

Serverless Application Center Getting Started 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

STencent 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

Getting Started

Installing Serverless Framework

Quick Deployment

Quickly Deploying a Function Template

Quick Creation of Application Template

Deployment in Console

Getting Started Installing Serverless Framework

Last updated : 2023-11-06 16:05:54

Overview

You can quickly install Serverless Framework through npm or binary installation.

Installing

Method 1. Installation through npm

Prerequisites

Before installing through npm, you need to make sure that Node.js (above v10) and npm have been installed in your environment (for more information, please see Node.js Installation Guide).

\$ node -v
v12.18.0
\$ npm -v
7.0.10

Note :

To ensure the installation speed and stability, we recommend you use cnpm for installation: download and install cnpm first, and then replace all the npm commands used below with cnpm commands.

Installation steps

Run the following command on the command line:

npm install -g serverless

Note :



If macOS prompts that you have no permission, you need to run sudo npm install -g serverless for installation.

If you have already installed Serverless Framework, you can run the following command to upgrade it to the latest version:

```
npm update -g serverless
```

Viewing version information

After the installation is completed, run the serverless -v command to view the version information of Serverless Framework:

serverless -v

Method 2. Binary installation

If Node.js is not installed in your local environment, you can install it directly in binary mode:

macOS/Linux

Open the command line and enter the following command:

curl -o- -L https://slss.io/install | bash

If you have already installed a binary version, you can run the following command to upgrade it:

serverless upgrade

Windows

Windows supports installation through Chocolatey. Open the command line and enter the following command:

```
choco install serverless
```

If you have already installed a binary version, you can run the following command to upgrade it:

choco upgrade serverless

Viewing version information



After the installation is completed, run the serverless -v command to view the version information of Serverless Framework:

serverless -v

Relevant Operations

Next step: getting started

• Quick Deployment of Function Template

Quick Deployment

Last updated : 2021-01-05 15:12:15

Overview

This task shows you how to use Serverless Framework to quickly create, configure, and deploy a Serverless application on Tencent Cloud.

i Note:

- Serverless Framework provides a Visualized Page for you to view and manage the resources of a Serverless application.
- Serverless Framework Component V2 has been released. We advise you to use the latest version.

Prerequisites

- Ensure that Serverless Framework 1.67.2 or later has been installed in advance.
- If your Tencent Cloud account is the root account, you can continue implementing deployment. If your account is a sub-account, please go to Account and Permission Configuration to get permission before implementing deployment.

Directions

Creating an Application

You can run the sls init command to quickly create an SCF, Express.js, or static website hosting application. For example, to create an Express.js application:

sls init express-demo

i Note:

```
You can run the sls registry command to view the supported demo.
```

Go to the express-demo directory and perform deployment:



cd express-demo && sls deploy

After the deployment is completed, visit the URL returned from the CLI to access the deployed application.

Note :

To view the detailed information during deployment, you can add the --debug parameter.

Viewing the Deployment Information

To view the application deployment statuses and resources again, go to the directory that is successfully deployed and run the following command:

```
$ cd express-app #Go to the project directory.
$ sls info
```

Note :

sls is short for the serverless command.

Developing and Debugging

Run the sls dev command to enable the real-time deployment log. This capability automatically detects the updates of the local code and deploys the updates on cloud. In addition, the invoked log can be output in real time. For a Node.js 10 application, cloud debugging is available. For more information, please see "In-cloud Debugging: Node.js 10+" in Development Mode and In-cloud Debugging

```
$ cd express-app
$ sls dev
```

Removing the Project

Run the sls remove command to remove all on-cloud resources, as shown below:

```
$ cd express-app #Go to the project directory.
$ sls remove
serverless $ framework
Action: "remove" - Stage: "dev" - App: "scfApp" - Instance: "scfdemo"
6s > scfdemo > Success
```

i Note:

To view the detailed information during removal, you can add the --debug parameter.

Configuring Account Information (Optional)

To configure persistent environment variables or key information, please refer to Account and Permission Configuration.

FAQs

- Question 1: What do I do if instructions are not displayed when I enter serverless ?
 Solution: Add the SERVERLESS_PLATFORM_VENDOR=tencent configuration to the .env file.
- Question 2: What do I do if the deployment with an overseas network is very slow after I enter sls deploy ? Solution: Add the GLOBAL_ACCELERATOR_NA=true configuration to the .env file to accelerate the overseas network.
- Question 3: What do I do if network error occurs during deployment after I enter sls deploy ? Solution: Add the following proxy configurations to the .env file.

