



DevOpsDays

Shanghai

— 2017.8.18-8.19 —

上海龙之梦酒店（长宁区延安西路1116号）

主办单位： 高效运维社区
GreatOPS Community  Best Practice
最佳实践





无服务器化的微服务持续交付

顾宇

ThoughtWorks高级咨询师



ThoughtWorks 高级咨询师。在 ThoughtWorks 分别以 微服务架构师、微服务咨询师、DevOps 咨询师、DevOps 工程师以及软件开发工程师的角色参与了国内外共计 7 个产品和项目的微服务咨询和实施。现在主要参与 ThoughtWorks 国内外交付项目解决方案的架构设计与实施，以及国内 DevOps 以及微服务相关咨询。并长期专注于 DevOps、微服务相关最佳实践以及全功能产品团队发展。

在加入 ThoughtWorks 之前，曾经参与中国移动 10086 呼叫中心以及中国联通省级 BOSS 系统的研发、实施和割接。曾任项目经理，维护经理，开发工程师等职务，拥有丰富的大型系统生产环境实战经验。

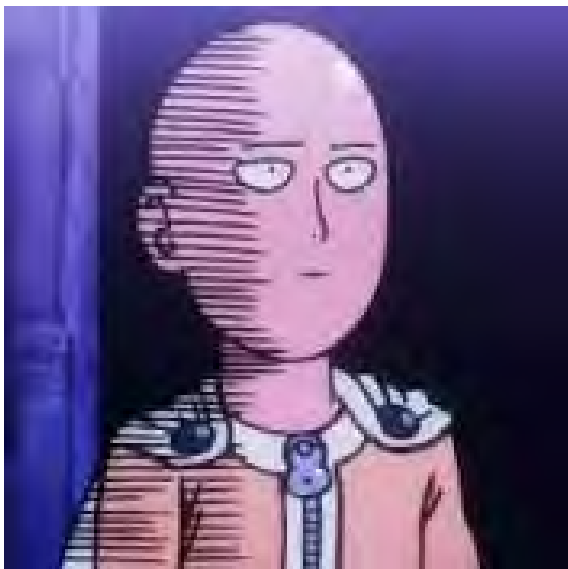
于 2015 年和 ThoughtWorks 其它同事翻译并出版《七周七 Web 框架》，并在等多个平台和媒体发表 DevOps 和微服务领域相关文章。



目录

- ➔ **1** 什么是无服务器架构？
- 2** 基于 AWS Lambda 的无服务器架构
- 3** 打造无服务器化的微服务持续交付流水线
- 4** 无服务器化的挑战
- 5** 云计算竞争的下一个阶段

什么是无服务器架构？



什么是无服务器架构？

一脸懵逼



<http://readwrite.com/2012/10/15/why-the-future-of-software-and-apps-is-serverless/>

