

# 云计算的另一块战场：Serverless初谈

原创：Leo\_D BespinGlobal 8月23日

历数当前热门技术，除了物联网、区块链、AI，必不可少Serverless（无服务器）。它究竟是怎样一门技术？

-01-

## Serverless的含义

首先，我们看一下Wikipedia对于Serverless是怎么解释的：

//

Serverless computing is a cloud-computing execution model in which the cloud provider acts as the server, dynamically managing the allocation of machine resources. Pricing is based on the actual amount of resources consumed by an application, rather than on pre-purchased units of capacity. It is a form of utility computing.

[https://en.wikipedia.org/wiki/Serverless\\_computing](https://en.wikipedia.org/wiki/Serverless_computing)

//

Serverless字面意思比较晦涩，但它并不是无源之水、无本之木。就像虚拟化等技术，Serverless也运行在硬件之上，但是抽象层级却远高于硬件。

-02-

## 从虚拟化、容器到Serverless

让我们先从IT技术近几年的发展讲起，长话短说。

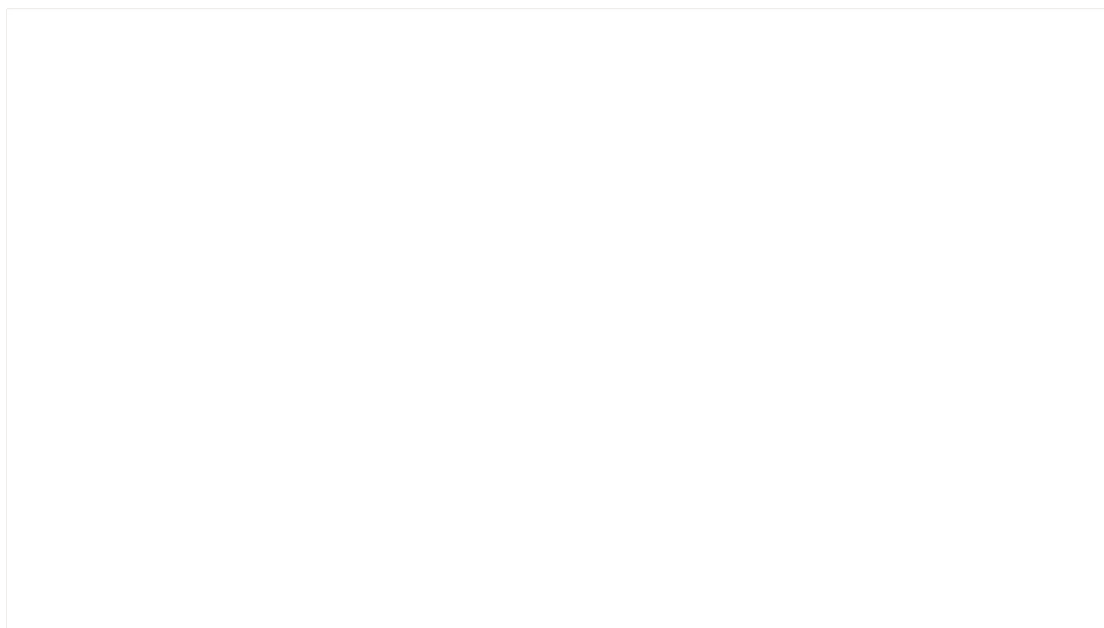
### Step1. 虚拟化

虚拟化技术抽象了底层资源，形成了Hypervisor层，让操作系统可以运行于其上。自此，操作系统与底层硬件解耦。

### Step2. 容器

几年前，容器技术大火。其中典型代表就是Docker。Docker通过宿主机OS层上的Docker Engine，使应用可以独立运行在容器的最小化镜像内，完全占有独立的

Bins/Libs。下面的两幅图对比了传统虚拟化技术与容器技术。很明显，容器技术消除了 Guest OS对资源的严重消耗，并且将资源以更细粒度的分割方式进行划分，从而让应用消耗的资源达到最小化的程度。自此，**应用程序与操作系统解耦**。



如果再将容器平台以服务的形式提供出来，就是大家常说的CaaS (Container as a Service) 。

### Step3. Serverless

正当CaaS方兴未艾之时，Serverless技术异军突起。随着技术从容器发展到CaaS再到FaaS (Functions as a Service) ，对于资源的抽象越来越高。Serverless更是将对资源的抽象程度提高到了只需关注业务代码的层面。利用Serverless，IT部门的主要精力将更加聚焦在研发上。

在Serverless环境中，IT工作的重心完全移到了编写业务代码上。Serverless架构中的runtime layer及其以下均由云供应商维护。云供应商利用自身专业的技术提供了高可用、高并发的性能。操作者只需对环境进行简单的配置，即可开始编写业务代码。

