无服务器云函数

产品简介

产品文档

腾讯云

文档目录

产品简介
- 产品概述
- 工作原理
- 产品优势

产品简介
产品概述

最近更新时间：2018-08-28 16:22:18

腾讯云无服务器云函数（Serverless Cloud Function）帮助您在没有购买和管理服务器时仍能运行代码。您只需要进行核心代码的编写及设置代码运行的条件，代码即可在腾讯云基础设施上自动、安全地运行，是许多场景下理想的计算平台。

计算资源的变迁

随着云服务的发展，计算资源高度抽象化，腾讯云提供了从物理服务器到云函数的、横跨各种抽象程度的计算资源供用户选择。

- 黑石物理服务器：以物理机为扩展单位。用户完全拥有整台实体计算资源，安全性最好。
- 云服务器：以云服务器为扩展单位，虚拟化硬件设备。用户和其他租户共享物理机资源，仍可自行配置CVM的各项指标，相对部署和迭代更加简单。
- 容器：以服务为扩展单位，虚拟化操作系统。测试和生产环境完全一致，测试和部署非常轻松。
- 无服务器云函数：以函数为扩展单位，虚拟化运行时环境（Runtime）。是现有计算资源的最小单位，具有完全自动、一键部署、高度可扩展等特点，是轻量级服务部署非常好的选择。

什么是无服务器Serverless

Serverless并不代表没有了服务器，只是用户不再需要关心这些底层资源了。当然，这也意味着用户无法登录服务器，并且也不需要想办法优化它。开发者只需关心最核心的代码片段，跳过复杂的、无聊的其他工作。这些代码完全由事件触发（event-trigger），平台根据请求自动平行调整服务资源，拥有近乎无限的扩容能力，空闲时则没有任何资源在运行。代码运行无状态，可以轻易实现快速迭代、极速部署。

腾讯云无服务器云函数简介

腾讯云云函数是腾讯云提供的无服务器（Serverless）执行环境。您只需编写简单的、目的单一的云函数，并将它与您的腾讯云基础设施及其他云服务产生的事件关联起来。
使用云函数时，用户只需关注自己的代码。腾讯云完全管理底层计算资源，包括服务器CPU、内存、网络和其他配置/资源维护、代码部署、弹性伸缩、负载均衡、安全升级、资源运行情况监控等，用户只需使用平台支持的语言（目前支持Python，Nodejs，Java）提供代码。同时，这意味着您无法登录或管理服务器、无法自定义系统和环境。

代码在执行时将根据请求负载扩缩容，无需人工配置和介入即可满足不同情景下服务的可用性和稳定性，从每天几个请求到每秒数千个请求，都由云函数底层自行伸缩。云函数自动地在地域内的多个可用区部署，提供极高的容错性。用户只需为运行中的云函数付费，代码未运行时不产生任何费用。

您可以自定义何时运行您的代码，比如在COS Bucket上传或删除了文件时、应用程序通过SDK调用了代码时，或自行指定代码定期执行。因此，您可以使用云函数作为COS服务的数据处理触发程序，轻松实现IFTTT逻辑。也可构建灵活的定时自动化任务来覆盖手工完成的操作，轻松构建灵活可控的软件架构。

如果用户需要管理您的计算资源，腾讯云还为您提供以下计算服务：

- 云服务器提供各种规格的实例配置，您可以自定义计算资源的CPU、内存、操作系统、网络、安全等各项配置，但您需要自定义策略完成跨可用区容错、资源伸缩等。
- 容器服务提供高度可扩展的容器管理服务，您可以根据资源需求和可用性要求在您的集群中安排容器，满足业务或应用程序的特定要求。

工作原理

最近更新时间：2018-08-28 16:22:28

函数运行时的容器模型

SCF 将在事件触发时代表您执行 SCF 函数，根据您的配置信息（如内存大小等）进行资源分配，并启动和管理容器（即函数的执行环境）。SCF 平台负责所有函数运行容器的创建、管理和删除操作，用户没有权限对其进行管理。

在容器启动时需要一些时间，这会使得每次调用函数时增加一些延迟。但是，通常仅在首次调用函数、更新函数、或长时间未调用时重新调用时会察觉到此延迟，因为平台为了尽量减少此启动延时，会尝试对后续调用重用容器，在调用函数后容器仍会存留一段时间，预期用于下次调用。在此段时间内的调用会直接重用该存留的容器。

容器重用机制的意义在于：

- 用户代码中位于执行方法外部的任何声明保持已初始化的状态，再次调用函数时可以直接重用。例如，如果您的函数代码中建立了数据库连接，容器重用时可以直接使用原始连接。您可以在代码中添加逻辑，在创建新连接之前检查是否已存在连接。
- 每个容器在 `/tmp` 目录中提供部分磁盘空间。容器存留时该目录内容会保留，提供可用于多次调用的暂时性缓存。再次调用函数时有可能可以直接使用该磁盘内容，您可以添加额外的代码来检查缓存中是否有您存储的数据。

> 注意事项：
> 请勿在函数代码中假定始终重用容器，因为是否重用和单次实际调用相关，无法保证是创建新容器还是重用现有容器。

临时磁盘空间

SCF 函数在执行过程中，都拥有一块 512MB 的临时磁盘空间 `/tmp` ，用户可以在执行代码中对该空间进行一些读写操作，但这部分数据可能 *不会* 在函数执行完成后保留。因此，如果您需要对执行过程中产生的数据进行持久化存储，请使用 COS 或 Redis/Memcached 等外部持久化存储。

