

7月3日-4日·中国·北京

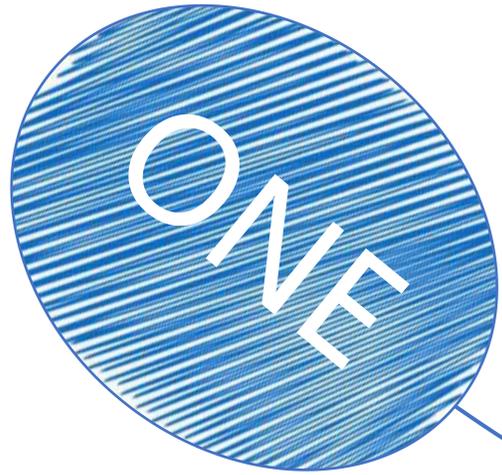
2019云计算开源产业大会

《函数即服务标准解读》

演讲人：刘如明

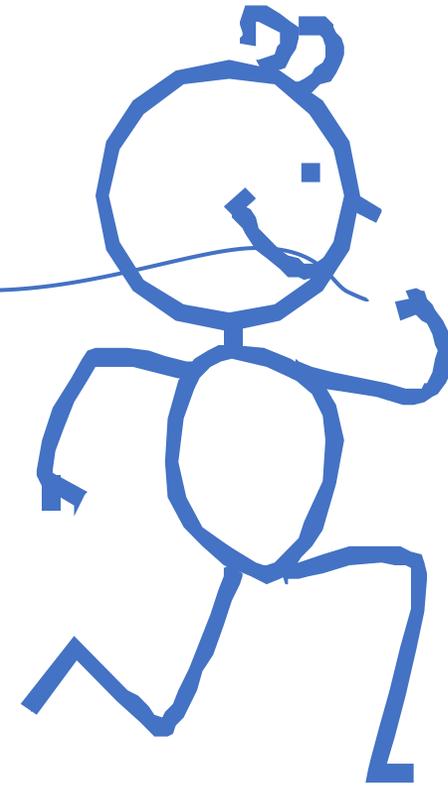


- 1 Serverless架构技术简介
- 2 Serverless产业发展近况
- 3 《函数即服务》标准简介
- 4 2019下半年工作方向



Serverless架构技术

为什么会出现Serverless
Serverless是什么
Serverless的组成部分
应用场景



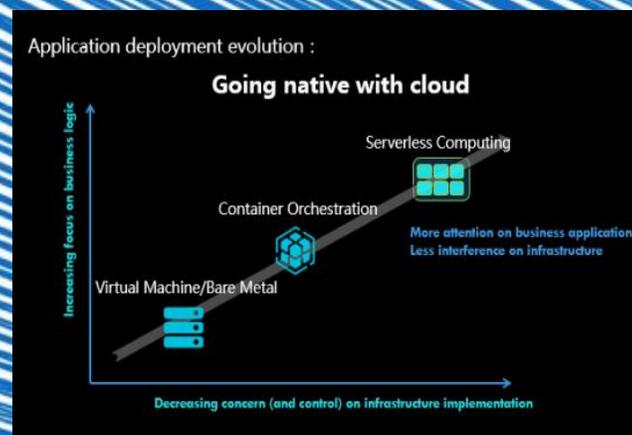
Serverless源起

ONE

应用架构的变化刺激产生新的计算抽象
成熟的计算抽象模型促进应用架构转变

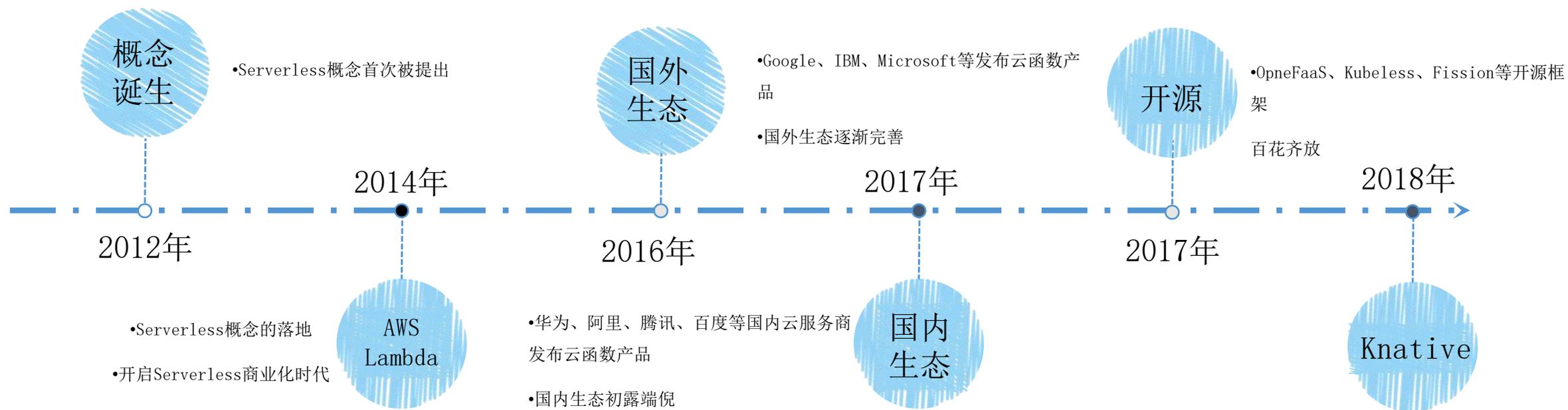


应用架构逐渐解耦
计算的抽象逐步加强
专注业务逻辑开发，弱化基础设施管理



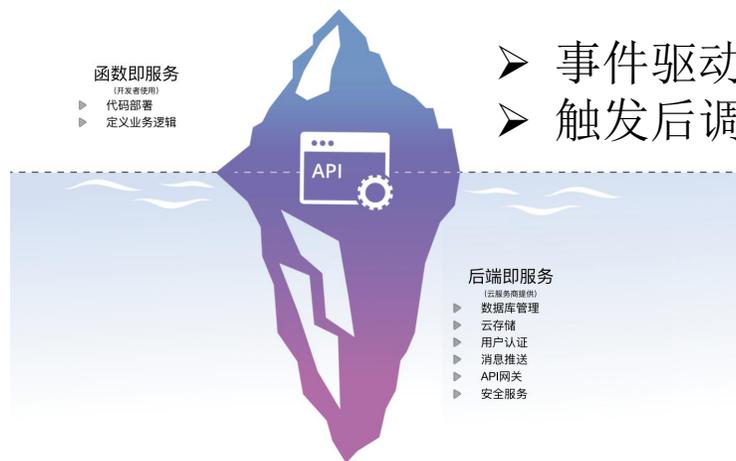
Serverless大事记

ONE



Serverless=FaaS+BaaS

ONE



函数即服务
(开发者使用)
▶ 代码部署
▶ 定义业务逻辑

- ▶ 事件驱动
- ▶ 触发后调用处理后释放

后端即服务
(云服务商提供)
▶ 数据库管理
▶ 云存储
▶ 用户认证
▶ 消息推送
▶ API网关
▶ 安全服务

- ▶ 无需构建直接使用
- ▶ API交互

FaaS

Serverless

BaaS

- ▶ 没有一直存在的应用
- ▶ 没有一直存在的服务器

Serverless适用场景

ONE

应用后端服务

- 移动应用后端服务
- Web应用后端服务
- IoT后端服务

简化后端服务，无需构建、运维
专注业务价值更高的前端开发

大规模数据处理 和计算类

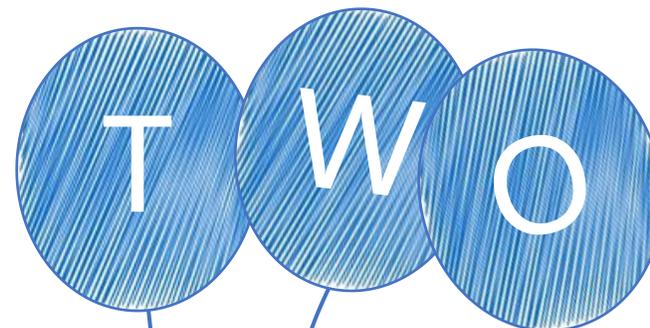
- 人工智能推理预测
- 批处理或计划任务

波峰波谷访问明显
最大效能的利用资源

基于事件的内容 处理类应用

- 实时文件处理
- 定制事件触发

事件触发



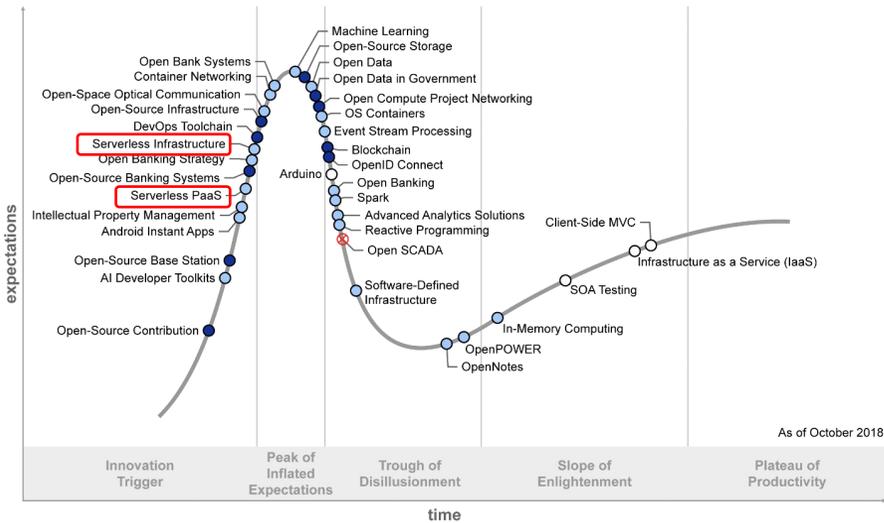
Serverless产业发展近况

- 产业趋势
- 技术趋势
- 国内现状



TWO

Serverless技术及市场发展趋势

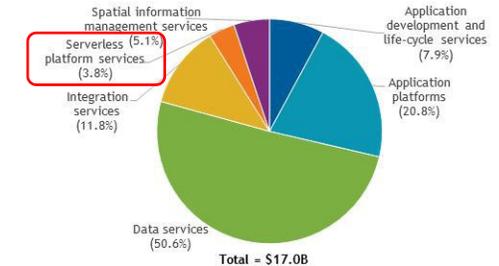


