

Serverless 应用中心

Serverless 组件

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

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

私有网络 VPC 组件

Layer 组件

数据库 PostgreSQL 组件

Serverless 组件

Components 概述

最近更新时间：2020-09-11 14:24:08

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

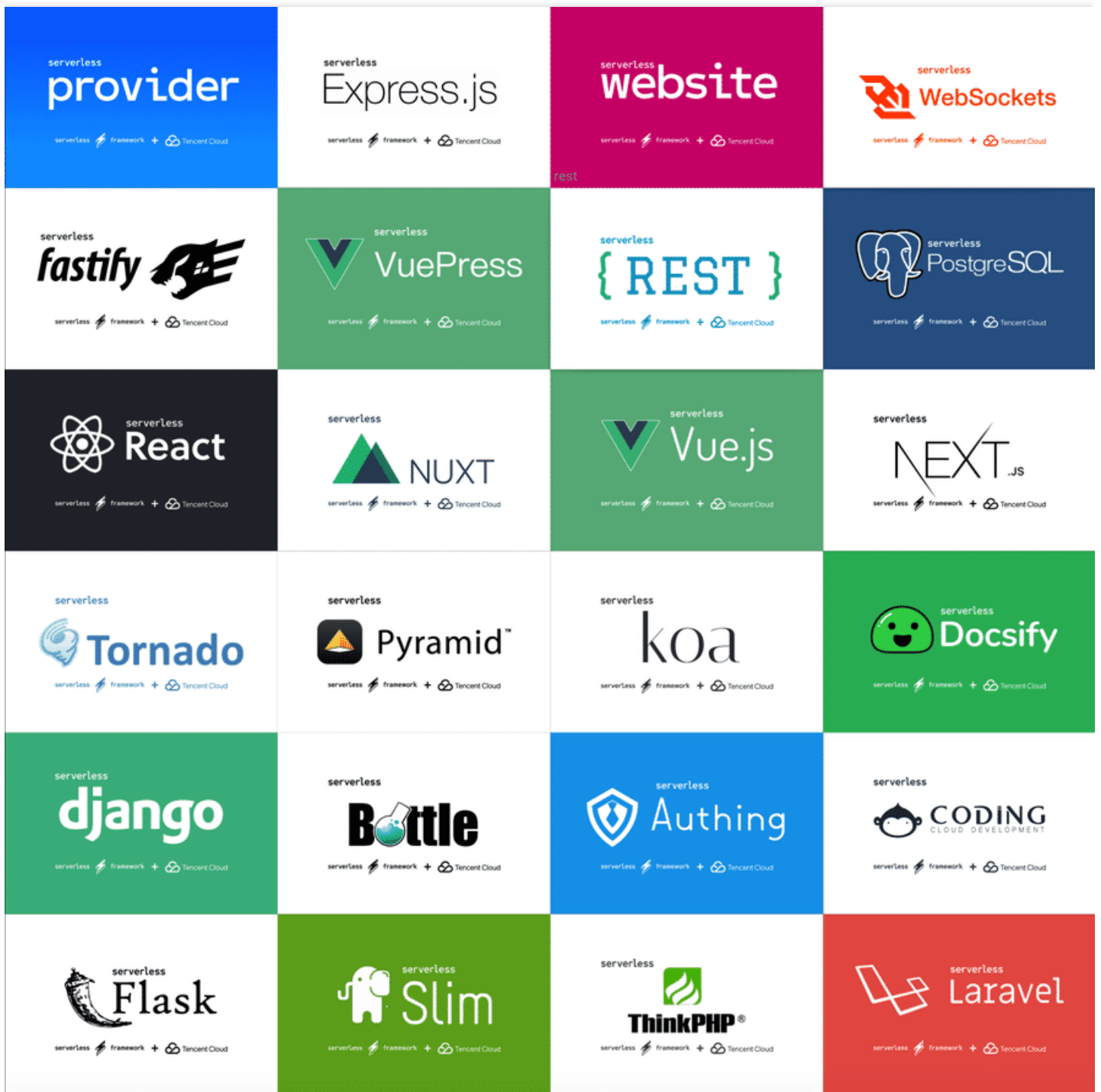
高阶组件：

- [@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
- [@tw39/tencent-fastify](#) - 快速部署基于 fastify.js 的后端服务到腾讯云函数的组件
- [@tw39/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：2021-03-05 15:46:21

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
```

① 说明：

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
```

① 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`。

对象存储 COS 组件

最近更新时间：2020-07-14 11:41:58

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](#)（Node.js 版本需不低于 8.6，建议使用 Node.js10.0 及以上版本）

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
```

① 说明：

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
```

① 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：2021-03-05 15:42:14

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。
- 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`。

私有网络 VPC 组件

最近更新时间：2020-07-10 11:47:10

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
```

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
```

① 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：2021-03-05 15:29:01

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：

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。

数据库 PostgreSQL 组件

最近更新时间：2021-03-29 15:51:15

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。

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
```

📄 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。

配置

创建并进入一个全新目录：

```
$ mkdir tencent-postgreSQL && cd tencent-postgreSQL
```

在新目录创建 `serverless.yml` 文件：

```
$ touch serverless.yml
```

在 `serverless.yml` 中进行如下配置：

```
# serverless.yml
component: postgresql # (必填) 引用 component 的名称, 当前用到的是 postgresql 组件
name: serverlessDB # (必填) 该 postgresql 组件创建的实例名称
org: test # (可选) 用于记录组织信息, 默认值为您的腾讯云账户 appid
app: serverlessDB # (可选) 该 sql 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 # 可选 ap-guangzhou, ap-shanghai, ap-beijing
zone: ap-guangzhou-2 # 可选 ap-guangzhou-2, ap-shanghai-2, ap-beijing-3
dbInstanceName: serverlessDB
vpcConfig:
vpcId: vpc-xxxxxxx
subnetId: subnet-xxxxxxx
extranetAccess: false
```

PostgreSQL 组件支持0配置部署，您可以直接通过配置文件中的默认值进行部署。您依然可以修改更多可选配置来进一步开发该项目。

[查看详细配置文档 >>](#)

⚠ 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

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：

① 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。