

Serverless Cloud Function

Getting Started

Product Documentation



Copyright Notice

©2013-2019 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

Getting Started

- Getting Started

- Creating functions using the console

- Creating Functions on CLI

Getting Started

Getting Started

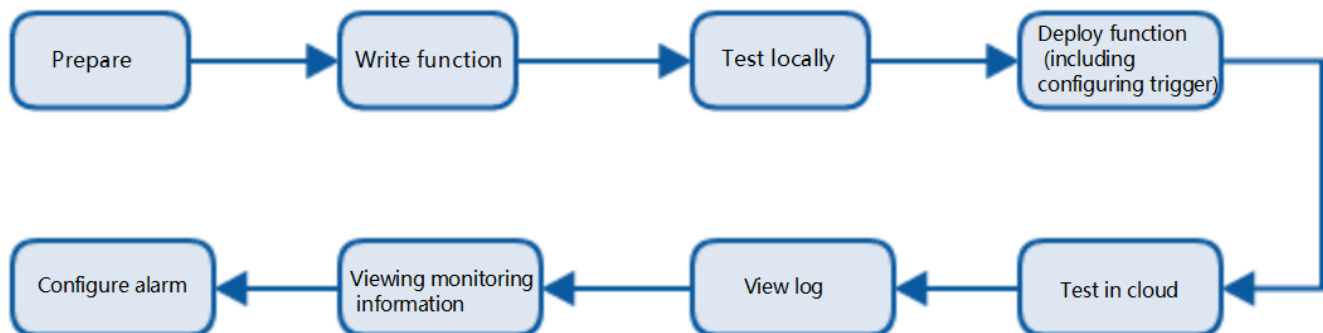
Last updated : 2020-05-11 11:33:11

Tencent Cloud Serverless Cloud Function (SCF) is a serverless execution environment that enables you to build and run applications without having to purchase and manage servers. Simply code in a supported language and set the execution conditions, and your code can be run on the Tencent Cloud infrastructure elastically and securely.

This document describes the usage process of SCF and provides multiple examples for function creation, deployment, and testing as well as service building through the SCF Console, SCF CLI, and SCF VS Code plugin.

Directions

The following is the flow chart and basic steps of using SCF:



1. **Prepare:** [sign up for a Tencent Cloud Account](#), activate the SCF service, configure a basic development environment, etc.
2. **Write function:** a function is the basic unit of scheduling and operation, and you must follow the function API specifications when writing a function.
3. **Test locally:** you can debug the code locally and then deploy it in the cloud.
4. **Deploy function (including configuring trigger):** after the code is deployed in the cloud, SCF can execute the function after a triggering condition are configured. The execution condition of a function is called a trigger. You can configure various types of triggers, such as timer, API Gateway, and COS triggers.

5. **Test in cloud:** after the function is deployed in the cloud, you can test it through the configured trigger.
6. **View log:** SCF supports displaying historical or real-time function logs in various ways.
7. **View monitoring information:** you can stay on top of the function running status by viewing monitoring metrics.
8. **Configure alarm:** alarming is critical for online production business. After an alarm is configured, you can receive alarm messages promptly when an exceptional situation occurs in the business.

Getting Started

The getting started document can guide you through the complete process of using SCF. You can further familiarize yourself with SCF and developer tools through the following examples:

- [Creating Functions in Console](#)
- [Creating Functions on Serverless Framework CLI](#) (recommended)

Developer Tools

The Tencent Cloud Serverless team provides a wide variety of developer tools to help you use SCF:

- [SCF Console](#)
- [Serverless Framework CLI](#)

You can also see [How SCF Works](#), [Development Guide](#), and [Best Practice](#) to learn more about how to use SCF to build production systems.

Creating functions using the console

Last updated : 2020-09-10 16:20:16

Overview

This document describes how to develop a simple `Hello World` web service in the Serverless Cloud Function (SCF) Console.