应用要上线了，当 Ops 提出一堆问题



权限？

防火墙？

吞吐量？

SDK 版本？



数据库锁？

资源开销？

组策略？

性能？

安全补丁？

程序员眼里的生产环境



“资源是无限的” —— CPU，内存，磁盘等资源都是无限的

“老子就是 root，怎么着吧” —— 完全没有安全意识

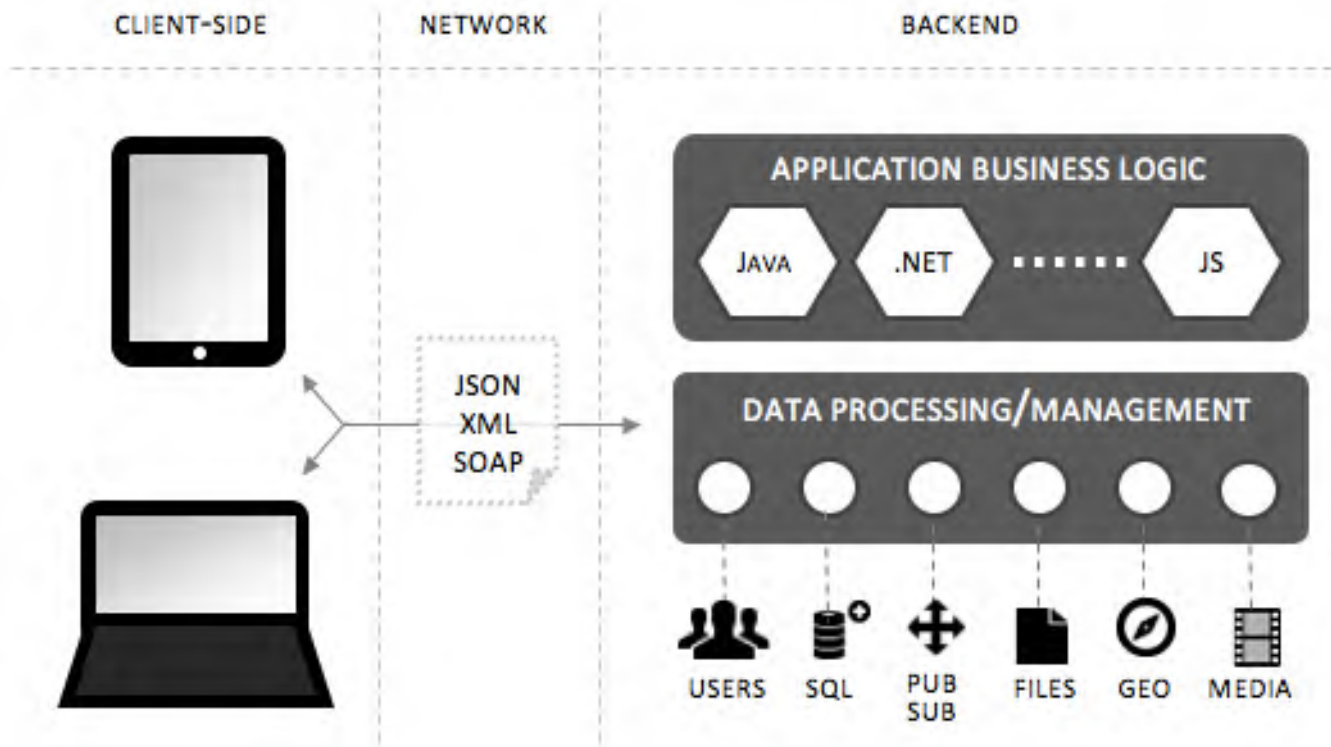
“在我机子上运行好着呢呀” —— 用自己的PC 当做生产环境

无服务器 —— 老子就想安安静静写代码



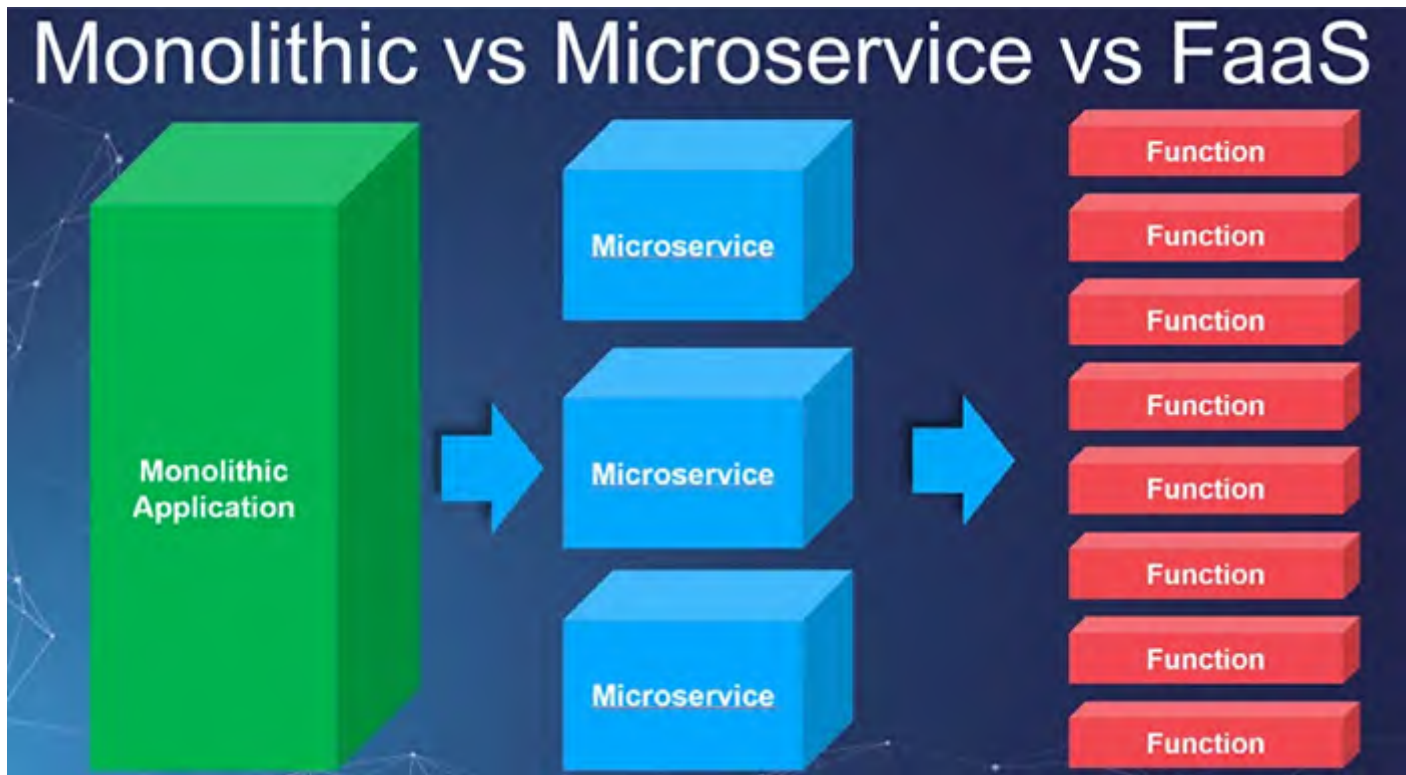
无服务器架构 —— 定义1：BaaS

后端应用即服务 —— 移动互联网的兴起



无服务器架构 —— 定义2：FaaS

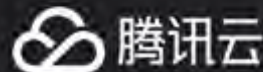
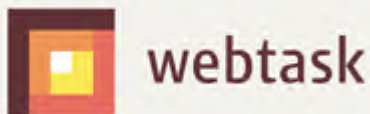
函数即服务 —— 云平台的极致



