Serverless 应用中心

Serverless 组件

腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2025 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或95716。

文档目录

Serverless 组件
- 云函数 SCF 组件
- API 网关组件（即将下线）
- 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

版权所有：腾讯云计算（北京）有限责任公司

Serverless 组件
云函数 SCF 组件

最近更新时间：2025-07-01 14:44:31

组件概述

腾讯云 SCF 云函数组件通过使用 Tencent Serverless Cloud Framework，基于云上 Serverless 服务（云函数及触发器等），实现“0”配置，便捷开发，极速部署您的第一个云函数，该组件支持丰富的配置扩展，提供了目前最易用、低成本并且弹性伸缩的云函数的开发、配置及部署能力。

快速入门

前提条件

- 已经安装 Serverless Cloud Framework 。详情请参见 安装 Serverless Cloud Framework 。
- 账号开通 Serverless 相关权限。详情请参见 账号和权限配置 。

操作步骤

创建

- 方式1：按照 快速部署 操作，选择 SCF 项目模板进行创建。
- 方式2：直接使用 `scf init` 命令创建。快速创建一个 nodejs 的 SCF 示例：

```
scf init scf-nodejs
```

> **说明：**
> 目前 SCF 组件支持的 helloworld 模板有：scf-golang、scf-php、scf-python，只需要将命令中的 `scf-nodejs` 更换为前面列出的模板名称，即可快速初始化相应语言模板。

部署

执行以下命令，将会弹出二维码，直接扫码授权进行部署：

```
scf deploy
```

> **说明：**
> 如果鉴权失败，请参见 权限配置 进行授权。

查看

执行以下命令，查看您部署的项目信息：

```
scf info
```

移除

执行以下命令，移除您已经部署的项目：

```
scf remove
```

进阶指导

serverless.yml

执行 `scf deploy` 时，根据 serverless.yml 文件中的配置对云函数资源进行创建或更新。一份简单的 serverless.yml 文件如下：

> **说明：**
> 配置详情请参见 全量配置文档 。

```
#scf组件配置样例

#组件信息
component: scf # （必填）引用 component 的名称，当前用到的是 tencent-scf 组件
name: scfdemo # （必填）创建的实例名称，请修改成您的实例名称

#组件参数
inputs:
  name: ${name}-${stage}-${app} #函数名称
  src: ./  #代码路径
  handler: index.main_handler #入口
  runtime: Nodejs10.15 # 云函数运行时的环境
  region: ap-guangzhou # 云函数所在区域
  events: # 触发器
    - apigw: # 网关触发器
        parameters:
          endpoints:
            - path: /
              method: GET
```

如您使用的是函数 URL，请将 `events` 修改为：

```
events:
  - http:
      parameters:
        netConfig:
          enableIntranet: true
          enableExtranet: true
        qualifier: $DEFAULT
        authType: NONE
```

serverless.yml 文件包含的信息：

组件信息

组件名	是否必选	说明
component	必填	component 的名称，使用 `scf registry` 命令查询您可以引入的组件。
name	必填	创建的实例名称，每个组件在部署时将创建一个实例。

参数信息

inputs 下的参数为组件配置参数。一个简单的 SCF 组件参数配置包含以下内容：

参数名	说明
name	云函数名称。由于云函数名称又是资源 ID，为了保证资源的唯一性，建议采用 `${name}-${stage}-${app}` 变量方式。
src	代码路径。
handler	函数处理方法名称。
runtime	云函数运行环境，目前支持： Python2.7、Python3.6、Nodejs6.10、Nodejs8.9、Nodejs10.15、Nodejs12.16、PHP5、PHP7、Go1、Java8 和 CustomRuntime。
region	云函数所在的区域。
events	触发器。支持的触发器为：timer、apigw（即将下线）、cos、cmq、ckafka。

账号权限

部署实例时需要账号授权去操作具体的云资源，目前可以通过两种方式进行授权：**扫码授权**和**密钥授权**。

版权所有：腾讯云计算（北京）有限责任公司

- **扫码授权**：能快速进行授权部署，但生成的凭证是临时凭证，过期后需要重新扫码。
- **密钥授权**：能够获得永久授权，需要预先配置账号的 SecretId 和 SecretKey 。

配置详情请参见 权限配置 。

开发调试

在 `serverless.yml` 文件所在的目录下运行 `scf dev` 可以实时输出云端日志，每次部署完毕后，对项目进行访问，即可在命令行中实时输出调用日志，便于查看业务情况和排障。Node.js 支持开启开发调试能力，将会对本地代码的改动进行检测和自动上传。详情请参见 开发与调试 。

应用管理

Serverless 部署一个组件实例实质是部署了一个单组件实例的应用。在应用项目开发过程中，一个应用下可能会存在多个组件实例，关于管理组件实例进行应用项目开发的说明，详情请参见 应用管理 。

组件命令

云函数组件提供了组件级的命令。对于 `scf deploy` 命令部署成功的云函数，可以通过以下命令操作云函数。

> **说明：**
> 命令必须在 serverless.yml 同目录下执行。

函数发版本

将函数 my-function 云端的 $LATEST 版本发布为一个固定版本：

```
scf deploy --inputs publish=true  function=my-function
```

函数新增版本

将函数 my-function 云端的 $LATEST 版本新增一个版本名为 Version1 的函数版本：

```
 scf deploy --inputs publish=true qualifier=Version1 function=my-
function
```

触发函数

云端调用 functionName 函数，并传递 JSON 参数 `{"weights":{"2":0.1}}` ：

```
scf invoke  --inputs function=functionName  clientContext='{"weights":
{"2":0.1}}'
```

配置详情

全部配置

```
# serverless.yml

#应用组织信息
app: '' # 应用名称。留空则默认取当前组件的实例名称为app名称。
stage: '' # 环境名称。默认值是 dev。建议使用${env.STAGE}变量定义环境名称

#组件信息
component: scf # （必选）组件名称，在该实例中为scf
name: scfdemo # （必选）组件实例名称。

#组件参数配置
inputs:
  name: scfdemo # 云函数名称，默认为 ${name}-${stage}-${app}
  namespace: default
  role: exRole # 云函数执行角色
  # 1. 默认写法，新建特定命名的 cos bucket 并上传
  src: ./src
  # 2. src 为对象，并且制定忽略上传文件夹 node_modules
  # src:
  #   src: ./code
  #   exclude:
  #     - 'node_modules/**'
  # 3. 指定 bucket name 和文件的方式，直接上传 cos 中的文件部署云函数
  # src:
  #    bucket: tinatest    # bucket name 存储桶名称
  #    object: 'code.zip'       # bucket key 指定存储桶内的文件
  # 4. 指定本地文件到 bucket
  # src:
  #   bucket: tinatest    # bucket name
  #   src:          # 指定本地路径
  type: event # 函数类型，默认为 event（事件类型），web（web类型）
  handler: index.main_handler #入口（函数类型为事件类型时生效）
  entryFile: app.js #入口文件名（代码中无scf_bootstrap文件，且函数类型为web类型
时生效）
  runtime: Nodejs10.15 # 运行环境 默认 Nodejs10.15
  region: ap-guangzhou # 函数所在区域
  description: This is a function in ${app} application.
  memorySize: 128 # 内存大小，单位MB
  timeout: 20 # 函数执行超时时间，单位秒
```