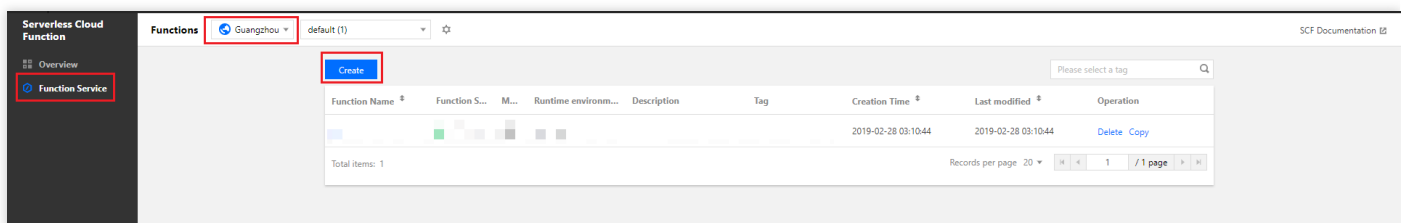
Prerequisites

- You have already registered a Tencent Cloud account. If you haven't done so yet, click [here](#) to sign up.
- You have already logged in to the [SCF Console](#).

Directions

Writing function

1. Click **Function Service** on the left sidebar to enter the "Function Service" page.
2. Select **Guangzhou** at the top of the page and click **Create** as shown below:



3. Enter the basic information of the function on the "Create Function" page and click **Next** as shown below:

Create a Function Create function using C

1 Basic Information > 2 Function configuration

Function name *
1. 2 to 60 characters
2. Starts with a letter and ends with a number or letter; supports a-z, A-Z, 0-9, -, _

Runtime environment

Create Method

- Function Template**
Creating Function Using Sample
Code Template
- Empty function
Create an empty function using
helloworld sample

Filter Separate multiple tags with carriage returns

helloworld [Learn More](#)