HTTP_PROXY=http://127.0.0.1:12345 #Your proxy HTTPS_PROXY=http://127.0.0.1:12345 #Your proxy

Quickly Deploying a Function Template

Last updated : 2023-05-05 15:04:43

Overview

This document describes how to use Serverless Cloud Framework to quickly create, configure, and deploy an Serverless Cloud Function (SCF) application on Tencent Cloud.

Prerequisites

You have installed Serverless Cloud Framework 1.67.2 or later. For more information, see Installing Serverless Cloud

Framework.

You have signed up for a Tencent Cloud account as instructed in Signing Up.

Note

If your account is a **Tencent Cloud sub-account**, first get the authorization from the root account as instructed in Account and Permission Configuration.

Directions

Quick deployment

In an **empty folder** directory, run the following command:





serverless-cloud-framework

Next, initialize the project as prompted. Select the scf-starter template for the application and select the runtime you want to use (Node.js is used as an example here):





serverless-cloud-framework: No Serverless project is detected. Do you want to creat serverless-cloud-framework: Select the Serverless application you want to create: s

react-starter - quickly deploys a React.js application restful-api - quickly deploys a RESTful API to use Python + API Gateway > scf-starter - quickly deploys an SCF function vue-starter - quickly deploys a basic Vue.js application website-starter - quickly deploys a static website eggjs-starter - quickly deploys a basic Egg.js application express-starter - quickly deploys a basic Express.js application

```
serverless-cloud-framework: Select the runtime of the application: scf-nodejs - qui
scf-golang - quickly deploys an SCF function in Go
> scf-nodejs - quickly deploys an SCF function in Node.js
scf-php - quickly deploys an SCF function in PHP
scf-python - quickly deploys an SCF function in Python
serverless-cloud-framework: Enter the project name: demo
serverless-cloud-framework: Installing the scf-nodejs application...
scf-nodejs > Created
```

The demo project has been successfully created!

Click **Deploy Now** to quickly deploy the initialized project to the SCF console:





serverless-cloud-framework: Do you want to deploy the project in the cloud now? Yes

```
Click the link below to log in:
https://slslogin.qcloud.com/*******
Logged in successfully!
serverless-cloud-framework
Action: "deploy" - Stage: "dev" - App: "scfApp" - Name: "scf-nodejs"
code:
    bucket: sls-cloudfunction-ap-guangzhou-code
```

Serverless Application Center



```
object: /scf_component_***********.zip
description:
                    This is a function in scfApp application
functionName:
                     scf-nodejs-dev-scfApp
                      index.main_handler
handler:
lastVersion:
                     $LATEST
memorySize:
                     128
                      default
namespace:
                      Nodejs10.15
runtime:
sourceCodeDownloadUrl: https://sls-app-code-prod-********.cos.ap-guangzhou.myqclo
triggers:
   NeedCreate: true
   apiList:
      _
       apiId:
                       api-*******
       apiName:
                       index
       authType:
                       NONE
       businessType:
                       NORMAL
                       true
       created:
       isBase64Encoded: false
                       GET
       method:
       path:
                        /
    created: true
    environment: release
    protocols: http
    serviceId: service-*******
    serviceName: serverless
    subDomain: service-*******************.gz.apiqw.tencentcs.com
    urls:
     - http://service-******************************.qz.apiqw.tencentcs.com/release/
type:
                      event
SCF console: https://serverless.cloud.tencent.com/apps/scfApp/scf-nodejs/dev
184s »scf-nodejs» Execution succeeded.
```

After deployment, complete the remote invocation of the function by running the following command:





scf invoke --inputs function=helloworld

Note

 scf is short for $\operatorname{serverless-cloud-framework}$.

Viewing the Deployment Information

To view the application deployment statuses and resources again, go to the directory that is successfully deployed and run the following command:





cd demo # Enter the project directory. Please change to your actual project's direc scf info

Viewing the directory structure

In the directory of the initialized project, you can see the most basic structure of a serverless function project:





```
    serverless.yml # Configuration file
    index.js # Entry function
        .env # Environment variable file
```

The serverless.yml configuration file implements the quick configuration of the basic function information. All the configuration items supported by the SCF console can be configured in the .yml file. For more information, see SCF Configuration Information.

index.js is the entry function of the project, which is the helloworld template here.



The .env file stores user login authentication information. You can also configure other environment variables in it.

Redeployment

In the local project directory, you can modify the function template and configuration file and then redeploy the project by running the following command:



scf deploy

Note

To view the detailed information during removal, you can add the --debug parameter.