版权所有：腾讯云计算（北京）有限责任公司

```
  initTimeout: 3 # 初始化超时时间，单位秒
  environment: #  环境变量
    variables: #  环境变量对象
      TEST: value
  publicAccess: true # 是否开启公网访问
  installDependency: false # 是否在线安装依赖
  vpcConfig: # 私有网络配置
    vpcId: vpc-xxx # 私有网络的Id
    subnetId: subnet-xxx # 子网ID
  instanceConcurrencyConfig: # 单实例多并发配置,仅web函数支持开启
    enable: true #是否开启
    dynamicEnabled: false #是否开启智能动态并发,true: 动态并发,false:表示静态并
发
    maxConcurrency: 1 #最大并发数，dynamicEnabled是false时，可进行设置
  aliasName: $DEFAULT   #别名及流量灰度配置，需发布或使用的函数别名
  aliasDescription: 'cli create alisa description'     #别名描述，非必填
  aliasFunctionVersion: $LATEST     # 别名下配置的版本1
  additionalVersionWeights:  #别名下配置的版本2及权重，自动为版本1分配剩余权重
    - version: 12
      weight: 0.4
  publish:  false #部署的同时是否发布新版本，true:发布；false:不发布
  publishDescription: 'cli publish version' #版本描述
  protocolType: 'WS'  #当前支持 WebSockets 协议，值为 WS，只有 type:web 时此
配置生效。
  protocolParams:
    wsParams:
      idleTimeOut: 60 #空闲超时时间， 单位秒，默认15s。可配置范围1-1800s
  cfs: # cfs配置
    - cfsId: cfs-123
      mountInsId: cfs-123
      localMountDir: /mnt/
      remoteMountDir: /
  deadLetter: # 死信队列配置
    type: deadLetterType
    name: deadLetterName
    filterType: deadLetterFilterType
  layers: #layer配置
    - name: scfLayer #  layer名称
      version: 1 #  版本
  cls: # 函数日志
    logsetId: ClsLogsetId
    topicId: ClsTopicId
  eip: false # 是否开启固定IP
```

```
  asyncRunEnable: false # 是否启用异步执行（长时间运行）
  traceEnable: false # 是否状态追踪
  msgTTL: 21600 # 消息保留时间，单位秒
  retryNum: 2 # 重试次数
  tags: #标签配置
    key1: value1
    key2: value2 # tags 的key value
  ignoreTriggers: false # 是否忽略触发器部署
  image: # 镜像配置
    registryId: tcr-xxx # 容器镜像服务名称，企业版必须
    imageType: personal # 镜像类型：personal - 个人，enterprise - 企业版，
public - 镜像模板
    imageUrl: ccr.ccs.tencentyun.com/sls-
scf/nodejs_test:latest@sha256:xxx
    command: node index.js # 容器启动命令
    args: test # 容器启动参数
    containerImageAccelerate: true # 是否开启镜像加速
    imagePort: 9000 #-1表示 job镜像， 9000 表示webServer镜像。不传或为空时，默
认取9000
  events: # 触发器
    - clb:
        parameters:
          loadBalanceId: string
          domain: string
          protocol: string
          port: string
          url: string
          weight: string
    - timer: # 定时触发器
        parameters:
          name: timer # 触发器名称，默认timer-${name}-${stage}，在YAML中指定
不同的timer name，可创建多个定时触发器
          qualifier: $DEFAULT # 别名配置
          cronExpression: '*/5 * * * * * *' # 每5秒触发一次
          enable: true
          argument: argument # 额外的参数
    - http:#函数URL
        parameters:
          netConfig:
            enableIntranet: true
            enableExtranet: true
          qualifier: $DEFAULT
          authType: NONE
```

```
    - apigw: # api网关触发器，已有apigw服务，配置触发器（即将下线）
        parameters:
          serviceName: serverless
          serviceId: service-8dsikiq6
          protocols:
            - http
          netTypes:
            - OUTER
          description: the serverless service
          environment: release
          endpoints:
            - path: /users
              method: POST
            - path: /test/{abc}/{cde}
              apiId: api-id
              apiName: index
              method: GET
              description: Serverless REST API
              enableCORS: true
              responseType: HTML
              serviceTimeout: 10
              isBase64Encoded: false
              isBase64Trigger: false
              base64EncodedTriggerRules:
                - name: Accept
                  value:
                    - image/jpeg
                - name: Content_Type
                  value:
                    - image/jpeg
              param:
                - name: abc
                  position: PATH
                  required: true
                  type: string
                  defaultValue: abc
                  desc: mytest
                - name: cde
                  position: PATH
                  required: true
                  type: string
                  defaultValue: abc
                  desc: mytest
```

```
            function:
              isIntegratedResponse: true
              functionQualifier: $DEFAULT
            usagePlan:
              usagePlanId: 1111
              usagePlanName: slscmp
              usagePlanDesc: sls create
              maxRequestNum: 1000
            auth:
              secretName: secret
              secretIds:
                - xxx
    - cos: # cos触发器
        parameters:
          qualifier: $DEFAULT # 别名配置
          bucket: cli-appid.cos.ap-beijing.myqcloud.com
          filter:
            prefix: filterdir/
            suffix: .jpg
          events: 'cos:ObjectCreated:*'
          enable: true
    - cmq: # CMQ Topic 触发器
        parameters:
          qualifier: $DEFAULT # 别名配置
          name: test-topic-queue
          enable: true
          filterType: 1 # 消息过滤类型，1为标签类型，2为路由匹配类型
          filterKey: # 当 filterType 为1时表示消息过滤标签，当 filterType 为2
时表示 Binding Key
            - key1
            - key2
    - ckafka: # ckafka触发器
        parameters:
          qualifier: $DEFAULT # 别名配置
          name: ckafka-xxx
          topic: test
          maxMsgNum: 999
          retry: 10000
          offset: latest
          timeout: 60
          enable: true
    - cls: # cls 触发器
        parameters:
```

```
        qualifier: '$DEFAULT' # 别名配置
        topicId: 'xxx-228b-42f5-aab5-7f740cc2fb11' # 日志主题 ID
        maxWait: 60 # 最长等待时间，单位秒
        enable: true
    - mps: # mps 触发器
        parameters:
        qualifier: '$DEFAULT' # 别名配置
        type: EditMediaTask # 事件类型
        enable: true
```

配置描述

主要的参数如下，详情请参见 创建函数 。

参数名称	必选	类型	默认值	描述
src	是	Src	-	函数代码路径。
type	否	string	`event`	函数类型，支持：event、web。
name	否	string	-	创建的函数名称。云函数名称，字段字符需满足只能包含字母、数字、下划线、连字符，以字母开头，以数字或字母结尾，2-60个字符。
namespace	否	string	`default`	函数命名空间。
handler	否	string	`index.main_handler`	函数处理方法名称，执行方法表明了调用云函数时需要从哪个文件中的哪个函数开始执行（函数类型为 web 时生效）。
entryFile	否	string	-	函数入口文件名，默认根据运行环境指定默认文件名。Nodejs 为 app.js，Python 环境为 app.py，PHP 环境为 hello.php（函数类型为 web 时生效）。
role	否	string	-	云函数绑定的运行角色。
runtime	否	string	`Nodejs10.15`	函数运行环境（web 函数目前仅支持 Nodejs、Python、PHP 三类环境）。目前仅支持 Nodejs6.10，Nodejs8.9，Nodejs10.15，Nodejs12.16，Nodejs14.18，Nodejs16.13，Python2.7，Python3.6，Python3.7，PHP5，PHP7，Go1，Java8，Java11 和

				CustomRuntime。使用 CustomRuntime 部署详情请参见 CustomRuntime。
region	否	string	`ap-guangzhou`	云函数所在区域。详情请参见产品支持的 地域列表 。
description	否	string	-	函数描述，最大支持1000个英文字母、数字、空格、逗号、换行符和英文句号，支持中文。
memorySize	否	number	`128`	函数运行时内存大小，可选范围64、128MB-3072MB，并且以128MB 为阶梯。
timeout	否	number	`3`	函数最长执行时间，单位为秒，可选值范围1-900秒。
initTimeout	否	number	`3`	函数初始化超时时间，单位为秒，可选值范围3-300秒。
eip	否	boolean	`false`	是否固定出口 IP。
publicAccess	否	number	`true`	是否开启公网访问。
environment	否	Environment	-	函数的环境变量。
vpcConfig	否	Vpc	-	函数的私有网络配置。
layers	否	Layer[]	-	云函数绑定的 layer。
deadLetter	否	DeadLetter	-	死信队列配置。
cls	否	Cls	-	函数日志配置。
eip	否	boolean	`false`	固定出口 IP。默认为 false，即不启用。
asyncRunEnable	否	boolean	`false`	是否启用异步执行（长时间运行），默认最大支持12小时，如果配置为 `true` ，cls（函数日志配置）必选。此参数只有在函数创建时才有效。
traceEnable	否	boolean	`false`	是否启用状态追踪，如果要配置为 `true` ，必须配置 `asyncRunEnable` 同时为 `true` 。

msgTTL	否	number	`21600`	异步执行消息保留时间，单位秒，仅 `asyncRunEnable` 为 `true` 时有效。
retryNum	否	number	`2`	重试次数，仅 `asyncRunEnable` 为 `true` 时有效。
installDependency	否	boolean	`false`	是否自动在线安装依赖。
tags	否	-	-	标签设置。可设置多对 key-value 的键值对。
cfs	否	Cfs[]	-	文件系统挂载配置，用于云函数挂载文件系统。
ignoreTriggers	否	boolean	`false`	是否忽略触发器，如果设置为 `true`，`events` 参数将不起作用，组件将只更新函数配置和代码。
events	否	Event[]	-	触发器配置。
image	否	Image	-	镜像配置。
protocolType	否	string	-	HTTP 函数支持的访问协议。当前支持 WebSockets 协议，值为 WS，只有 type:web 时此配置生效。
protocolParams	否	Protocol Params	-	HTTP 函数配置 ProtocolType 访问协议，当前协议可配置的参数，主要是配置空闲超时时间。
provisionedNum	否	number	-	预置并发数量。需注意所有版本的预置并发数总和存在上限限制。当前的上限是函数最大并发配额为100。
qualifier	否	number	-	函数的版本号，请注意：$LATEST 版本不支持预置并发。

Src

代码目录。

参数名称	必选	类型	默认值	描述
src	是	string	-	代码路径。与 object 不能同时存在。
exclude	否	string[]	-	不包含的文件或路径，遵守 glob 语法 。

bucket	否	string	-	存储桶名称，配置前需确认桶是否存在。
object	否	string	-	部署的代码在存储桶中的路径。

> **注意：**
> 如果配置了 `src`，表示部署 `src` 参数指定目录的代码并压缩成 `zip` 后上传到对应的存储桶中；如果配置了 `object`，表示获取对应存储桶中 `object` 对应的代码进行部署。