可以使用 Serverless 的云厂商



Google Cloud Platform



如果你想自己整一套



<https://github.com/apache/incubator-openwhisk>

目录

1 什么是无服务器架构？



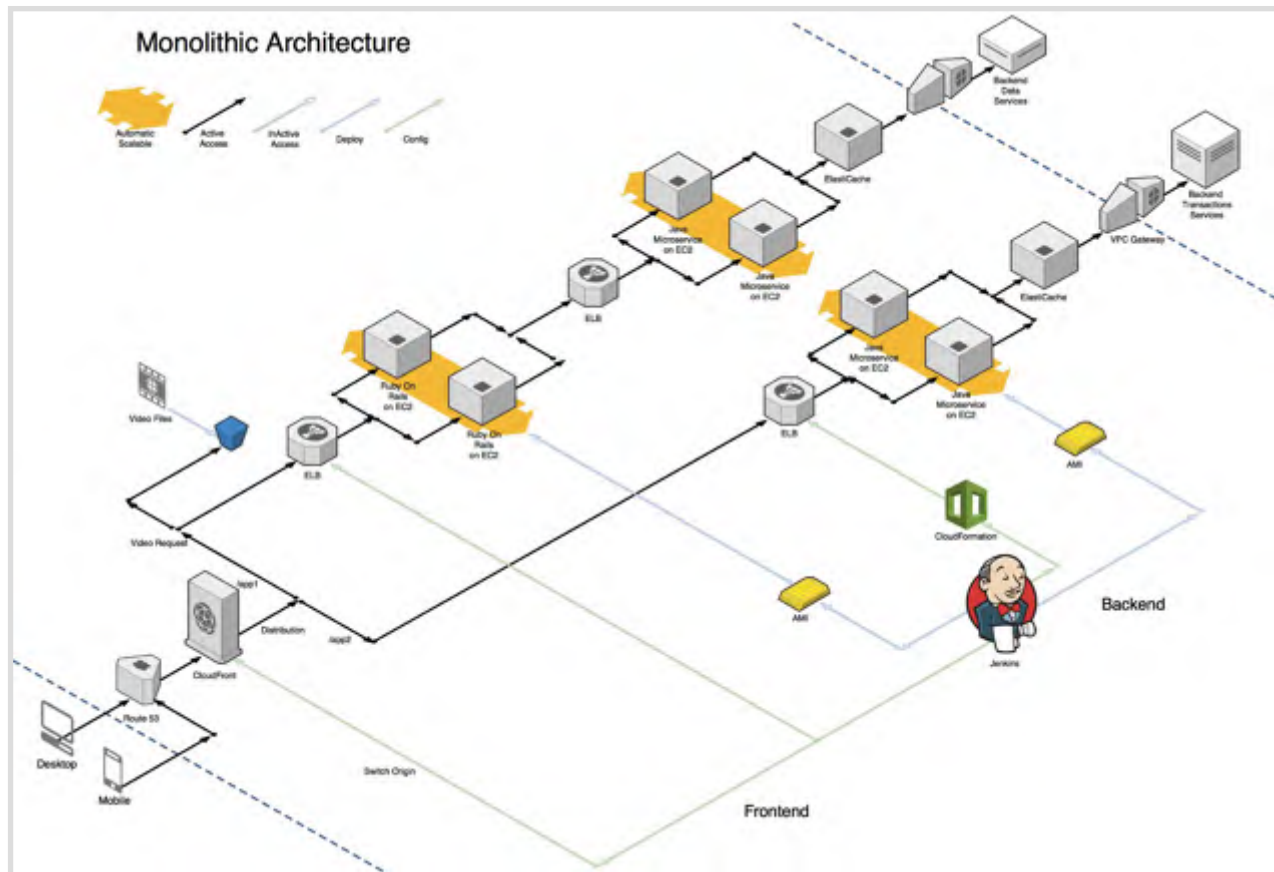
2 基于 AWS Lambda 的无服务器架构

3 打造无服务器化的微服务持续交付流水线

4 无服务器化的挑战

5 云计算竞争的下一个阶段

背景 —— 订单跟踪系统



架构痛点 —— 经典单体架构痛点



交付周期长 —— 一个功能上线需要经过 45 分钟

测试周期长 —— 无关的部分仍然需要测试, 20~27 分钟

