

Serverless 应用中心

快速入门

产品文档



腾讯云

【版权声明】

©2013-2023 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

- 安装 Serverless Framework

- 快速部署

- 快速部署函数模板

- 快速创建应用模板

- 控制台开发指南

快速入门

安装 Serverless Framework

最近更新时间：2021-12-31 17:59:07

操作场景

您可以通过 [NPM 安装](#) 或 [二进制安装](#) 的方式，快速安装 Serverless Framework。

安装方式

方式1：NPM 安装

安装前提

使用 npm 安装前，需要确保您的环境中已安装好了 Node（版本需要 > 10）以及 npm（查看 [Node.js 安装指南](#)）。

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

注意：

为保证安装速度和稳定性，建议您使用 [cnpm](#) 来完成安装：先下载安装 [cnpm](#)，然后将下面所有使用的 npm 命令替换为 [cnpm](#) 即可。

安装步骤

在命令行中运行如下命令：

```
npm install -g serverless
```

说明：

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

如果之前您已经安装过 `Serverless Framework`，可以通过以下命令升级到最新版。

```
npm update -g serverless
```

查看版本信息

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

```
serverless -v
```

方式2：二进制安装

如果您的本地环境没有安装 `Node.js`，您可以直接使用二进制的方式进行安装：

MacOS/Linux 系统

打开命令行，输入以下命令：

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

如果之前您已经安装过二进制版本，可以通过下列命令进行升级：

```
serverless upgrade
```

Windows 系统

Windows 系统支持通过 `chocolatey` 进行安装。打开命令行，输入以下命令：

```
choco install serverless
```

如果之前您已经安装过二进制版本，可以通过下列命令进行升级：

```
choco upgrade serverless
```

查看版本信息

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

```
serverless -v
```

相关操作

下一步：快速开始

- [快速部署函数模版](#)

快速部署

最近更新时间：2020-10-10 17:13:39

操作场景

该任务指导您通过 Serverless Framework，在腾讯云上快速创建、配置和部署一个 Serverless 应用。

说明：

- Serverless Framework 已提供已提供 [可视化页面](#)，您可以从 Serverless 应用的角度查看和管理资源。
- Serverless Framework V2 版本 Component 已经发布，我们推荐您使用最新版本。

前提条件

- 在使用之前，请确保已经 [安装 Serverless Framework 1.67.2 以上版本](#)。
- 如果您的腾讯云账号为主账号，则可以继续进行部署；如果您的账户为子账号，请参考 [账号和权限配置](#) 获得授权后再进行部署。

操作步骤

创建应用

在空文件夹下输入 `serverless` 命令，按照引导进行操作，即可部署一个 SCF、Express.js 或者静态网站托管应用。交互流程如下所示：

```
$ serverless
Serverless: 当前未检测到 Serverless 项目，是否希望新建一个项目？ (Y/n) y
Serverless: 请选择您希望创建的 Serverless 应用 (Use arrow keys or type to search)
> Express.js 项目模板
SCF 函数项目模板
Website 项目模板
Serverless: 请输入项目名称 express-app

express-app 项目已成功创建！
Serverless: 是否希望立即将该项目部署到云端？ (Y/n) y
Please scan QR code login from wechat.
Wait login...
Login successful for TencentCloud.
```

```
serverless ⚡ framework
Action: "deploy" - Stage: "dev" - App: "scfApp" - Instance: "scfdemo"

FunctionName: scfFunctionName
Description:
Namespace: default
Runtime: Nodejs10.15
Handler: index.main_handler
MemorySize: 128
Triggers:
apigw:
- https://service-9k0ggfbe-1250000000.gz.apigw.tencentcs.com/release/index

23s > scfdemo > Success
```

部署完毕后，访问命令行中输出的网页链接，即可访问已经部署成功的应用。

说明：

如果希望查看部署过程中的详细信息，可以增加 `--debug` 参数进行查看。

查看部署信息

如果希望再次查看应用的部署状态和资源，可以进入到部署成功的文件夹，运行如下命令，查看对应信息：

```
$ cd express-app #进入项目目录
$ sls info
```

说明：

`sls` 是 `serverless` 命令的简写。

开发调试

通过运行 `sls dev` 命令，可以开启部署的实时日志，该能力会自动监测本地代码的更新，并自动部署同步到云端，同时实时输出调用日志。针对 `Node.js 10` 的应用，还可以启用云端调试能力。详情参考 [Node.js 云端调试](#)。