Plateau will be reached:
 ○ less than 2 years ● 2 to 5 years ● 5 to 10 years ▲ more than 10 years ⊗ obsolete before plateau

Priority Matrix for Open-Source Software, 2018

benefit	years to mainstream adoption			
	less than 2 years	2 to 5 years	5 to 10 years	more than 10 years
transformational		Event Stream Processing In-Memory Computing Machine Learning Open Bank Systems Open Banking Serverless Infrastructure	Blockchain DevOps Toolchain	
high	Arduino Infrastructure as a Service (IaaS) SOA Testing	Advanced Analytics Solutions AI Developer Toolkits Open Banking Strategy Open Data Open-Space Optical Communication OS Containers Reactive Programming Software-Defined Infrastructure	Open Data in Government Open-Source Contribution Open-Source Infrastructure Open-Source Storage	
moderate		Android Instant Apps Intellectual Property Management OpenNotes Serverless PaaS Spark	Open Compute Project Networking OpenID Connect Open-Source Base Station	
low	Client-Side MVC	Container Networking OpenPOWER	Open-Source Banking Systems	

As of October 2018



Source: IDC, Worldwide Competitive Public Cloud Platform as a Service Market Shares, 2017

- Serverless技术正处于快速发展期
- 2017-2022年全球范围内Serverless平台服务的复合增长率超过接近40%

TABLE 2: Worldwide Competitive Public Cloud Platform as a Service Revenue by Submarket, 2017–2022 (\$M)

	2017	2018	2019	2020	2021	2022	2017 Share (%)	2017–2022 CAGR (%)	2022 Share (%)
Data	8,621.0	12,405.8	17,037.6	23,083.8	30,055.1	38,486.7	50.6	34.9	61.9
Application platforms	3,539.9	4,456.2	5,392.2	6,368.0	7,507.0	8,881.5	20.8	20.2	14.3
Integration	2,015.6	2,635.4	3,317.9	4,024.1	4,772.4	5,635.7	11.8	22.8	9.1
Application development and life cycle	1,338.9	1,668.3	2,044.3	2,482.7	2,988.0	3,588.2	7.9	21.8	5.8
Serverless platform services	640.4	969.4	1,403.9	1,911.3	2,535.9	3,273.8	3.8	38.6	5.3
Spatial information management	870.1	1,070.3	1,297.4	1,566.9	1,884.2	2,260.1	5.1	21.0	3.6
Total	17,025.8	23,205.3	30,493.3	39,436.9	49,742.6	62,126.0	100.0	29.5	100.0

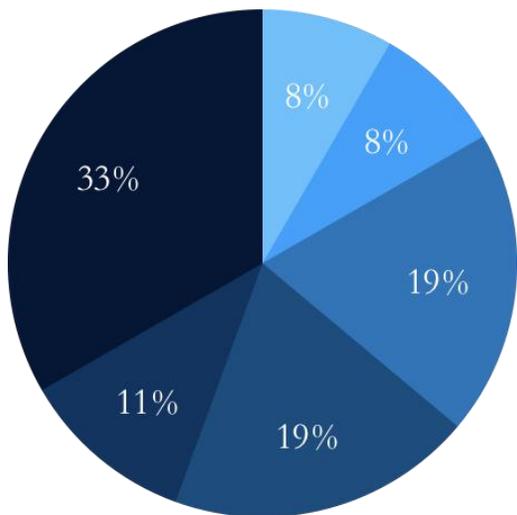
Source: IDC's Worldwide Semiannual Software Tracker, April 2018

TWO

Serverless内部调查

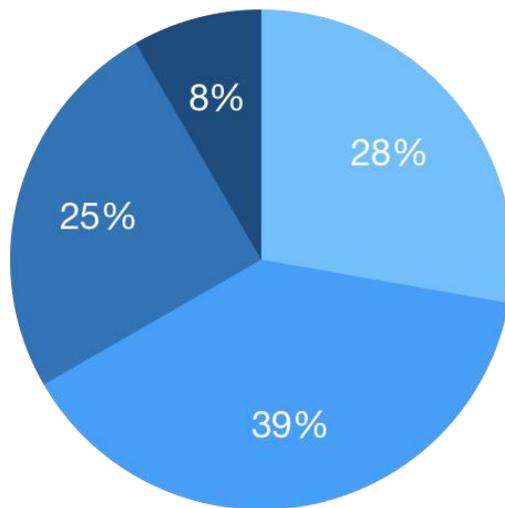
Serverless的主要推动者来自**大中型公司**
 业务以**公有云部署**为主，**多云环境**是主流

