

Serverless Framework

Serverless 组件

产品文档



腾讯云

【 版权声明 】

©2013–2021 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 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。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。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默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100。

文档目录

Serverless 组件

Components 概述

API 网关组件

对象存储 COS 组件

内容分发网络 CDN 组件

数据库 PostgreSQL 组件

私有网络 VPC 组件

Layer 组件

云原生数据库 TDSQL-C 组件

Serverless 组件

Components 概述

最近更新时间：2020-09-10 17:08:54

Serverless Components 是支持多个云资源编排和组织的场景化解决方案，主要基于客户的具体场景，如 Express 框架支持、网站部署等。Serverless Components 可以有效简化云资源的配置和管理，将网关、COS 和 CAM 等产品联动起来，让客户更多关注场景和业务。

详细介绍可以参考 [Github 上的 Serverless Components 项目](#)。

Serverless Components 优势

• 简便易用

Serverless Components 更多的围绕客户场景进行构建，如网站、博客系统、支付服务、图像处理场景等。通过抽象了底层的基础设施配置信息，开发者可以通过十分简单的配置实现场景。

• 可复用性

Serverless Components 可以通过非常简单的 `serverless.yml` 创建和部署，但同时也支持用十分简单的语法对 JavaScript 库 `serverless.js` 进行扩展编写和复用。

• 秒级部署

大多数 Serverless Components 比传统的配置工具部署快20倍左右，Components 可以通过快速的部署和远端验证，有效减少本地模拟和调试的环节。

Serverless Framework Components 最佳实践

- [@serverless/tencent-scf](#) - 腾讯云云函数组件
- [@serverless/tencent-express](#) - 快速部署基于 Express.js 的后端服务到腾讯云函数的组件
- [@serverless/tencent-website](#) - 快速部署静态网站到腾讯云的组件

Serverless Components 支持列表

当前 Serverless Components 支持丰富的多语言开发框架和应用，具体如下：

基础组件：

- [@serverless/tencent-postgresql](#) - 腾讯云 PG DB Serverless 数据库组件
- [@serverless/tencent-apigateway](#) - 腾讯云 API 网关组件
- [@serverless/tencent-cos](#) - 腾讯云对象存储组件
- [@serverless/tencent-scf](#) - 腾讯云云函数组件
- [@serverless/tencent-cdn](#) - 腾讯云 CDN 组件
- [@serverless/tencent-vpc](#) - 腾讯云 VPC 私有网络组件