调用类型

SCF 平台支持 `同步` 和 `异步` 两种调用方式来调用云函数。调用类型与函数本身的配置无关，只有在调用函数时才能控制调用类型。

- 同步调用函数将会在调用请求发出后持续等待函数的执行结果返回
- 异步调用将不会等待结果返回，只发出请求。

以下调用场景您可以自由定义函数的调用类型：

- 编写的应用程序调用 SCF 函数。如果您需要同步调用，请在 InvokeFunction 接口中传入参数 `invokeType=RequestResponse` ；如果您需要异步调用则请传入参数 `invokeType=Event`
- 手动调用 SCF 函数（使用 API 或 CLI）用于测试。调用时的参数区别同上。

但是，在您使用腾讯云其他云服务作为事件源时，云服务的调用类型是预定义的，用户无法在这种情况下自由指定调用类型。例如，COS 和定时器始终异步调用 SCF 函数。

用户限制

下表列出了内测期的资源限制。

资源	默认限制
单地域函数个数	20
新建函数时支持同时创建的触发器个数	1
单函数COS触发器数量	2
单函数定时触发器数量	2
临时磁盘容量（/tmp）	512MB
最大执行时长	300 秒
控制台上传的代码包大小	5MB
最大并发容器数	20，用户可以联系我们提高该限制

函数并发量

函数的并发数量是指在任意指定时间对函数代码的执行数量。对于当前的 SCF 函数来说，每个发布的事件请求就会执行一次。因此，这些触发器发布的事件数（即请求量）会影响函数的并发数。您可以使用以下公式来估算并发的函数实例总数目。

```
每秒请求量 * 函数执行时间（按秒）
```

例如，考虑一个处理 COS 事件的函数，假定函数平均用时 0.2 秒（即200毫秒），COS 每秒发布 300 个请求至函数。这样将同时生产 300*0.2=60 个函数实例。

并发限制

默认情况下，SCF 将会对每个函数的并发量限制为 20。您可以通过联系我们来调高此数值。

如果调用导致函数的并发数目超过了默认限制，则该调用会被阻塞，SCF 将不会执行这次调用。根据函数的调用方式，受限制的调用的处理方式会有所不同：

- 同步调用：如果函数被同步调用时受到限制，将会直接返回 429 错误
- 异步调用：如果函数被异步调用时受到限制，SCF 将在一定的时间内以固定的频率自动重试受限制的事件

重试机制

如果您的函数因为超过最大并发数目、或遇到平台内部资源不足等内部限制导致失败。此时，如果您的函数被同步调用，将会直接返回错误（见上面并发执行限制内容）。如果您的函数被内部云服务使用异步方式调用，则该次调用将会自动进入一个重试队列，SCF 将自动重试调用。

执行环境和可用库

当前 SCF 的执行环境建立在以下基础上：

- 标准 CentOS 7.2 镜像
- Python 2.7 运行时环境

以下库和依赖包在 SCF 执行环境下直接可用，您可以直接使用 `import` 语句导入它们：

- Python标准库
- Qcloud COS SDK
- Qcloud API SDK

产品优势

最近更新时间：2018-08-28 11:27:11

简单易用

减少组件开销

使用云函数时，用户只需编写最重要的“核心代码”，不再需要关心负载均衡、自动伸缩、网关等周边组件，极大地降低了服务架构搭建的复杂性。

自动扩缩容

无需任何手动配置，云函数即可根据请求量自动横向扩缩。不管您的应用每天只有几个请求（如日志统计等定期事务），还是每秒有几千上万个请求（如移动应用的后端），云函数均可自动安排合理的计算资源满足业务需求

高效又创造性地开发

加速开发

云函数不要求特定框架或依赖，开发者可以专注于核心代码的开发。同时开发人员可以组成多个小团队，单个模块的开发无需了解其他团队的代码细节。独立开发和迭代的速度变得前所未有的快，帮助用户把握住产品上线的黄金时间。

复用第三方服务

您可以使用云函数编写一些目的单一、逻辑独立的业务模块，因而可以完全复用已经成熟的第三方代码实现，比如使用oAuth实现登录模块。

简化运维

每个函数都是单独运行、单独部署、单独伸缩的，用户上传代码后即可自动部署，免除单体式应用部署升级难的问题。

稳定可靠

高可用部署

云函数可以自动在每个地域中随机地选择可用区来运行。如果某个可用区因灾害或电力故障等导致瘫痪，云函数会自动地选择其他可用区的基础设施来运行，免除单可用区运行的故障风险。

与其他计算服务相辅相成

常驻的工作负载可以通过云服务器 CVM 来承载，而由事件触发的工作负载可以使用云函数。不同云服务满足不同的业务场景和业务需求，使得您的服务架构更加健壮。

简化管理

简化安全配置

用户不再需要对OS入侵、登录风险、文件系统安全、网络安全和端口监听做复杂的配置和管理，一切交由平台处理，平台通过定制化的容器保证每个用户的隔离性。

可视化管理

用户可直接在控制台管理函数代码及函数何时运行（即函数触发器），无需复杂的配置文件即可一键部署和测试函数

大幅度降低开销

永远不为空闲时间付费

函数在未执行时不产生任何费用，对一些并非常驻的业务进程来说开销将大大降低。函数执行时按请求数和计算资源的运行时间收费，价格优势明显，对初创期的开发者十分友好。