

Spring 2018 CS304 Software Engineering

Function as a Service

Ziqiang Li Yulian Mao Xizi Ni Yilin Zheng Chenyu Zhou
{11510352, 11510086, 11510602, 11510506, 11510374}@mail.sustc.edu.cn
Stone Tencent
Yuqun Zhang zhangyq@sustc.edu.cn

June 15, 2018

1 Abstract

Serverless computing is recently raised hot fields of cloud computing. Serverless computing can hide the implementation of the back end from developers or users by using API gateway. The FaaS is a widely used model of serverless computing. In this project, we focus on developing serverless cloud functions(SCF) based on Tencent Cloud to provide a service for massive data, such as image processing. This service will finally be developed into to-business APIs with comprehensive documents.

2 Review

This part will roughly review some concepts to help better summarize our project. The concepts related to FaaS includes *serverless computing, function as a service, API Gateway*.

2.1 Serverless Computing

Serverless computing is a cloud computing model in which the cloud provider dynamically manages the allocation of machine resources. The pricing is based on the actual amount of resources consumed by an application or requests, instead of the units of capacity. Serverless computing does not mean no servers, it still requires servers. The name "serverless computing" is used because the

server management and capacity planning decisions are completely hidden from the developers or operators. Serverless code can be used in conjunction with code deployed in traditional styles, such as microservices. Alternatively, applications can be written to be purely serverless and use no provisioned servers at all[1].

2.2 Function as a Service

Function as a service (FaaS) is a category of cloud computing services that provides a platform allowing customers to develop, run, and manage application functionalities without the complexity of building and maintaining the infrastructure typically associated with developing and launching an app. Building an application following this model is one way of achieving a "serverless" architecture, and is typically used when building microservices applications[2].

The advantages of FaaS:

- Lower cost: save infrastructure costs, personnel cost, and development costs
- Strong expandability
- Simpler management
- High resource utilization

2.3 API Gateway

The Figures 2.3 shows the simple structure of API Gateway.

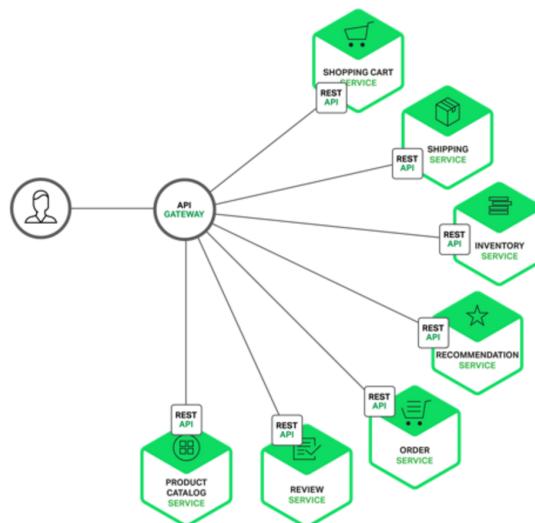


Figure 1: API Gateway

3 Features and Requirements

Our project is based on the problem that traditional service cannot handle a sharp increment of computing caused by large concurrent requests. It means, some situations have unstable computing requirements according to time or something others. This project is developed in order to deal with the unstable user demand. The providers can not expend their server unlimited. Therefore, the FaaS is suitable for their scenario. Based on this background, the stakeholder required us to implement some figure processing cloud function such as:

- Zoom out
- Rotation
- Compression
- Watermark (words/figures)
- Round Corner
- Overlap QR Codes
- Index Cutting
- Format Conversion

Besides, those functions are highly customized. Users can adjust many parameters to meet their own demand. These services are based on Tencent Cloud which is a strong back end.

The product is now planned to shift to face business customers and provides stable APIs in future.

4 Design and Architecture

The architecture can be referred as 2.

