



# 天翼云 · 函数计算

## 用户使用指南

中国电信股份有限公司云计算分公司

---

# 目 录

---

<b>1 简介</b> .....	<b>4</b>
1.1 函数计算介绍 .....	4
1.2 基本术语 .....	4
1.3 应用场景 .....	5
1.4 技术原理 .....	6
1.5 使用流程 .....	8
1.6 使用限制 .....	9
<b>2 产品功能</b> .....	<b>10</b>
2.1 总览 .....	10
2.2 应用管理 .....	10
2.3 函数管理 .....	10
2.4 其他核心功能 .....	11
<b>3 快速入门</b> .....	<b>12</b>
3.1 开通函数计算服务 .....	12
3.2 使用函数 .....	12
<b>4 最佳实践</b> .....	<b>16</b>
4.1 实践建议 .....	16
<b>5 函数计算管理</b> .....	<b>17</b>
5.1 总览 .....	17
5.2 应用管理 .....	18
5.2.1 创建应用 .....	18
5.2.2 编辑应用 .....	19
5.3 函数管理 .....	20
5.3.1 创建函数 .....	20
5.3.2 查看函数 .....	23
5.3.3 编辑函数 .....	23
5.3.4 触发器管理 .....	24
5.3.5 函数监控指标 .....	25
5.3.6 函数版本 .....	25

---

5.3.7 删除函数 .....	27
<b>6 常见问题.....</b>	<b>28</b>
6.1 函数执行失败是什么原因? .....	28
6.2 函数流量设置为什么不生效? .....	28
6.3 Nodejs 如何上传依赖包? .....	28
6.4 是否可以自定义内存? .....	28
6.5 提交的代码有什么要求? .....	29

# 1 简介

## 1.1 函数计算介绍

函数计算 ( Cloud Function , 简称 CF ) 是天翼云提供的一项事件驱动的全托管计算服务。通过函数计算, 开发者们不再需要购买、配置和维护基础设施, 只需要实现业务函数代码并上传。函数计算会以弹性的、可靠的方式运行代码, 且用户只需要按照资源使用量和执行次数付费。

产品特性:

- 1) 无运维: 使用函数计算, 您不用再购买、配置和维护基础设施, 省去复杂的运维工作。
- 2) 低成本: 函数计算根据请求量申请或释放资源, 根据请求次数和运行时长计费, 近乎 100% 的资源利用率让成本进一步降低
- 3) 更简单: 开发者们可以专注于业务核心代码, 通过多种事件触发器组合各种云资源, 轻松完成需求, 获得简单高效, 极致的开发体验

