

Serverless Framework

快速入门

产品文档



腾讯云

【版权声明】

©2013-2019 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

快速入门

Component 快速入门

Framework CLI 快速入门

快速入门

Component 快速入门

最近更新时间：2020-03-15 11:10:45

操作场景

该任务指导您使用 Serverless Framework 开源 Component 在腾讯云上部署一个云函数 + API 网关的服务，并完成创建、配置、部署等步骤。

说明：

- 通过 Serverless Framework 创建的资源，您可以在资源自身的控制台进行查看和管理，例如 [云函数控制台](#)、[API 网关控制台](#) 等。
- 预计2020年3月，Serverless Framework 将提供可视化的页面，您可以从 Serverless 应用的角度查看和管理资源。

前提条件

在使用之前，请确保如下软件已经安装：

- [Node.js](#) (6.x或以上的版本)
- [Serverless Framework](#) (1.57.0或以上的版本)

如果这些条件已经满足，您可以跳过此步骤，直接 [开始部署一个服务](#)。

安装 Node.js 和 NPM

1. 参考 [Node.js 安装指南](#) 根据您的系统环境进行安装。
2. 安装完毕后，通过 `node -v` 命令，查看安装好的 Node.js 版本信息：

```
$ node -v  
vx.x.x
```

3. 通过 `npm -v` 命令，查看安装好的 npm 版本信息：

```
$ npm -v  
x.x.x
```

安装 Serverless Framework

1. 在命令行中运行如下命令：

```
$ npm install -g serverless
```

说明：

如 Mac 系统提示无权限，则需要运行 `sudo npm install -g serverless` 进行安装。

2. 安装完毕后，通过运行 `serverless -v` 命令，查看 Serverless Framework CLI 的版本信息。

```
$ serverless -v  
X.X.X
```

操作步骤

完成上述安装准备后，通过如下步骤开始部署一个 Serverless 服务。

创建服务

1. 创建并进入目录：

```
$ mkdir my-function && cd my-function
```

2. 在目录中创建 `index.js` 作为云函数的入口函数：

```
$ touch index.js
```

3. 在 `index.js` 中增加如下代码：

```
'use strict';  
exports.main_handler = async (event, context, callback) => {  
  console.log("%j", event);  
  return "hello world"  
};
```

配置服务

在本地创建 `serverless.yml` 文件，

```
$ touch serverless.yml
```

在 `serverless.yml` 中进行如下配置：

```
# serverless.yml
myFunction:
  component: "@serverless/tencent-scf"
  inputs:
    name: myFunction
    codeUri: ./ # 代码目录
    handler: index.main_handler
    runtime: Nodejs8.9
    region: ap-guangzhou
    description: My Serverless Function
    memorySize: 128
  events: # 触发器配置
  - apigw:
    name: serverless
    parameters:
    protocols:
  - http
  endpoints:
  - path: /
    method: GET
```

说明：

您可以通过 [详细配置文档](#)，查看 `serverless.yml` 中所有可用属性的属性列表。

部署服务

如果您的账号未 [登录](#) 或 [注册](#) 腾讯云，您可以在运行该命令后，直接用[微信](#)扫描命令中弹出的二维码，对云账户进行授权登录和注册。

通过 `sls` 命令进行部署，并可以添加 `--debug` 参数查看部署过程中的信息：

说明：

`sls` 是 `serverless` 命令的简写。

```
sls --debug
```

```
DEBUG – Resolving the template's static variables.
```

```
DEBUG – Collecting components from the template.
DEBUG – Downloading any NPM components found in the template.
DEBUG – Analyzing the template's components dependencies.
DEBUG – Creating the template's components graph.
DEBUG – Syncing template state.
DEBUG – Executing the template's components graph.
Please scan QR code login from wechat.
Wait login...
Login successful for TencentCloud.
DEBUG – Compressing function myFunction file to /Users/tina/Desktop/live/scfcomponent/my-function/.serverless/myFunction.zip.
DEBUG – Compressed function myFunction file successful
DEBUG – Uploading service package to cos[sls-cloudfunction-ap-guangzhou-code].sls-cloudfunction-default-myFunction-1582797244.zip
DEBUG – Uploaded package successful /Users/tina/Desktop/live/scfcomponent/my-function/.serverless/myFunction.zip
DEBUG – Creating function myFunction
DEBUG – Created function myFunction successful
DEBUG – Setting tags for function myFunction
DEBUG – Creating trigger for function myFunction
DEBUG – Starting API-Gateway deployment with name myFunction.serverless in the ap-guangzhou region
DEBUG – Service with ID service-qs0cud0s created.
DEBUG – API with id api-irl0q216 created.
DEBUG – Deploying service with id service-qs0cud0s.
DEBUG – Deployment successful for the api named myFunction.serverless in the ap-guangzhou region.
DEBUG – Deployed function myFunction successful

myFunction:
Name: myFunction
Runtime: Nodejs8.9
Handler: index.main_handler
MemorySize: 128
Timeout: 3
Region: ap-guangzhou
Description: My Serverless Function
APIGateway:
- serverless - http://service-qs0cud0s-1300862921.gz.apigw.tencentcs.com/release

22s › myFunction › done
```