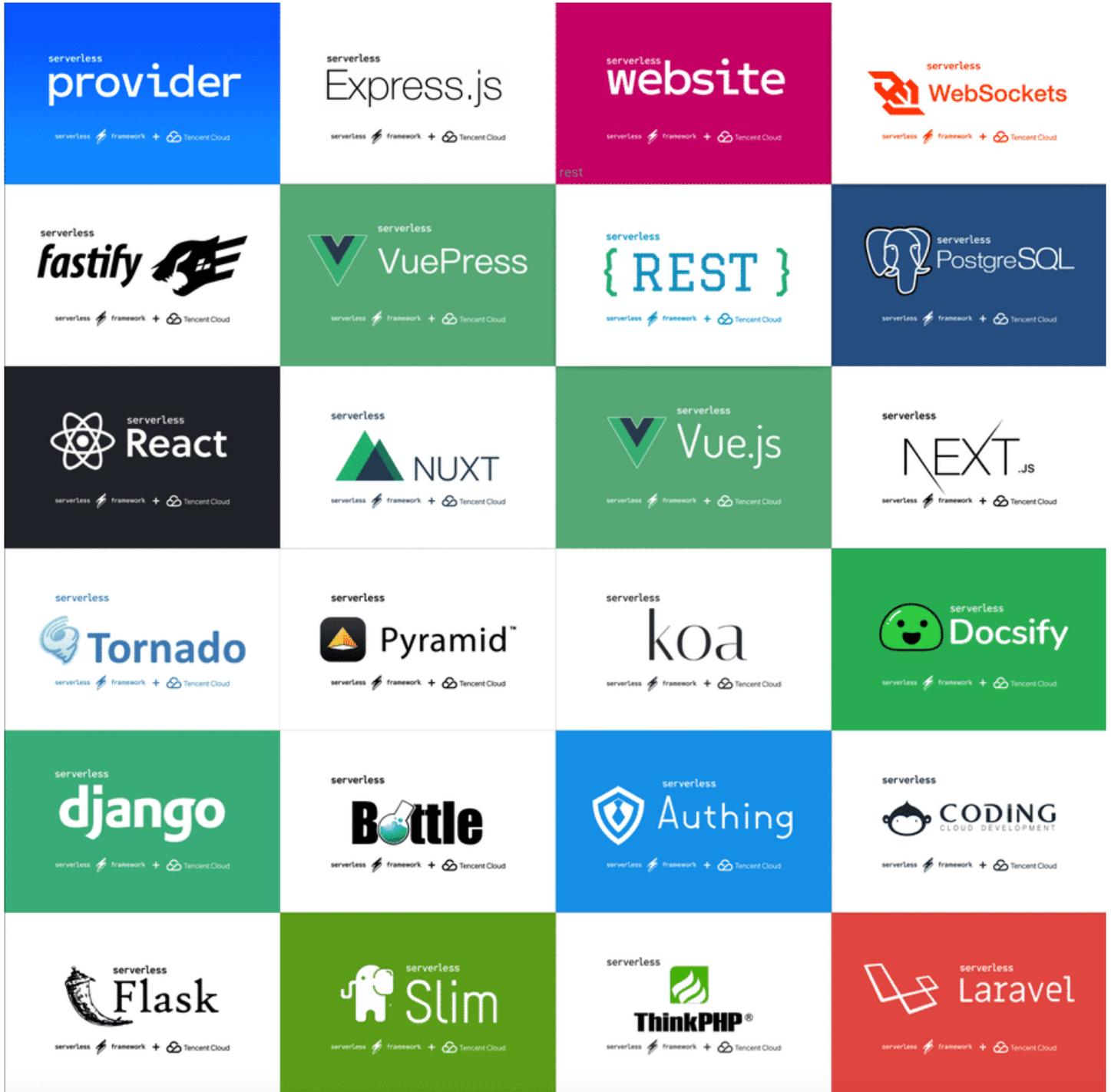
高阶组件:

- [@serverless/tencent-nextjs](#) – 快速部署基于 Next.js 框架到腾讯云函数的组件
- [@serverless/tencent-nuxtjs](#) – 快速部署基于 Nuxt.js 框架到腾讯云函数的组件
- [@serverless/tencent-express](#) – 快速部署基于 Express.js 的后端服务到腾讯云函数的组件
- [@serverless/tencent-egg](#) – 快速部署基于 Egg.js 的后端服务到腾讯云函数的组件
- [@serverless/tencent-koa](#) – 快速部署基于 Koa.js 的后端服务到腾讯云函数的组件
- [@serverless/tencent-flask](#) – 腾讯云 Python Flask RESTful API 组件
- [@serverless/tencent-django](#) – 腾讯云 Python Django RESTful API 组件
- [@serverless/tencent-laravel](#) – 腾讯云 PHP Laravel RESTful API 组件
- [@serverless/tencent-thinkphp](#) – 腾讯云 ThinkPHP RESTful API 组件
- [@serverless/tencent-website](#) – 快速部署静态网站到腾讯云的组件

第三方贡献:

- [@authing/serverless-oidc](#) – 快速部署基于 Authing 的身份认证组件
- [@twn39/tencent-fastify](#) – 快速部署基于 fastify.js 的后端服务到腾讯云函数的组件
- [@twn39/tencent-php-slim](#) – 快速部署基于 Slim PHP 微框架的后端服务到腾讯云函数的组件