## 1.2 基本术语

### 应用

应用指的是支撑某个特定业务, 功能或服务相对内聚的一个逻辑单元, 一个应用可以由多个函数组成。在函数计算平台中, 主要用于聚合函数。

### 函数

函数是函数计算平台管理和运行的最小单元, 也指用户提交的一组代码。当事件 ( 请求 ) 到达时, 平台会启用函数容器执行函数代码处理请求。

### 运行环境

开发语言对应的运行时环境。当前平台支持的是 Nodejs 12。

### 函数入口

代码执行入口，类似于 Java 的 main。

### 内存

指函数容器运行时向平台申请的内存。

### 超时时间

触发事件（请求）时，期望在阈值内得到响应的的时间值，超过则终止。

### 触发器

触发函数执行的事件源组件。常见的触发器有定时触发器，各种云服务（例如对象存储）的触发器。

### 版本

函数运行时的快照。利用版本可以支持灰度发布。

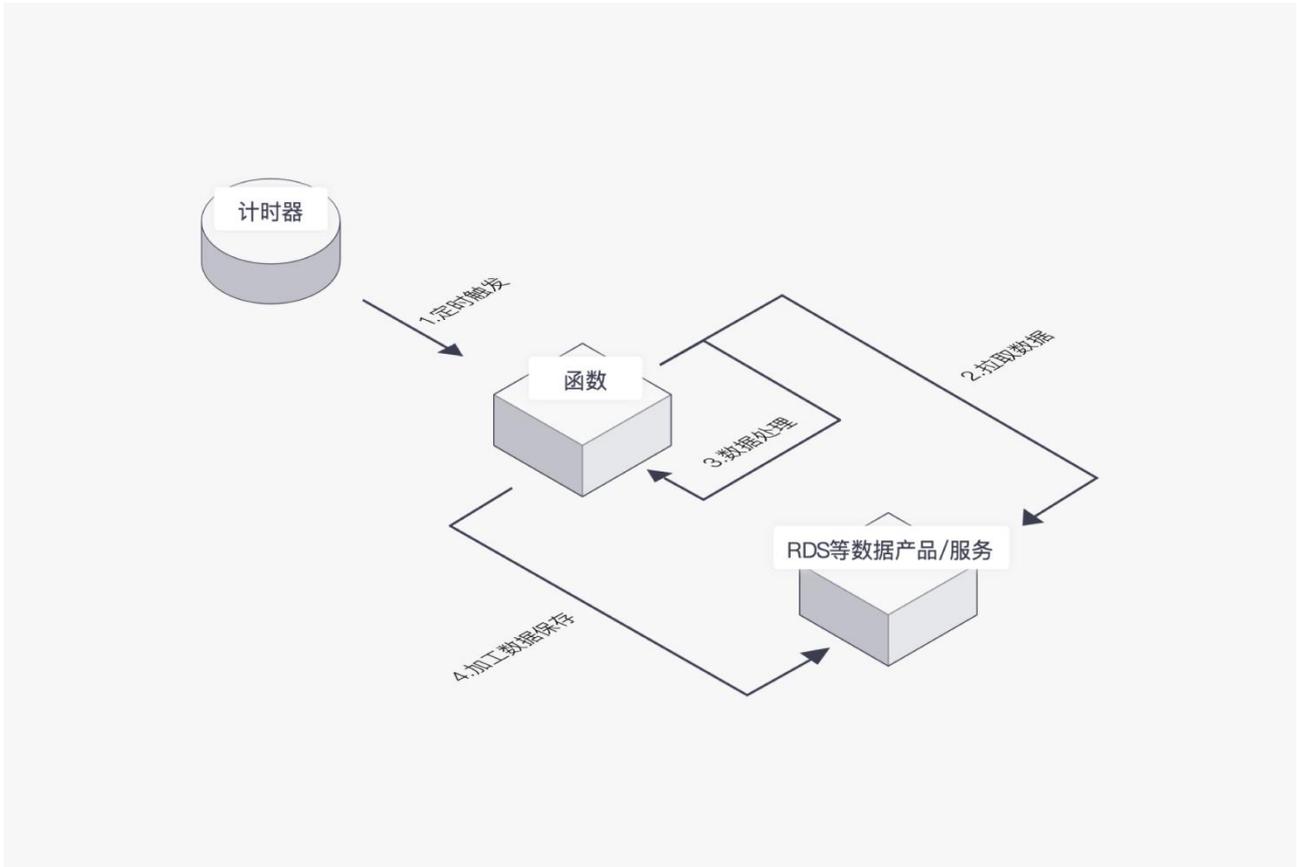
## 1.3 应用场景

天翼云函数计算（Cloud Function，CF）处于持续迭代中，随着对接产品的持续增长，函数计算能够适配的应用场景将越来越丰富。

### 数据处理：

业务运行过程中往往会产生许许多多的业务数据，这些数据需要定期的处理，提炼，以便发现更有价值的信息或作为决策依据。

例如，公众号，定时采集文章的浏览情况数据，汇总，生产报表数据。例如，网上商城，每天统计前一天商品销售量最高的前 5，分析商品的热度，根据库存情况生成采购单，将业务流转自动化。函数只有在设定的时间启动运行，相比传动模式，这样的模式更经济。



## 1.4 技术原理

函数计算是基于事件请求的服务。当收到事件请求时，函数计算平台将触发您创建的函数，根据函数的配置信息（内存大小等）进行资源分配，启动和管理容器（即函数的运行环境）。期间，函数计算平台将负责容器的创建、运行和销毁，用户无权对其进行管理。

容器启动的时候需要一定的时间，这会增加函数调用的响应时间。为了尽量缩小这部分耗时，容器处理完请求会不会马上删除，而是会留存一段时间，当有临近的请求再次到来时，即可马上交付给函数处理，无需等待容器启动。

函数运行时您无需关心诸如扩缩容之类的工作，但函数本身也存在一些限制，了解这些限制对正确使用函数很有帮助：

#### 临时磁盘空间：

函数运行过程中会分配一个 512MB 的磁盘空间/tmp。值得注意的是该空间在函数容器被删除时会被回收。所以任何中间信息或计算结果请存储到外部 Redis、MySQL 等存储服务中。

#### 函数并发量：

函数的并发量指的是在任意指定时间对函数代码的访问量。函数的并发量受发布的事件数（即请求数）影响。您可以通过如下公式估算函数的并发数量。

每秒请求数 \* 函数执行时间（按秒）

#### 执行环境和可用库：

当前函数计算 CF 的执行环境建立在一下基础上：

- Debian GNU/Linux 9

#### 运行内存可选规格：

目前可选的内存选项有 128MB、256MB、384MB... 2048MB，按照 128MB 递增

#### 依赖包：

天翼云函数计算目前支持 Nodejs，也继承了部分常用的包。用户依赖的包可以随代码一起提交（以 ZIP 包方式），以 msyql 为例目录格式如下

自带依赖包，压缩包里的文件结构如下：

```
├─ node_modules
|  └─ mysql
└─ test-mysql.js
```

#### 语言的支持：

目前平台支持 Nodejs。

Nodejs 运行环境默认携带的包有：

winston@2.2.0

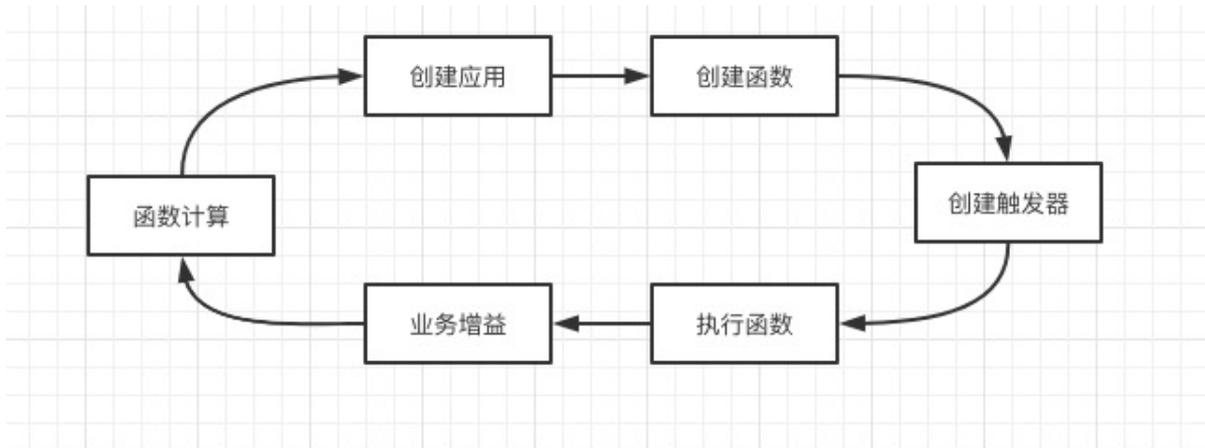
raw-body@2.3.2

body@5.1.0

mkdirp@0.5.1  
winston-daily-rotate-file@1.2.0  
express@4.14.0  
body-parser@1.15.2  
co@4.6.0  
istanbul@0.4.4  
js-beautify@1.6.12  
mocha@3.0.2  
uuid@8.3.0  
mysql@2.18.1