测试服务

在浏览器中打开输出链接，或替换如下命令中的链接地址，通过 curl 对其进行测试，该链接可以在 sls 命令执行后获取得到。

```
$ curl -X GET http://service-qs0cud0s-1300862921.gz.apigw.tencentcs.com/release
```

移除服务

如果您不再需要此服务，可以通过如下命令一键移除服务，该命令会清理相应函数和触发器资源。

```
serverless remove --debug
```

```
DEBUG – Flushing template state and removing all components.
```

```
DEBUG – Removing any previously deployed API. api-irl0q216
```

```
DEBUG – Removing any previously deployed service. service-qs0cud0s
```

```
DEBUG – Removing function
```

```
DEBUG – Removed function myFunction successful
```

```
9s > myFunction1 > done
```

配置账户信息（可选）

当前默认支持部署时扫描微信二维码登录，如您希望配置持久的环境变量/密钥信息，也可以参考 [配置账号](#) 文档。

Framework CLI 快速入门

最近更新时间：2020-01-16 11:49:39

操作场景

该任务指导您使用 Serverless Framework 开源 CLI 在腾讯云上部署一个服务，并完成配置、创建、测试、部署等步骤。

说明：

- 通过 Serverless Framework 创建的资源，您可以在资源自身的控制台进行查看和管理，例如 [云函数控制台](#)、[API 网关控制台](#) 等。
- 预计2020年3月，Serverless Framework 将提供可视化的页面，您可以从 Serverless 应用的角度查看和管理资源。

前提条件

在使用之前，请确保如下软件已经安装：

- [Node.js](#) (6.x或以上的版本)
- [Serverless Framework CLI](#) (1.57.0或以上的版本)

如果这些条件已经满足，您可以跳过此步骤，直接 [开始部署一个服务](#)。

安装 Node.js 和 NPM

- 参考 [Node.js 安装指南](#) 根据您的系统环境进行安装。
- 安装完毕后，通过 `node -v` 命令，查看安装好的 Node.js 版本信息：

```
$ node -v
vx.x.x
```

- 通过 `npm -v` 命令，查看安装好的 npm 版本信息：

```
$ npm -v
x.x.x
```

安装 Serverless Framework CLI

- 在命令行中运行如下命令：

```
npm install -g serverless
```

- 安装完毕后，通过运行 `serverless -v` 命令，查看 Serverless Framework CLI 的版本信息。

```
$ serverless -v  
X.X.X
```

操作步骤

完成上述安装准备后，通过如下步骤开始部署一个 Serverless 服务。

通过模板创建服务

- 使用 Serverless Framework 的 `tencent-nodejs` 模板创建一个新的服务。

通过运行如下命令进行创建，`--path` 可以指定服务的路径：

```
# 创建一个serverless服务  
$ serverless create --template tencent-nodejs --path my-service
```

- 安装依赖。

进入服务所在路径，运行如下命令安装依赖：

```
$ cd my-service  
$ npm install
```

配置触发器

云函数需要通过触发器的事件调用进行触发，因此可以在 `serverless.yml` 中增加对触发器的配置，以 API 网关触发器为例，配置如下：

```
service: my-service # service name  
  
provider: # provider information  
name: tencent  
runtime: Nodejs8.9  
credentials: ~/credentials #如不使用二维码一键登录，密钥字段需要和 credentials 文件路径一致  
  
plugins:  
- serverless-tencent-scf
```

functions:

hello_world: # 函数名称

handler: `index.main_handler`**runtime:** Nodejs8.9

events:

- **apigw:**

name: hello_world_apigw

parameters:

stageName: release

serviceld:

httpMethod: ANY

说明：

- 您可以通过 [详细配置文档](#)，查看 `serverless.yml` 中所有可用属性的属性列表。
- Serverless Framework 会为控制台中实际部署的函数增加前缀组成函数名称，前缀规范为 `service-stage-function`，默认的stage为 `dev`。以上述配置为例，配置文件中的函数名称 `hello_world` 在控制台中的函数名称对应为 `my-service-dev-hello_world`。

部署服务

通过该命令部署或更新您创建的函数和触发器，资源配置会和 `serverless.yml` 中保持一致。

```
serverless deploy
```

如您的账号未 [登录](#) 或 [注册](#) 腾讯云，您可以在运行该命令后，直接用[微信](#)扫描命令中弹出的二维码，对云账户进行授权登录和注册。

更多部署详情参考 [部署服务](#) 文档。

测试服务

替换如下命令中的链接地址，通过 `curl` 对其进行测试，该链接可以在 `sls deploy` 命令执行后获取得到。

```
$ curl -X POST https://service-xxxx-1300000000.ap-guangzhou.apigateway.myqcloud.com/release/
```

云端调用

通过以下命令云端调用函数并且获得日志信息的返回。

```
serverless invoke -f hello_world
```

更多部署详情参考 [云端调用](#)。

获取函数日志

单独开启一个命令行，输入如下命令，再次云端调用函数，并实时获取函数 `hello_world` 的云端调用日志信息。

```
serverless logs -f hello_world -t
```

移除服务

如果您不再需要此服务，可以通过如下命令一键移除服务，该命令会清理相应函数和触发器资源。

```
serverless remove
```

配置账户信息

当前默认支持 CLI 扫描二维码登录，如您希望配置持久的环境变量/密钥信息，也可以参考 [配置账号](#) 文档。