此外，所有的 Serverless Components 均可在 [Github 仓库](#) 中查看，查看时请注意切换至最新的v2版本。



API 网关组件

最近更新时间：2021-03-01 15:37:18

操作场景

API 网关组件是 serverless-tencent 组件库中的基础组件之一，您可以通过该组件快速且方便地创建、配置和管理腾讯云的 API 网关产品。

操作步骤

通过 API 网关组件，您可以对一个 API 服务/接口进行完整的创建、配置、部署和删除等操作，支持的命令如下：

安装

通过 npm 安装 Serverless：

```
npm install -g serverless
```

配置

本地创建 serverless.yml 文件：

```
touch serverless.yml
```

在 serverless.yml 中进行如下配置：

```
# serverless.yml
component: apigateway # (必填) 组件名称, 此处为 apigateway
name: apigwDemo # (必填) 实例名称
app: appDemo # (可选) 该 next.js 应用名称
stage: dev # (可选) 用于区分环境信息, 默认值是 dev
inputs:
  region: ap-guangzhou
protocols:
  - http
  - https
serviceName: serverless
environment: release
```

```
endpoints:  
  - path: /  
protocol: HTTP  
method: GET  
apiName: index  
function:  
  functionName: myFunction
```

[查看详细配置文档>>](#)

部署

执行以下命令进行扫码授权部署：

```
sls deploy
```

🔗 说明：

微信扫码授权部署有过期时间，如果想要持久授权，请参考 [账号配置](#)。

移除

执行以下命令移除部署的服务：

```
sls remove
```

账号配置（可选）

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，也可以本地创建 `.env` 文件：

```
touch .env # 腾讯云的配置信息
```

在 `.env` 文件中配置腾讯云的 `SecretId` 和 `SecretKey` 信息并保存：

```
# .env  
TENCENT_SECRET_ID=123  
TENCENT_SECRET_KEY=123
```


对象存储 COS 组件

最近更新时间：2021-01-12 16:58:08

操作场景

对象存储 COS 组件是 serverless-tencent 组件库中的基础组件之一。通过对象存储 COS 组件，可以快速且方便的创建、配置和管理腾讯云的 COS 存储桶。

前提条件

已安装 [Node.js](#)

⚠ 注意：

2020年9月1日起，Serverless 组件不再支持 Node.js10.0 以下版本，请注意升级。

操作步骤

安装

通过 npm 安装 Serverless：

```
npm install -g serverless
```

如果之前您已经安装过 Serverless Framework，可以通过下列命令升级到最新版：

```
npm update -g serverless
```

配置

本地创建 `serverless.yml` 文件，在其中进行如下配置：

```
touch serverless.yml
```

```
# serverless.yml
org: orgDemo
app: appDemo
```

```
stage: dev
component: cos
name: cosDemo
inputs:
bucket: my-bucket
region: ap-guangzhou
```

[查看详细配置文档 >>](#)

部署

执行以下命令进行部署，返回信息如下：

```
[root@iZh8dhuyhmexn3Z demo]# sls deploy
serverless < framework
Action: "deploy" - Stage: "dev" - App: "appDemo" - Instance: "cosDemo"
region: ap-guangzhou
bucket: my-bucket-xxxxxxx
url: http://my-bucket-xxxxxxx.cos.ap-guangzhou.myqcloud.com
Full details: https://serverless.cloud.tencent.com/instances/appDemo%3Adev%3AcosDemo
3s > cosDemo > Success
```

说明：

微信扫码授权部署有过期时间，如果想要持久授权，请参考 [账号配置](#)。

移除

执行 `sls remove` 命令移除部署的存储桶，返回信息如下：

```
[root@iZh8dhuyhmexn3Z demo]# sls remove
serverless < framework
Action: "remove" - Stage: "dev" - App: "appDemo" - Instance: "cosDemo"
3s > cosDemo > Success
```

账号配置（可选）

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，也可以本地创建 `.env` 文件：

```
touch .env # 腾讯云的配置信息
```