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

移除项目

通过 `sls remove` 命令可以移除云端部署的所有资源，如下所示：


```
$ cd express-app #进入项目目录
$ sls remove

serverless ✂ framework
Action: "remove" - Stage: "dev" - App: "scfApp" - Instance: "scfdemo"

6s > scfdemo > Success
```

① 说明：

如果希望查看移除过程中的详细信息，可以增加 `--debug` 参数进行查看。

配置账户信息（可选）

配置持久的环境变量/密钥信息，请参考 [配置账号](#) 文档。

常见问题

- 问题1：输入 `serverless` 时没有默认弹出中文引导。
解决方案：在 `.env` 文件中增加配置 `SERVERLESS_PLATFORM_VENDOR=tencent` 即可。
- 问题2：在境外网络环境，输入 `sls deploy` 后部署十分缓慢。
解决方案：在 `.env` 文件中增加配置 `GLOBAL_ACCELERATOR_NA=true` 则开启境外加速。
- 问题3：输入 `sls deploy` 后部署报网络错误。
解决方案：在 `.env` 文件中增加以下代理配置。

```
HTTP_PROXY=http://127.0.0.1:12345 #您的代理
HTTPS_PROXY=http://127.0.0.1:12345 #您的代理
```

快速部署函数模板

最近更新时间：2023-05-05 15:05:13

操作场景

该任务指导您通过 Serverless Cloud Framework，在腾讯云上快速创建、配置和部署一个 SCF 云函数应用。

前提条件

已经安装 Serverless Cloud Framework 1.67.2 以上版本。详情见 [安装 Serverless Cloud Framework](#)。