Languages Python2.7

Description This is helloworld function

Tag

Deploy 50940 times

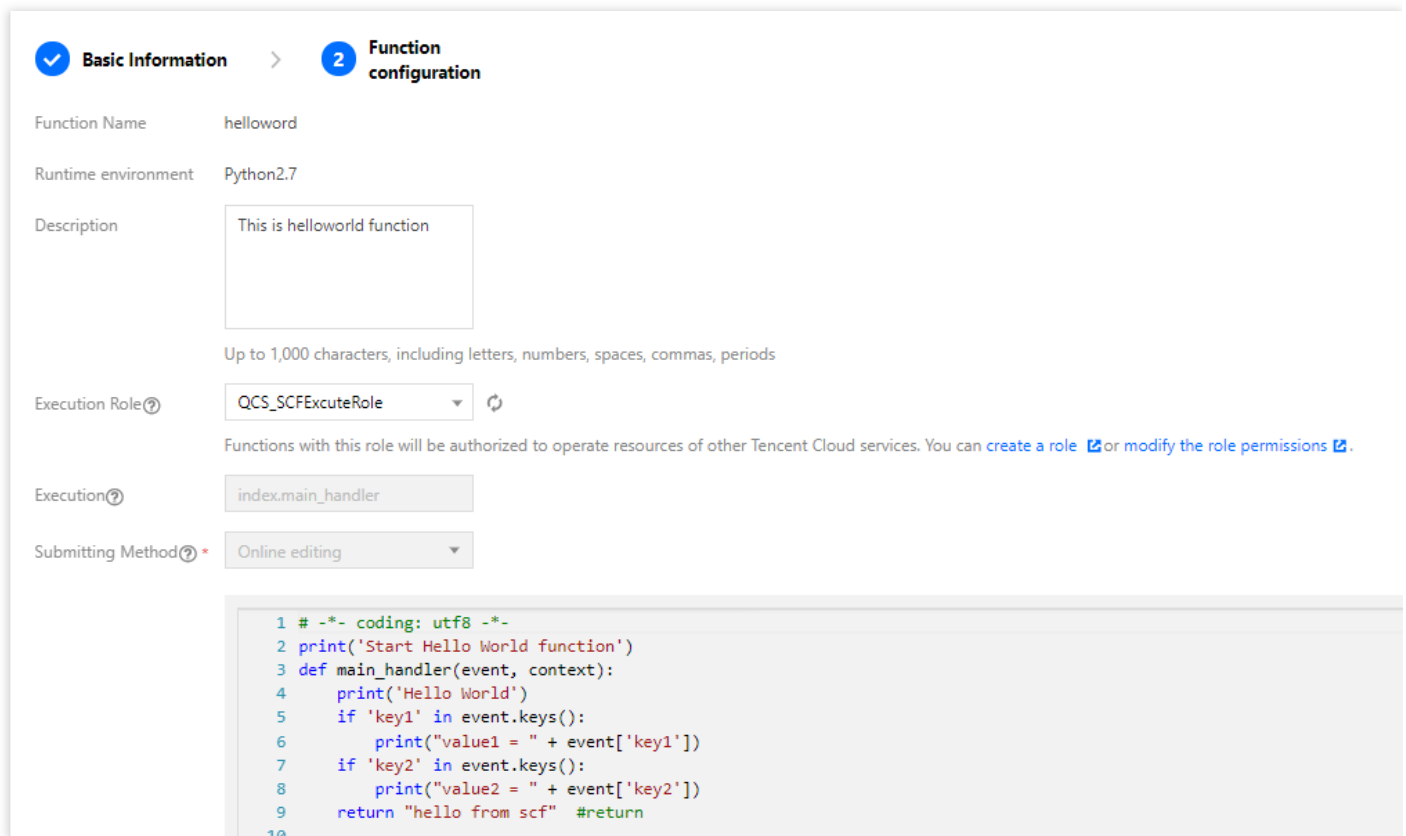
Total items: 1

Records per page 12 ▾ « « 1 / 1 page » »

Next

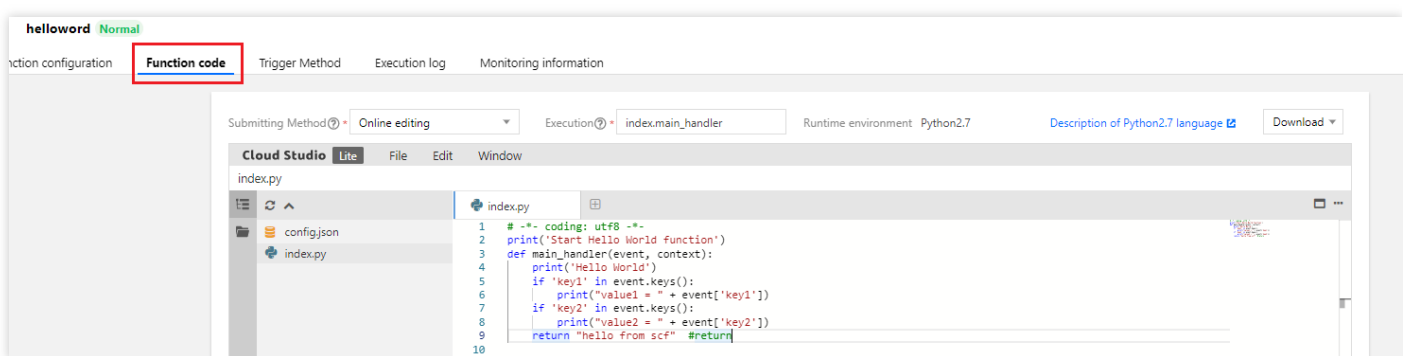
- **Function name:** enter `helloworld`.
 - **Runtime environment:** select "Python 2.7".
 - **Create Method:** select "Function Template".
 - **Filter:** after entering `helloworld`, press "Enter" to search for and select the "helloworld" template.
4. Keep the default function configuration and click **Complete** as shown below:
After the function is created, you will be automatically redirected to its "Function configuration"

page where you can view its configuration information.



- "Execution" is set to "index.main_handler", indicating that the SCF Console will automatically save this snippet of code as an `index.py` file, compress it, and upload it to SCF for function creation.
- `main_handler` in the sample code is the entry function, which has the following main parameters:
 - `event` parameter: gets information of the triggering source.
 - `context` parameter: gets the environment and configuration information of this function.

5. Select the **Function code** tab to view or edit the function code online as shown below:



Deploying function (with trigger configured)

1. After editing the function code online, click **Save** and the function will be deployed.
2. On the details page of the created function, select **Trigger Management** on the left and click **Create a Trigger**.
3. In the **Create a Trigger** window that pops up, set "Trigger Method" to "API Gateway Trigger" and keep other parameters as default as shown below:

Add Trigger

Trigger Method ⓘ * **API Gateway Trigger**

For API gateway triggers, the format of contents returned from SCF should be constructed in integration response method. For details, please see [here](#).