Continuous development

After the deployment is completed, Serverless Cloud Framework allows you to run different commands to implement capabilities such as continuous development and grayscale release for the project. You can also use this component in conjunction with other components to manage the deployment of multi-component applications. For more information, see Application Management and List of Supported Commands.

FAQs

 Question 1: What should I do if the wizard does not pop up by default after I enter
 serverless-cloud

 framework
 .

 Solution: Add the
 SERVERLESS_PLATFORM_VENDOR=tencent
 configuration to the
 .env
 file.

Question 2: What should I do if the deployment is very slow in an overseas network after I enter sls

Solution: Add the GLOBAL_ACCELERATOR_NA=true configuration to the .env file to accelerate the overseas network.

Question 3: What should I do if a network error occurs during deployment after I enterscf deploy?Solution: Add the following proxy configurations to the.envfile.





HTTP_PROXY=http://127.0.0.1:12345 # Replace "12345" with your proxy port HTTPS_PROXY=http://127.0.0.1:12345 # Replace "12345" with your proxy port

Quick Creation of Application Template

Last updated : 2022-05-16 11:43:29

Overview

This document describes how to create, configure, and deploy a web framework application in Tencent Cloud through Serverless Framework.

Prerequisites

• Install Serverless Framework 1.67.2 or later versions.

```
npm install -g serverless
```

• Register on Tencent Cloud and complete identity verification.

Note:

If your account is a **Tencent Cloud sub-account**, please get the authorization from the root account first as instructed in Account and Permission Configuration.

Directions

Quick deployment

In an empty folder directory, run the following command:

```
serverless
```

Next, follow the prompts to initialize the project. Select the application framework template you want to deploy (Express is used as an example here):

```
Serverless: No serverless project is detected. Do you want to create one? Yes
Serverless: Please select the serverless application you want to create: express-
starter
```



```
eggjs-starter - quickly deploys a basic Egg.js application
> express-starter - quickly deploys a basic Express.js application
flask-starter - quickly deploys a basic Flask application
fullstack - quickly deploys a full stack application (Vue.js + Express + Postgre
s)
koa-starter - quickly deploys a basic Koa.js application
laravel-starter - quickly deploys a basic Laravel application
nextjs-starter - quickly deploys a basic Next.js application
Serverless: Please enter the project name: demo
Serverless: Installing the express-starter application...
express-starter > Created
```

The demo project has been successfully created!

Select **Deploy Now** to deploy the initialized project to Tencent Cloud:

```
Serverless: Do you want to deploy the project in the cloud now? Yes
XXXXXXXX
x OR x
x CODE x
XXXXXXXX
Please scan the QR code above with WeChat or click the link below to log in
https://slslogin.gcloud.com/XKYUcbaK
Logged in successfully!
serverless framework
Action: "deploy" - Stage: "dev" - App: "demo1" - Instance: "expressDemo"
region: ap-guangzhou
apigw:
serviceId: service-xxxxx
subDomain: service-xxxx.gz.apigw.tencentcs.com
environment: release
url: https://service-xxxx.gz.apigw.tencentcs.com/release/
scf:
functionName: express_component
runtime: Nodejs10.15
namespace: default
lastVersion: $LATEST
traffic: 1
26s > expressDemo > Success
```

After the deployment is completed, click the API Gateway link output by the command line to quickly access the successfully deployed web framework application.

Viewing deployment information

If you want to check the deployment status and resources of the application again, you can go to the folder where the project is successfully deployed and run the following command to view the corresponding information:

```
cd demo # Enter the project directory. Please change to your actual project's dir
ectory name here
sls info
```

Note :

sls is short for the serverless command.

Viewing directory structure

In the directory of the initialized project, you can see the most basic structure of an Express project:

```
serverless.yml # Configuration file
index.js # Entry function
package.json # Project dependencies
.env # Environment variable file
```

- The serverless.yml configuration file implements the quick configuration of the basic function information. All the configuration items supported by the SCF console can be configured in the .yml file (for more information, please see SCF Configuration Information).
- index.js is the entry function of the project, which is the helloworld template here.
- package is the dependent file of the project, which records the dependency package that should be installed for the Express project.
- The .env file stores user login authentication information. You can also configure other environment variables in it.

Redeployment

In the local project directory, you can modify the function template and configuration file, reinstall the dependencies, and then redeploy the project by running the following command:

npm install && sls deploy

Note:

If you want to view the details during the removal process, you can add the --debug parameter after sls deploy .