部署周期长 —— 多层 EC2 实例部署, 部署需要15分钟

依赖功能多 —— 一个功能的改动阻塞其它部门开发和测试

架构师守则 No1 : 不要和康威定理对着干



单体架构就是一个灰犀牛问题





单体架构就是一个灰犀牛问题

罗马不是一天建成的，泥球也不是一天滚出来的

管理层和开发人员对架构风险和技术债的漠视

可靠性设计 并不是 过度设计

组织痛点 —— Ops 成本极大



AWS 需要专业的知识和培训

Ops 用的工具远多于编程语言

Ops 拥有着阻碍 DevOps 的权限

DevOps Team 在一定程度上会变成 DevOps反模式

什么时候拆微服务？



系统边际功能的交付成本 \geq 系统的总维护成本

组织痛点

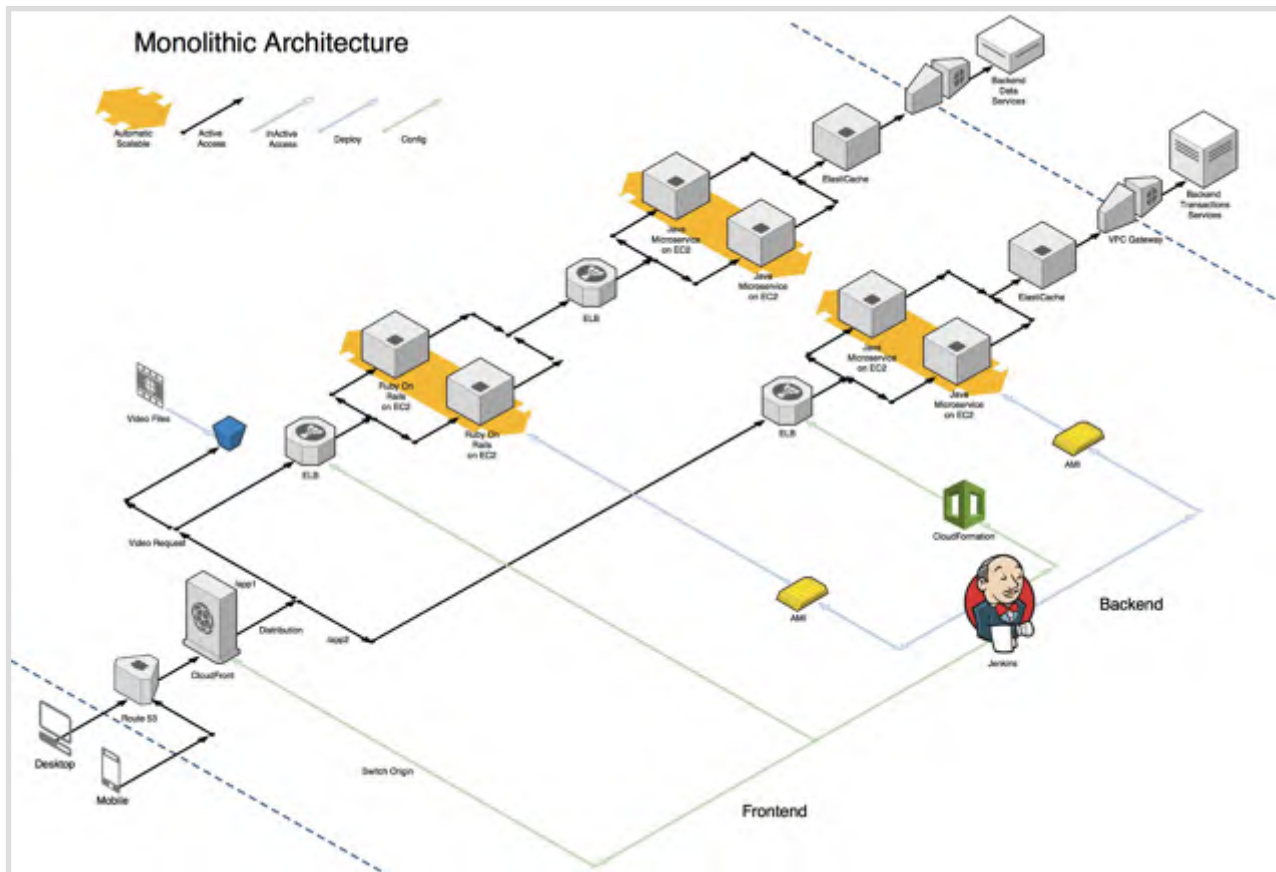


架构痛点

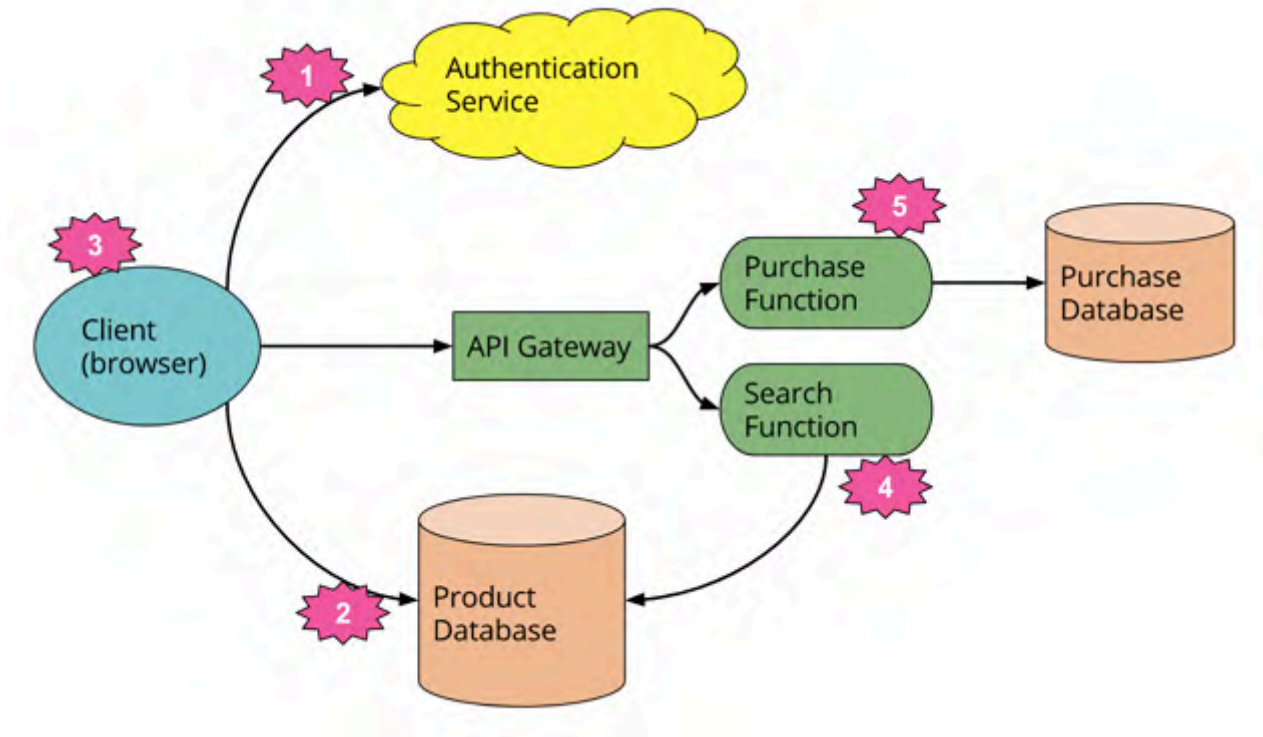


部署周期太长

订单跟踪系统



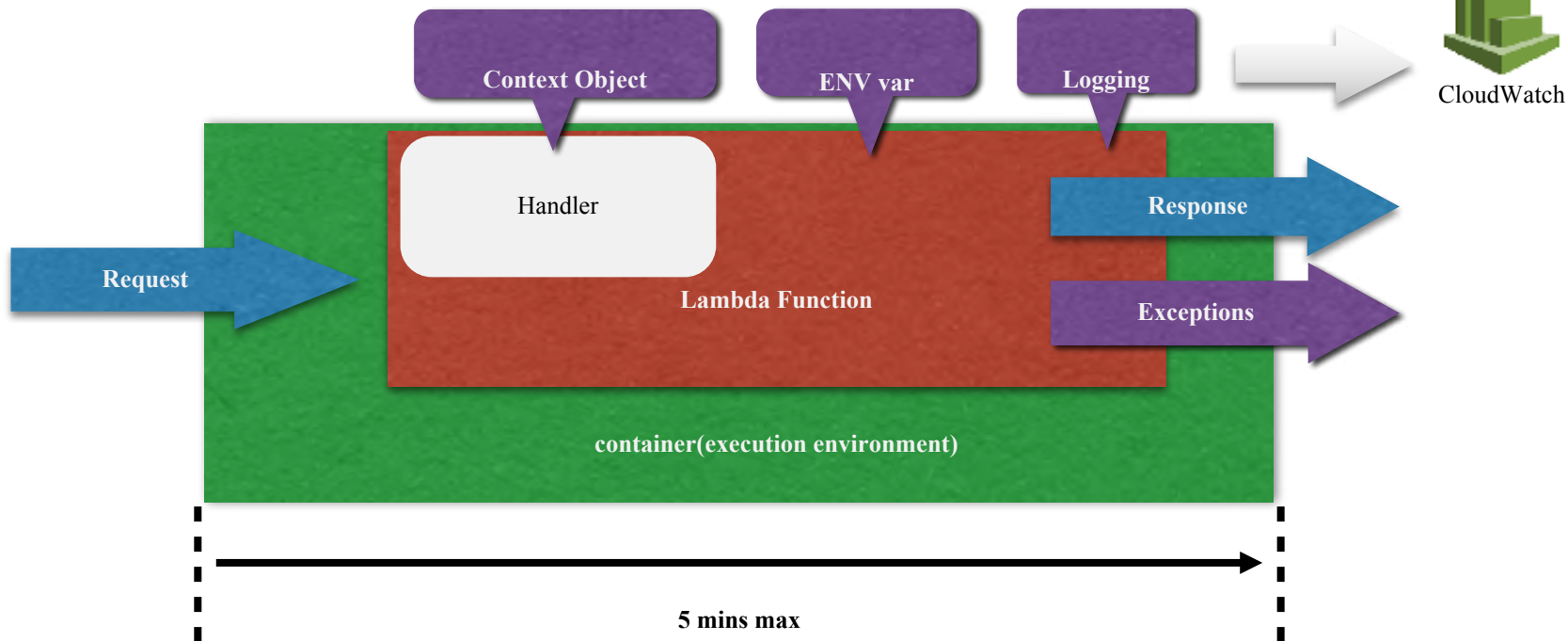
常见架构：用户界面驱动的应用程序



常见架构：消息驱动的应用程序



AWS Lambda 的编程模型

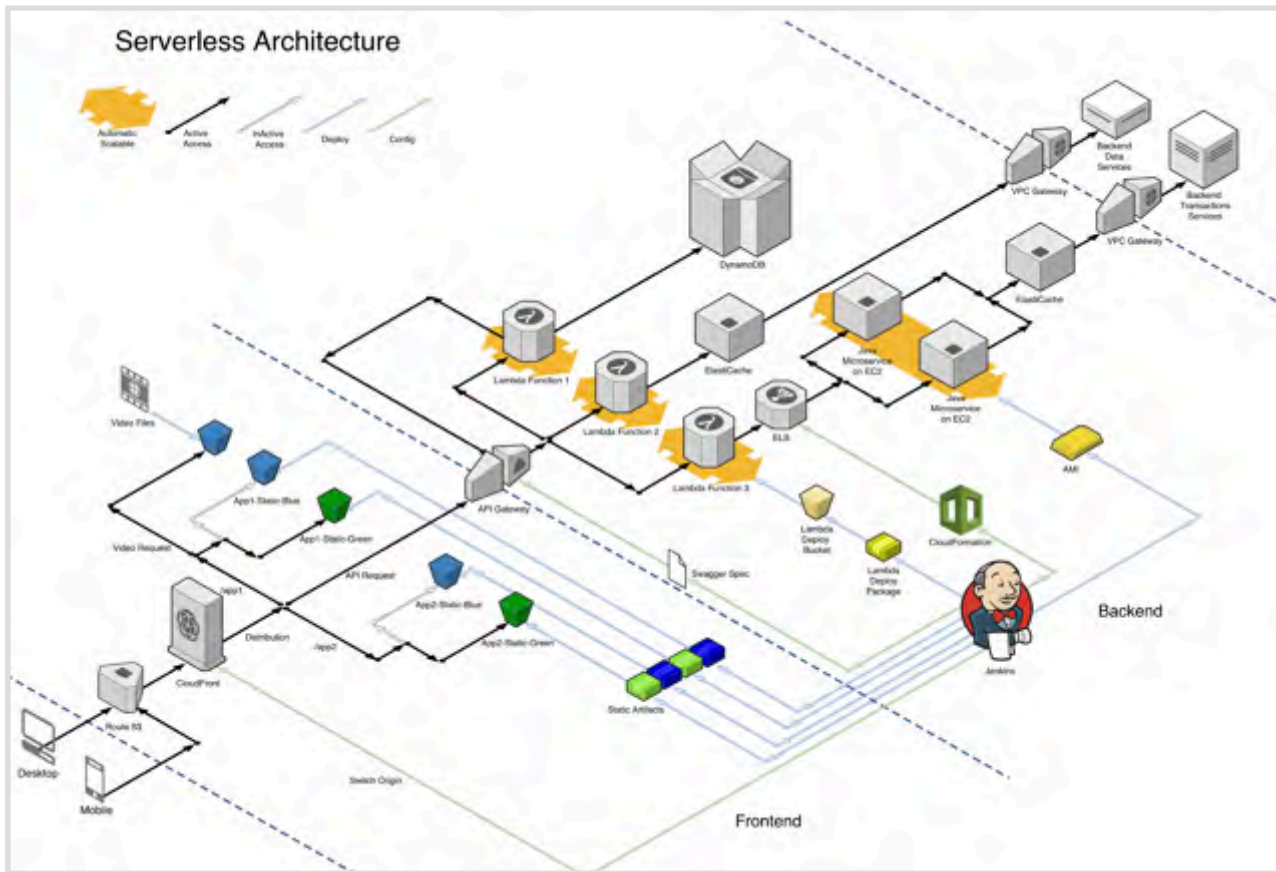


AWS Lambda 简介 —— 优势



- 无需初始化和管理服务器。
- 按请求收费，不用考虑负载均衡。
- 根据代码运行时间付费。
- 自动处理内存，CPU，网络和其它资源。
- 包括了初始化，监控，自动打安全补丁，日志。