Environment

环境变量。

参数名称	类型	描述
variables	Array of Variable	环境变量参数，包含多对 key-value 的键值对。

VPC

私有网络。

参数名称	必选	类型	描述
vpcId	否	string	私有网络的 ID
subnetId	否	string	子网的 ID

DeadLetter

死信队列。

名称	必选	类型	描述
Type	是	string	死信队列模式
Name	是	string	死信队列名称
FilterType	否	string	死信队列主题模式的标签形式

Layer

层配置。

参数名称	必选	类型	描述

版权所有：腾讯云计算（北京）有限责任公司

name	是	string	层名称
version	是	number	层版本号

CLS

函数日志。

参数名称	必选	类型	描述
logsetId	否	string	日志集 ID
topicId	否	string	日志主题 ID

CFS

文件系统，使用文件系统必须配置 私有网络，并保证 CFS 文件系统与云函数在同一个私有网络下。

参数名称	必选	类型	描述
cfsId	是	String	文件系统实例 ID
mountInsId	是	String	文件系统挂载点 ID
localMountDir	是	String	本地挂载点
remoteMountDir	是	String	远程挂载点

Event

> ⚠ **注意：**
> 触发器部署逻辑说明
> 当通过 CLI 部署触发器时，如果本地配置文件 (YAML) 与云端的函数配置存在配置项差异，并且该触发器的类型不支持更新操作，系统将采取以下强制操作：
> 1. 自动删除现有的云端触发器。
> 2. 根据本地 YAML 配置文件重新创建一个新的触发器。
>
> 请注意：
> 不支持更新的触发器类型包括： CLS 触发器（日志服务）、CLB 触发器（负载均衡）、COS 触发器（对象存储）、CMQ 触发器（消息队列）。
> 请仔细评估此操作（删除旧触发器）影响，避免影响业务使用。

触发器，触发器配置为数组，按照配置的 name 和 param 创建触发器。支持以下触发器：

触发器类型	描述
timer	定时触发器
cos	COS 触发器
cmq	CMQ 主题订阅触发器
ckafka	CKafka 触发器
cls	CLS 触发器
clb	CLB 触发器
http	函数 URL

> **注意：**
> 对于 API 网关触发器，如果没有配置网关服务 ID（serviceId），则自动创建一个 API 网关服务，对于其他触发器仅执行配置触发器，不涉及服务资源创建。

参数名称	必选	类型	默认值	描述
parameters	是	object	-	根据触发器类型，参考触发器参数表。
name	否	string	触发器类型-${name}-${stage}-${app}	触发器名称。

详情请参见 官方触发器配置描述。

定时触发器

详情请参见 定时触发器说明 。

参数名称	必选	类型	默认值	描述
name	否	string	timer-${name}-${stage}	触发器名称，可自定义不同的定时触发器名称，创建多个定时触发器。
qualifier	否	string	`$DEFAULT`	触发版本，默认为 `$DEFAULT` ，即默认流量。

cronExpression	是	string	-	触发时间，为 Cron 表达式。
argument	否	object	-	附加信息参数。
enable	否	boolean	`false`	触发器是否启用。

COS 触发器

详情请参见 COS 触发器说明 。

参数名称	必选	类型	默认值	描述
qualifier	否	string	`$DEFAULT`	触发版本，默认为 `$DEFAULT` ，即默认流量。
bucket	是	string	-	配置的 COS Bucket，仅支持选择同地域下的 COS 存储桶。
filter	是	CosFilter	-	COS 文件名的过滤规则。
events	是	string	-	详情请参见 COS 的事件类型 。
enable	否	boolean	`false`	触发器是否启用。

CMQ 触发器

> **说明：**
> 添加 CMQ 触发器，需要给 SLS_QcsRole 添加 QcloudCMQFullAccess 策略。

参数名称	必选	类型	默认值	描述
qualifier	否	string	`$DEFAULT`	触发版本，默认为 `$DEFAULT` ，即默认流量。
name	是	string	-	CMQ Topic 主题队列名称。
filterType	否	number	-	消息过滤类型，1为标签类型，2为路由匹配类型。
filterK	否	string	-	当 filterType 为1时表示消息过滤标签，当 filterType

ey		[]		为2时表示 Binding Key。
enable	否	boolean	false	触发器是否启用。

CKafka 触发器

> **说明：**
> 添加 CKafka 触发器，需要给 SLS_QcsRole 添加 QcloudCKafkaFullAccess 策略。

参数名称	必选	类型	默认值	描述
qualifier	否	string	$DEFAULT	触发版本，默认为 $DEFAULT ，即默认流量。
name	是	string	-	配置连接的 CKafka 实例，仅支持选择同地域下的实例。
topic	是	string	-	支持在 CKafka 实例中已经创建的 Topic。
maxMsgNum	是	number	100	5秒内每汇聚 maxMsgNum 条 CKafka 消息，则触发一次函数调用。
offset	是	string	latest	offset 为开始消费 CKafka 消息的位置，目前只能填写 latest 。
retry	是	number	10000	重试次数，函数调用失败时的最大重试次数。
timeout	是	number	60	单次触发的最长等待时间，最大60秒。
enable	否	boolean	false	触发器是否启用。

API 网关触发器

> **说明：**
> 如果配置多个 API 网关触发器，需要配置不同的 serviceName。

参数名称	必选	类型	默认值	描述

environment	否	string	release	发布的环境，填写 release 、test 或 prepub ，不填写默认为 release 。
serviceId	否	string	-	网关 Service ID（不传入则新建一个 Service）。
instanceId	否	string	-	网关实例 ID，填写则使用独享型实例创建 API 网关，否则创建共享型实例（该项只能在创建时指定，创建后无法修改）。
protocols	否	string[]	['http']	前端请求的类型，如 http，https，http 与 https。
netTypes	否	string[]	['OUTER']	网络类型，如 ['OUTER'] ，['INNER'] 与 ['OUTER', 'INNER']
serviceName	否	string	-	网关 API 名称。如果不传递则默认新建一个名称与触发器名称相同的 Apigw API 名称。
description	否	string	-	网关 API 描述。
endpoints	是	object[]	-	详情见 endpoints 参数。

endpoints 参数

详情请参见 创建 API 接口 。

参数名称	必选	类型	默认值	描述
path	是	string	-	API 的前端路径，如 /path 。
method	否	string	-	API 的前端请求方法，如 GET，支持的参数有：["ANY","GET","HEAD","POST","PUT","DELETE"]
apiId	否	string	-	API ID。如果不传递则根据 path 和 method 创建一个，传递了直接忽略 path 和 method 参数。
apiName	否	string	-	API 名称。
description	否	string	-	API 描述。

enableCORS	否	boolean	false	是否需要开启跨域。
responseType	否	string	-	自定义响应配置返回类型，现在只支持 HTML、JSON、TEST、BINARY、XML（此配置仅用于生成 API 文档提示调用者）。
serviceTimeout	否	number	15	API 的后端服务超时时间，单位是秒，取值范围：1-1800 秒。
param	否	Parameter	-	前端参数。
function	否	Function	-	SCF 配置。
usagePlan	否	UsagePlan	-	使用计划。
auth	否	Auth	-	API 密钥配置。
authType	否	string	NONE	`NONE` 或 `APP`。
app	否	App	-	API 绑定 APP 配置。
isBase64Encoded	否	boolean	false	是否开启 Base64 编码，只有后端为 scf 时才会生效。
isBase64Trigger	否	boolean	false	是否开启 Base64 编码的 header 触发，只有后端为 scf 时才会生效。
base64EncodedTriggerRules	否	Base64Rule[]	[]	Header 触发 Base64 编码规则，总规则数不能超过10，只有 `isBase64Trigger` 设置为 `true` 才有效。

Parameter

前端参数。

参数名称	必选	类型	默认值	描述
name	否	string	-	API 的前端参数名称。

position	否	string	-	API 的前端参数位置。当前仅支持 PATH、QUERY、HEADER 。
required	否	boolean	-	API 的前端参数是否必填，true：表示必填，false：表示可选。
type	否	string	-	API 的前端参数类型，如 String、Int 等。
defaultValue	否	string	-	API 的前端参数默认值。
desc	否	string	-	API 的前端参数备注。

Function

SCF 配置。

参数名称	必选	类型	默认值	描述
isIntegratedResponse	否	boolean	`false`	是否启用 SCF 集成响应。
functionQualifier	否	string	`$DEFAULT`	触发器关联的 SCF 版本 。

UsagePlan

使用计划，详情请参见 创建使用计划 。

参数名称	必选	类型	描述
usagePlanId	否	string	用户自定义使用计划 ID。
usagePlanName	否	string	用户自定义的使用计划名称。
usagePlanDesc	否	string	用户自定义的使用计划描述。
maxRequestNum	否	number	请求配额总数，如果为空，将使用-1作为默认值，表示不开启。