已经 [注册腾讯云账号](#)。

说明

如果您的账号为**腾讯云子账号**，请先联系主账号，参考 [账号和权限配置](#) 进行授权。

操作步骤

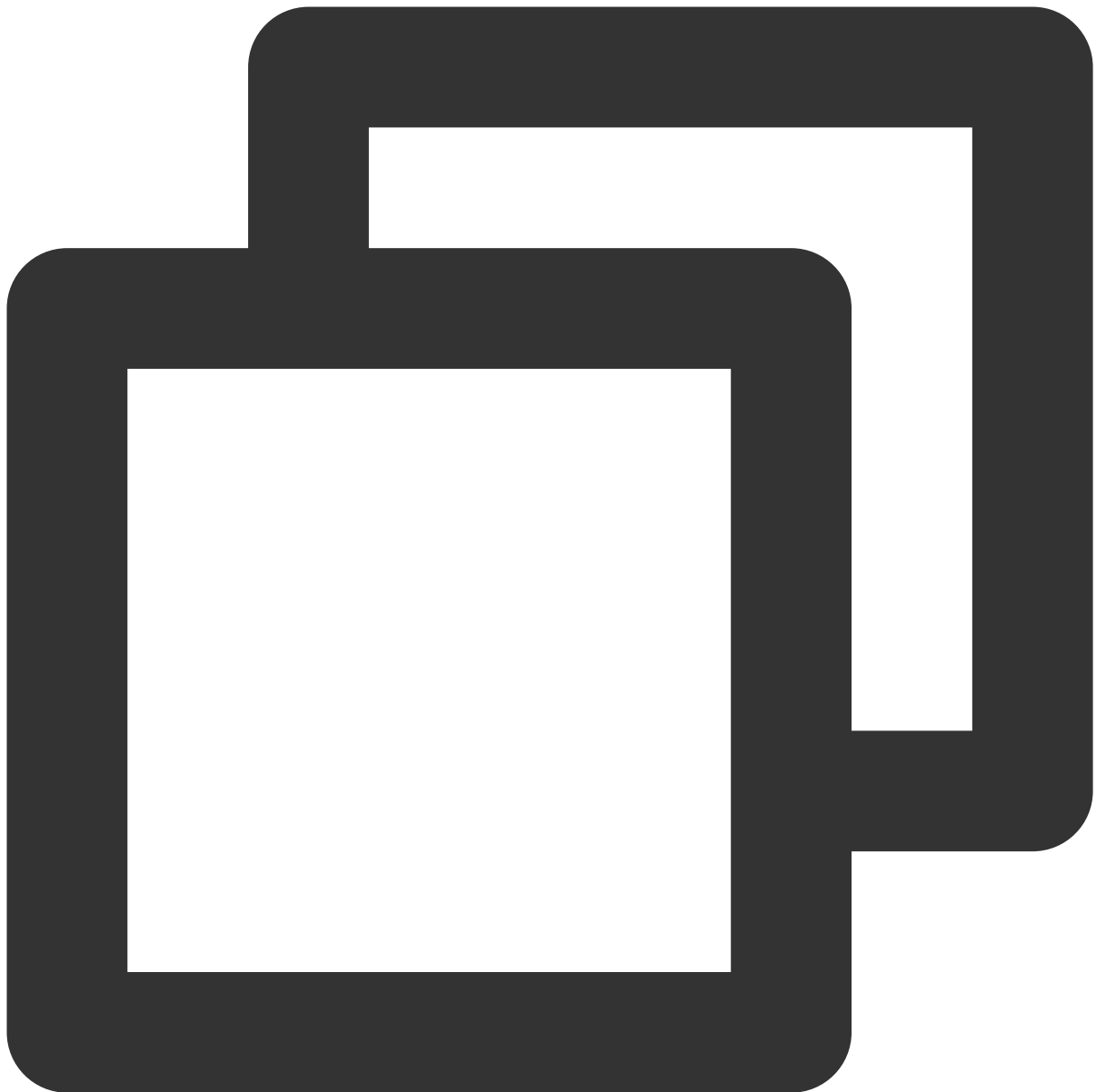
快速部署

在**空文件夹**目录下，执行如下指令：



```
serverless-cloud-framework
```

接下来按照交互提示，完成项目初始化，应用请选择 `scf-starter` 模板，并选择您希望用的运行时（此处以 Node.js 为例）：



```
serverless-cloud-framework: 当前未检测到 Serverless 项目, 是否希望新建一个项目? Yes
serverless-cloud-framework: 请选择您希望创建的 Serverless 应用 scf-starter - 快速部署一个
```

```
  react-starter - 快速部署一个 React.js 应用
  restful-api - 快速部署一个 REST API 使用 python + API gateway
  > scf-starter - 快速部署一个云函数
  vue-starter - 快速部署一个 Vue.js 基础应用
  website-starter - 快速部署一个静态网站
  eggjs-starter - 快速部署一个Egg.js 基础应用
  express-starter - 快速部署一个 Express.js 基础应用
```

```
serverless-cloud-framework: 请选择应用的运行时 scf-nodejs - 快速部署一个 nodejs 云函数

  scf-golang - 快速部署一个 golang 云函数
  > scf-nodejs - 快速部署一个 nodejs 云函数
  scf-php - 快速部署一个 PHP 云函数
  scf-python - 快速部署一个 python 云函数

serverless-cloud-framework: 请输入项目名称 demo
serverless-cloud-framework: 正在安装 scf-nodejs 应用...

scf-nodejs > Created

demo 项目已成功创建!
```

选择**立即部署**，将已经初始化好的项目快速部署到云函数控制台：



```
serverless-cloud-framework: 是否希望立即将该项目部署到云端? Yes
```

请点击下方链接登录

```
https://slogin.qcloud.com/*****
```

登录成功!

```
serverless-cloud-framework
```

```
Action: "deploy" - Stage: "dev" - App: "scfApp" - Name: "scf-nodejs"
```

```
code:
```

```
bucket: sls-cloudfunction-ap-guangzhou-code
```