基于 AWS Lambda 的无服务器架构



目录

1 什么是无服务器架构？

2 基于 AWS Lambda 的无服务器架构

→ 3 打造无服务器化的微服务持续交付流水线

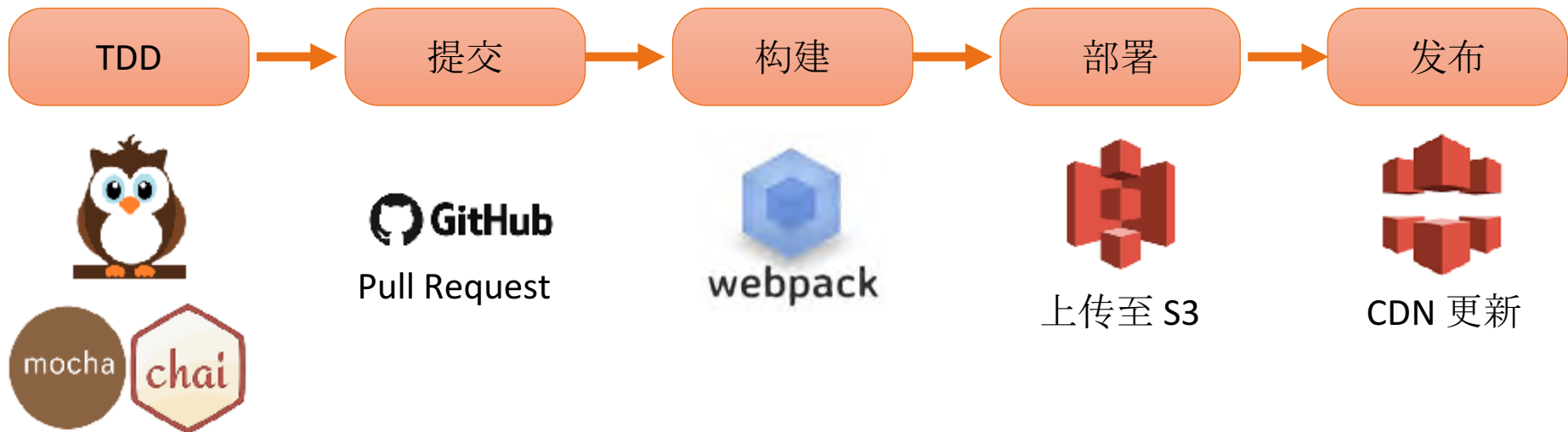
4 无服务器化的挑战

5 云计算竞争的下一个阶段

基于的无服务器架构流水线总览



基于 AWS Lambda 的无服务器架构 —— 前端



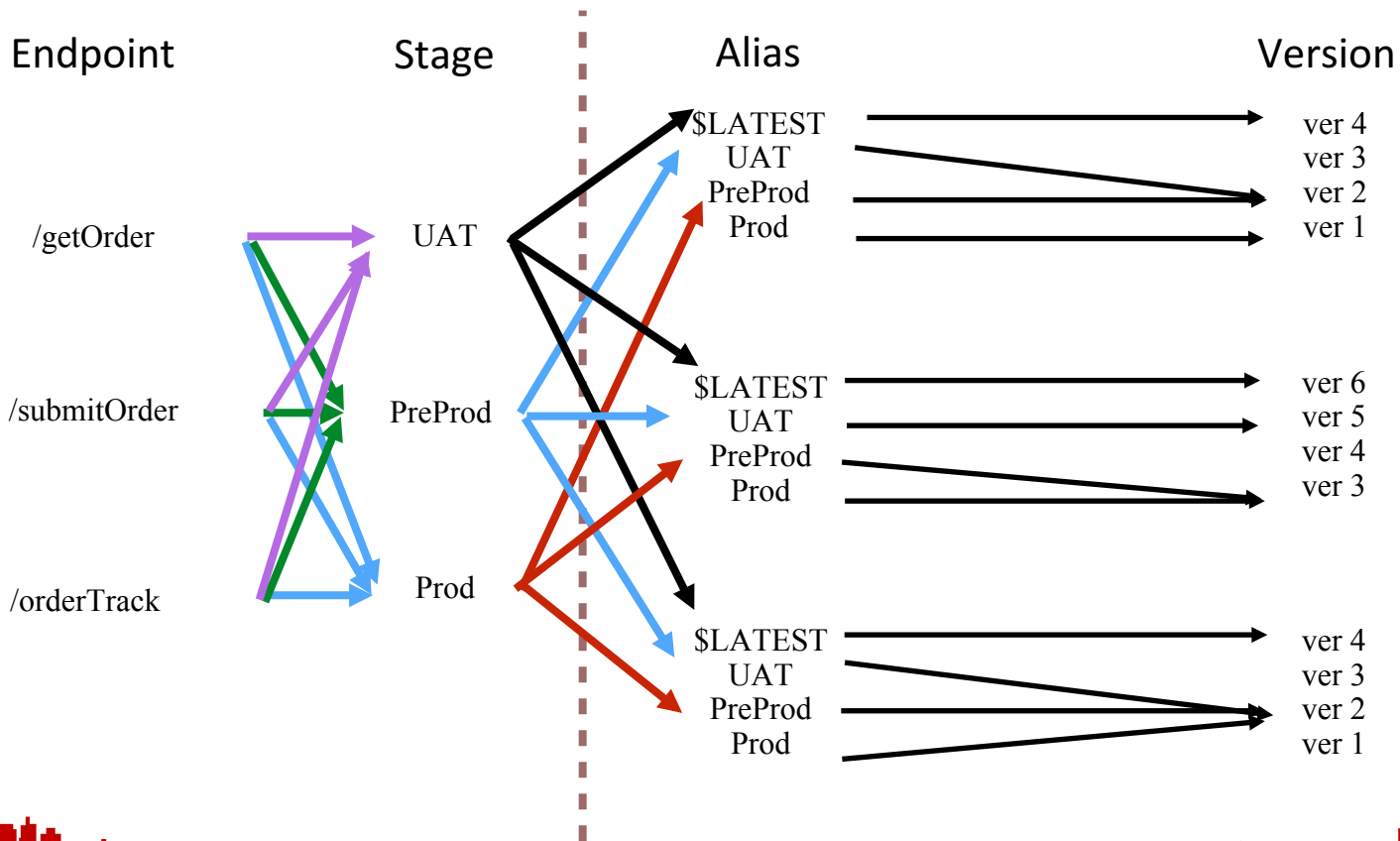
基于 AWS Lambda 的无服务器架构 —— 后端



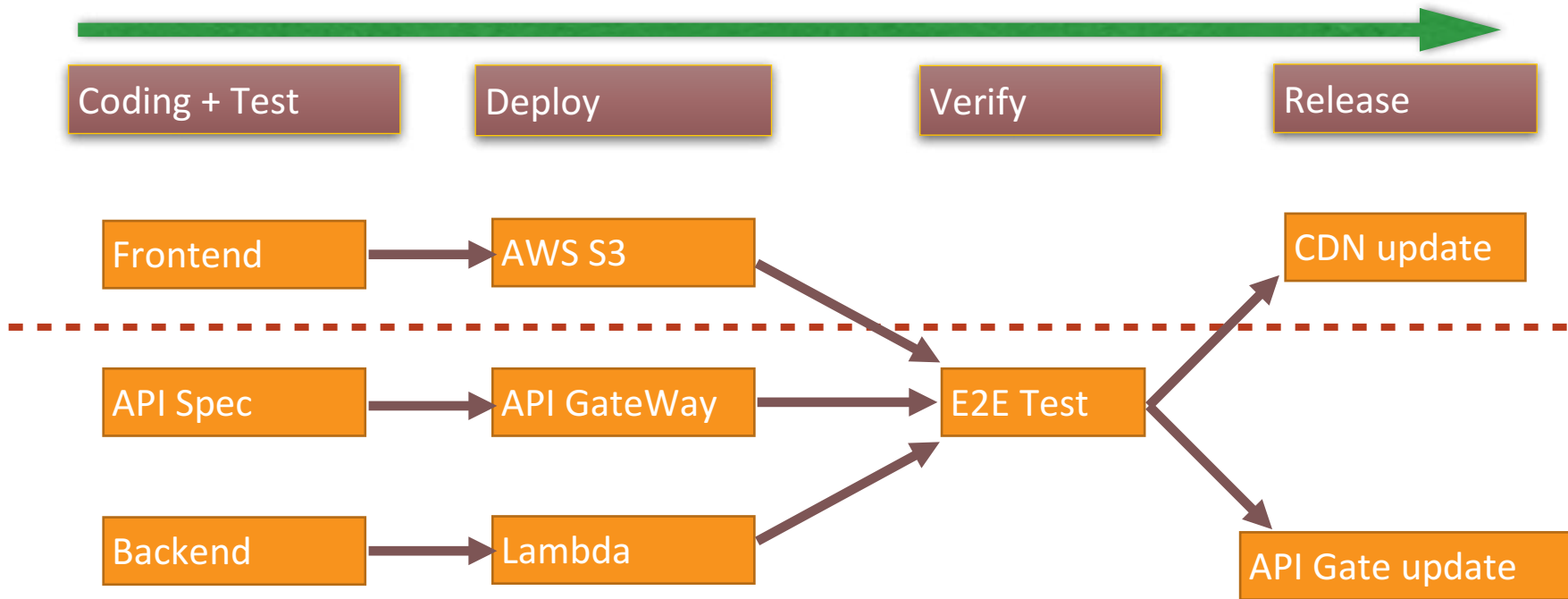
Pull Request



构造无服务器架构的流水线



构造无服务器架构的流水线



整个过程只有一台服务器：CI




Jenkins