API Service Type ⓘ * Create API Service Use Existing API Service

API Service * SCF_API_SERVICE

Up to 50 chars. It supports a-z, A-Z, 0-9 and -. It should starts with a letter and ends with a number or letter

Request method ⓘ * ANY

Up to 50 chars. It supports a-z, A-Z, 0-9 and -. It should starts with a letter and ends with a number or letter

Publishing Environment ⓘ * Publish

Authentication Method ⓘ * No authentication

Enable integration response ⓘ

Save **Cancel**

4. Click **Submit** to complete the function deployment and trigger configuration.

Cloud test

Testing function deployment

Select **Function code** and click **Test** to run the code and return the test result as shown below:

Note :

- If you need to replace the test template or its content, you can directly edit the function content or select **Current Test Template**, replace it, and then click **Save**.
- Different test templates simulate different trigger message sources, and the messages passed between different triggers and SCF are data structures agreed upon in advance. For more information, please see [Trigger Overview](#).

helloworld Normal

Function configuration **Function code** Trigger Method Execution log Monitoring information

Submitting Method: Online editing Execution: index.main_handler Runtime environment: Python2.7 [Description of Python2.7 language](#) Download

```

1 # -*- coding: utf8 -*-
2 print('Start Hello World function')
3 def main_handler(event, context):
4     print('Hello World!')
5     if 'key1' in event.keys():
6         print("value1 = " + event['key1'])
7     if 'key2' in event.keys():
8         print("value2 = " + event['key2'])
9     return "hello from scf" #return
10

```

Save **Test** Current Testing Template: Hello World event template

During this test, SCF will get the data structures of the "Hello world event template" in the `event` parameter of the `main_handler`.

```

{
  "key1": "test value 1",
  "key2": "test value 2"
}

```

Testing trigger configuration

1. After a trigger is successfully created, an access path will be generated on the "Trigger Method" page of the function as shown below:

helloworld Normal

Function configuration Function code **Trigger Method** Execution log Monitoring information

Add Trigger

API Gateway Trigger

API Name: SCF_API_SERVICE

serviceId: service-...

apiId: api-1...

Request method: ANY

Publishing Environment: release

Authentication Method: No authentication

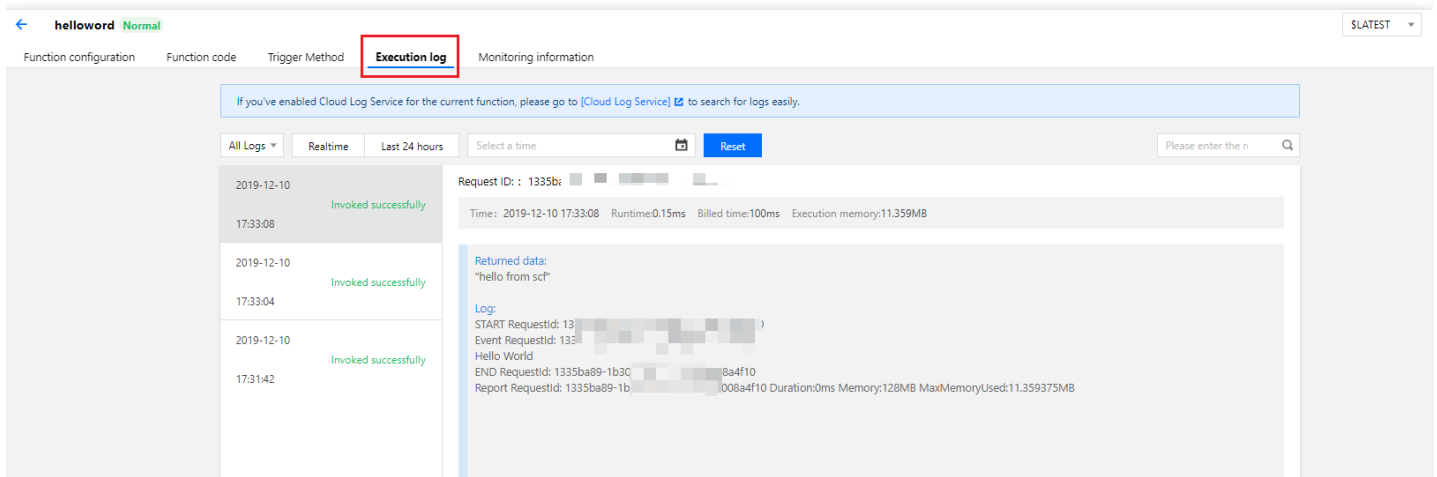
Enable integration response: Disabled

Access path: <https://service-4j75ixs-1253831162.../release/helloworld>

- Open the access path in a browser. If "hello from scf" is displayed, the function is successfully deployed.