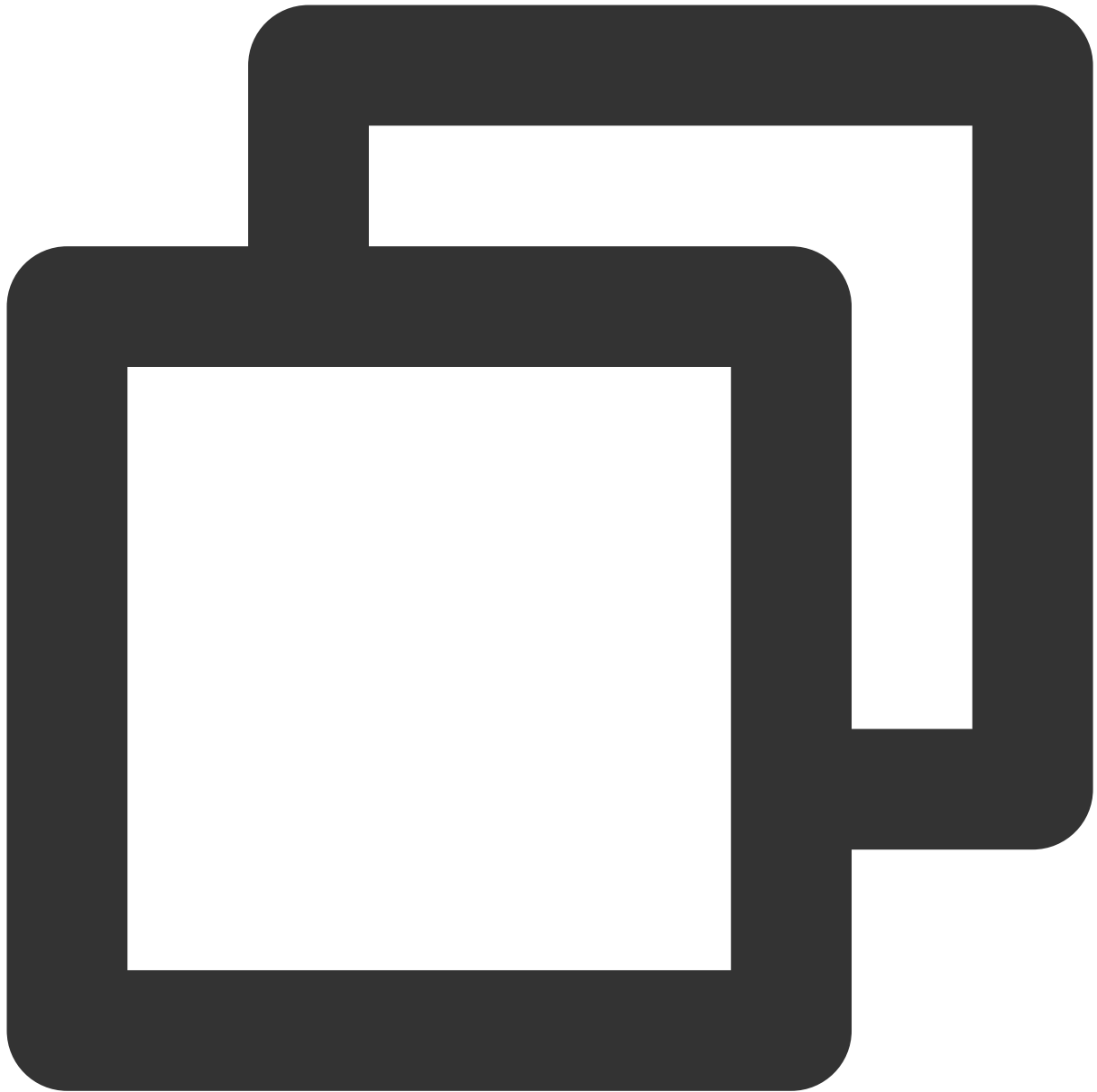
```

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

应用控制台: <https://serverless.cloud.tencent.com/apps/scfApp/scf-nodejs/dev>

184s »scf-nodejs» 执行成功

部署完毕后，通过以下指令，完成函数的远程调用：



