

Serverless Application Center

Serverless Framework Component

Product Documentation



Copyright Notice

©2013-2019 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



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

Serverless Framework Component

Components Overview

API Gateway Component

COS Component

CDN Component

VPC Component

Layer Component

PostgreSQL Component

Serverless Framework Component Components Overview

Last updated : 2020-09-15 15:16:09

Serverless Components is a scenario-based solution that supports orchestration and organization of multiple cloud resources for different use cases such as Express framework integration and website deployment. It can greatly simplify the configuration and management of cloud resources while interconnecting Tencent Cloud products such as gateways, COS, and CAM, so that you can focus more on your business development.

For more information, please see [Serverless Components on GitHub](#).

Serverless Components Advantages

- **Ease of use**

Serverless Components is built around your scenarios (e.g., websites, blogs, payment systems, and image services). It abstracts underlying infrastructure configuration and enables you to implement your business scenarios with simple configurations.

- **Reusability**

A serverless component can be easily created and deployed through a very simple `serverless.yml` file. Plus, its JavaScript library `serverless.js` supports extension and reuse with simple syntax.

- **Fast deployment**

The deployment of most serverless components is about 20 times faster than traditional configuration tools. They allow rapid deployment and remote verification which help effectively reduce the workload of local emulation and debugging.

Serverless Framework Components Best Practices

- [@serverless/tencent-scf](#) - SCF component
- [@serverless/tencent-express](#) - Component used to quickly deploy Express.js-based backend services in SCF
- [@serverless/tencent-website](#) - Component used to quickly deploy static websites in SCF

Supported Serverless Components

Currently, Serverless Components supports a rich set of development frameworks and applications in various programming languages as detailed below:

Basic components:

- [@serverless/tencent-postgresql](#) - TencentDB for PostgreSQL serverless component
- [@serverless/tencent-apigateway](#) - Tencent Cloud API Gateway component

- [@serverless/tencent-cos](#) - Tencent Cloud COS component
- [@serverless/tencent-scf](#) - Tencent Cloud SCF component
- [@serverless/tencent-cdn](#) - Tencent Cloud CDN component
- [@serverless/tencent-vpc](#) - Tencent Cloud VPC component