Viewing logs

On the details page of a created function, select **Log Query** on the left to view the detailed logs of the function as shown below:



The screenshot shows the 'Execution log' tab for a function named 'helloworld'. The interface includes a navigation bar with tabs for 'Function configuration', 'Function code', 'Trigger Method', 'Execution log', and 'Monitoring information'. The 'Execution log' tab is active and highlighted with a red box. Below the navigation bar, there is a message: 'If you've enabled Cloud Log Service for the current function, please go to [Cloud Log Service] to search for logs easily.' The main content area displays a list of logs on the left and a detailed view of a specific log on the right. The log list shows three entries, all with the status 'Invoked successfully'. The detailed view for the first entry (Request ID: 1335ba89-1b30) shows the following information: Time: 2019-12-10 17:33:08, Runtime: 0.15ms, Billed time: 100ms, Execution memory: 11.359MB. The returned data is 'hello from scf'. The log content includes: 'Log: START RequestId: 1335ba89-1b30, Event RequestId: 1335ba89-1b30, Hello World, END RequestId: 1335ba89-1b30, Report RequestId: 1335ba89-1b30, 8a4f10, 008a4f10 Duration: 0ms Memory: 128MB MaxMemoryUsed: 11.359375MB'.

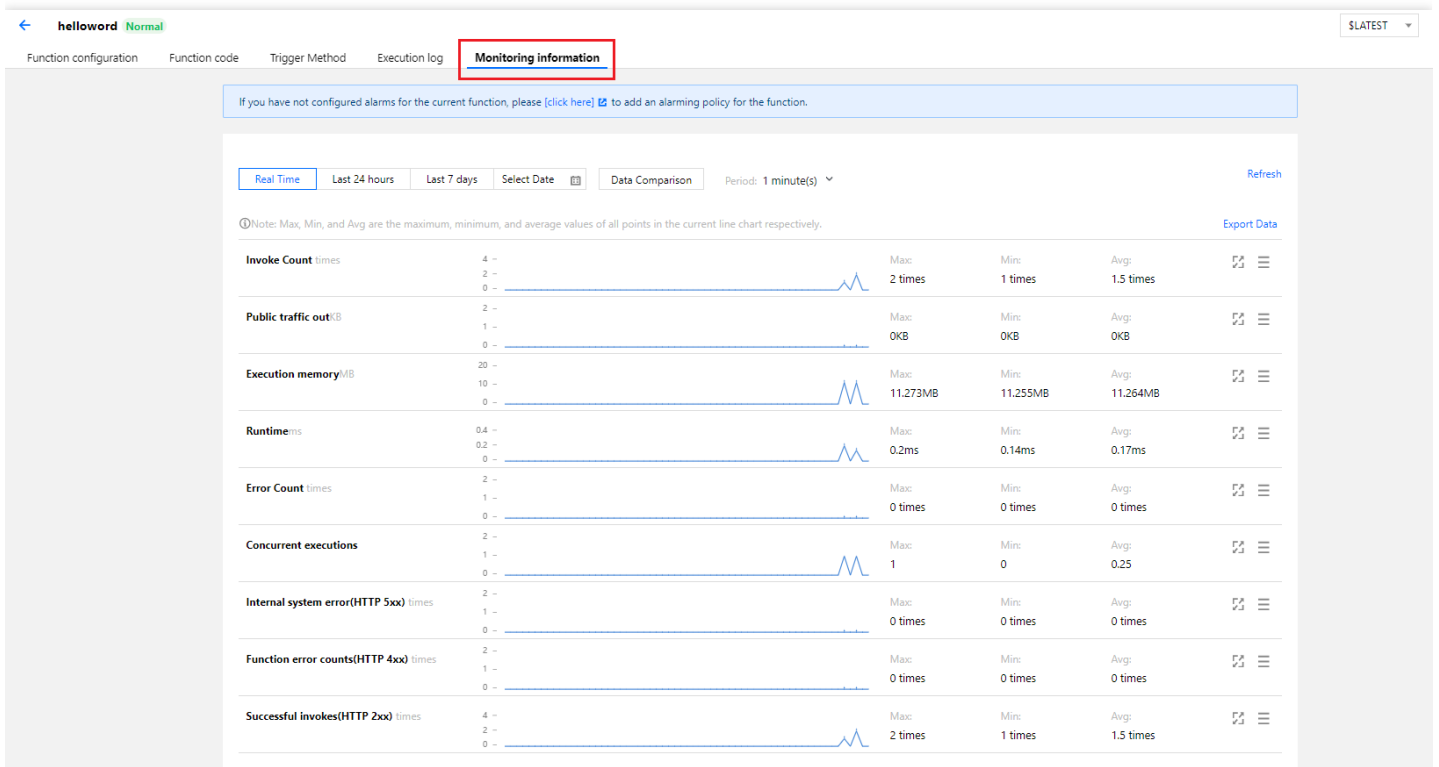
For more information on logs, please see [Viewing Execution Logs](#).