```
scf invoke --inputs function=helloworld
```

说明

`serverless-cloud-framework` 命令的简写为 `scf`。

查看部署信息

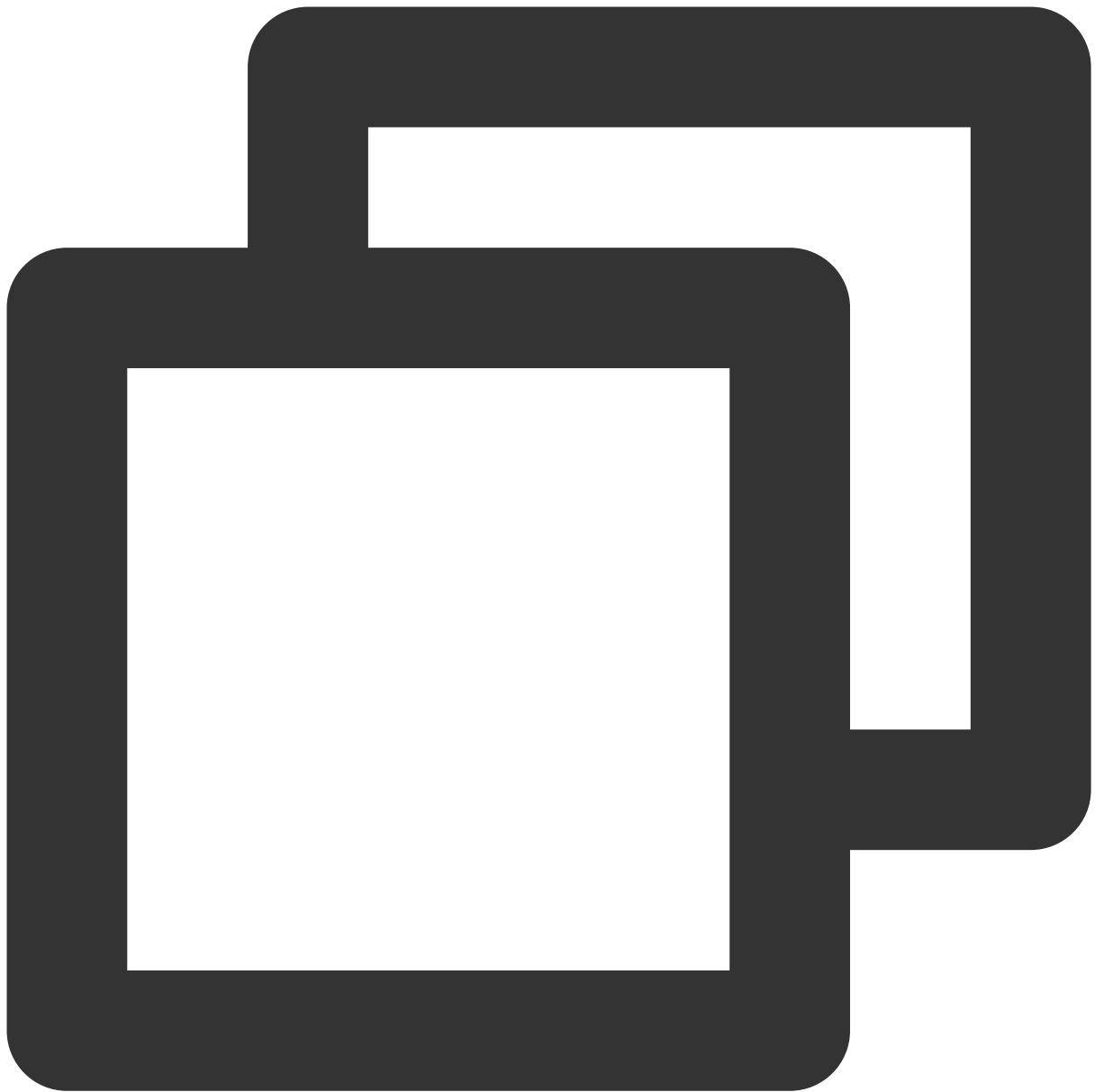
如果希望再次查看应用的部署状态和资源，可以进入到部署成功的文件夹，运行如下命令，查看对应信息：



```
cd demo #进入项目目录，此处请改为您的项目目录名称
scf info
```

查看目录结构

在初始化的项目目录下，可以看到一个 Serverless 函数项目的最基本结构：