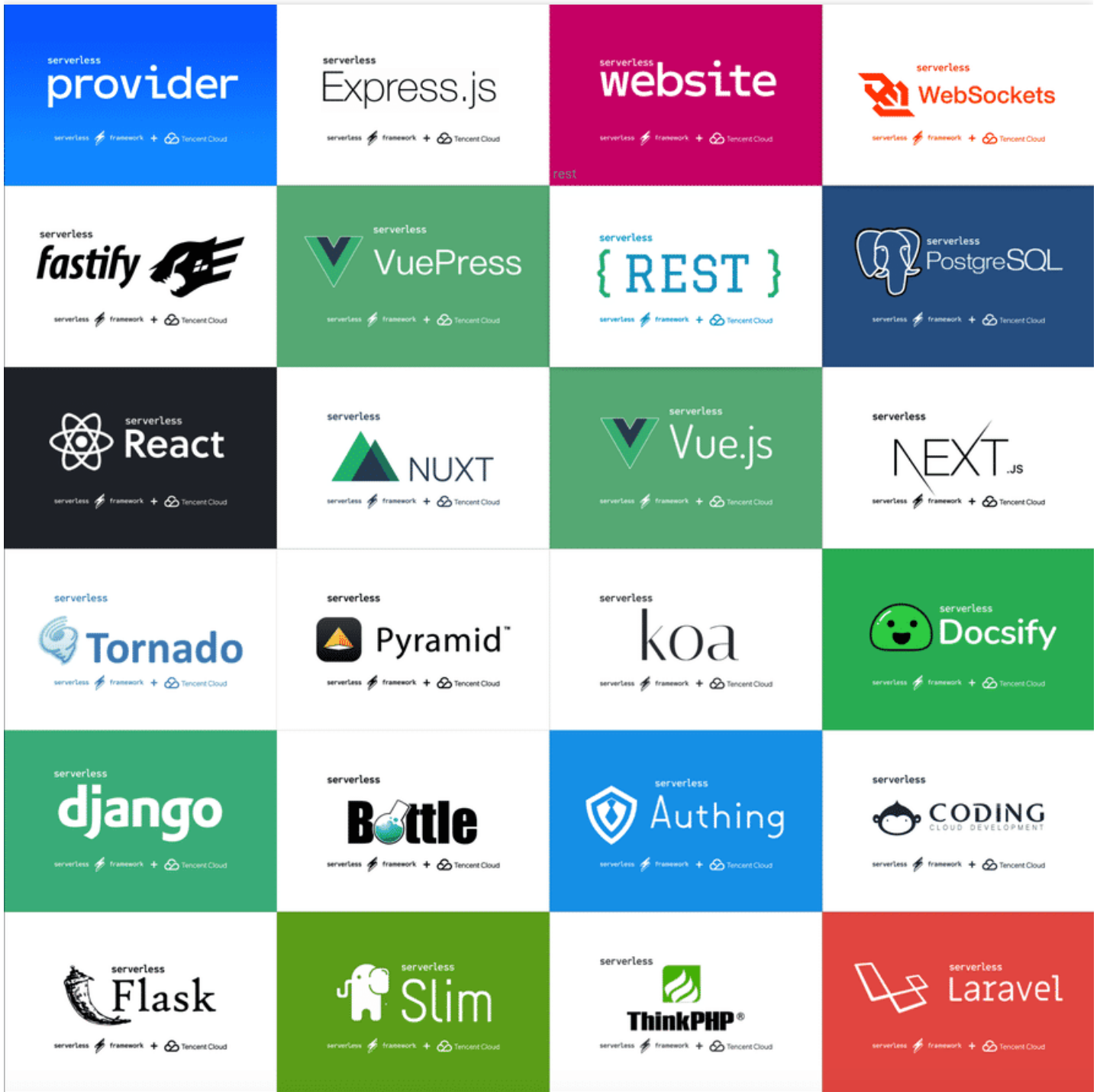
Advanced components:

- [@serverless/tencent-nextjs](#) - Component used to quickly deploy Next.js-based applications in SCF
- [@serverless/tencent-nuxtjs](#) - Component used to quickly deploy Nuxt.js-based applications in SCF
- [@serverless/tencent-express](#) - Component used to quickly deploy Express.js-based backend services in SCF
- [@serverless/tencent-egg](#) - Component used to quickly deploy Egg.js-based backend services in SCF
- [@serverless/tencent-koa](#) - Component used to quickly deploy Koa.js-based backend services in SCF
- [@serverless/tencent-flask](#) - Tencent Cloud Python Flask RESTful API component
- [@serverless/tencent-django](#) - Tencent Cloud Python Django RESTful API component
- [@serverless/tencent-laravel](#) - Tencent Cloud PHP Laravel RESTful API component
- [@serverless/tencent-thinkphp](#) - Tencent Cloud ThinkPHP RESTful API component
- [@serverless/tencent-website](#) - Component used to quickly deploy static websites in SCF

Third-party components:

- [@authing/serverless-oidc](#) - Component used to quickly deploy Authing-based authentication
- [@tw39/tencent-fastify](#) - Component used to quickly deploy Fastify.js-based backend services in SCF
- [@tw39/tencent-php-slim](#) - Component used to quickly deploy backend services based on Slim PHP microframework in SCF

In addition, you can view all Serverless Components in the [GitHub repository](#). Be sure to switch to the **v2** version when viewing the components.



API Gateway Component

Last updated : 2021-03-05 15:45:55

Operation Scenarios

The API Gateway component is one of the basic components in the `serverless-tencent` component library. Through this component, you can create, configure, and manage API gateways with speed and ease.