Auth

版权所有：腾讯云计算（北京）有限责任公司

API 密钥配置，详情请参见 创建密钥 。

参数名称	必选	类型	描述
secretName	否	string	密钥名称
secretIds	否	string	密钥 ID

APP

API 绑定 APP 配置，详情请参见 应用管理 。

参数名称	必选	类型	描述
name	否	string	用户自定义 APP 名称
id	否	string	APP ID
description	否	string	用户自定义 APP 描述

Base64Rule

Header 触发 Base64 编码规则，总规则数不能超过 10，只有 `isBase64Trigger` 设置为 `true` 时生效。详情请参见 Base64 编码 。

参数名称	类型	描述
name	string	进行编码触发的 header，可选值 "Accept" 和 "Content_Type" 对应实际数据流请求 header 中的 Accept 和 Content-Type。
value	string[]	进行编码触发的 header 的可选值数组，数组元素的字符串最大长度为40，元素可以包括数字，英文字母以及特殊字符，特殊字符的可选值为： . + * - / _

例如 `value` 可以配置为：

```
value:
  - application/zip
```

CLS 触发器

> **说明：**
> 添加 CLS 触发器，需要给 SLS_QcsRole 添加 QcloudCLSFullAccess 策略。

参数名称	必选	类型	默认值	描述
qualifier	否	string	`$DEFAULT`	触发版本，默认为 `$DEFAULT`，即默认流量。
topicId	是	string	-	CLS 日志主题 ID。
maxWait	否	number	`60`	最长等待时间，单位秒。
enable	否	boolean	`false`	触发器是否启用。

MPS 触发器

> **说明：**
> 添加 MPS 触发器，需要给 SLS_QcsRole 添加 QcloudMPSFullAccess 策略。

参数名称	必选	类型	默认值	描述
qualifier	否	string	`$DEFAULT`	触发版本，默认为 `$DEFAULT`，即默认流量。
type	是	string	-	事件类型。 • WorkflowTask：工作流任务 • EditMediaTask：视频编辑任务
enable	否	boolean	`false`	触发器是否启用。

关于 API 网关 Base64 编码

> **注意：**
> 开启 API 网关 Base64 编码的后端必须是云函数。

如果需要开启 API 网关 Base64 编码，必须配置 `isBase64Encoded` 为 `true`，此时每次请求的请求内容都会被 Base64 编码后再传递给云函数。如果想要部分请求 Base64 编码，可以通过配置 `isBase64Trigger` 为 `true`，配置 `base64EncodedTriggerRules` Header 触发规则，此时 API 网关将根据触发规则对请求头进行校验，只有拥有特定 Content-Type 或 Accept 请求头的请求会被 Base64 编码后再传递给云函数，不满足条件的请求将不进行 Base64 编码，直接传递给云函数。详情请参见官方介绍文档 Base64 编码 。

Image

镜像相关配置如下：

参数名称	必选	类型	默认值	描述
imageUrl	是	string	-	镜像版本 URL。
registryId	否	string	-	容器镜像服务实例 ID，使用企业版镜像时必选。
imageType	否	string	`personal`	镜像类型，支持：personal、enterprise、public。
command	否	string	-	容器启动命令，默认使用镜像中的 `Entrypoint` 或者 `CMD` 。
args	否	string	-	容器启动参数，默认使用镜像中的 `CMD` 。

`imageUrl` 拼接格式为 `<仓库地址>:<镜像版本>@<镜像ID(sha256)>` ，示例如下：

```
ccr.ccs.tencentyun.com/sls-scf/nodejs_test:latest@sha256:xxx
```

ProtocolParams

详情请参见 创建函数 。HTTP 函数支持其他访问协议的参数如下：

参数名称	必选	类型	描述
wsParams	否	WSParams	WebSockets 协议支持的参数。

WSParams

HTTP 函数通过 WebSockets 协议访问时的参数如下：

参数名称	必选	类型	描述
idleTimeOut	否	int	空闲超时时间, 单位秒，默认15s。可配置范围1-1800s。

版权所有：腾讯云计算（北京）有限责任公司

API 网关组件（即将下线）

最近更新时间：2025-06-16 16:52:32

> **注意：**
> 由于 API 网关产品计划于2025年6月30日停止服务，云函数 SCF 和 Serverless 应用的 API 网关触发器将进行如下调整。具体内容请参见 云函数 SCF 与 Serverless 应用平台 API 网关触发器下线通知 。
> 原本使用 API 网关组件的用户可以采用函数 URL 组件作为替代方案。如果您在短期内仍需要新建 API 网关组件，请 提交工单 至 API 网关团队申请开白。

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。

前提条件

已完成函数的创建，此处以在广州地域创建好的 `myFuntion` 函数为例。

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。

安装

已经安装 Serverless Cloud Framework 。详情请参见 安装 Serverless Cloud Framework 。

配置

1. 本地创建 `serverless.yml` 文件：

Mac 系统

```
touch serverless.yml
```

Linux 系统

```
touch serverless.yml
```

Windows 系统

```
echo. > serverless.yml
```

2. 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
```

⚠ 注意：

1. 配置中的 `region` 需与前提条件中已创建好的函数地域相同。
2. 配置中的 `functionName` 需与前提条件中创建好的函数名字相同。

查看详细配置文档>>

部署

执行以下命令进行扫码授权部署：

```
scf deploy
```

> **说明：**
> 微信扫码授权部署有过期时间，如果想要持久授权，请参见 账号配置 。
> 如执行部署报错，提示“未找到指定的函数”，请再次确认账号下是否已在相关地域创建好对应的函数。

移除

执行以下命令移除部署的服务：

- 仅删除云函数相关的配置、代码。
- 关联的其他云资源（如 COS、CLS 等），平台均不会关联删除，您可以前往对应产品控制台删除，避免不必要的计费。

```
scf 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。

配置详情

完整配置

```
# 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） 该应用名称
stage: dev #（可选） 用于区分环境信息，默认值为 dev
component: apigateway # （必填） 组件名称，此处为 apigateway
name: apigwDemo # （必填） 实例名称

inputs:
  serviceId: service-8dsikiq6
  region: ap-shanghai
  protocols:
    - http
    - https
  serviceName: serverless
  serviceDesc: the serverless service
  environment: release
  netTypes:
    - OUTER
    - INNER
  customDomains:
    - domain: abc.com
      # 如要添加https，需先行在腾讯云-SSL证书进行认证获取cettificateId
      certificateId: abcdefg
      isForcedHttps: true # 是否强制https，如果为true，必须配置 certificateId
(SSL证书 ID)
      # 如要设置自定义路径映射，请设置为 false
      isDefaultMapping: false
      pathMappingSet:
        - path: /
          environment: release
      protocols:
        - http
        - https
  endpoints:
    # 前端类型: WEBSOCKET, 后端类型: SCF
    - path: /
      method: GET
      protocol: WEBSOCKET
      function:
        # 前端类型为WEBSOCKET且后端为SCF时, transportFunctionName 为是
```

```
      transportFunctionName: myFunction
      registerFunctionName: myFunction
      cleanupFunctionName: myFunction
  # 前端类型: WEBSOCKET, 后端类型: HTTP
  - path: /ws
    protocol: WEBSOCKET
    apiName: 'test-ws'
    method: GET
    serviceType: WEBSOCKET
    serviceConfig:
      url: 'ws://www.test.com'
      path: /
      method: GET
  # 前端类型: HTTP, 后端类型: SCF
  - path: /test/{abc}/{cde}
    apiId: api-id
    apiDesc: Serverless REST API
    method: GET
    enableCORS: true
    responseType: HTML
    serviceTimeout: 10
    isBase64Encoded: false
    isBase64Trigger: false
    base64EncodedTriggerRules:
      - name: Accept
        value:
          - image/jpeg
      - name: Content_Type
        value:
          - image/jpeg
    param:
      - name: abc
        position: PATH
        required: 'TRUE'
        type: string
        defaultValue: abc
        desc: mytest
      - name: cde
        position: PATH
        required: 'TRUE'
        type: string
        defaultValue: abc
        desc: mytest
```

版权所有：腾讯云计算（北京）有限责任公司

```
      function:
        isIntegratedResponse: true
        functionQualifier: $LATEST
        functionName: myFunction
      usagePlan:
        usagePlanId: 1111
        usagePlanName: slscmp
        usagePlanDesc: sls create
        maxRequestNum: -1
        maxRequestNumPreSec: 1000
      auth:
        secretName: secret
        secretIds:
          - xxx
    # 前端类型: HTTP, 后端类型: MOCK
    - path: /mo
      protocol: HTTP
      method: GET
      apiName: 'mock-api'
      serviceType: MOCK
      serviceMockReturnMessage: 'mock response content'
    # 前端类型: HTTP, 后端类型: HTTP
    - path: /rest
      protocol: HTTP
      apiName: 'test-http'
      method: GET
      serviceType: HTTP
      serviceConfig:
        url: 'http://www.test.com'
        path: /test
        method: GET
    # 下面两个为互相关联的 oauth2.0 接口示例
    # 参考文档 https://cloud.tencent.com/document/product/628/38393
    - path: '/oauth'
      protocol: 'HTTP'
      method: 'GET'
      apiName: 'oauthapi'
      authType: 'OAUTH'
      businessType: 'OAUTH'
      serviceType: 'HTTP'
      serviceConfig:
        method: 'GET'
        path: '/check'
```

```
        url: 'http://10.64.47.103:9090'
      oauthConfig:
        loginRedirectUrl: 'http://10.64.47.103:9090/code'
        publicKey: '{"e":"AQAB","kty":"RSA","n":"dkdd"}'
        tokenLocation: 'method.req.header.authorization'
        # // tokenLocation: 'method.req.header.cookie',
    - path: '/oauthwork'
      protocol: 'HTTP'
      method: 'GET'
      apiName: 'business'
      authType: 'OAUTH'
      businessType: 'NORMAL'
      authRelationApi:
        path: '/oauth'
        method: 'GET'
      serviceType: 'MOCK'
      serviceMockReturnMessage: 'helloworld'
```