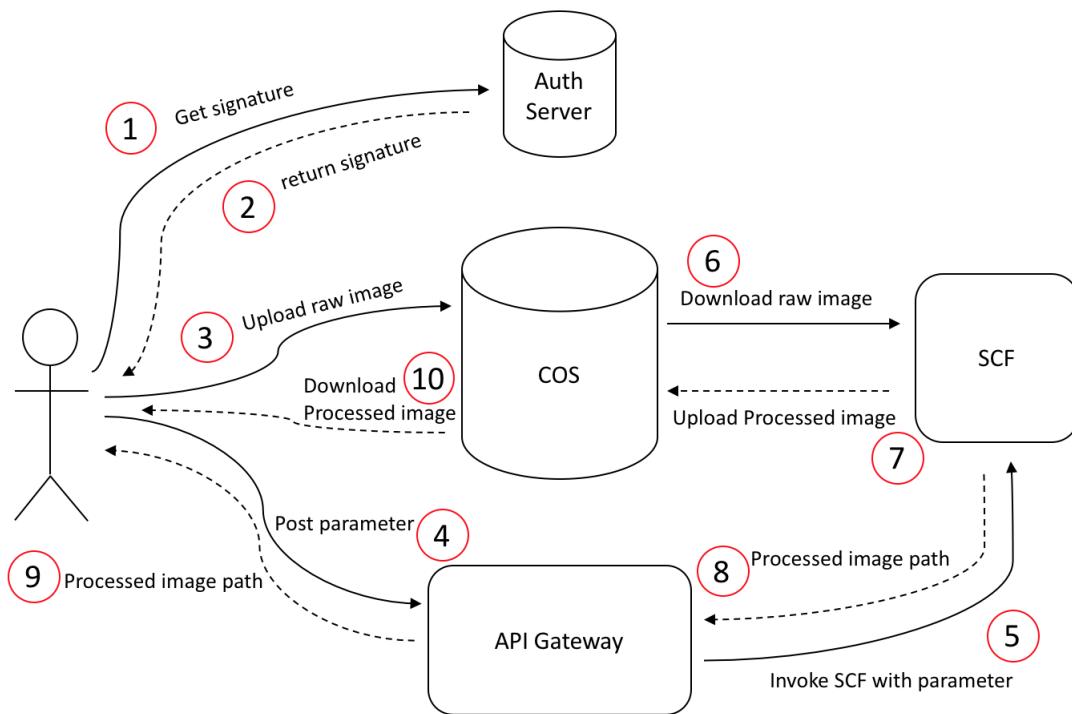


Figure 2: Architecture

We use Tencent Cloud SCF service to develop our cloud function, use their COS to store figures and use API Gateway to custom our APIs to release to users.

The workflow includes:

1. Get the authorized secret key to upload the image

For safety reasons, the user cannot get the authorization for uploading the image as the administrator. Users' image will be checked and users will be returned an authorized signature for uploading an image to COS.

2. Upload image with the signature

User upload image with the signature.

3. Post parameter to image processing

User post the image processing query to the API, which includes the parameters. A processed image path will be returned.

4. Download processed image

The user can preview or download the processed image from the previous path.

5 Implementation

5.1 Team assignment

- Front End: Yulian Mao, Chenyu Zhou
- Back End: Ziqiang Li, Xizi Ni, Yilin Zheng
- Documentation: Yilin Zheng, Ziqiang Li
- Reports: Yilin Zheng
- PPTs: Xizi Ni, Yilin Zheng

5.2 Platform choices

The platform we choices includes Tencent Cloud Servers, API Gateway, and COS Object storage. The system is Linux.

6 User Interface

This is a back end hosted service which normally should have no front end but we just established front end pages for demo and the captures can be seen in Appendix A.

We have developed a comprehensive documents on *Github pages* and *Readthedocs* with *Sphinx*. The full document can be access from:

- Github pages: faasdev.github.io
- Readthedocs: <https://faasdevgithubio.readthedocs.io>

7 Conclusion

This project is a serverless function development based on Tencent Cloud. This service provides basic image processing, and can be further added more SCFs if necessary. This service is serverless so the users have not burden of managing and maintaining servers. Besides, the pricing is counted by the times of requests but not whole servers which largely saves their cost. We have planned to further develop this service to APIs for business.