Directions

Through the API Gateway component, you can perform a complete set of operations on an API service/API, such as creation, configuration, deployment, and deletion. The supported commands are as follows:

Installation

Install Serverless through npm:

```
npm install -g serverless
```

Configuration

Create the `serverless.yml` file locally:

```
touch serverless.yml
```

Configure `serverless.yml` as follows:

```
# serverless.yml

component: apigateway # Component name, which is required. `apigateway` is used i
n this example
name: apigwDemo # Instance name, which is required
app: appDemo # Next.js application name, which is optional
stage: dev # Information for identifying environment, which is optional. The defa
ult value is `dev`

inputs:
  region: ap-guangzhou
  protocols:
    - http
```

```
- https
serviceName: serverless
environment: release
endpoints:
- path: /
protocol: HTTP
method: GET
apiName: index
function:
functionName: myFunction
```

[Detailed Configuration >>](#)

Deployment

Run the following command to deploy by scanning code:

```
sls deploy
```

Note :

To grant persistent permission, please see [Account Configuration](#).

Removal

You can run the following command to remove the deployed service:

```
sls remove
```

Account configuration (optional)

Currently, you can scan a QR code to log in to the CLI by default. If you want to configure persistent environment variables/key information, you can also create a local `.env` file:

```
touch .env # Tencent Cloud configuration information
```

Configure Tencent Cloud's `SecretId` and `SecretKey` information in the `.env` file and save it:

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


Note :

- If you don't have a Tencent Cloud account yet, please [sign up](#) first.
- If you already have a Tencent Cloud account, you can get `SecretId` and `SecretKey` in [API Key Management](#).

COS Component

Last updated : 2020-07-14 11:41:57

Operation Scenarios

The COS component is one of the basic components in the `serverless-tencent` component library. Through this component, you can create, configure, and manage COS buckets with speed and ease.

Prerequisites

You have installed [Node.js](#) (v8.6 or above; v10.0 or above is recommended).

Directions

Installation

Install Serverless through npm:

```
npm install -g serverless
```

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

```
npm update -g serverless
```

Configuration

Create the `serverless.yml` file locally and configure it as follows:

```
touch serverless.yml

# serverless.yml

org: orgDemo
app: appDemo
stage: dev
component: cos
```

```
name: cosDemo

inputs:
bucket: my-bucket
region: ap-guangzhou
```

[Detailed Configuration >>](#)

Deployment

Deploy by running the following command, and the information below will be returned:

```
[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
```

Note :

To grant persistent permission, please see [Account Configuration](#).

Removal

Run the `sls remove` command to remove the deployed bucket, and the following information will be returned:

```
[root@iZh8dhuyhmexn3Z demo]# sls remove

serverless ↗ framework
Action: "remove" - Stage: "dev" - App: "appDemo" - Instance: "cosDemo"

3s > cosDemo > Success
```

Account configuration (optional)

Currently, you can scan a QR code to log in to the CLI by default. If you want to configure persistent environment variables/key information, you can also create a local `.env` file:

```
touch .env # Tencent Cloud configuration information
```

Configure Tencent Cloud's `SecretId` and `SecretKey` information in the `.env` file and save it:

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

Note :

- If you don't have a Tencent Cloud account yet, please [sign up](#) first.
- If you already have a Tencent Cloud account, you can get `SecretId` and `SecretKey` in [API Key Management](#).

CDN Component

Last updated : 2021-03-05 15:41:46

Operation Scenarios

The CDN component is one of the basic components in the `serverless-tencent` component library. Through this component, you can create, configure, and manage CDN services with speed and ease.

Prerequisites

- You have installed [Node.js](#) (v8.6 or above; v10.0 or above is recommended).
- You have activated [CDN](#).