Continuous development

After the deployment is completed, you can log in to the SLS console to view the monitoring metric data output after project deployment, such as the basic information, the number of project requests, and project error statistics. View the monitoring metric data output after project deployment, such as the basic information, the number of project requests, and project error statistics, and implement the continuous development and deployment of the project.

For more information, see Console Deployment Guide.

Serverless Framework supports running different commands to help you implement continuous development, deployment, and grayscale release for the project. You can also use this component in conjunction with other advanced capabilities such as **layer** and **custom domain name** to configure advanced features for the application.

FAQs

- Problem 1: The wizard does not pop up by default when serverless is entered.
 Solution: Add the SERVERLESS_PLATFORM_VENDOR=tencent configuration item to the .env file.
- Problem 2: After sls deploy is entered in a network environment outside the Chinese mainland, the deployment is very slow.

Solution: Add the GLOBAL_ACCELERATOR_NA=true configuration item to the .env file to enable acceleration outside the Chinese mainland.

• Problem 3: Ater sls deploy is entered, the deployment reports a network error. Solution: Add the following proxy configuration to the .env file.

HTTP_PROXY=http://127.0.0.1:12345 # Replace "12345" with your proxy port HTTPS_PROXY=http://127.0.0.1:12345 # Replace "12345" with your proxy port

Deployment in Console

Last updated : 2022-07-22 15:16:16

Overview

For common framework components, you can quickly develop and deploy your applications in the Tencent Cloud SLS console.

Prerequisites

Before deploying your application in the console, you must complete the following permission configuration:

Authorizing by root account

- 1. Log in to the SLS console and click Authorize to enter the CAM console.
- 2. On the **Role** list page in the CAM console, check whether the **SLS_QcsRole** and **CODING_QCSRole** service roles have been created successfully.

Note :

```
If you have already created CODING_QCSRole, please check whether it has complete permissions. It requires the following basic policies: QcloudSLSFullAccess, QcloudSSLFullAccess, and QcloudAccessForCODINGRole. If any policy is missing, please add it manually.
```

3. You can start to use the service after making sure that both the roles and permissions meet the requirements.

Granting permissions to sub-account

If the **Tencent Cloud SLS** and **Coding DevOps** services haven't been activated yet, please contact the root account to activate them and create the roles (as instructed in Directions).

Directions

Step 1. Create an application

- 1. Log in to the SLS console.
- 2. Click Create Application to enter the project creation page.

3. Enter the basic application information as prompted on the page.

- Application Name: it can contain 2–63 lowercase letters, digits, and hyphens and must start with a lowercase letter and end with a digit or lowercase letter. It cannot be changed once the application is created.
- Environment: select dev, test, or prod. Custom environment is also supported.
- Region: it is the same as the region supported by SCF. For more information, see Region List.
- Creation Method: you can create an application from an application template or by importing an existing project. You can choose either method based on your actual needs.

Note:

If you choose to create an application by importing an existing project, you need to make certain modifications to some frameworks. For more information, please see the relevant framework migration documentation.

4. Click **Create**, and the application will be automatically deployed for you. You can view the deployment log of the project.

Creating application from application template

If you choose to create an application from a template, you can quickly create a web application by selecting a project template provided in the console. During template-based deployment, the following configuration will be completed for you by default:

- 1. Create a layer (for the Node.js framework only) and store the project dependency package node_modules
 in the layer. For more information on how to use a layer, please see Overview.
- 2. Create a COS bucket (for Next.js and Nuxt.js frameworks only), separate the static resources, and host them in the COS bucket.

You can also configure more capabilities for your project in **Advanced Configuration**, such as custom domain name and detailed function configuration.

Note :

When configuring a custom domain name, make sure that an ICP filing has been obtained and a CNAME record has been configured for it. For more information, see Custom Domain Name and Certificate.

Importing existing project



The Serverless console allows you to migrate an existing project through code hosting and folder upload.

Code hosting

Currently, **GitHub**, **GitLab**, **Gitee**, and public custom code repositories are supported. You can have your application automatically updated by selecting the trigger method for it.

• Folder upload

You can directly import a local project by uploading the folder. For the Node.js framework, Serverless Framework will automatically create a layer and pass the dependency package <code>node_modules</code> into the layer to complete the deployment.

Step 2. Manage resources

In the SLS console, click the target application to enter the application details page and view the basic information output after the project is deployed as well as monitoring metric data such as the number of requests and error statistics.

Step 3. Deploy for development

Click **Deploy** at the top of the application details page, and you can quickly modify the configuration of your project and upload it for re-deployment. Three methods are supported: **local upload, code hosting, and CLI development**.