```
.
├── serverless.yml # 配置文件
├── index.js      # 入口函数
└── .env         # 环境变量文件
```

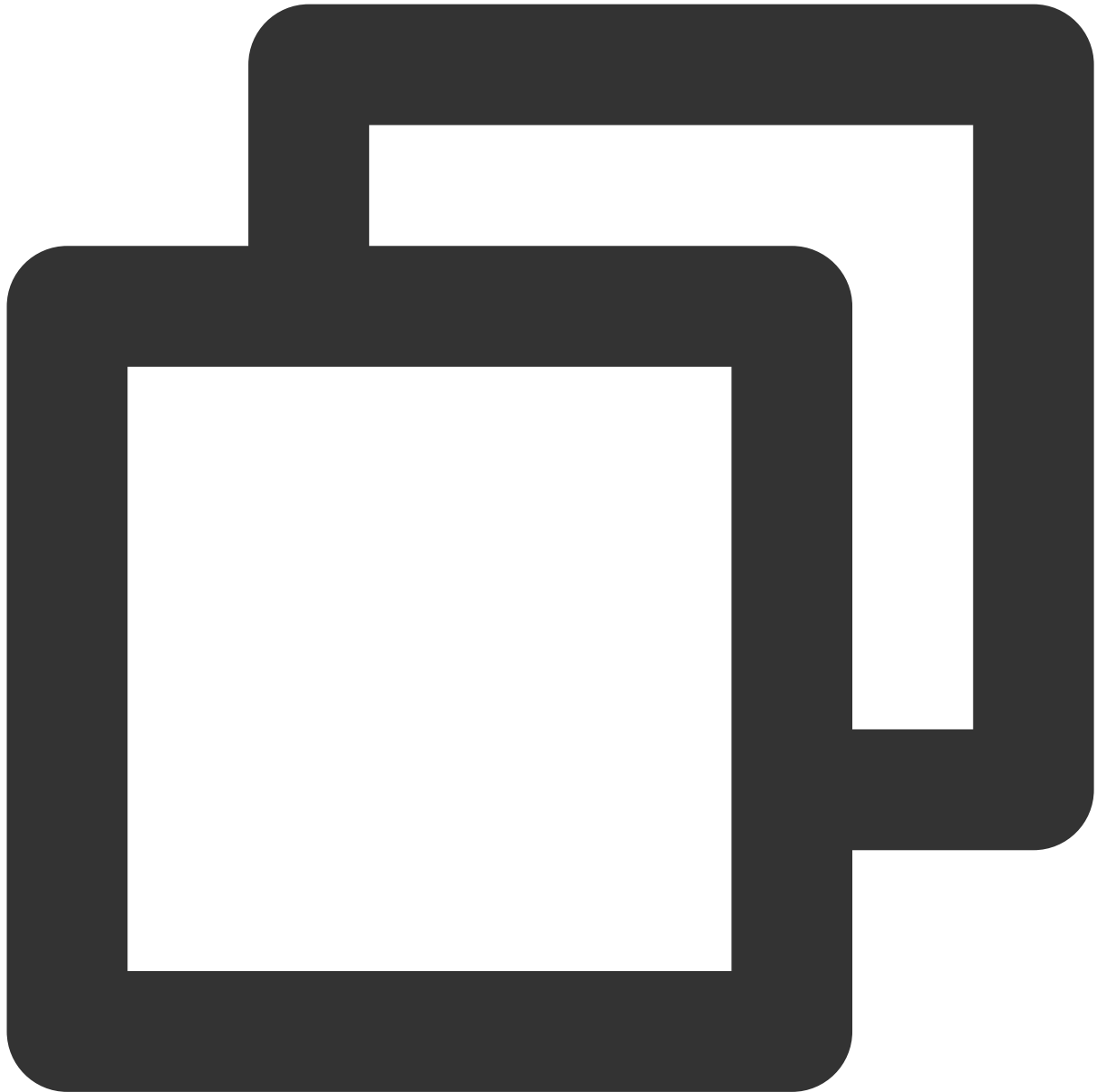
`serverless.yml` 配置文件实现了函数基本信息的快速配置，函数控制台支持的配置项都支持在 `yml` 文件里配置（查看[云函数的全量配置信息](#)）。

`index.js` 为项目的入口函数，此处为 `helloworld` 模板。

`.env` 文件里存放了用户登录的鉴权信息，您也可以在里面配置其它环境变量。

重新部署

在本地项目目录下，您可以对函数模板项目内容与配置文件进行修改，并通过以下指令进行重新部署：



```
scf deploy
```

说明

如需查看移除过程中的详细信息，可以增加 `--debug` 参数进行查看。

持续开发

部署完成后，Serverless Cloud Framework 支持通过不同指令，帮助您完成项目的持续开发部署、灰度发布等能力，您也可以结合其它组件一起使用，完成多组件应用的部署管理。

详情请参考文档 [应用管理](#) 与 [支持命令列表](#)。

常见问题

问题1：输入 `serverless-cloud-framework` 时没有默认弹出中文引导。

解决方案：在 `.env` 文件中增加配置 `SERVERLESS_PLATFORM_VENDOR=tencent` 即可。

问题2：在境外网络环境，输入 `scf deploy` 后部署十分缓慢。

解决方案：在 `.env` 文件中增加配置 `GLOBAL_ACCELERATOR_NA=true` 则开启境外加速。

问题3：输入 `scf deploy` 后部署报网络错误。

解决方案：在 `.env` 文件中增加以下代理配置。