调查者所在公司规模



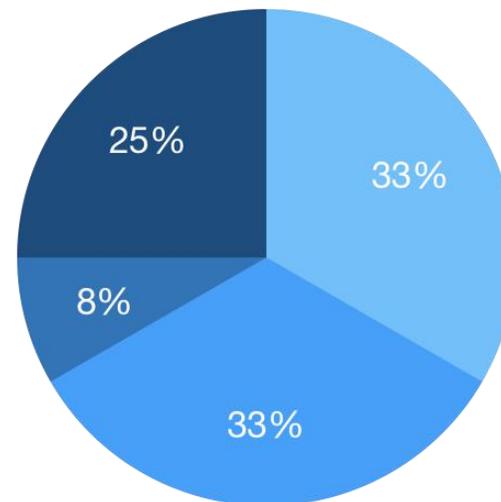
- 10人以下
- 10-50人
- 50-150人
- 150-500人
- 500-1000人
- 大于1000人

现有业务在公有云的部署情况



- 主要部署在公有云
- 部分部署在公有云
- 很少部署在公有云
- 不使用公有云

现有业务是否存在多云环境



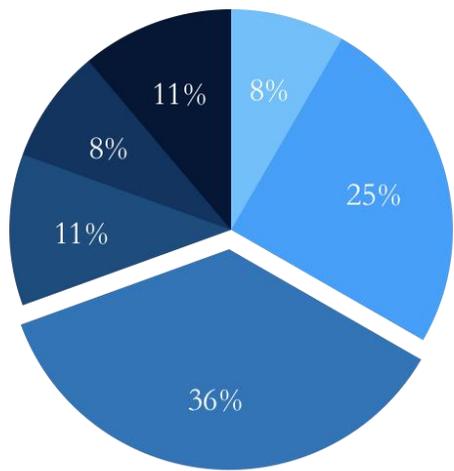
- 一套云环境
- 两套云环境
- 三套云环境
- 三套以上云环境

TWO

Serverless内部调查

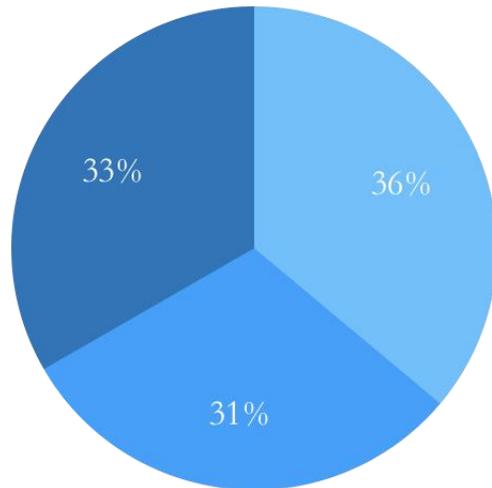
Serverless处于小范围试水阶段 公有云应用与私有云部署均衡考虑

Serverless技术的应用现状



- 这对公司核心业务很重要
- 小范围适用该技术
- 暂未使用, 计划未来1年使用
- 很重要, 但未触及核心业务
- 只做项目或个人研究使用
- 暂时使用, 也没有计划使用