A Front End



Figure 3: Welcome page

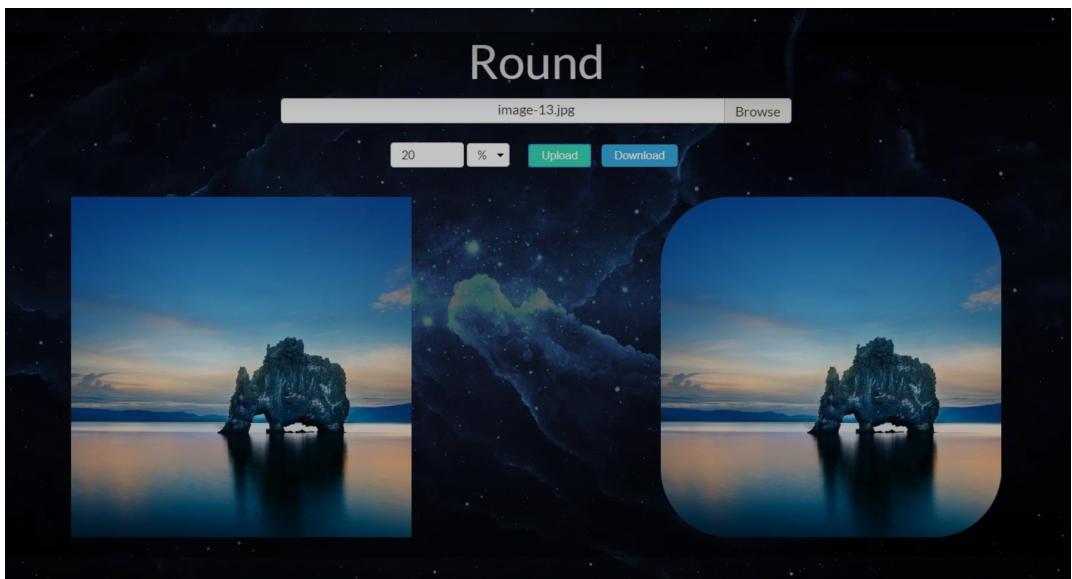


Figure 4: Round corner

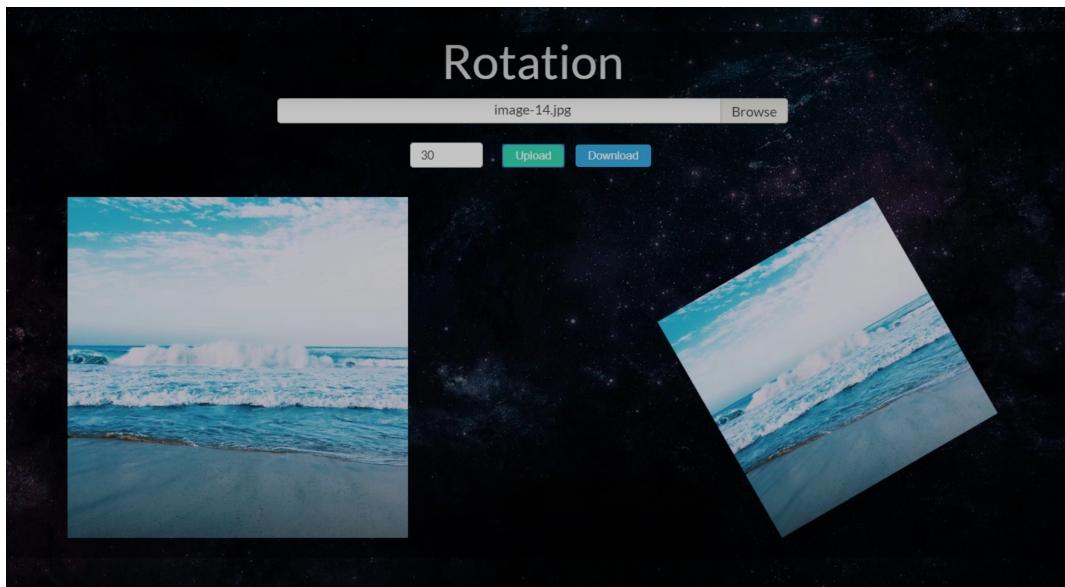


Figure 5: Rotation

B Back End

B.1 SCF

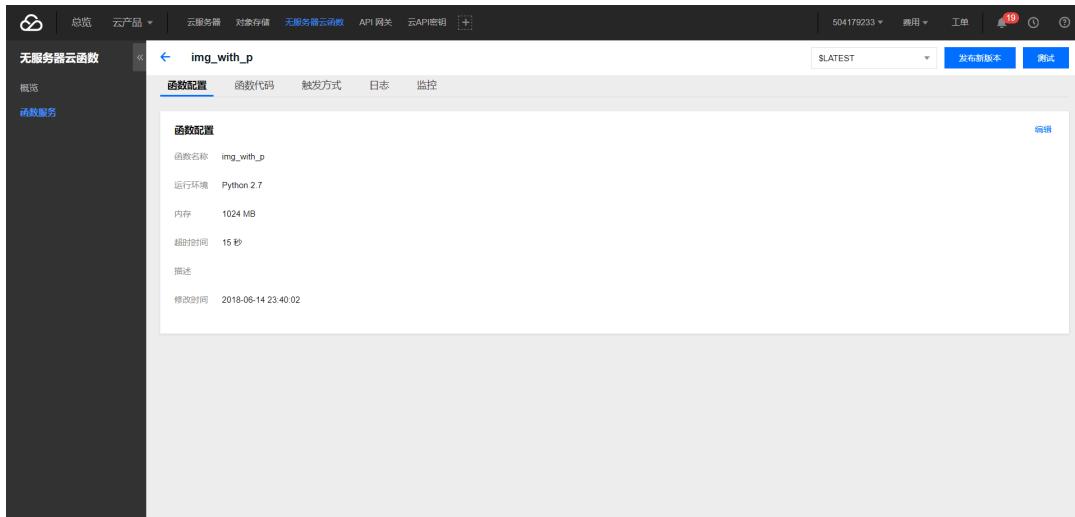


Figure 6: SCF setting

The screenshot shows the QCloud SCF code editor interface. At the top, there are tabs for '函数配置' (Function Configuration), '函数代码' (Function Code), '触发方式' (Trigger Type), '日志' (Logs), and '监控' (Monitoring). The '函数代码' tab is selected. Below it, there are three input type options: '在线编辑' (Online Editor) (selected), '本地上传zip包' (Upload local zip package), and '通过COS上传zip包' (Upload zip package via COS). The '执行方法' (Execution method) field contains 'req_main_handler'. The code editor displays the following Python code:

```
1 # -*- coding: utf8 -*-
2 import uuid
3 import json
4 import os
5 import logging
6 import re
7 import commands
8 import datetime
9 import credential
10 import math
11 from image_processing import *
12 from qcloud_cos import CosClient
13 from qcloud_cos import DownloadFileRequest
14 from qcloud_cos import UploadFileRequest
15 from PIL import Image, ImageDraw, ImageFont
16
17
18 def round_image(input_path, output_path, rad, fixed):
19     #op == 0
20     print("rad",rad,"fix",fixed)
21     image = round_corner(input_path, rad, fixed)
```