## 1.5 使用流程

函数计算的使用流程如下：



- 1、创建应用
- 2、创建函数，编写代码，将代码提交到函数计算平台中
- 3、以事件源的方式触发函数
- 4、查看服务的统计与监控

## 1.6 使用限制

限制项	最大数量
函数个数	400
函数版本	10
内存大小	2048MB
磁盘空间 ( tmp 目录 )	512MB
函数运行时扩容上限	60

# 2 产品功能

## 2.1 总览

功能项	说明
统计数据	统计数据包括“函数数量”、“本月调用次数”及“本月资源使用量”
监控数据	监控数据包括调用次数和资源使用量的监控。用户可选择 1 小时/24 小时/7 天内或者指定时间范围内的监控数据

## 2.2 应用管理

功能项	说明
创建应用	用户可以通过应用管理弹窗创建应用
修改应用	用户可以通过应用管理弹窗修改应用，且可以批量的修改
删除应用	用户可以通过应用管理弹窗修改应用，且可以批量的删除

## 2.3 函数管理

功能项	说明
创建函数	用户通过选择运行环境，提交代码，设置内存大小和超时等信息来创建函数
修改函数	修改函数，支持超时时间、内存大小等信息的修改
删除函数	用户可自行进行函数删除操作

触发器管理	在函数的管理页面的触发器 tab 页面中，支持创建、删除触发器。触发器用于产生事件源（请求）
创建版本	对当前运行的函数创造快照
设置流量	基于版本设置流量比例，可以实现灰度发布