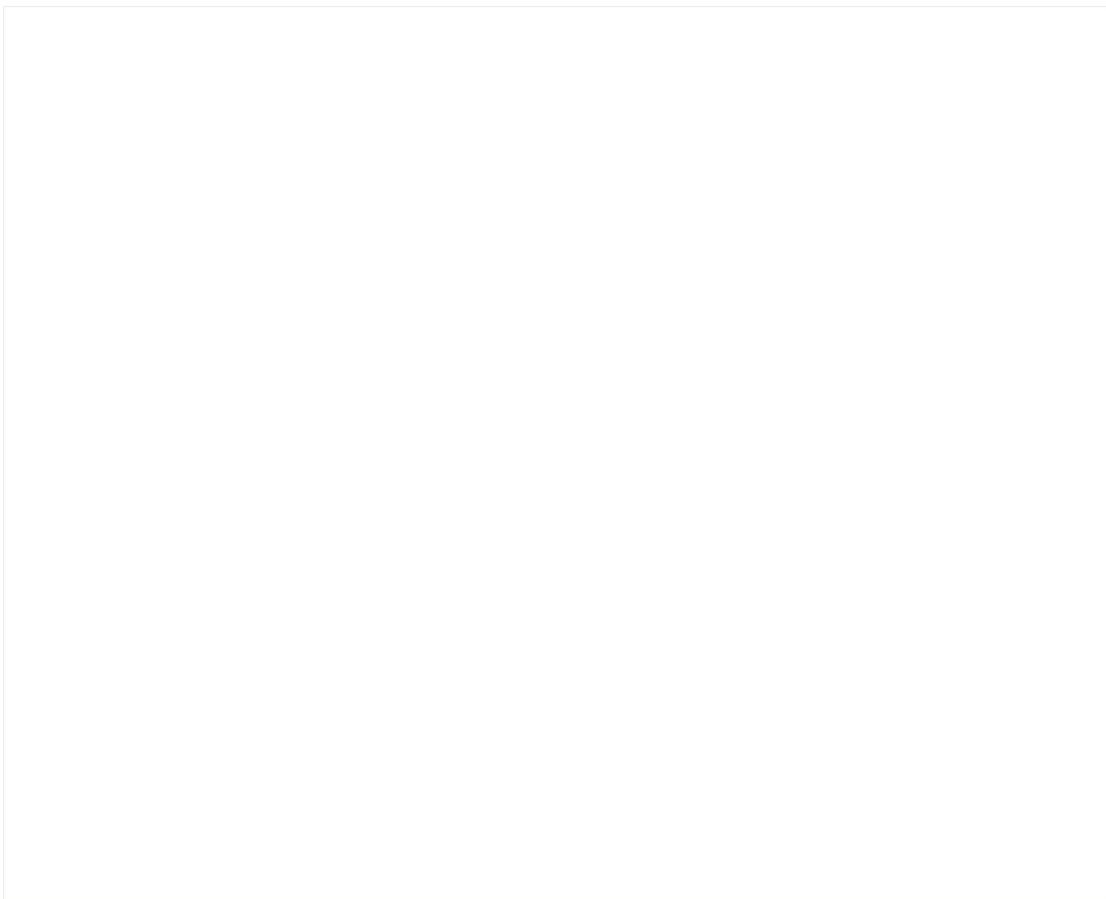
-03-

## Serverless的应用

以AWS Lambda服务为例进行简单介绍。Lambda服务是业界公认的Serverless的典型代表，开启了FaaS的时代。下图是在Lambda的console界面编写业务代码的示例。

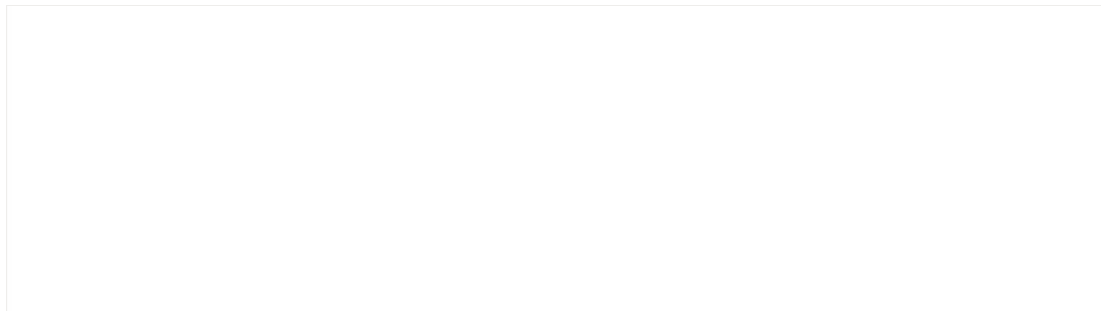


为了让代码更有效、安全的运行，操作者需要对内存、运行时、超时时间、网络环境、环境变量等参数进行配置。配置过程很简单，只需要点点鼠标、敲敲键盘。



最后，我们搭建了一个演示场景，发布了一个API Gateway + Lambda + DynamoDB的产品组合，形成纯粹的Serverless架构。其中，API Gateway是AWS另一个Serverless产品，用于路由API请求，将特定请求发送给运行在Lambda上的微服务。然后，微服务进行响应，读取DynamoDB表中的数据。

DynamoDB是AWS提供的另一个Serverless产品，是在读写性能和存储容量上提供超高可扩展能力的no-SQL数据库。



一次典型的请求过程简单来说分为以下几步：

1. 浏览器向API Gateway的API URL地址发送GET请求。
2. API Gateway将API请求路由到相关的Lambda函数。
3. Lambda执行代码，读取DynamoDB表中的数据。
4. Lambda将执行结果返回给API Gateway，API Gateway再将其返回给浏览器。

以上就是对Serverless的优点、表现形式、典型架构等方面的入门介绍。Serverless简单易用，但是想用好也绝非一朝一夕之功。

**更多精彩，欢迎继续关注Serverless的进阶篇。**

**-- 快速下单，单单有惊喜 --**

点击[阅读原文](#)，在BespIn Global阿里云市场快速下单，更有免费机会等你拿哦！



[阅读原文](#)