如何构建现有的Serverless环境



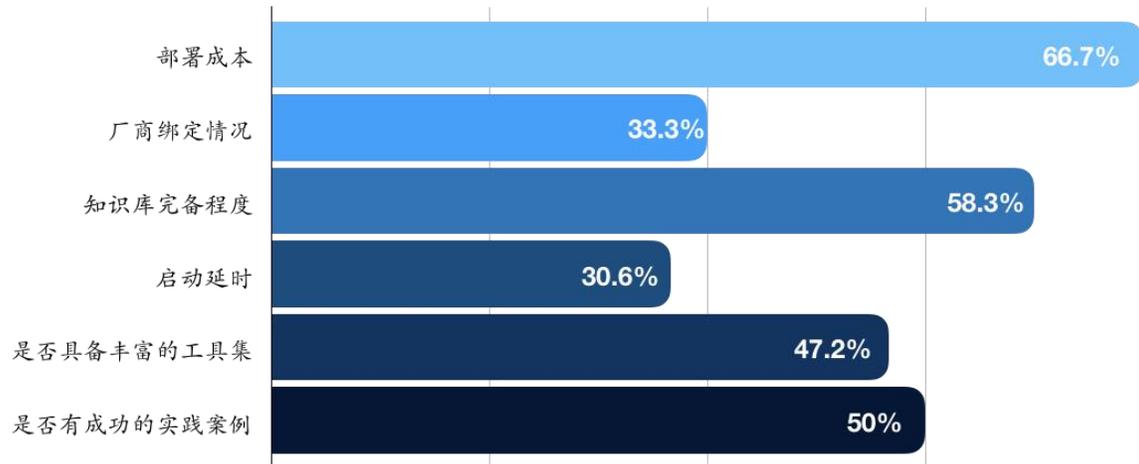
- 采用公有云函数服务
- 应用开源Serverless框架构建
- 二者均有

TWO

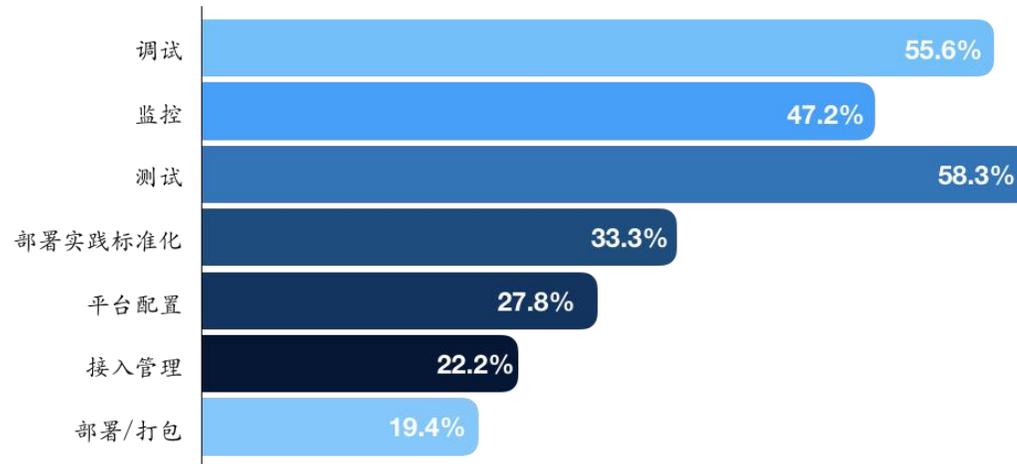
Serverless内部调查

部署成本是第一考量因素
能用、好用与具备落地实例是重要参考

采纳Serverless架构技术前，主要考虑的因素



业务Serverless化部署过程中，下述哪些部分挑战较大？

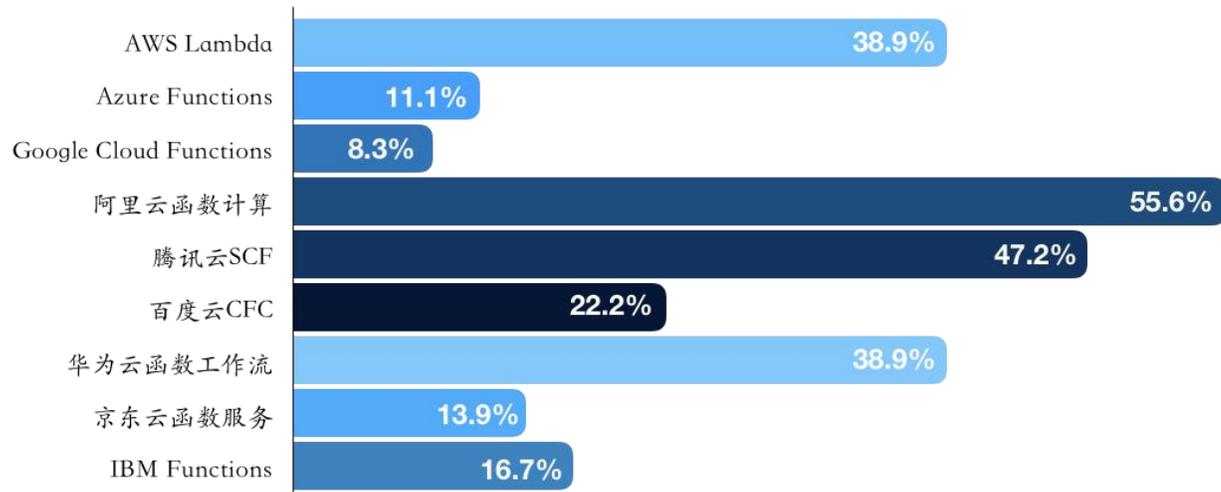


TWO

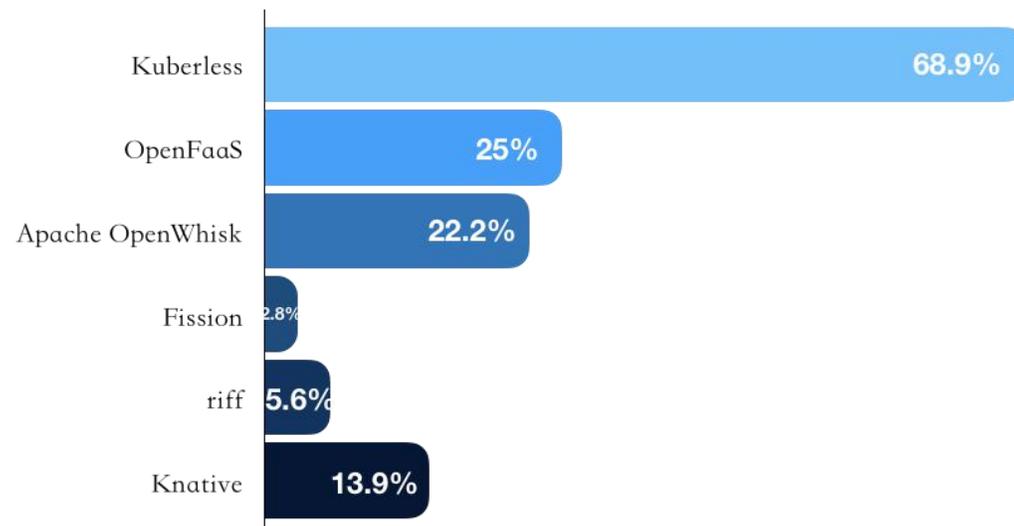
Serverless内部调查

阿里云、腾讯云、华为云是公有云部署的主要考虑对象
Kuberless是构建私有化环境的主要参考

正在使用或未来一年内计划使用如下哪种产品:



正在使用或未来一年内计划使用如下哪种框架:





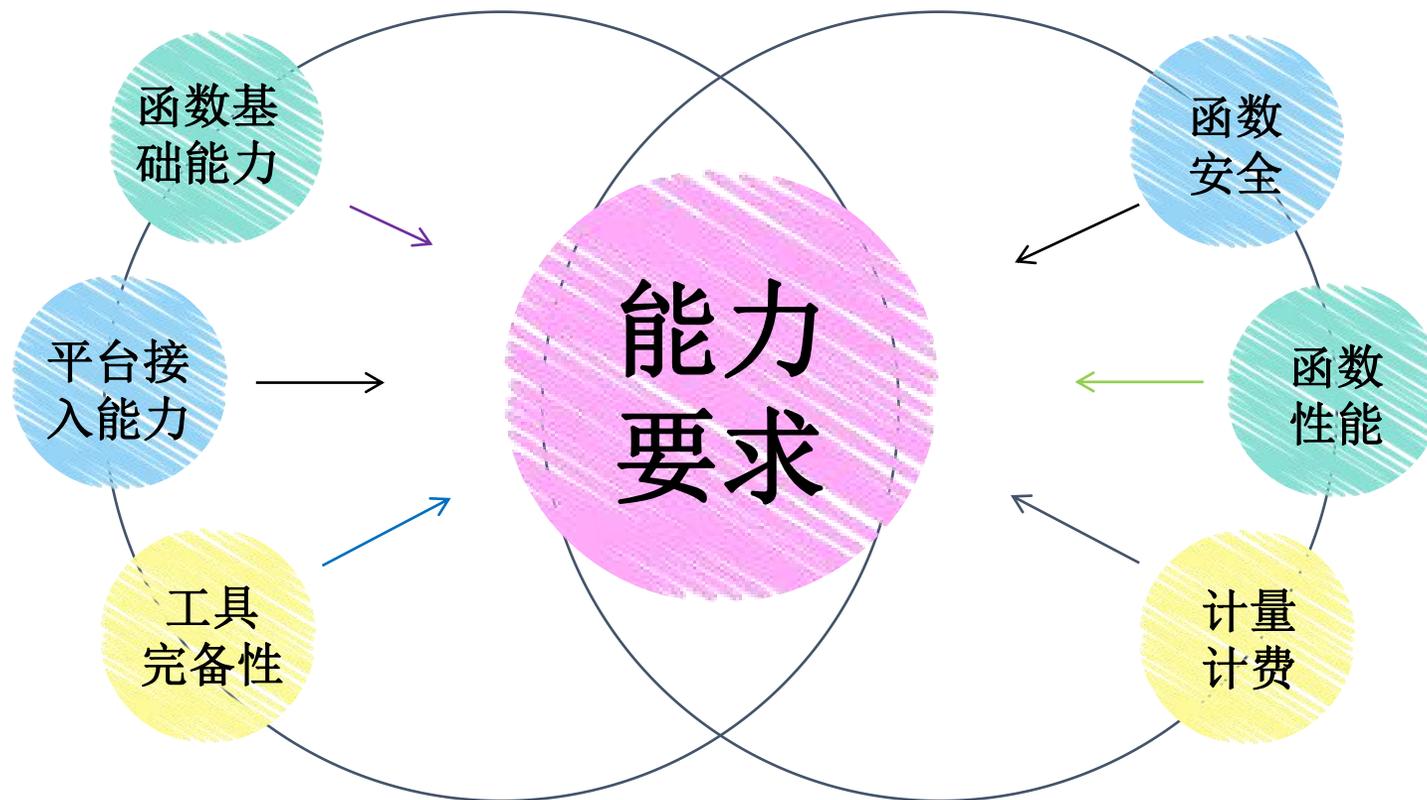
《函数即服务》标准解读

标准框架

细化指标能力要求

函数即服务能力要求

THREE



函数基础能力要求

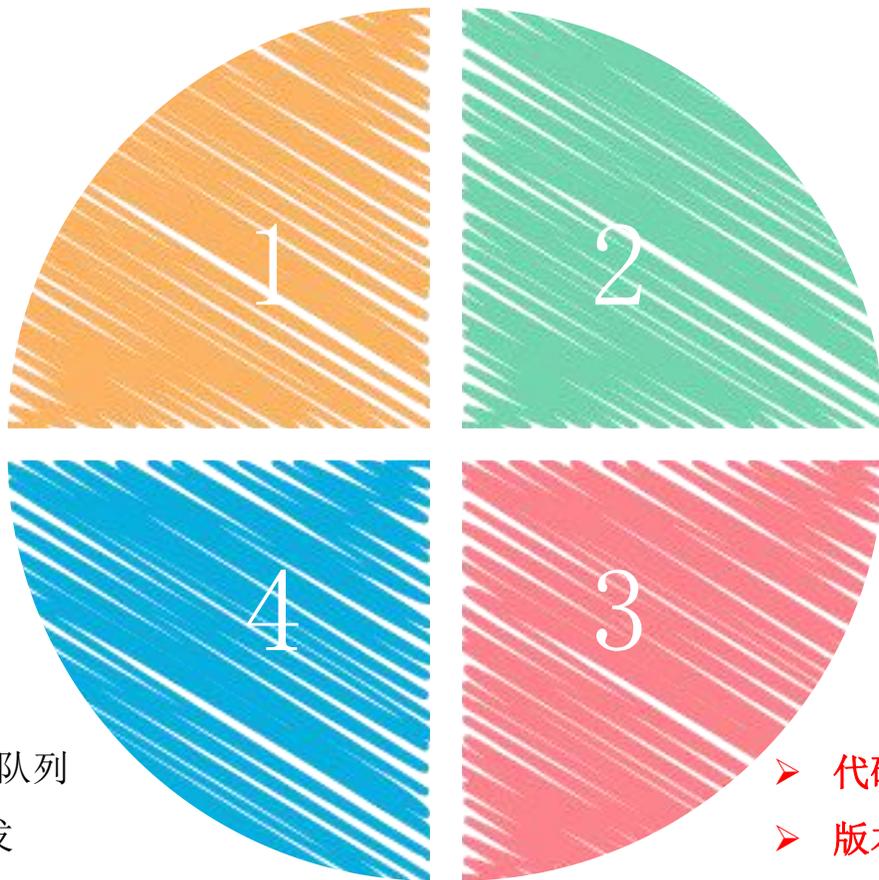
THREE

函数生命周期管理

- **函数创建:** 流程创建、模板创建
- **函数管理:** 增、删、改、查
- **函数调用:** 事件、函数间调用

事件触发

- **触发源:** API网关、Http、对象存储、消息队列
- **触发方式:** 同步触发、异步触发、定时触发
- **触发器管理:** 增删改、版本关联



函数配置