## 2.4 其他核心功能

这部分功能不需要用户操作，但却是很重要的功能，由平台提供。

- 1) 平台以加密的方式传输和存储代码，以保证您源码的安全；
- 2) 函数运行时将自动根据请求量扩容和缩容。缩容的下限为 0，扩容的上限为 60。

# 3 快速入门

## 3.1 开通函数计算服务

使用函数之前还需要到订购页开通租户的函数计算服务。打开界面，选择区域，阅读并勾选服务协议，点击“授权开通”。

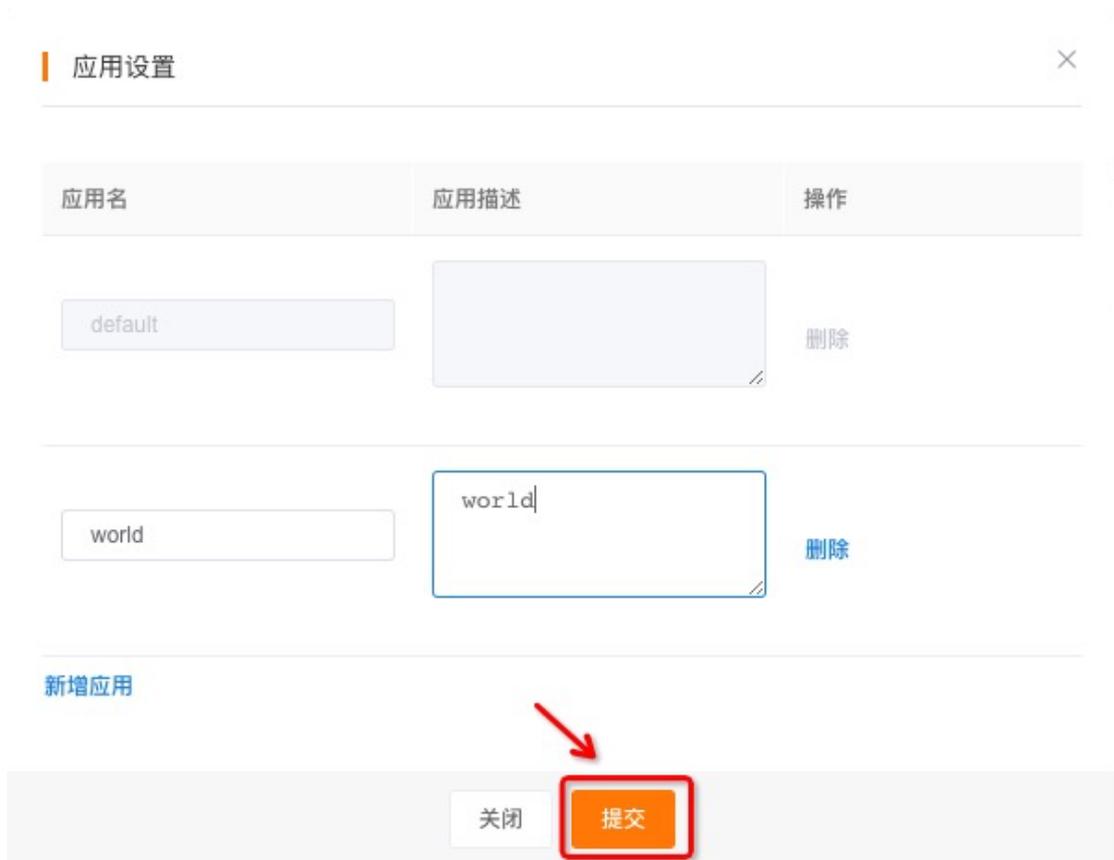


## 3.2 使用函数

步骤 1：创建应用，从控制台界面的上方进入；



填写应用名称和应用描述（可选）。应用名称支持字母、数字、-，且以字母开头。



步骤 2：创建函数，点击“创建函数”，填写如下函数信息，并点击“确认”创建函数。

**函数名称**：标示函数，支持支持字母、数字、中划线（-），并以字母开头。

**所属应用**：选择对应的应用。

**运行环境**：选择函数运行时的语言环境。

完成之后点击“下一步”。

**函数入口**：函数入口是代码执行的入口。

Nodejs，格式为[文件名].[Nodejs 函数名称]，默认为 main.handler，用户可以根据实际情况调整。

**上传代码**：编写代码，以 Nodejs 为例，编写 main.js 源代码，如下

```

exports.handler = function(event, context, callback) {

    // do something.

    callback(null, "This is a function");

};
    
```