Figure 7: SCF code

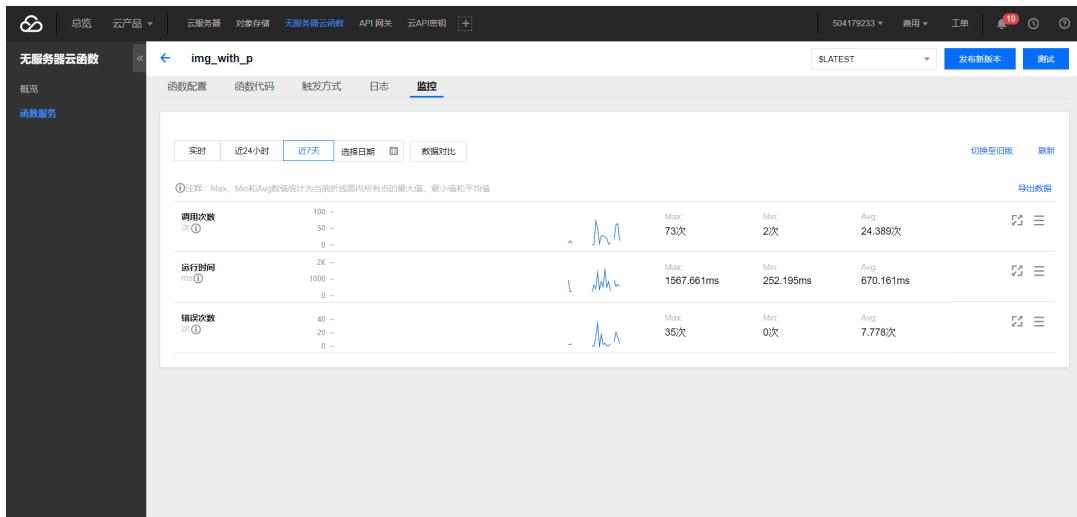


Figure 8: SCF monitoring

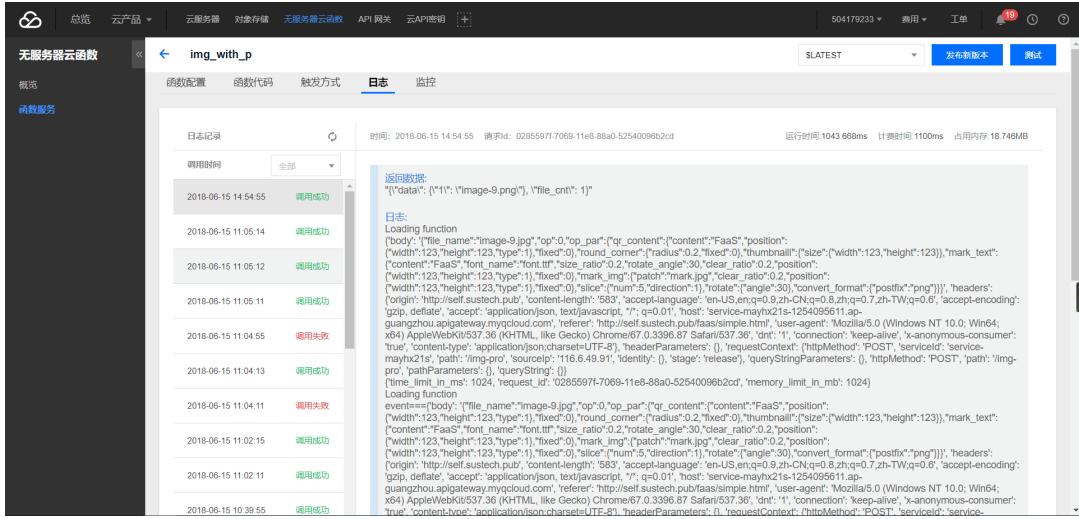


Figure 9: SCF log

B.2 API Gateway

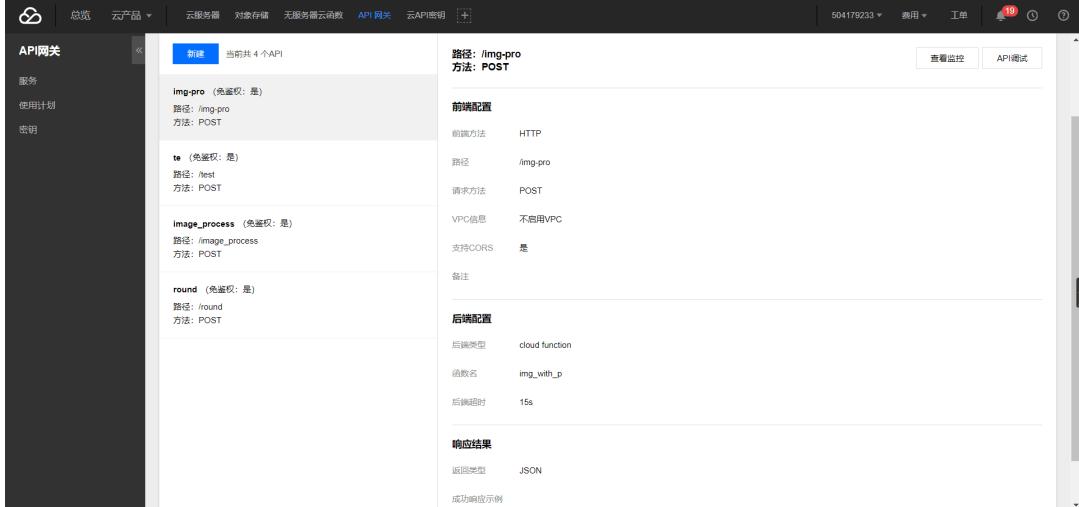


