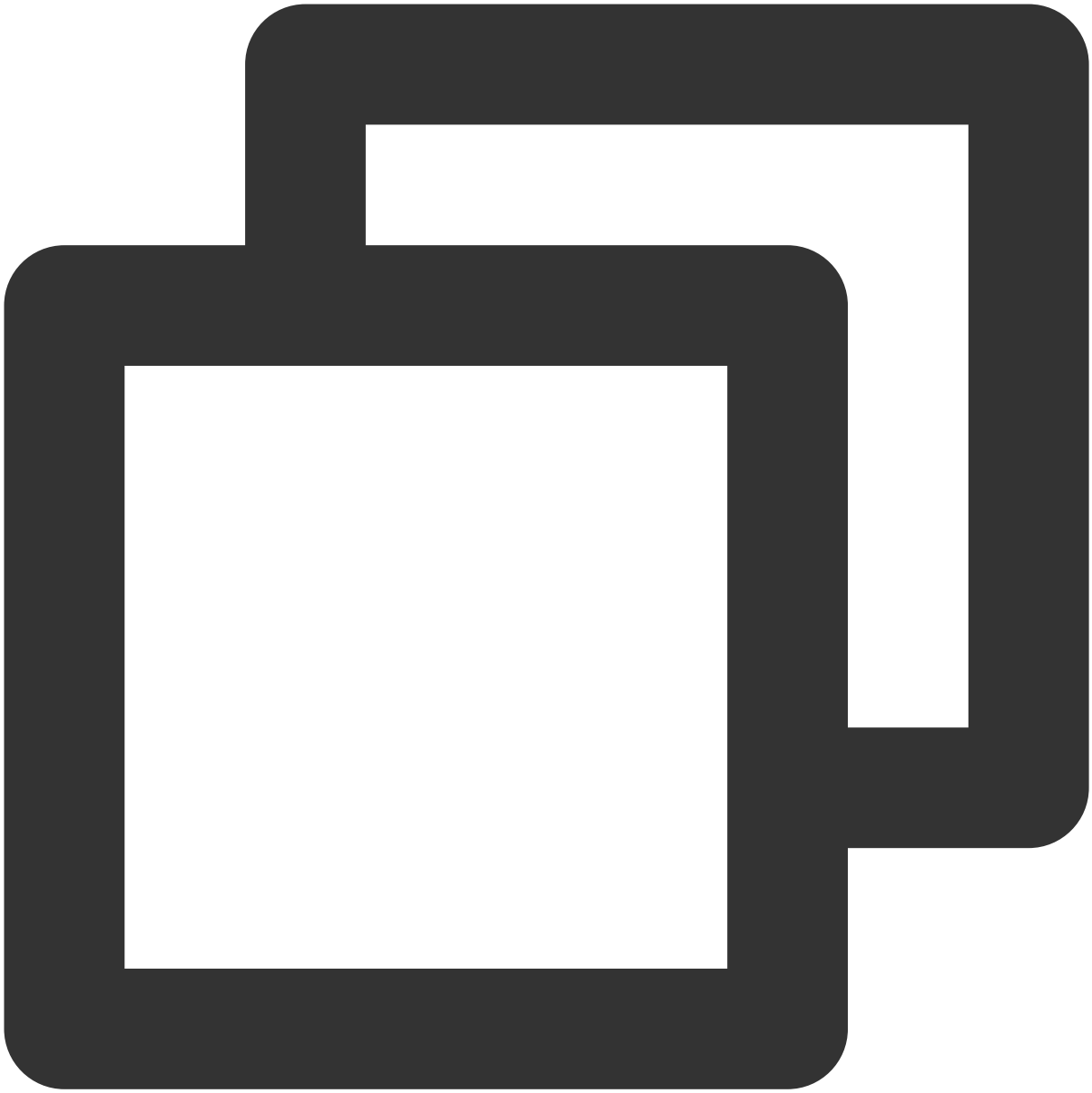
1. Log in to the COS Console and click **Bucket List** on the left sidebar.
2. Select the ID of the created bucket, click **Files**, and select **output** as shown below:

BatchCompute copies the output data from the local directory to the remote directory. The execution results of `3_StoreMapping.py` are stored in `result.txt`, which is automatically synced to COS.



`result.txt` is composed as below:

©2013-2022 Tencent Cloud. All rights reserved.



```
45
```

©2013-2022 Tencent Cloud. All rights reserved.