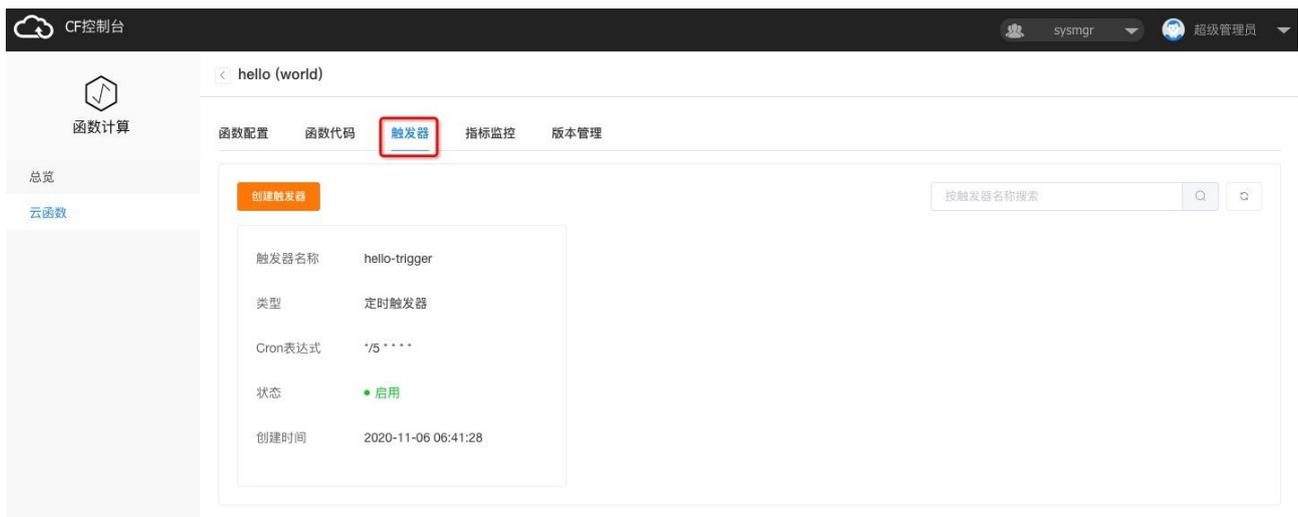
选择代码并点击“上传到服务器”上传。

**内存**：选择运行时需要的内存大小。注意，如果少于实际需要的内存空间，函数可能无法运行。

**超时时间**：设置调用超时的时间。

步骤 3：创建触发器，触发器用来触发函数执行。

点击函数名称，进入函数详情页，点击“触发器”tab 页面，点击“创建触发器”。

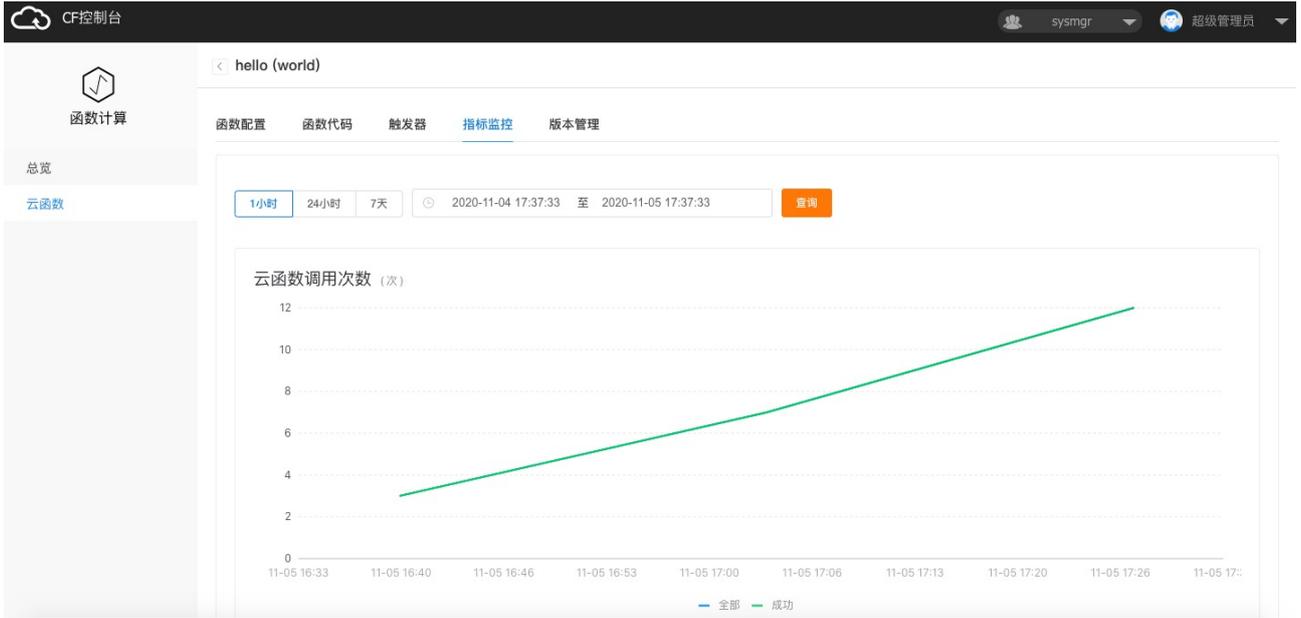


**触发器类型**：目前支持定时触发器。

**触发器名称**：触发器的名称，支持支持字母、数字、中划线 (-)，并以字母开头。

**Cron 表达式**：例如，\*/5 \* \* \* \*，每 5 分钟执行一次

当触发器发出请求时，可以在函数的管理页面的指标监控中看到处理的监控情况。



# 4 最佳实践

## 4.1 实践建议

**根据函数计算的特别，我们推荐您这样使用：**

- 编写无状态的函数代码，将中间信息通过缓存、消息等方式“存储”，将最终计算结果保存到数据库等服务中；
- 请对您上传的文件授以可读、可执行的权限，即+rx，以确保代码能够执行；
- 在源码中多使用 log 语句，输出有价值的信息，以便您调试；
- 建议在外部使用代码管理工具（例如 Git 等），进行核心代码的版本管理，确保代码的完备和可追溯。

# 5 函数计算管理

## 5.1 总览

总览，是用户使用身份认证信息登录函数计算控制台后出现的默认页面。概览，大致分为统计数据、监控数据两个部分。



图 5-1

统计数据包括“函数数量”、“本月调用次数”及“本月资源使用量”。函数数量为用户创建的函数的总数量。本月调用次数，为用户创建的所有函数在当前月份的被调用次数之和。本月资源使用量，是用户创建的所有函数在当前月份执行所消耗的资源量，以 GB\*s 为单位。每个用户每个月有免费使用的调用次数和资源使用量，超过免费额度的才进行计费。