配置说明

主要参数说明

参数	必选	参数类型	默认值	描述
serviceId	否	string	-	网关服务 ID。
instanceId	否	string	-	网关实例 ID，填写则使用独享型实例创建 API 网关，否则创建共享型实例。（该项只能在创建时指定，创建后无法修改）
region	是	string	`ap-guangzhou`	服务的部署区域。
protocols	是	string[]	`['http']`	服务的前端请求类型，http 和 https。
serviceName	否	string	`serverless`	用户自定义的服务名称。如果该参数未传递，则由系统自动生成一个唯一名称。
netTypes	否	string[]	`['OUTER']`	网络类型列表，用于指定支持的访问类型，INNER 为内网访问，OUTER 为外网访问。

serviceDesc	否	string	–	用户自定义的服务描述说明。
environment	是	string	`release`	服务要发布的环境的名称，支持三种环境: test、prepub、release。
endpoints	是	Endpoint[]	–	–
customDomains	否	CustomDomain[]	`[]`	自定义域名。

Endpoint

API 参数说明：

参数	必选	类型	默认值	描述
apiId	否	string	–	API 的唯一 ID。
protocol	否	string	`HTTP`	指定的前端 API 类型，支持 `HTTP` 、`WEBSOCKET` 。
path	是	string	–	API 路径。
method	是	string	–	请求方法。
serviceType	否	string	`SCF`	指定的后端类型，支持： `SCF` 、`HTTP` 、MOCK。
description	否	string	–	API 描述。
enableCORS	否	boolean	`false`	是否启用跨域访问。 • true：启用。 • false：不启用。
function	是	Function	–	对应的 Serverless 云函数。
usagePlan	否	UsagePlan	–	基于 API 维度的使用计划。
auth	否	SecretAuth	–	API 密钥鉴权设置。
serviceTimeout	否	number	`15`	API 的后端服务超时时间，单位为秒。

responseType	否	string	–	返回类型：`HTML`、`JSON`、`TEST`、`BINARY`、`XML`
param	否	RequestParameter	–	前端请求参数。
serviceConfig	否	ServiceConfig	–	API 的后端服务配置。
serviceMockReturnMessage	否	string	–	Mock 接口类型返回结果，如果 `serviceType` 设置为 `MOCK`，此参数必须。
authType	否	string	`NONE`	鉴权类型，支持：`NONE`(免鉴权)、`SECRET`(密钥对)，`OAUTH`(Oauth2.0)，`APP`(应用鉴权)
app	否	App	`NONE`	API 绑定 APP 配置
businessType	否	string	`NORMAL`	业务类型，支持：`NORMAL`、`OAUTH`
oauthConfig	否	OauthConfig	–	Oauth2.0 鉴权，授业 API 后端配置，当 `authType` 为 `OAUTH`，并且 businessType 为 `OAUTH` 时，此参数必选。
authRelationApi	否	AuthRelationApi	–	Oauth2.0 鉴权，业务 API 关联授业 API 配置，当 `authType` 为 `OAUTH`，并且 businessType 为 `NORMAL` 时，此参数必选。
isBase64Encoded	否	boolean	`false`	是否开启 Base64 编码，只有后端为 scf 时才会生效。
isBase64Trigger	否	boolean	`false`	是否开启 Base64 编码的 header 触发，只有后端为 scf 时才会生效。
base64EncodedTriggerRules	否	Base64Rule[]	[]	Header 触发 Base64 编码规则，总规则数不能超过 10，只有 `isBase64Trigger` 设置为 `true` 才有效。

API 类型补充说明：

前端 API 类型 (参数:protocol)	后端服务类型 (参数:serviceType)
HTTP (默认)	SCF (默认)
	HTTP
	MOCK
WEBSOCKET	SCF (默认)
	WEBSOCKET

Function

关联云函数参数配置：

> **说明：**
> 此时 `serviceType` 必须为 `SCF` 。

参数	必选	类型	默认值	描述
isIntegratedResponse	否	boolean	`false`	是否开启响应集成，当前端类型为 `HTTP` 时生效。
functionQualifier	否	string	`$DEFAULT`	scf 函数版本。
functionName	是	string	-	云函数的名称。
transportFunctionName	否	string	-	传输函数的名称，`protocol` 为 `WEBSOCKET` 时必须。
registerFunctionName	否	string	-	注册函数的名称，`protocol` 为 `WEBSOCKET` 时必须。
cleanupFunctionName	否	string	-	清理函数的名称，`protocol` 为 `WEBSOCKET` 时必须。

ServiceConfig

API 的后端服务配置：

参数	必选	类型	默认值	描述
url	是	string	-	API 的后端服务 url，如果 `serviceType` 是 `HTTP`，则此参数必传。
path	是	string	-	API 的后端服务路径，如果 `serviceType` 是 `HTTP`，则此参数必传。
method	是	string	-	API 的后端服务请求方法，如果 `serviceType` 是 `HTTP`，则此参数必传。

UsagePlan

使用计划参数说明：

参数	必选	类型	默认值	描述
usagePlanId	是	string	-	用户自定义的基于 API 的使用计划 ID。
usagePlanName	是	string	-	用户自定义的基于 API 的使用计划名称。
usagePlanDesc	是	string	-	用户自定义的基于 API 的使用计划描述。
maxRequestNum	是	number	`-1`	允许的请求总数。默认情况下将使用 `-1`，表示禁用。
maxRequestNumPreSec	是	number	`-1`	每秒最大请求数。默认情况下将使用 `-1`，表示禁用。

SecretAuth

密钥鉴权参数说明：

参数	必选	类型	默认值	描述
secretName	是	string	-	密钥名称。
secretIds	否	string[]	-	密钥SecretId。

RequestParameter

前端请求参数说明：

参数	必选	类型	默认值	描述
name	是	string	-	请求参数名称。
position	是	string	-	参数位置，仅支持 `PATH` ， `QUERY` 和 `HEADER` 类型。
type	是	string	-	参数类型，如 String 和 int.
defaultValue	是	string	-	参数默认值。
required	是	boolean	-	参数是否必填，TRUE：表示必填，FALSE：表示可选
desc	是	string	-	参数备注/描述

CustomDomain

自定义域名：

参数	必选	类型	默认值	描述
domain	是	string	-	需要绑定的自定义域名
certificateId	否	string	-	自定义域名的证书，如果设置为 https，则为必需。
isDefaultMapping	否	boolean	`true`	是否使用默认路径映射。 如果要自定义路径映射，请设为 `false`
pathMappingSet	否	PathMap[]	`[]`	自定义路径映射，当 `isDefaultMapping` 为 `false` 时，此参数必须
protocols	否	string[]	`['http']`	绑定自定义域协议类型，支持 http 和 https
isForcedHttps	否	boolean	`false`	是否强制 HTTPS。

PathMap

参数	必选	类型	默认值	描述

path	是	string	-	自定义映射路径
environment	是	string	-	自定义映射环境

OauthConfig

参数	必选	类型	默认值	描述
loginRedirectUrl	是	string	-	重定向地址，用于引导用户登录操作
publicKey	是	string	-	公钥，用于验证用户 token
tokenLocation	是	string	-	token 传递位置，支持: `method.req.header.authorization` 、`method.req.header.cookie`

有关授业 API 的公钥生成，请参见 OAuth2.0 。

AuthRelationApi

Oauth2.0 鉴权，业务 API 关联授业 API 配置，当 `authType` 为 `OAUTH` ，并且 businessType 为 `NORMAL` 时，此参数为必选。

参数	必选	类型	默认值	描述
path	是	string	-	关联授业 API 的请求路径
method	是	string	-	关联授业 API 的请求路径

APP

API 绑定 APP 配置，详情见 应用管理 。

参数名称	必选	类型	描述
name	否	string	用户自定义 APP 名称
id	否	string	APP ID
description	否	string	用户自定义 APP 描述

Base64Rule

Header 触发 Base64 编码规则，总规则数不能超过10，只有 `isBase64Trigger` 设置为 `true` 才有效。

版权所有：腾讯云计算（北京）有限责任公司

参数名称	类型	描述
name	string	进行编码触发的 header，可选值 "Accept" 和 "Content_Type" 对应实际数据流请求 header 中的 Accept 和 Content-Type
value	string[]	进行编码触发的 header 的可选值数组, 数组元素的字符串最大长度为 40，元素可以包括数字，英文字母以及特殊字符，特殊字符的可选值为： . + * - / _