Directions

Installation

Install Serverless through npm:

```
npm install -g serverless
```

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

```
npm update -g serverless
```

Configuration

Create the `serverless.yml` file locally:

```
touch serverless.yml
```

Configure `serverless.yml` as follows:

```
# serverless.yml

component: cdn
name: cdnDemo
```

```
app: appDemo
stage: dev

inputs:
  area: overseas
  domain: mysite.com # Domain name
  origin:
    origins:
      - xxx.cos.ap-guangzhou.myqcloud.com # Origin server, which can be a domain name or an 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'
```

[Detailed Configuration >>](#)

Deployment

Run the following command to deploy by scanning code:

```
sls deploy
```

Note :

- Make sure that you have activated [CDN](#).
- To grant persistent permission, please see [Account Configuration](#).

Removal

Run the following command to remove the deployed CDN configuration:

```
sls remove
```

Account configuration (optional)

Currently, you can scan a QR code to log in to the CLI by default. If you want to configure persistent environment variables/key information, you can also create a local `.env` file:

```
touch .env # Tencent Cloud configuration information
```

Configure Tencent Cloud's `SecretId` and `SecretKey` information in the `.env` file and save it:

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

Note :

- If you don't have a Tencent Cloud account yet, please [sign up](#) first.
- If you already have a Tencent Cloud account, you can get `SecretId` and `SecretKey` in [API Key Management](#).

VPC Component

Last updated : 2020-07-31 14:23:39

Operation Scenarios

Tencent Cloud VPC component supports configuring `serverless.yml` to quickly create VPCs and subnets with specified names and output `VPCID` and `SubnetID` in order to facilitate the configuration of network information required by other components.

Directions

Installation

Install the latest version of Serverless Framework through npm:

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

Configuration

Create a `vpcDemo` directory and create a `serverless.yml` file in it.

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

Configure `serverless.yml` as follows:

```
# serverless.yml
org: orgDemo # Organization information, which is optional. The default value is
the `appid` of your Tencent Cloud account.
app: appDemo # VPC application name, which is optional.
stage: dev # Information for identifying environment, which is optional. The defa
ult value is `dev`.

component: vpc # Name of the imported component, which is required. The `tencent-
vpc` component is used in this example
name: vpcDemo # Name of the instance created by this component, which is require
d.

inputs:
region: ap-guangzhou
```



```
zone: ap-guangzhou-2
vpcName: serverless
subnetName: serverless
```

[Detailed Configuration >>](#)

Deployment

Run `sls deploy` to deploy:

```
$ 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
```

Note :

`sls` is short for the `serverless` command.

Information viewing

Run `sls info` to view the information of successful deployment:

```
$ 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
```

Removal

You can run the following commands to remove the deployed VPC:

```
$ sls remove

serverless ↗ framework
Action: "remove" - Stage: "dev" - App: "appDemo" - Instance: "vpcDemo"

6s > vpcDemo > Success
```

Account configuration (optional)

Currently, you can scan a QR code to log in to the CLI by default. If you want to configure persistent environment variables/key information, you can also create a local `.env` file:

```
$ touch .env # Tencent Cloud configuration information
```

Configure Tencent Cloud's `SecretId` and `SecretKey` information in the `.env` file and save it:

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

Note :

- If you don't have a Tencent Cloud account yet, please [sign up](#) first.
- If you already have a Tencent Cloud account, you can get `SecretId` and `SecretKey` in [API Key Management](#).

Layer Component

Last updated : 2021-03-05 15:28:01

Overview

The Layer component is one of the basic components in the `serverless-tencent` component library. Through this component, you can create, configure, and manage SCF layer resources with speed and ease.

Prerequisites

[Node.js](#) has been installed.

Note :

Starting from September 1, 2020, Serverless components no longer support Node.js versions below 10.0. Please upgrade if needed.

Directions

Installation

Install Serverless through npm:

```
npm install -g serverless
```

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

```
npm update -g serverless
```

Configuration

Create the `serverless.yml` file locally and configure it as follows:

```
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
```

[Detailed Configuration >>](#)

Deployment

Run the following command to deploy by scanning code:

```
sls deploy
```

Note :

The QR code has a validity period. To grant persistent permission, please see [Account Configuration](#).