函数数量为实时刷新数据。其余指标为当月1号到近1个小时的数据汇总



图 5-2

监控数据，包括调用次数和资源使用量的监控。用户可选择 1 小时/24 小时/7 天内或者指定时间范围内的监控数据。

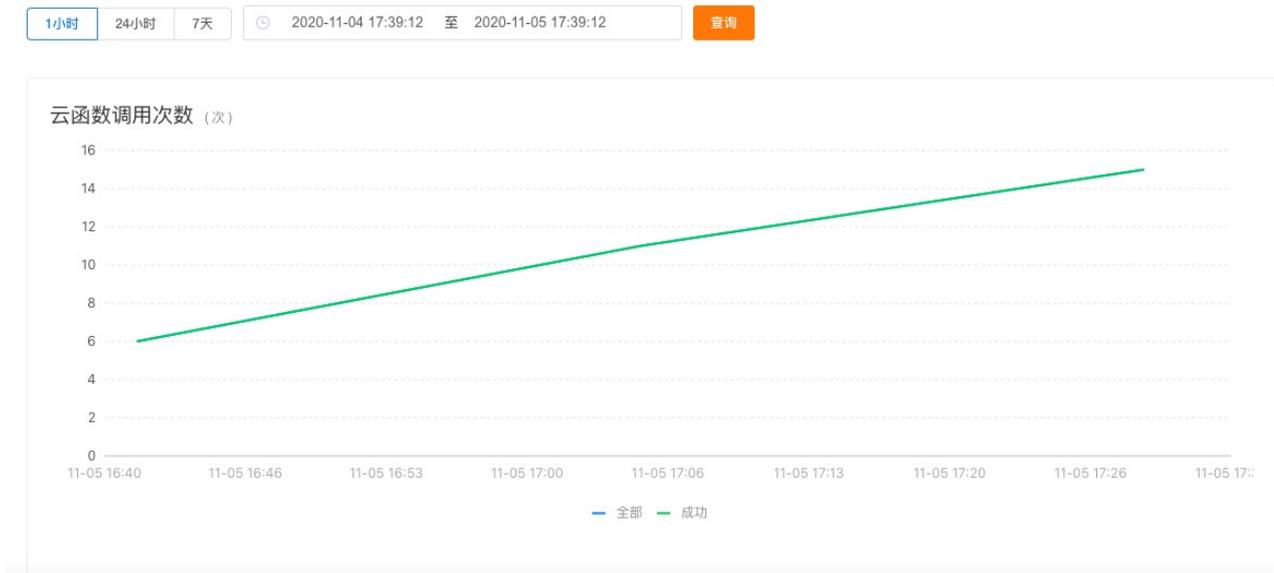


图 5-3

## 5.2 应用管理

### 5.2.1 创建应用

函数必须归属于某个应用。系统默认有“default”应用。用户可以根据业务特点，创建应用来管理函数。

如图 5-4 所示，在“云函数”页面点击图标，进入应用管理页面（如图 5-5），点击“新增应用”，填写应用名称和应用描述。



图 5-4

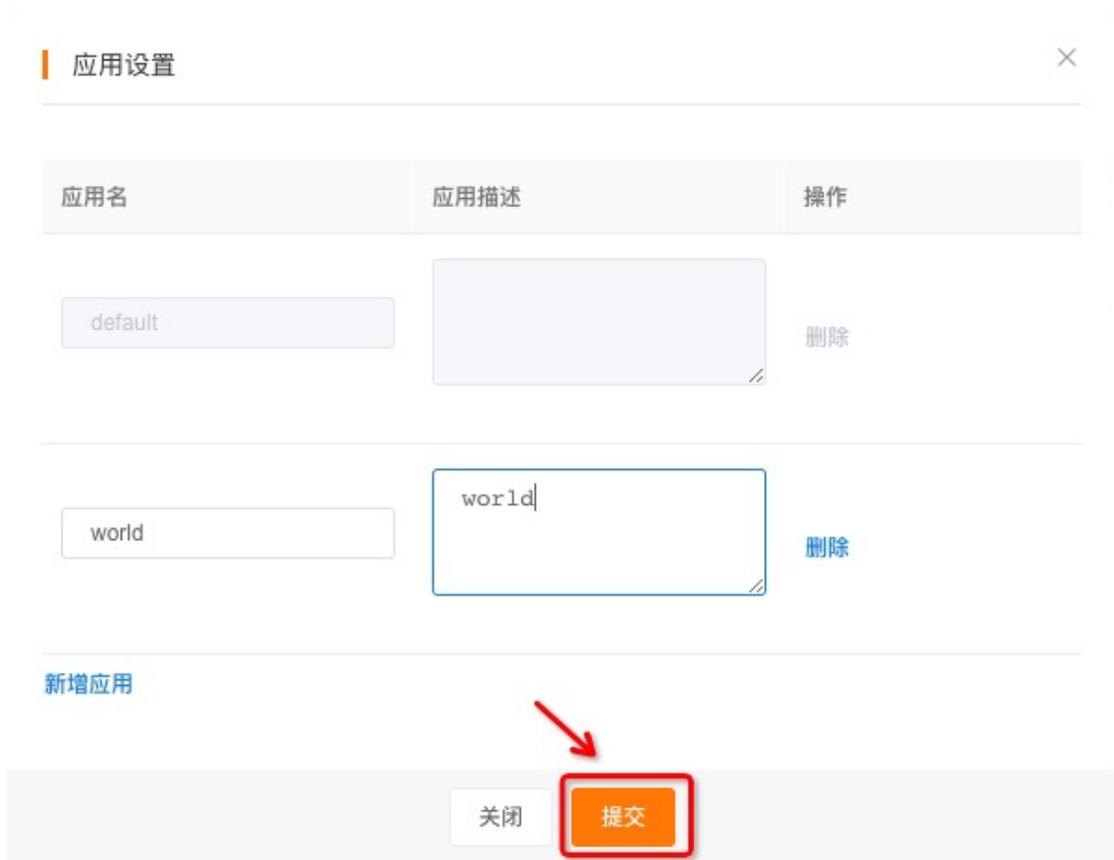


图 5-5

## 5.2.2 编辑应用

如图 5-4 所示，在“云函数”页面点击图标，进入应用管理页面，可以对应用的描述进行修改及删除应用。

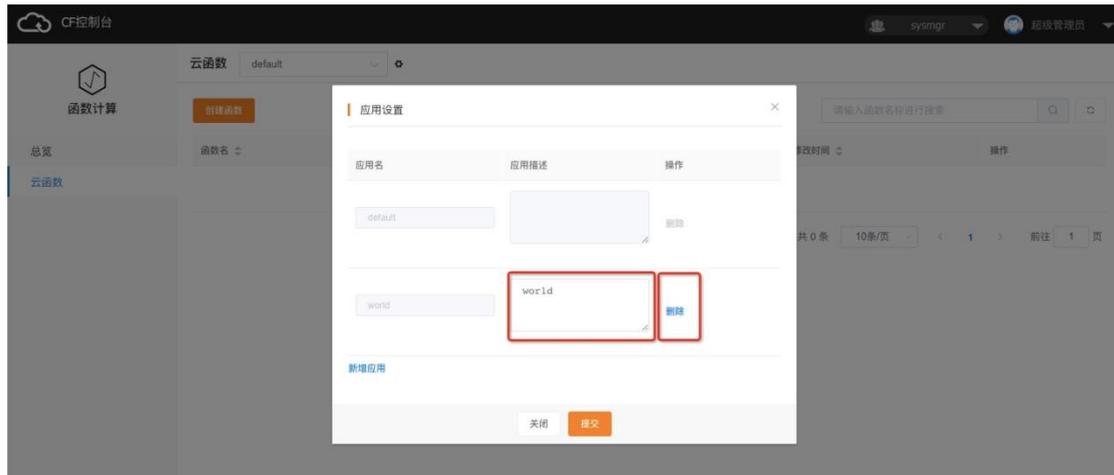


图 5-6