- **基本配置:** 内存、运行时、运行超限、角色、标签、死信队列
- **环境变量配置:** 变量的更改与设置、加密

代码管理

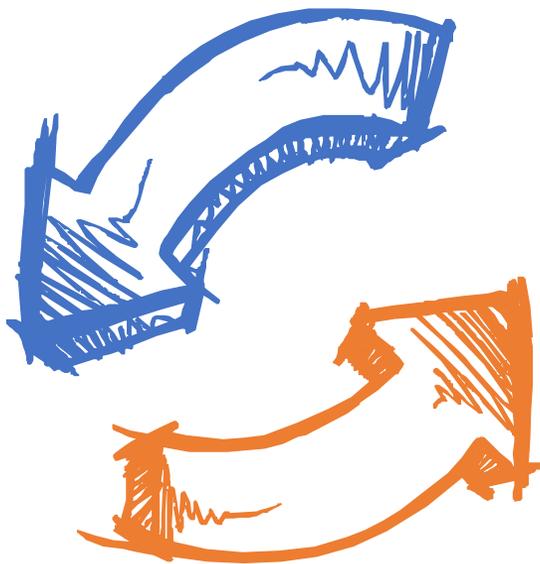
- **代码上传:** 控制台录入、zip包、OSS
- **版本管理:** 多版本并存、指定版本、别名、灰度流控
- **代码调试:** 在线、本地IDE、命令行、预览

网络接入能力要求

THREE

1 鉴权管理

- **基于身份的策略管理：**多用户角色创建、平台服务访问策略管理、服务、触发器访问控制
- **基于资源的策略管理：**函数权限的访问策略管理、向其他云服务或账户开放函数访问权限