例如 `value` 可以配置为：

```
value:
  - application/zip
```

关于 API 网关 Base64 编码

> **说明：**
> 开启 API 网关 Base64 编码的后端必须是云函数。

如果需要开启 API 网关 Base64 编码，必须配置 `isBase64Encoded` 为 `true`，此时每次请求的请求内容都会被 Base64 编码后再传递给云函数。如果想要部分请求 Base64 编码，可以通过配置 `isBase64Trigger` 为 `true`，配置 `base64EncodedTriggerRules` Header 触发规则，此时 API 网关将根据触发规则对请求头进行校验，只有拥有特定 Content-Type 或 Accept 请求头的请求会被 Base64 编码后再传递给云函数，不满足条件的请求将不进行 Base64 编码，直接传递给云函数。详情见 Base64 编码 。

对象存储 COS 组件

最近更新时间：2025-06-13 15:03:32

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，可以快速且方便地创建、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

已经安装 Serverless Cloud Framework 。详情请参见 安装 Serverless Cloud Framework 。

配置

1. 新建一个 cosDemo 文件夹；

```
mkdir cosDemo
```

2. 在 cosDemo 中创建 `serverless.yml` 文件；

Mac 系统

```
cd cosDemo
touch serverless.yml
```

Linux 系统

```
cd cosDemo
touch serverless.yml
```

Windows 系统

```
cd cosDemo
echo. > serverless.yml
```

3. 在创建好的 `serverless.yml` 文件中，进行以下配置：

```
# 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]# scf deploy

serverless-cloud-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，请参见 账号配置 。

移除

执行以下命令移除部署的服务：

- 仅删除云函数相关的配置、代码。
- 关联的其他云资源（如 COS、CLS 等），平台均不会关联删除，您可以前往对应产品控制台删除，避免不必要的计费。

```
scf 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：
以Linux系统/Mac系统为例，执行以下命令：

```
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。

配置详情

完整配置

```yaml
# serverless.yml

app: appDemo # （否）该应用名称
stage: dev # （否）用于区分环境信息，默认值为 dev

component: cos # （是）组件名称，此处为 cos
name: cos-demo # （是）实例名称

inputs:
  src:
    src: ./
    exclude:
      - .env
  bucket: my-bucket
  targetDir: /
  protocol: https
  region: ap-guangzhou
  replace: false # 部署时会先清理桶内所有文件，谨慎使用！
  website: false
  websiteConfig:
    index: index.html
    error: index.html
    ignoreHtmlExt: false # 是否忽略 html 扩展名，默认 false
    disableErrorStatus: false # 是否禁用错误码，默认 false
  acl:
    permissions: private
    grantRead: id="1234567"
    grantWrite: id="1234567"
    grantFullControl: id="1234567"
  cors:
    - maxAgeSeconds: 0
      allowedMethods:
        - GET
      allowedOrigins:
        - '*'
      allowedHeaders:
        - '*'
      exposeHeaders:
        - ''
```

```
  tags:
    - key: abc
      value: xyz
  policy:
    - Principal:
        qcs:
          - 'qcs::cam::anyone:anyone'
      Effect: 'Allow'
      Action:
        - 'name/cos:HeadBucket'
        - 'name/cos:ListMultipartUploads'
        - 'name/cos:ListParts'
        - 'name/cos:GetObject'
        - 'name/cos:HeadObject'
        - 'name/cos:OptionsObject'
      Resource:
        - qcs::cos:ap-guangzhou:uid/123456789:my-bucket-123456789/*
    - Principal:
        qcs:
          - 'qcs::cam::uin/10023456789:uin/10023456789'
      Effect: 'Deny'
      Action:
        - 'name/cos:*'
      Resource:
        - qcs::cos:ap-guangzhou:uid/123456789:my-bucket-123456789/*
```

配置说明

主要参数说明如下：

参数	必选	类型	默认值	描述
bucket	是	string	-	存储桶名称，如若不添加 AppId 后缀，则系统会自动添加，后缀为大写
region	是	string	-	存储桶所属的区域
src	否	string	-	要上传到存储桶的文件或目录
targetDir	否	string	`/`	要上传到存储桶的目标目录，默认目录是根路径 `/`
protocol	否	string	`https`	访问协议

acl	否	Acl	-	访问控制配置
cors	否	Cors[]	-	跨域资源共享配置
tags	否	Tag[]	-	标签配置
policy	否	Policy	-	策略控制配置
replace	否	boolean	`false`	是否是替换式部署，如果为 `true`，部署时将先删除对应 `bucket` 的所有旧文件。
website	否	boolean	`false`	是否开启静态网站能力
websiteConfig	否	WebsiteConfig	-	静态网站相关配置，只有 `website` 为 `true` 时才生效

Acl

参数	必选	类型	默认值	描述
permissions	否	string	`private`	定义存储桶的访问控制列表(ACL)属性。有关枚举值（如 private、public-read），请参见 ACL 概述 中的“存储桶的预设 ACL”部分。
grantRead	否	string	-	授予读取权限，以 `id ="OwnerUin"` 格式授权对存储桶读取的权限，例如 id="100000000001"
grantWrite	否	string	-	授予写入权限，以 `id ="OwnerUin"` 格式授权对存储桶写入的权限，例如 id="100000000001"
grantFullControl	否	string	-	授予全权控制权限，以 `id ="OwnerUin"` 格式授权对存储桶全权控制的权限，例如 id="100000000001"

Cors

参数	必选	类型	描述
id	是	string	配置规则的 id。
maxAgeSeconds	是	number	设置 OPTIONS 请求得到结果的有效期。

allowedMethods	是	string[]	允许的 HTTP 操作，枚举值：GET，PUT，HEAD，POST，DELETE。
allowedOrigins	是	string[]	允许的访问来源，支持通配符 `*`，格式为：`协议://域名[:端口]`，例如：`http://www.qq.com`。
allowedHeaders	是	string[]	在发送 OPTIONS 请求时告知服务端，接下来的请求可以使用哪些自定义的 HTTP 请求头部，支持通配符 `*`。
exposeHeaders	是	string[]	设置浏览器可以接收到的来自服务器端的自定义头部信息。

Tag

参数	必选	类型	描述
key	是	string	标签的 Key，长度不超过 128 字节，支持英文字母、数字、空格、加号、减号、下划线、等号、点号、冒号、斜线。
value	是	string	标签的 Value，长度不超过 256 字节，支持英文字母、数字、空格、加号、减号、下划线、等号、点号、冒号、斜线。

Policy

详情见 COS API 授权策略使用指引 和 访问策略语言概述。

> **说明：**
> `Principal`、`Effect`、`Action`、`Resource` 均为首字母大写。

WebsiteConfig

静态网站相关配置如下：

参数	必选	类型	描述
index	否	string	`index.html`
error	否	string	`error.html`
disableErrorStatus	否	boolean	`false`
ignoreHtmlExt	否	boolean	`false`

内容分发网络 CDN 组件

最近更新时间：2025-06-13 15:03:32

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

已经安装 Serverless Cloud Framework 。详情请参见 安装 Serverless Cloud Framework 。

配置

1. 新建一个 `cdnDemo` 文件夹。

```
mkdir cdnDemo
```

2. 在 `cdnDemo` 中创建 `serverless.yml` 文件；

Mac 系统

```
cd cdnDemo
touch serverless.yml
```

Linux 系统

```
cd cdnDemo
```

```
touch serverless.yml
```

Windows 系统

```
cd cdnDemo
echo. > serverless.yml
```

3. 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：

```
scf 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：

- 仅删除云函数相关的配置、代码。
- 关联的其他云资源（如 COS、CLS 等），平台均不会关联删除，您可以前往对应产品控制台删除，避免不必要的计费。

```
scf 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。

配置详情

全部配置

```yaml
# serverless.yml

component: cdn
name: cdnDemo
org: orgDemo
app: appDemo
stage: dev

inputs:
  domain: abc.com
  area: overseas
  async: true
  onlyRefresh: false
  serviceType: web
  origin:
    origins:
      - origin.site.com
    originType: cos
    originPullProtocol: https
  https:
    switch: on
    http2: on
    certInfo:
      certId: 'abc'
  refreshCdn:
    urls:
      - https://abc.com
  pushCdn:
    urls:
      - https://abc.com
  cache:
    simpleCache:
      followOrigin: on
      cacheRules:
        - cacheType: all
          cacheContents:
            - '*'
          cacheTime: 1000
  cacheKey:
    fullUrlCache: on
  referer:
    switch: on
    refererRules:
```