Removal

You can run the following command to remove the deployed service:

```
sls remove
```

Account configuration (optional)

Currently, you can scan a QR code to log in to the CLI by default. If you want to configure persistent environment variables/key information, you can also create a local `.env` file:

```
touch .env # Tencent Cloud configuration information
```

Configure Tencent Cloud's `SecretId` and `SecretKey` information in the `.env` file and save it:

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

```
TENCENT_SECRET_KEY=123
```

Note :

- If you don't have a Tencent Cloud account yet, please [sign up](#) first.
- If you already have a Tencent Cloud account, you can get `SecretId` and `SecretKey` in [API Key Management](#).

PostgreSQL Component

Last updated : 2021-03-29 15:51:15

Overview

PostgreSQL for Serverless (ServerlessDB) is a database product that allocates resources on demand based on PostgreSQL. Its database automatically allocates resources based on your actual number of requests. With PostgreSQL for Serverless, you can create a database instance for easy use without caring about the instance specifications. You only need to pay for the actual usage when the database is active.

Through the PostgreSQL for Serverless component, you can create, configure, and manage PostgreSQL instances with speed and ease.

Features:

- **Pay-as-you-go billing:** fees are charged based on the request usage, and you don't need to pay anything if there is no request.
- **Zero configuration:** the default configuration will be done by Serverless.
- **Fast deployment:** you can create or update your database in just a few seconds.
- **Convenient collaboration:** the database status information and deployment logs in the cloud make multi-person collaborative development easier.

Directions

Installation

Use npm to install [Serverless CLI](#) globally:

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

Account configuration

Create the `.env` file locally:

```
$ touch .env # Tencent Cloud configuration information
```

Configure Tencent Cloud's `SecretId` and `SecretKey` information in the `.env` file and save it:

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

Note :

- If you don't have a Tencent Cloud account yet, please [sign up](#) first.
- If you already have a Tencent Cloud account, you can get `SecretId` and `SecretKey` in [API Key Management](#).

Configuration

Create a directory and enter it:

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

Create a `serverless.yml` file in a new directory:

```
$ touch serverless.yml
```

Configure `serverless.yml` as follows:

```
# serverless.yml
component: postgresql # Name of the imported component, which is required. The `postgresql` component is used in this example
name: serverlessDB # Name of the instance created by this component, which is required
org: test # Organization information, which is optional. The default value is the `appid` of your Tencent Cloud account
app: serverlessDB # SQL application name, which is optional
stage: dev # Information for identifying environment, which is optional. The default value is `dev`

inputs:
  region: ap-guangzhou # Valid values: ap-guangzhou, ap-shanghai, ap-beijing
  zone: ap-guangzhou-2 # Valid values: ap-guangzhou-2, ap-shanghai-2, ap-beijing-3
  dbName: serverlessDB
  vpcConfig:
    vpcId: vpc-xxxxxxx
    subnetId: subnet-xxxxxxx
    extranetAccess: false
```

The PostgreSQL component supports "zero" configuration deployment, that is, it can be deployed directly through the default values in the configuration file. Nonetheless, you can also modify more optional configuration items to further customize your project.

[Detailed Configuration >>](#)

Note :

Currently, PostgreSQL for Serverless is available for creation and deployment only in **Beijing Zone 3**, **Guangzhou Zone 2**, and **Shanghai Zone 2**. Therefore, when entering the region and AZ information in the `yaml` file, please be sure to use the correct region and corresponding VPC and subnet information.

Deployment

Deploy by running the `sls` command, and you can add the `--debug` parameter to view the information during the deployment process:

Note :

`sls` is short for the `serverless` command.

```
$ sls deploy
```

Removal

You can run the following commands to remove the deployed database instance:

```
$ sls remove
```

Best Practice

After deploying the PostgreSQL Serverless database, you can refer to [Deploying Full-Stack Website with Vue + Express + PostgreSQL](#) to use this database instance.

More Components

You can view more component information in the repository of [Serverless Components](#).