## 5.3 函数管理

### 5.3.1 创建函数

在“概览”页点击“创建函数”（如图 5-7），或者在“云函数”页面点击“创建函数”（如图 5-8），可以创建函数。



图 5-7



图 5-8

“新建函数”页面需要填写“函数名称”，选择归属的应用，选择函数的运行环境。

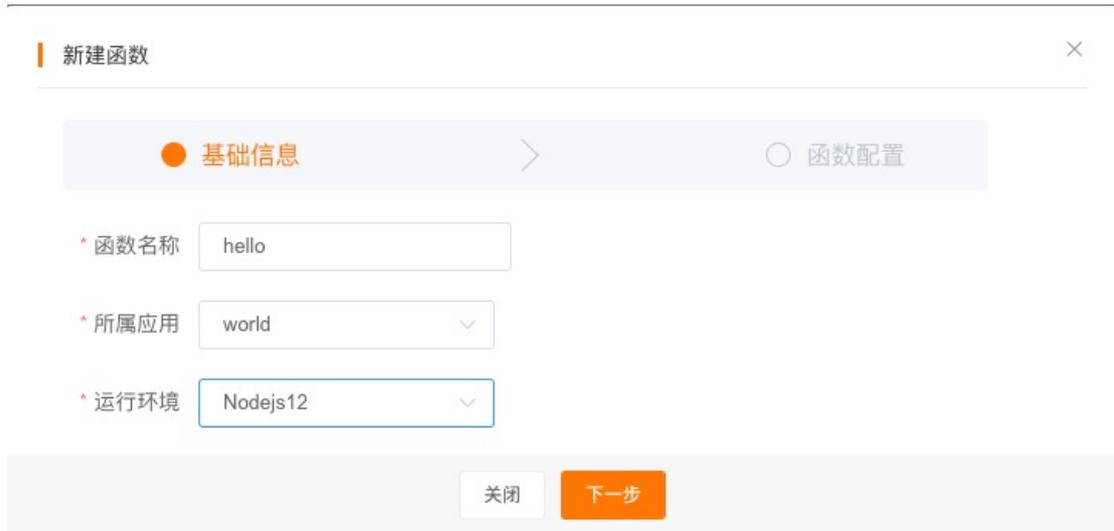


图 5-9

“函数配置”页面，函数入口格式为 [文件名].[函数名]。例如创建函数时指定的函数入口为 index.handler，那么文件名为 index.js，入口函数为 handler。将代码文件拖到“代码上传”框或者点击上传按钮，进行代码的上传。用户还需根据函数的特点，指定函数运行的内存上限，以及超时时间。如果函数运行所需的内存超过填写的内存值，会导致函数运行失败。

### 新建函数 ×

---

☑ 基础信息 ➤ ● 函数配置

函数名称 hello

所属应用 world

运行环境 Nodejs12

描述

\* 函数入口

提交方法

代码上传



将文件拖到此处, 或 [点击上传](#)