在 `.env` 文件中配置腾讯云的 `SecretId` 和 `SecretKey` 信息并保存。

```
# .env
TENCENT_SECRET_ID=123
TENCENT_SECRET_KEY=123
```

🔗 说明：

- 如果没有腾讯云账号，请先 [注册新账号](#)。
- 如果已有腾讯云账号，可以在 [API 密钥管理](#) 中获取 `SecretId` 和 `SecretKey`。

内容分发网络 CDN 组件

最近更新时间：2021-03-01 15:34:53

操作场景

腾讯云 CDN 组件是 serverless-tencent 组件库中的基础组件之一。通过 CDN 组件，您可以快速方便的创建、配置和管理腾讯云的 CDN 产品。

前提条件

- 已安装 [Node.js](#) (Node.js 版本需不低于 8.6，建议使用 Node.js10.0 及以上版本)。
- 需要开通 [内容分发网络](#) 服务。

操作步骤

安装

通过 npm 安装 Serverless:

```
npm install -g serverless
```

如果之前您已经安装过 Serverless Framework，可以通过下列命令升级到最新版:

```
npm update -g serverless
```

配置

本地创建 serverless.yml 文件:

```
touch serverless.yml
```

在 serverless.yml 中进行如下配置:

```
# serverless.yml
component: cdn
name: cdnDemo
app: appDemo
```

```
stage: dev
inputs:
area: overseas
domain: mysite.com #域名
origin:
origins:
- xxx.cos.ap-guangzhou.myqcloud.com #源站, 可以是域名或 IP
originType: cos
originPullProtocol: https
serviceType: web
forceRedirect:
switch: on
redirectType: https
redirectStatusCode: 301
https:
switch: on
http2: on
certInfo:
certId: 'abc'
# certificate: 'xxx'
# privateKey: 'xxx'
```

[查看详细配置文档 >>](#)

部署

执行以下命令进行扫码授权部署:

```
sls deploy
```

🔗 说明:

- 请确认您已经开通 [内容分发网络](#) 服务。
- 微信扫码授权部署有过期时间, 如果想要持久授权, 请参考 [账号配置](#)。

移除

执行以下命令移除部署的 CDN 配置:

```
sls remove
```

账号配置（可选）

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，也可以本地创建 `.env` 文件：

```
touch .env # 腾讯云的配置信息
```

在 `.env` 文件中配置腾讯云的 `SecretId` 和 `SecretKey` 信息并保存：

```
# .env
TENCENT_SECRET_ID=123
TENCENT_SECRET_KEY=123
```

🔗 说明：

- 如果没有腾讯云账号，请先 [注册新账号](#)。
- 如果已有腾讯云账号，可以在 [API 密钥管理](#) 中获取 `SecretId` 和 `SecretKey`。

数据库 PostgreSQL 组件

最近更新时间：2021-03-01 15:33:10

操作场景

PostgreSQL for Serverless (ServerlessDB) 是一款基于 PostgreSQL 数据库实现的按需分配资源的数据库产品，其数据库将根据您的实际请求数来自动分配资源。PostgreSQL for Serverless 仅需创建实例，即可正常使用，您无需关心数据库实例规格，仅需要在数据库处于活动状态期间按照实际用量进行付费，不需要为数据库的闲时进行付费。详情参考 [ServerlessDB](#) 文档。

通过 PostgreSQL ServerlessDB 组件，您可以快速方便地创建、配置和管理腾讯云的 PostgreSQL 实例。

特性介绍：

- **按需付费** – 按照请求的使用量进行收费，没有请求时无需付费。
- **"0"配置** – 默认配置将由 Serverless 完成。
- **极速部署** – 仅需几秒，创建或更新您的数据库。
- **便捷协作** – 通过云端数据库的状态信息和部署日志，方便的进行多人协作开发。

操作步骤

安装

通过 npm 全局安装 [Serverless CLI](#)：

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

账号配置

本地创建 `.env` 文件：

```
$ touch .env # 腾讯云的配置信息
```

在 `.env` 文件中配置腾讯云的 `SecretId` 和 `SecretKey` 信息并保存：

```
# .env
TENCENT_SECRET_ID=123
TENCENT_SECRET_KEY=123
```