```
      - ruleType: all
        rulePaths:
          - '*'
        refererType: blacklist
        allowEmpty: true
        referers:
          - 'qq.baidu.com'
          - '*.baidu.com'
  ipFilter:
    switch: on
    filterType: blacklist
    filters:
      - '1.2.3.4'
      - '2.3.4.5'
  forceRedirect:
    switch: on
    redirectType: https
    redirectStatusCode: 301
```

配置说明

CDN 配置

参数名称	是否必选	默认值	描述
domain	是	-	CDN 域名。
origin	是	-	源站配置，详情见 origin。
area	否	`mainland`	域名加速区域： • mainland：中国境内加速 • overseas：中国境外加速 • global：全球加速 （使用中国境外加速、全球加速时，需要先开通中国境外加速服务）
serviceType	否	`web`	加速域名业务类型。 • web：静态加速 • download：下载加速 • media：流媒体点播加速
async	否	`false`	是否为异步操作，如果为 true，则不会等待 CDN 创建或更新

			成功再返回。
onlyRefresh	否	`false`	是否只刷新预热 CDN，如果为 `true`，那么只进行刷新操作，不会更新 CDN 配置。
refreshCdn	否	-	刷新 CDN 相关配置，详情见 refreshCdn，如果想刷新 CDN，必须在同步部署情况下，也就是 async 必须配置为 false。
pushCdn	否	-	预热 CDN 相关配置，详情见 pushCdn，如果想预热 CDN，必须在同步部署情况下，也就是 async 必须配置为 false。
https	否	-	Https 加速配置，详情见 https。
cacheKey	否	-	节点缓存键配置，详情见 cacheKey。
cache	否	-	缓存过期时间配置，详情见 cache。
referer	否	-	防盗链设置，详情见 referer。
ipFilter	否	-	IP 黑白名单配置，详情见 IpFilter。

注意：

`async` 参数对于配置多个 CDN 域名需求，或者在 CI 流程中时，建议配置成 `true`，不然会导致 serverless cli 执行超时，或者 CI 流程超时。

origin

详情见 文档。

参数名称	是否必选	默认	描述
origins	是	-	主源站列表
originType	是	`domain`	主源站类型。 • domain：域名类型 • cos：对象存储源站
originPullProtocol	是	`http`	回源协议配置。 • http：强制 http 回源 • follow：协议跟随回源 • https：强制 https 回源，https 回源时仅支持源站443端口

serverName	否	-	回主源站时 Host 头部，不填充则默认为加速域名。若接入的是泛域名，则回源 Host 默认为访问时的子域名。
cosPrivateAccess	否	-	originType 为对象存储（COS）时，可以指定是否允许访问私有 bucket。注意：需要先授权 CDN 访问该私有 Bucket 的权限后，才可开启此配置。
backupOrigins	否	-	备源站列表。
backupOriginType	否	`domain`	备源站类型。 • domain：域名类型 • cos：对象存储源站
backupServerName	否	-	回备源站时 Host 头部，不填充则默认为加速域名。若接入的是泛域名，则回源 Host 默认为访问时的子域名。

refreshCdn

参数名称	是否必选	默认	描述
urls	否	[]	需要刷新的 CDN 目录

pushCdn

参数名称	是否必选	默认	描述
urls	否	[]	需要预热的 CDN URLs

更多配置，请参见 新增加速域名 。

数据库 PostgreSQL 组件

最近更新时间：2025-03-28 15:53:52

> **注意：**
> 云数据库 PostgreSQL 的 serverless 实例创建能力将下线，详情请参见 产品公告 。
> Serverless 应用的数据库 PostgreSQL 组件下线计划如下：
> - 2025年1月6日起，所有用户将无法新建组件。
> - 对于半年内没有请求的用户，2025年1月15日起组件将停止服务。
> - 对于半年内有请求的用户，组件的具体下线时间将根据云数据库 PostgreSQL 产品公告 统一通知。
>
> 请根据业务需求选择合适的数据库组件进行迁移，例如云原生数据库 TDSQL-C 组件 、云数据库 MySQL 等，以避免因 PostgreSQL 组件下线对业务造成影响。

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。详情请参见 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。ServerlessDB 的特性介绍如下：

- **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，方便进行多人协作开发。

操作步骤

安装

已经安装 Serverless Cloud Framework 。详情请参见 安装 Serverless Cloud Framework 。

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：

Mac 系统

```
touch serverless.yml
```

Linux 系统

```
touch serverless.yml
```

Windows 系统

```
echo. > 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-shanghai  # 可选 ap-shanghai, ap-beijing
  zone: ap-shanghai-2  # 可选 ap-shanghai-2, ap-beijing-3
  dBInstanceName: serverlessDB
  vpcConfig:
    vpcId: vpc-xxxxxxx
    subnetId: subnet-xxxxxx
  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**，**上海二区**两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

如您的账号未 登录 或 注册 腾讯云，您可以直接通过**微信**扫描命令行中的二维码进行授权登录和注册。
通过 `scf` 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：**
> serverless-cloud-framework 命令的简写为 scf。

```
scf deploy
```

移除

执行以下命令移除部署的服务：

- 仅删除云函数相关的配置、代码。
- 关联的其他云资源（如 COS、CLS 等），平台均不会关联删除，您可以前往对应产品控制台删除，避免不必要的计费。

```
scf remove
```

账号配置（可选）

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。

实践教程

部署 PG Serverless DB 之后，您可以使用该 DB 实例。

配置详情

完整配置

```
# serverless.yml
component: postgresql # （必填）组件名称，此处为 postgresql
name: serverlessDB # （必填）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）该应用名称
stage: dev # （可选）用于区分环境信息，默认值为 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
  projectId: 0
  dBVersion: 10.4
  dBCharset: UTF8
  vpcConfig:
    vpcId: vpc-123
    subnetId: subnet-123
  extranetAccess: false
```

配置说明

主要参数说明如下：

参数	必填/可选	类型	默认值	描述
region	必填	String	-	数据库的所属地区
zone	必填	String	-	数据库所在地区的区域
dBInstanceNam	必填	String	-	数据库实例名称，对于用户必须唯一

e				
dBVersion	可选	string	`10.4`	PostgreSQL 版本号，目前支持：10.4
dBCharset	可选	String	`UTF8`	数据库的字符集编码
projectId	可选	Integer	`0`	项目的 ID
vpcConfig.vpcId	必填	String	-	VPC 的 ID
vpcConfig.subnetId	可选	String	-	Subnet 的 ID
extranetAccess	可选	Boolean	`false`	是否开启 serverlessDB 实例外网访问

注意：
部署时，若出现 “Unavailable Zone or SOLD OUT” 的错误提示，则代表该区域暂不可用或售罄，请您更换至其他地区，并选择该地区下的区域。

私有网络 VPC 组件

最近更新时间：2024-08-23 17:22:11

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

已经安装 Serverless Cloud Framework 。详情请参见 安装 Serverless Cloud Framework 。

配置

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

Mac 系统

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

Linux 系统

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

Windows 系统

```
mkdir vpcDemo && cd vpcDemo
echo. > serverless.yml
```

2. 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 `scf deploy` 命令进行部署：

```
scf deploy
serverless-cloud-framework
Action: "deploy" - Stage: "dev" - App: "appDemo" - Instance: "vpcDemo"

region:     ap-guangzhou
zone:       ap-guangzhou-2
vpcId:      vpc-xxxxxxxx
vpcName:    serverless
subnetId:   subnet-xxxxxxxx
subnetName: serverless


3s › vpcDemo › Success
```

如您的账号未 登录 或 注册 腾讯云，您可以直接通过微信扫描命令行中的二维码进行授权登录和注册。

> **说明：**
> `scf` 是 `serverless-cloud-framework` 命令的简写。

查看信息

运行 `scf info` 命令查看部署成功的信息：

```
scf info

serverless-cloud-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

执行以下命令移除部署的服务：

- 仅删除云函数相关的配置、代码。
- 关联的其他云资源（如 COS、CLS 等），平台均不会关联删除，您可以前往对应产品控制台删除，避免不必要的计费。

```
scf 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。

配置详情

完整配置

```yaml
# serverless.yml

component: vpc # （必填）组件名称，此处为 vpc
name: vpcDemo # （必填）实例名称
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）该应用名称
stage: dev # （可选）用于区分环境信息，默认值为 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
  vpcName: serverless
  subnetName: serverless
  cidrBlock: 10.0.0.0/16
  enableMulticast: 'FALSE'
  enableSubnetBroadcast: 'FALSE'
  dnsServers:
    - 127.0.0.1
  domainName: demo
  tags:
    - Key: City
      Value: guangzhou
  subnetTags:
    - Key: City
      Value: guangzhou
```

配置说明

主要参数说明如下：

参数	必填/可选	类型	默认值	描述
region	必填	String	-	VPC 的所属地区
zone	必填	String	-	VPC 所在地区的区域
vpcName	必填	String	-	VPC 的名称
subnetName	必填	String	-	Subnet 的名称
cidrBlock	可选	String	`10.0.0.0/16`	VPC 和 Subnet 的 IPv4 CIDR，例如：10.0.0.0/16，172.16.0.0/16，192.168.0.0/16
enableMulticast	可选	String	`FALSE`	是否启用 VPC 组播
dnsServers	可选	Array	-	VPC DNS 地址，最大数量为 4，第一个为 master
domainName	可选	String	-	VPC 域名相应的 cvm 域名后缀
tags	可选	Array	-	VPC 绑定的标签键值对，例如：[{"Key": "city", "Value": "shanghai"}]
subnetTags	可选	Array	-	Subnet 绑定的标签键值对，例如：[{"Key": "city", "Value": "shanghai"}]
enableSubnetBroadcast	可选	String	`FALSE`	是否启用子网广播