注: 可一次上传多个文件, 只能上传js文件

main.js  
上传到服务器

图 5-10

### 环境配置

委托

\* 内存

\* 超时时间  秒

上一步 确认

图 5-11

### 5.3.2 查看函数

点击“云函数”，可以看到 default 应用下的函数列表。通过选择应用，可以看到不同应用下的函数列表。



图 5-12

### 5.3.3 编辑函数

进入“云函数”页面，选择应用，点击需要编辑的函数右侧的“编辑”按钮，如图 5-12 所示，可以进入函数的编辑页面。



图 5-13

函数配置，可以修改内存及超时时间，如图 5-14 所示。

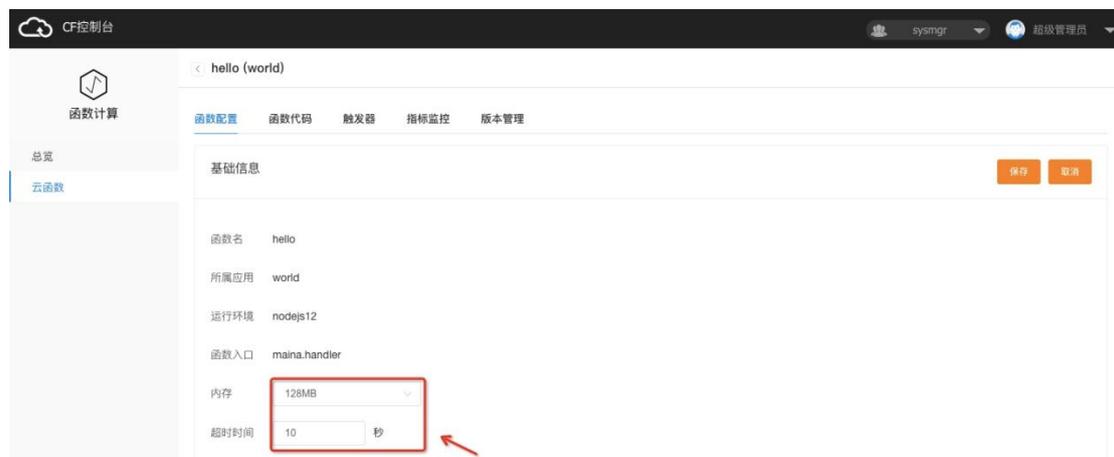


图 5-14

函数代码页面，点击“编辑”，可以更新函数代码。

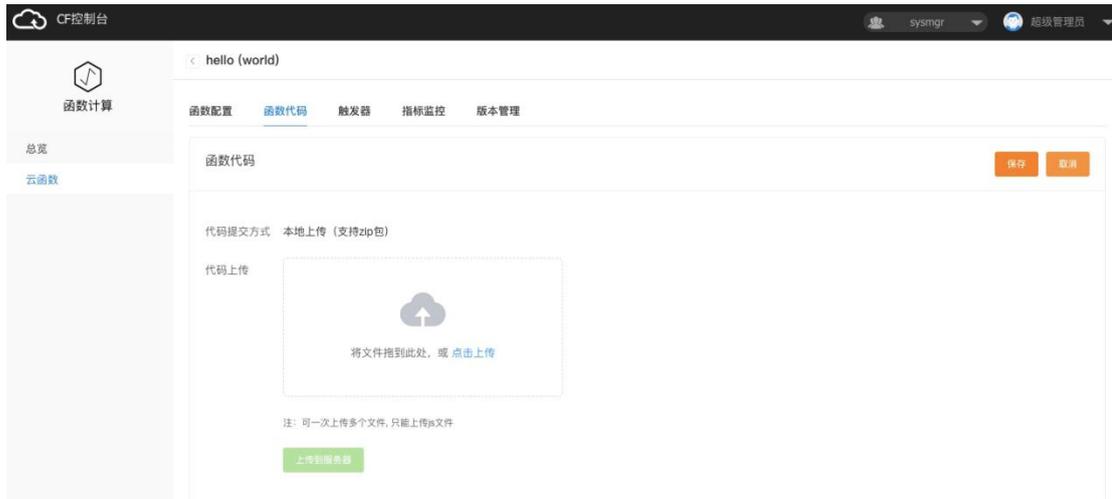


图 5-15

### 5.3.4 触发器管理

触发器页面的入口在函数详情页面的触发器 tab 中，可以创建触发器，目前支持定时器。触发器用户产生和发送请求到对应的函数，触发函数执行。定时触发器的作用为定时向函数发送请求。

目前触发器仅支持创建和删除，不支持修改。



图 5-16



图 5-17

### 5.3.5 函数监控指标

指标监控页面在函数详情的指标监控中，可以查看函数的调用次数和资源使用量的监控数据。

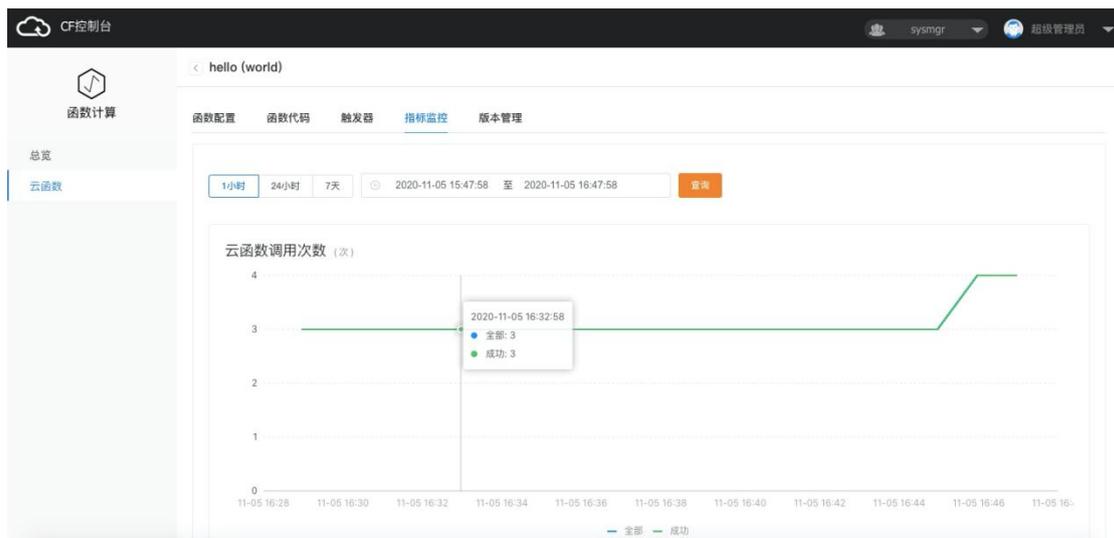


图 5-18

### 5.3.6 函数版本

点击函数详情链接，进入版本管理的 tab 页，点击“发布版本”按钮可以发布版本。发布版本相当于以 \$LATEST 的函数为快照发布，且函数运行相关的配置都“定格”在发布生效的那一刻，发布过的版本不支持修改。点击“流量”设置，可以在两个版本中进行流量的分配，实现灰度发布等。



图 5-19



图 5-20



图 5-21

### 5.3.7 删除函数

选择应用，点击要删除函数的右侧的“删除”按钮可以将函数删除。



图 5-22

# 6 常见问题

## 6.1 函数执行失败是什么原因？

函数运行失败的原因有多种，请依次检查：

- 1) 检查提交的代码是否合规，即符合语言的语法规范；
- 2) 检查依赖包（非官方自带）是否存在，是否符合规定的目录结构
- 3) 检查配置的内存大小是否满足实际需求

## 6.2 函数流量设置为什么不生效？

函数流量会被分配给各个函数实例，故需确保各个版本的函数运行正常（见函数调用失败的检查步骤）。

## 6.3 Nodejs 如何上传依赖包？

用户依赖的包可以随代码一起提交（以 ZIP 包方式），以 mysql 为例目录格式如下：

自带依赖包，压缩包里的文件结构如下：

```
├─ node_modules
│   └─ mysql
└─ test-mysql.js
```

## 6.4 是否可以自定义内存？

目前暂不支持。目前可选的内存选项有 128MB、256MB、384MB... 2048MB，按照 128MB 递增。

## 6.5 提交的代码有什么要求？

- 编写无状态的函数代码，将中间信息通过缓存、消息等方式“存储”，将最终计算结果保存到数据库等服务中。
- 请对您上传的文件授以可读、可执行的权限，即+rx，以确保代码能够执行。