Figure 10: API Gateway

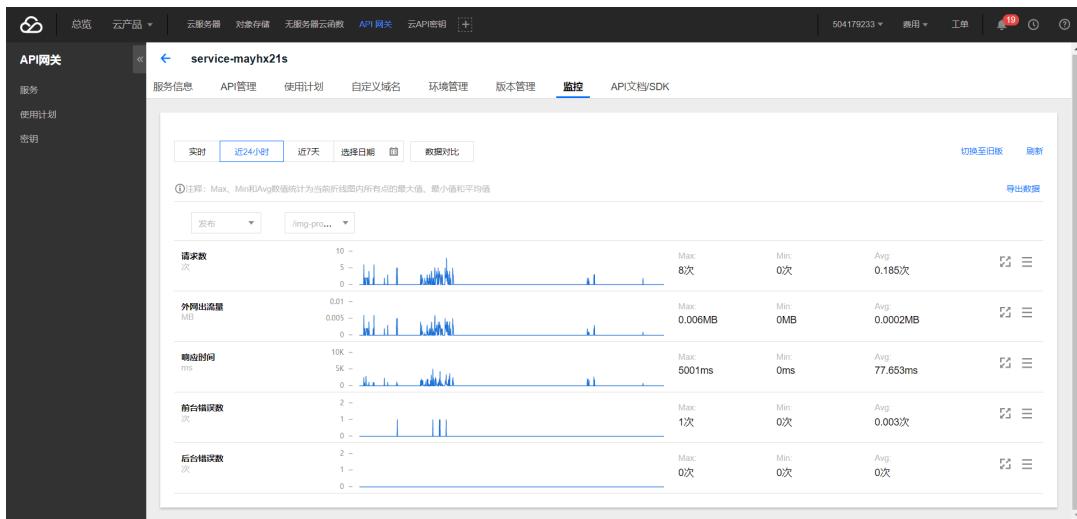


Figure 11: API Gateway monitoring

B.3 COS

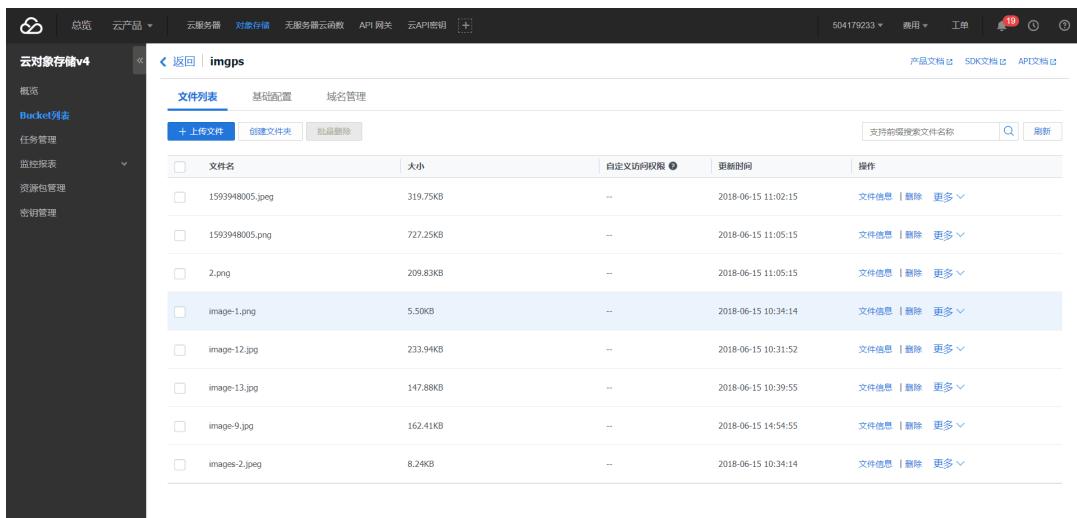


Figure 12: COS