⚠ 注意:

当前 PGSQL for Serverless 仅支持北京三区，广州二区，上海二区三个地域的创建和部署，因此在填写 yaml 中的地域可用区时需要注意填写为正确的地域和对应的 VPC 子网信息。

部署

如果您的账号未 [登录](#) 或 [注册](#) 腾讯云，您可以直接通过微信扫码命令行中的二维码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 deploy
```

移除

通过以下命令移除部署的 DB 实例：

```
$ sls remove
```

最佳实践

部署 PG Serverless DB 之后，您可以参考 [部署支持数据库操作的全栈网站](#) 使用该 DB 实例。

更多组件

您可以在 [Serverless Components repo](#) 中查询更多组件的信息。

私有网络 VPC 组件

最近更新时间：2021-03-01 15:32:45

操作场景

腾讯云 VPC 组件支持通过 `serverless.yml` 配置，快速创建指定名称的私有网络和子网，并输出 VPCID 和 SubnetID，便于配置其他组件所需的网络信息。

操作步骤

安装

通过 npm 安装最新版本的 Serverless Framework:

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

配置

新建一个目录 `vpcDemo`，在 `vpcDemo` 下创建 `serverless.yml` 文件:

```
$ mkdir vpcDemo && cd vpcDemo
$ touch serverless.yml
```

在 `serverless.yml` 中进行如下配置:

```
# serverless.yml
org: orgDemo # (可选) 用于记录组织信息，默认值为您的腾讯云账户 appid.
app: appDemo # (可选) 该VPC应用名称.
stage: dev # (可选) 用于区分环境信息，默认值是 dev.
component: vpc # (必填) 引用 component 的名称，当前用到的是 tencent-vpc 组件.
name: vpcDemo # (必填) 该组件创建的实例名称.
inputs:
  region: ap-guangzhou
  zone: ap-guangzhou-2
  vpcName: serverless
  subnetName: serverless
```

[查看详细配置文档 >>](#)

部署

运行 `sls deploy` 进行部署:

```
$ sls deploy
serverless ↗ framework
Action: "deploy" - Stage: "dev" - App: "appDemo" - Instance: "vpcDemo"
region: ap-guangzhou
zone: ap-guangzhou-2
vpcId: vpc-xxxxxxx
vpcName: serverless
subnetId: subnet-xxxxxxx
subnetName: serverless
3s > vpcDemo > Success
```

如果您的账号未 [登录](#) 或 [注册](#) 腾讯云, 您可以通过微信扫码命令行中的二维码进行授权登录和注册。

🔗 说明:

`sls` 是 `serverless` 命令的简写。

查看信息

运行 `sls info` 进行查看部署成功的信息:

```
$ sls info
serverless ↗ framework
Status: active
Last Action: deploy (5 minutes ago)
Deployments: 2
region: ap-guangzhou
zone: ap-guangzhou-2
vpcId: vpc-xxxxxxx
vpcName: serverless
subnetId: subnet-xxxxxxx
subnetName: serverless
vpcDemo > Info successfully loaded
```

移除

通过以下命令移除部署的 VPC:

```
$ sls remove
serverless < framework
Action: "remove" - Stage: "dev" - App: "appDemo" - Instance: "vpcDemo"
6s > vpcDemo > Success
```

账号配置 (可选)

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, 也可以本地创建 `.env` 文件:

```
$ touch .env # 腾讯云的配置信息
```

在 `.env` 文件中配置腾讯云的 `SecretId` 和 `SecretKey` 信息并保存:

```
# .env
TENCENT_SECRET_ID=123
TENCENT_SECRET_KEY=123
```

🔍 说明:

- 如果没有腾讯云账号, 请先 [注册新账号](#)。
- 如果已有腾讯云账号, 可以在 [API 密钥管理](#) 中获取 `SecretId` 和 `SecretKey`。

Layer 组件

最近更新时间：2021-02-23 14:44:42

操作场景

Layer 组件是 serverless-tencent 组件库中的基础组件之一。您可以通过该组件快速且方便地创建、配置和管理腾讯云函数的层资源。

前提条件

已安装 [Node.js](#)

⚠ 注意：

2020年9月1日起，Serverless 组件不再支持 Node.js10.0 以下版本，请注意升级。

操作步骤

安装

通过 npm 安装 Serverless：

```
npm install -g serverless
```

如果之前您已经安装过 Serverless Framework，可以通过下列命令升级到最新版：

```
npm update -g serverless
```

配置

本地创建 `serverless.yml` 文件，在其中进行如下配置：

```
touch serverless.yml
```

```
# serverless.yml
component: layer
name: layerDemo
```

```
app: appDemo
stage: dev
inputs:
region: ap-guangzhou
name: layerDemo
src: ./layer-folder
runtimes:
- Nodejs10.15
```

[查看详细配置文档 >>](#)

部署

执行以下命令进行扫码授权部署：

```
sls deploy
```

🔗 说明：

微信扫码授权部署有过期时间，如果想要持久授权，请参考 [账号配置](#)。

移除

执行以下命令移除部署的服务：

```
sls remove
```

账号配置（可选）

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，也可以本地创建 `.env` 文件：

```
touch .env # 腾讯云的配置信息
```

在 `.env` 文件中配置腾讯云的 `SecretId` 和 `SecretKey` 信息并保存。

```
# .env
TENCENT_SECRET_ID=123
TENCENT_SECRET_KEY=123
```

🔍 说明:

- 如果没有腾讯云账号，请先 [注册新账号](#)。
- 如果已有腾讯云账号，可以在 [API 密钥管理](#) 中获取 SecretId 和 SecretKey。

云原生数据库 TDSQL-C 组件

最近更新时间：2021-02-23 14:45:06

操作场景

该教程指导您通过 Serverless Framework 组件，快速创建一个 TDSQL-C Serverless 数据库实例。

操作步骤

1. 安装 Serverless Framework

通过 npm 全局安装最新版本的 Serverless Framework：

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

2. 创建新目录

创建并进入一个全新目录：

```
$ mkdir tencent-tdsqlc && cd tencent-tdsqlc
```

3. 配置文件

在新目录下创建 serverless.yml 文件：

```
$ touch serverless.yml
```

在 serverless.yml 文件中进行如下配置 ([查看全量配置](#))：

```
# serverless.yml
component: cynosdb
name: cynosdbDemo
inputs:
  region: ap-guangzhou
  zone: ap-guangzhou-4
vpcConfig:
```



```
# serverless.yml
app: appDemo
stage: dev
component: cynosdb
name: cynosdbDemo

inputs:
  region: ap-guangzhou
  zone: ap-guangzhou-4
  enablePublicAccess: true
vpcConfig:
  vpcId: vpc-xxx
  subnetId: subnet-xxx
```

然后重新执行部署：

```
$ sls deploy
```

4.2 重置密码

组件只支持重置 root 用户密码。例如，需要将密码重置为 123456@abc 只需运行如下命令：

```
$ sls resetpwd --inputs adminPassword=123456@abc
```

5. 查看状态

在 serverless.yml 文件所在的目录下，通过如下命令查看部署状态：

```
$ sls info
```

账号配置（可选）

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，也可以本地创建 .env 文件：

```
$ touch .env # 腾讯云的配置信息
```

在 .env 文件中配置腾讯云的 SecretId 和 SecretKey 信息并保存：

```
# .env
TENCENT_SECRET_ID=123
TENCENT_SECRET_KEY=123
```

🔗 说明:

- 如果没有腾讯云账号，可以在此 [注册新账号](#)。
- 如果已有腾讯云账号，可以在 [API 密钥管理](#) 中获取 SecretId 和 SecretKey。