版权所有：腾讯云计算（北京）有限责任公司

Layer 组件

最近更新时间：2025-06-20 11:26:42

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

已经安装 Serverless Cloud Framework 。详情请参见 安装 Serverless Cloud Framework 。

配置

请按照如下步骤完成配置：
步骤一：在本地创建文件夹，此处以 `layerDemo` 为例；
步骤二：在 `layerDemo` 文件夹内，创建 `serverless.yml` 文件；
步骤三：在 `layerDemo` 文件夹内，创建 `layer-folder` 文件夹；
步骤四：在 `layer-folder` 文件夹内，创建 `index.xx文件` ，并放置您的代码；
完成以上步骤后，再打开步骤二所创建的 `serverless.yml` 文件，并参考如下内容进行配置：

> **注意：**
> `src` 后的文件夹路径需要与步骤三中所创建的文件夹路径一致，否则会导致 src 目录无效。

```
# 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：

```
scf deploy
```

> **说明：**
> 微信扫码授权部署有过期时间，如果想要持久授权，请参见 账号配置 。

移除

执行以下命令移除部署的服务：

- 仅删除云函数相关的配置、代码。
- 关联的其他云资源（如 COS、CLS 等），平台均不会关联删除，您可以前往对应产品控制台删除，避免不必要的计费。

```
scf 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。

配置详情

全部配置

```
# serverless.yml

component: layer
name: layerDemo
org: orgDemo
app: appDemo
stage: dev

inputs:
  name: test
  region: ap-guangzhou
  src: ./node_modules
  # src:
  #   src: ./node_modules
  #   targetDir: /node_modules
  #   exclude:    # 被排除的文件或目录
  #     - .env
  #     - node_modules
  # src:
  #   bucket: layers
  #   object: sls-layer-test-1584524206.zip
  #   exclude:    # 被排除的文件或目录
  #     - .env
  #     - node_modules
  runtimes:
    - Nodejs10.14
  description: test project layer
```

配置描述

参数名称	是否必填	参数类型	默认值	描述

region	是	String	-	地区
name	是	String	-	层名称
src	是	String	-	默认为当前目录，如果是对象，配置参数参考 执行目录 。
runtimes	是	String[]	-	层支持的运行环境
description	否	String	-	描述

执行目录

参数名称	是否必填	参数类型	默认值	描述
src	否	String	-	代码路径。与 object 不能同时存在。
exclude	否	Array of String	-	不包含的文件或路径，遵守 glob 语法 。
bucket	否	String	-	bucket 名称。如果配置了 src，表示部署 src 的代码并压缩成 zip 后上传到 bucket-appid 对应的存储桶中；如果配置了 object，表示获取 bucket-appid 对应存储桶中 object 对应的代码进行部署。
object	否	String	-	部署的代码在存储桶中的路径。

版权所有：腾讯云计算（北京）有限责任公司

云原生数据库 TDSQL-C 组件

最近更新时间：2024-08-28 16:45:21

操作场景

该教程指导您通过 Serverless Cloud 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 Cloud Framework

使用 npm 安装 Serverless Cloud Framework，详情请参见 安装 Serverless Cloud Framework 。

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：

Mac 系统

```
touch serverless.yml
```

Linux 系统

```
touch serverless.yml
```

Windows 系统

```
echo. > 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:
    vpcId: vpc-xxx
    subnetId: subnet-xxx
```

> **注意：**
> 当前仅支持**北京三区、广州四区、上海二区、南京一区**四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填写为正确的地域和对应的 VPC 子网信息。

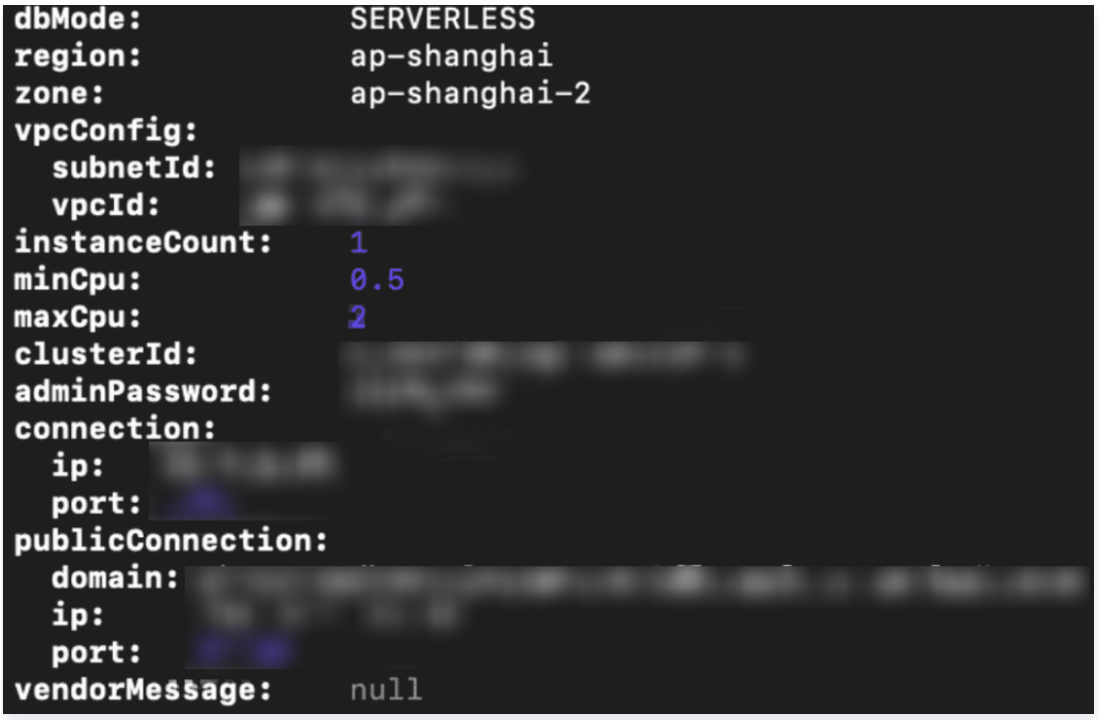
4. 部署

如您的账号未 登录 或 注册 腾讯云，您可以直接通过**微信**扫描命令行中的二维码进行授权登录和注册。
通过 `scf` 命令进行部署，并可以添加 `--debug` 参数查看部署过程中的信息。

```
$ scf deploy
```

> **说明：**
> `scf` 命令是 `serverless` 命令的缩写。

部署完成后，可以在命令行看到创建的数据库实例信息：



4.1 开启外网访问

如果需要数据库实例开启外网访问，只需添加 `enablePublicAccess` 配置为 `true`，如下：

```
# 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：

```
$ scf 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：

```
$ scf 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：

```
$ scf info
```

6. 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。

7. 移除

执行以下命令移除部署的服务：

- 仅删除云函数相关的配置、代码。
- 关联的其他云资源（如 COS、CLS 等），平台均不会关联删除，您可以前往对应产品控制台删除，避免不必要的计费。

```
scf remove
```

配置详情

完整配置

```
app: appDemo # （可选）该应用名称，字符串
stage: dev # （可选）用于区分环境信息，默认值为 dev，字符串
component: cynosdb # （必填）组件名称，此处为 cynosdb
name: cynosdbDemo # （必填）实例名称

inputs:
  region: ap-shanghai # 可选 ap-guangzhou, ap-shanghai, ap-nanjing
  zone: ap-shanghai-2 # 可选 ap-guangzhou-4, ap-shanghai-2, ap-beijing-3,
ap-nanjing-1
  enablePublicAccess: false
  vpcConfig:
    vpcId: vpc-123
    subnetId: subnet-123
  # 如果只创建 serverless 版本，以下两个参数可忽略
  dbMode: SERVERLESS
  payMode: 0
```

配置说明

主要参数说明如下：

参数	必选	类型	默认值	描述
region	是	string	-	数据库的所属地区
zone	是	string	-	数据库所在地区的区域
vpcConfig.vpcId	是	string	-	VPC 的 ID
vpcConfig.subnetId	是	string	-	Subnet 的 ID
enablePublicAccess	否	boolean	`false`	是否开启外网访问
dbMode	否	string	`SERVERLESS`	数据库类型
payMode	number	number	`0`	付费类型

说明：

> Serverless Cynosdb 当前支持可用区为：`ap-guangzhou-4`，`ap-shanghai-2`，`ap-beijing-3`，`ap-nanjing-1`。

dbMode 说明

```
SERVERLESS - serverless 版本
NORMAL     - 正常版本
```

payMode 说明

只有在 `dbMode` 配置为 `NORMAL` 时，才生效。

```
0          - 按量计费
1          - 包年包月，目前只支持购买一个月
```