```
HTTP_PROXY=http://127.0.0.1:12345 #请将'12345'替换为您的代理端口  
HTTPS_PROXY=http://127.0.0.1:12345 #请将'12345'替换为您的代理端口
```

快速创建应用模板

最近更新时间：2021-12-31 18:03:30

操作场景

该任务指导您通过 Serverless Framework，在腾讯云上快速创建、配置和部署一个 Web 框架应用。

前提条件

- 已经 [安装 Serverless Framework 1.67.2](#) 以上版本。

```
npm install -g serverless
```

- 已经 [注册腾讯云账号](#) 并完成 [实名认证](#)。

说明：

如果您的账号为[腾讯云子账号](#)，请先联系主账号，参考 [账号和权限配置](#) 进行授权。

操作步骤

快速部署

在空文件夹目录下，执行如下指令：

```
serverless
```

接下来按照交互提示，完成项目初始化，应用请选择您希望部署的应用框架模版（此处以 Express 为例）：

```
Serverless: 当前未检测到 Serverless 项目，是否希望新建一个项目？ Yes
Serverless: 请选择您希望创建的 Serverless 应用 express-starter
eggjs-starter - 快速部署一个Egg.js 基础应用
> express-starter - 快速部署一个 Express.js 基础应用
flask-starter - 快速部署一个 Flask 基础应用
fullstack - 快速部署一个 Full Stack 应用，vuejs + express + postgres
koa-starter - 快速部署一个 Koa.js 基础应用
laravel-starter - 快速部署一个 Laravel 基础应用
nextjs-starter - 快速部署一个 nextjs 应用
```

```
Serverless: 请输入项目名称 demo
Serverless: 正在安装 express-starter 应用...

express-starter > Created

demo 项目已成功创建！
```

选择**立即部署**，将已经初始化好的项目快速部署腾讯云平台：

```
Serverless: 是否希望立即将该项目部署到云端？ Yes
xxxxxxxx
x QR x
x CODE x
xxxxxxxx
请使用微信扫码上方二维码或者点击下方链接登录
https://slogin.qcloud.com/XKYUcbaK
登录成功！
serverless ↗framework
Action: "deploy" - Stage: "dev" - App: "demo1" - Instance: "expressDemo"
region: ap-guangzhou
apigw:
serviceId: service-xxxxx
subDomain: service-xxxxx.gz.apigw.tencentcs.com
environment: release
url: https://service-xxxxx.gz.apigw.tencentcs.com/release/
scf:
functionName: express_component
runtime: Nodejs10.15
namespace: default
lastVersion: $LATEST
traffic: 1
26s > expressDemo > Success
```

部署完毕后，单击命令行输出的 API 网关链接，即可快速访问已部署好的 Web 框架应用：

查看部署信息

如果希望再次查看应用的部署状态和资源，可以进入到部署成功的文件夹，运行如下命令，查看对应信息：

```
cd demo #进入项目目录，此处请改为您的项目目录名称
sls info
```

说明：

sls 是 serverless 命令的简写。

查看目录结构

在初始化的项目目录下，可以看到一个 Express 项目的最基本结构：

```
.
├── serverless.yml # 配置文件
├── index.js # 入口函数
├── package.json # 项目依赖
└── .env # 环境变量文件
```

- `serverless.yml` 配置文件实现了函数基本信息的快速配置，函数控制台支持的配置项都支持在 `yml` 文件里配置（查看 [云函数的全量配置信息](#)）。
- `index.js` 为项目的入口函数，此处为 `helloworld` 模版。
- `package` 为项目依赖文件，记录了该 `Node.js` 框架项目需要安装的依赖包。
- `.env` 文件里存放了用户登录的鉴权信息，您也可以在里面配置其它环境变量。

重新部署

在本地项目目录下，您可以对函数模版项目内容与配置文件进行修改，重新安装依赖后，通过以下指令进行重新部署：

```
npm install && sls deploy
```

说明：

如需查看移除过程中的详细信息，可以在 `sls deploy` 后增加 `--debug` 参数进行查看。

持续开发

部署完成后，登录 [Serverless 应用控制台](#)，查看项目部署后输出的基本信息、项目请求次数、项目报错统计等多项监控指标

查看项目部署后输出的基本信息、项目请求次数、项目报错统计等多项监控指标，并实现项目持续开发与部署。

详情请参考 [控制台开发文档](#)。

Serverless Framework 支持通过不同指令，帮助您完成项目的持续开发部署、灰度发布等能力，您也可以结合 [层](#)、[自定义域名](#) 等其它高级能力一起使用，实现应用的高级能力配置。

常见问题

输入 `serverless` 时没有默认弹出中文引导。

解决方案：在 `.env` 文件中增加配置 `SERVERLESS_PLATFORM_VENDOR=tencent` 即可。

在境外网络环境，输入 `sls deploy` 后部署十分缓慢。

解决方案：在 `.env` 文件中增加配置 `GLOBAL_ACCELERATOR_NA=true` 则开启境外加速。

输入 `sls deploy` 后部署报网络错误。

解决方案：在 `.env` 文件中增加以下代理配置。