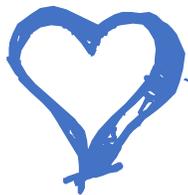


2 VPC网络接入

VPC网络访问：VPC网络访问控制、函数可访问VPC内网应用

工具完备性能力要求

THREE



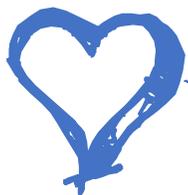
代码工具

支持SDK、WEB发布方式
支持命令行工具编写、调试、发布



日志工具

日志采集、聚合
日志标签
日志关键词搜索



函数监控工具

执行时间
并发执行状态
调用次数、错误次数
内存状态
函数计量数据



告警工具

自定义告警规则
邮件、短信等多渠道通知

安全性能力要求

THREE

租户和运行环境隔离

- 容器级别的租户运行环境隔离
- 虚拟机级别的租户运行环境隔离

函数运行时安全

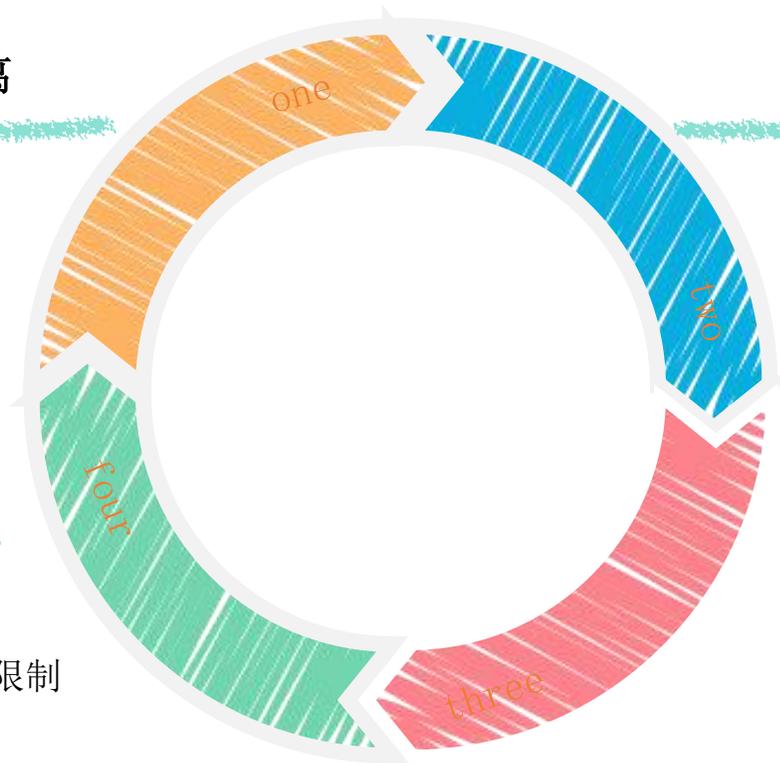
- 函数运行过程中程序系统调用权限限制
- 函数对系统资源访问权限的限制

代码安全

- 支持用户上传代码的加密存储
- 支持用户代码读取的鉴权管理

操作审计

- 函数运行的系统调用过程审计
- 用户行为审计



计费、性能、可用性能力要求

THREE

➤ 计费模型:

- 支持按调用次数的计量模型
- 支持内存*时长的计费模型
- 支持按流量计量计费模型

➤ 计费详单:

- 支持函数资源的度量统计
- 支持提供扣费明细及资源使用详单

1 | 计量计费能力

2 | 函数性能

3 | 平台可用性

- 公开SLA大于99.95%
- 支持错误重试机制
- 同步、异步错误重试

- 50M以内代码包的首次调用（同步调用）毫秒级响应
- 非首次的函数调用在30ms内响应
- -函数的异步调用在1小时内响应

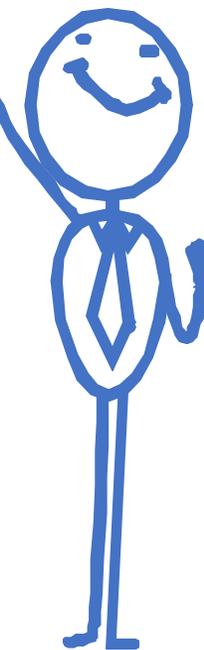
FOUR

后续工作计划

函数性能

容器性能

Service Mesh



后续工作安排

FOUR



函数性能能力要求

函数性能指标调研

框架搭建

测试基准制定



容器性能基准测试

基准测试框架研讨

测试基准制定



Service Mesh研究

白皮书

主流技术预研

标准规范

Contact Us

FOUR



刘如明

Email:

liuruming@caict.ac.cn

Nub:18500716640



无服务器架构
技术白皮书



云原生技术实践
白皮书

CAICT 中国信通院

7月3日-4日-中国·北京

2019云计算开源产业大会

THANKS