Viewing monitoring information

On the details page of a created function, select **Monitoring information** to view metrics such as function invocations and execution duration as shown below:

Note :

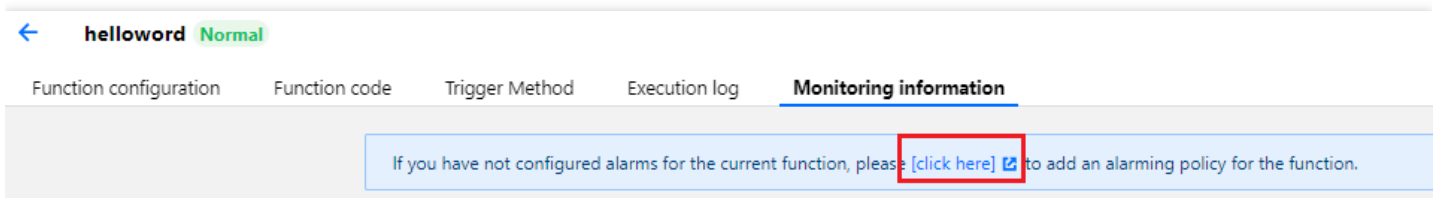
The minimal granularity of monitoring statistics collection is 1 minute. You need to wait for 1 minute before you can view the current monitoring record.



For more information on monitoring, please see [Description of Monitoring Metrics](#).

Configuring alarm

On the details page of a created function, click **click here** to configure an alarm policy for the function to monitor its running status as shown below:



For more information on how to configure an alarm, please see [Configuring Alarm](#).

Creating Functions on CLI

Last updated : 2020-11-26 18:25:52

Overview

This document describes how to use the SCF component provided by Serverless Framework to quickly create and deploy an SCF project. For more information on components and how to use them, please see [Components Overview](#)

Note :

The SCF CLI tool has been disused since February 2020. We recommend you use the more rich-featured and convenient Serverless Framework CLI tool for project development.

Prerequisites

- Serverless Framework has been installed. For more information, please see [Installing Serverless Framework](#).
- Your account has the Serverless Framework permissions. For more information, please see [Account and Permission Configuration](#).

Directions

Creating function

Run the following command to quickly create a function in the Node.js language:

```
sls init scf-demo
```

Note :

`scf-demo` in the command can be replaced with a template for another programming language. Currently, the SCF component supports the following components: `go1-helloworld` , `nodejs1015-helloworld` , `php72-helloworld` , and `python36-helloworld` .

Deploying function

Run the following command in the `scf-demo` directory to deploy the function:

```
sls deploy
```

A QR code will pop up. Please scan it to authorize and start deployment. After successful deployment, SCF resources will be automatically created.

Note :

If authentication fails, please authorize as instructed in [Account and Permission Configuration](#).

View function information

Run the following command to view the information of the deployed SCF resources:

```
sls info
```

Removing function

Run the following command to remove the deployed SCF resources:

```
sls remove
```

Note :

For more information on how to use Serverless Framework CLI to manipulate SCF functions, please see [Serverless Framework CLI](#).