```
HTTP_PROXY=http://127.0.0.1:12345 #请将'12345'替换为您的代理端口
HTTPS_PROXY=http://127.0.0.1:12345 #请将'12345'替换为您的代理端口
```

控制台开发指南

最近更新时间：2022-07-22 15:22:53

操作场景

针对常用框架组件，您可以通过 [Serverless 应用控制台](#)，快速实现完整的应用开发部署流程。

前提条件

在使用控制台部署前，您需要先完成以下权限配置：

主账号授权

1. 登录 [Serverless 应用控制台](#)，单击[前往授权](#)进入访问管理控制台。
2. 在访问管理控制台的[角色](#)列表页，查看 **SLS_QcsRole** 和 **CODING_QCSRole** 服务角色是否创建成功。

注意：

如果您已经创建过 CODING_QCSRole，请检查角色拥有权限是否完整，该角色需要基本策略列表如下：
QcloudSLSFullAccess、QcloudSSLFullAccess、QcloudAccessForCODINGRole，如有缺少，请手动添加。

3. 确定角色与权限都符合要求后，即可开始使用服务。

子账号授权

如果未开通 **Serverless Framework** 和 **Coding DevOps** 的服务，请先与主账号联系，完成服务开通与角色创建（[操作说明](#)）。

操作步骤

步骤1：创建应用

1. 登录 [Serverless 应用控制台](#)。
2. 单击[新建应用](#)，进入项目创建页面。
3. 根据页面提示，填写应用基本信息。

- 应用名：2 - 63个字符，只能包含小写字母、数字及分隔符“-”，且必须以小写字母开头，数字或小写字母结尾。创建后不可更改。
- 环境：选择 dev、test、prod 任一种方式，也支持自定义环境。
- 地域：与云函数支持地域相同，详情请参考 [地域列表](#)。
- 创建方式：支持 [应用模版](#) 创建和 [导入已有项目](#) 两种方式，您可以根据自己的实际情况，选择相应的创建方案。

说明：

导入已有项目时，部分框架需要做一定简单的改造，请参考相关框架迁移文档，完成项目改造。

4. 单击**创建**，将为您自动部署应用，您可以查看项目的部署日志。

应用模版创建

如果选择模版创建，您可以通过选择控制台提供的项目模版，快速创建一个 web 应用，模版部署时，将为默认您完成以下配置：

1. 新建层（**仅限 Node.js 框架**），并将项目依赖包 node_modules 存放在层中，层的使用请参考 [层管理](#)。
2. 新建 COS 存储桶（**仅限 Next.js、Nuxt.js 框架**），拆分静态资源，将静态资源托管到 COS 桶中。

您还可以在**高级配置**部分，为您的项目进行自定义域名、函数详细配置等更多能力的配置。

说明：

配置自定义域名时，请确保您的域名已在腾讯云备案并配置了 CNAME 解析，详细步骤参考 [自定义域名配置](#)。

导入已有项目

Serverless 控制台支持您通过**代码托管导入**和**文件夹上传**两种方式实现已有项目迁移。

- 代码托管
目前支持 **GitHub**、**GitLab**、**Gitee** 的代码仓库地址，也支持公开的自定义代码库，您可以通过选择应用的触发方式，完成应用的自动更新。
- 文件夹上传
您可以通过上传文件夹的方式直接导入本地项目，对于 Node.js 框架，Serverless Framework 将自动为您创建层，并将依赖包 node_modules 传入层中完成部署。

步骤2：资源管理

在 [Serverless 应用](#) 页面，单击目标应用进入应用详情页，查看项目部署后输出的基本信息、项目请求次数、项目报错统计等多项监控指标，方便您轻松实现项目的管理运维。

步骤3：开发部署

在应用详情页顶部，单击**开发部署**，您可以轻松地实现应用的配置修改与二次部署上传，支持**本地上传**、**代码托管**、**CLI 开发**三种方式。