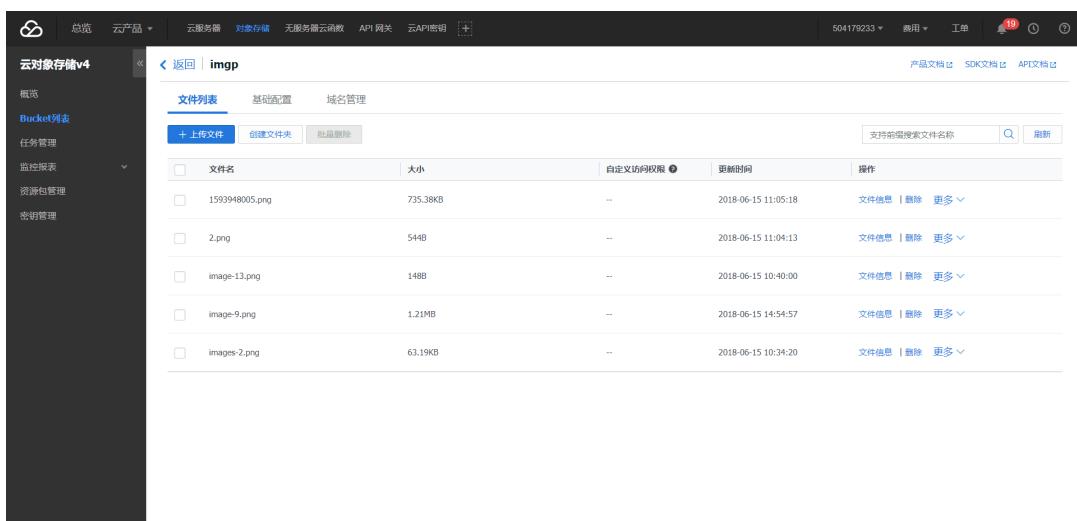


Figure 13: COS

C Documents

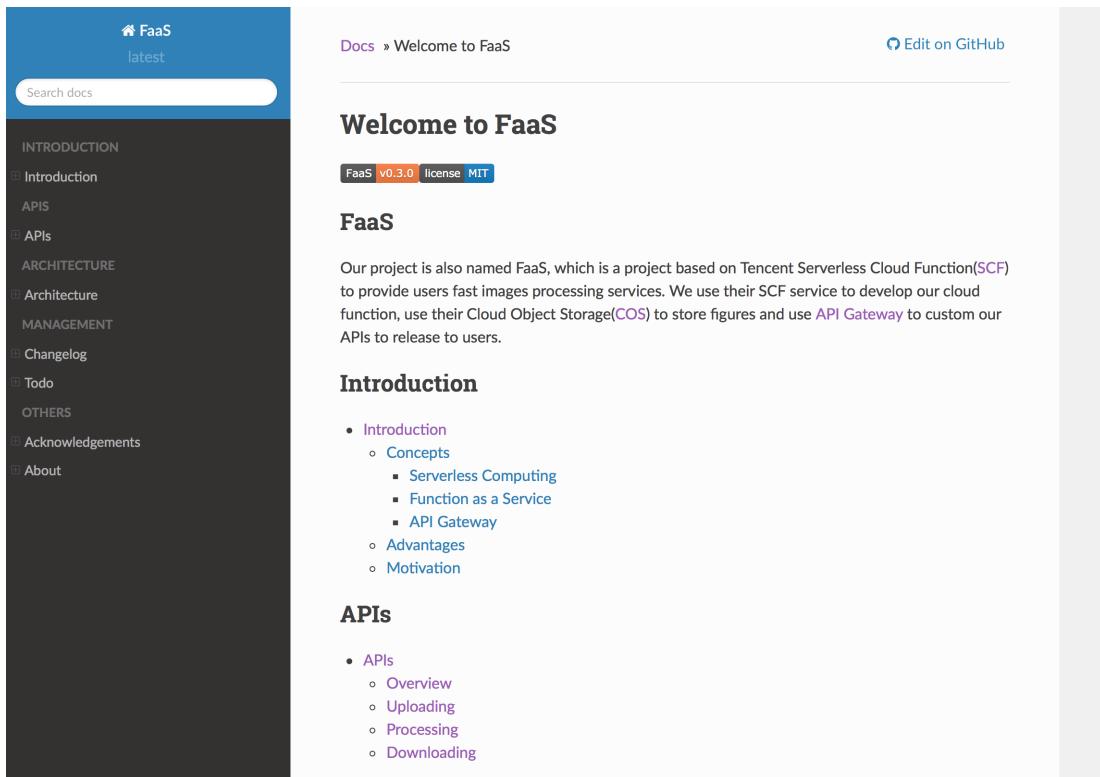


Figure 14: Document

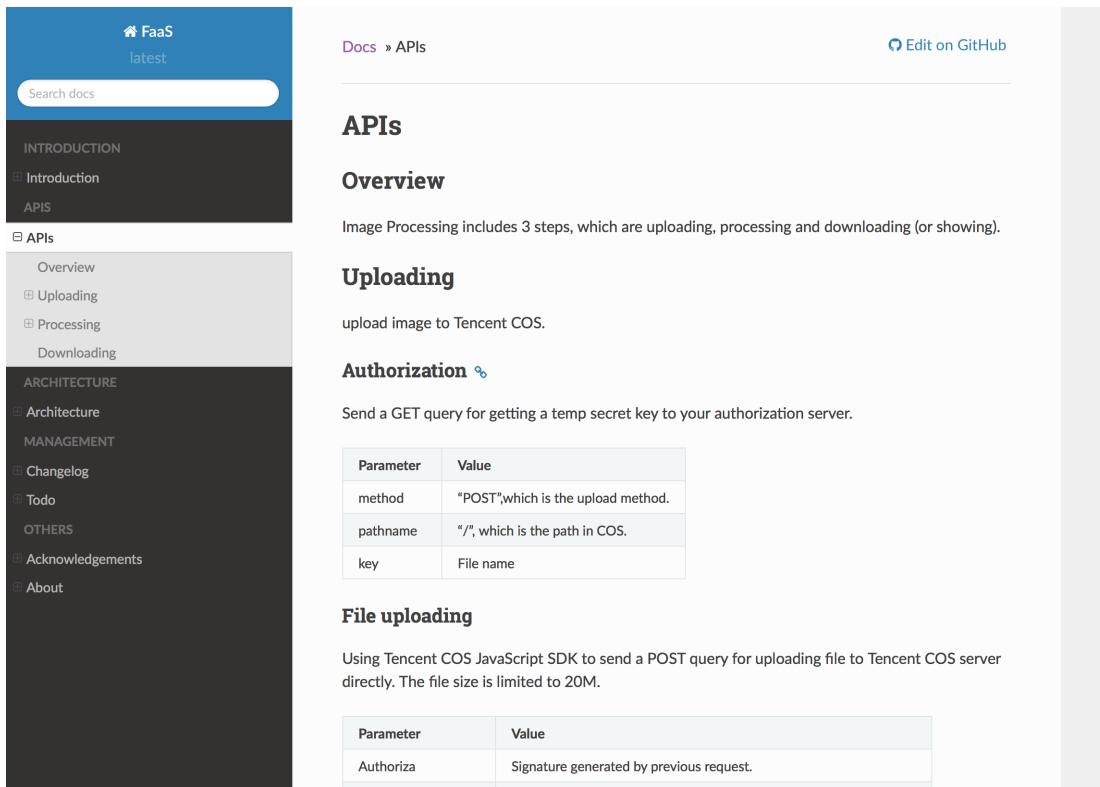


Figure 15: Document

References

- [1] https://en.wikipedia.org/wiki/Serverless_computing
- [2] https://en.wikipedia.org/wiki/Function_as_a_service
- [3] <https://mesosphere.com/blog/iaas-vs-caas-vs-paas-vs-faas/>