LambCI - 无服务器持续集成



LambCI 

Serverless continuous integration

 New York

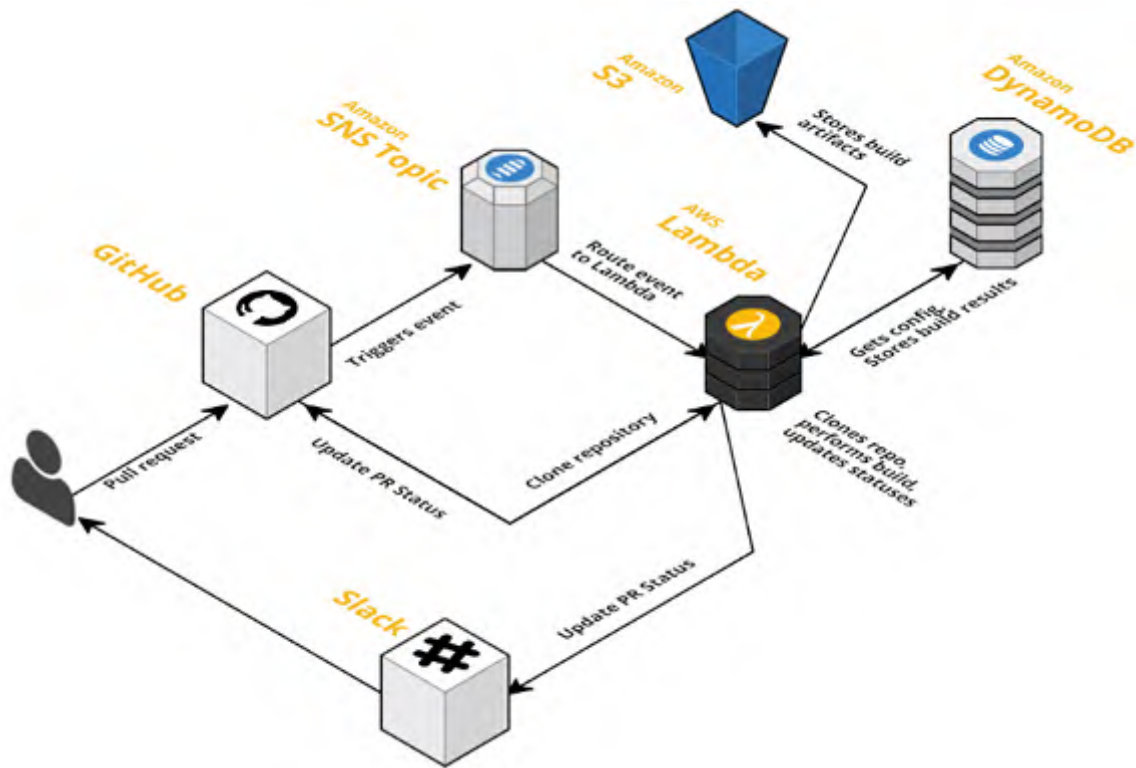


<http://lambci.org>



info@lambci.org

LambCI - 无服务器持续集成



效率的提升



理论开发效率提升了至少 8 倍 = 16 人月 (8人8周) to 2 人月 (4人2周)

基础设施即代码配置减少了 82% = 4732 行 to 822 行

从提交到部署的时间缩短了 84.4% = 45min to 7 min

基础设施导致的故障率降低了 % ?

Ops 的工作剩下了三件事



架构



流水线



监控

目录

1 什么是无服务器架构？

2 基于 AWS Lambda 的无服务器架构

3 打造无服务器化的微服务持续交付流水线

➔ 4 无服务器化的挑战

5 云计算竞争的下一个阶段

函数的开发

函数的开发测试部署非常快，几乎不需要等待。

用更多的时间沟通，用更少的时间编程。需求分析完，功能就开发完了。

更简单和面向资源的抽象方式。 (What - How - Where)

对复杂问题的拆解更简单。

Serverless



SERVERLESS FRAMEWORK VERSION 1.0

Build auto-scaling, pay-per-execution,
event-driven apps on AWS Lambda

▶ WATCH THE VIDEO

📖 READ THE DOCS

```
# Install serverless globally
$ npm install serverless -g

# Create an AWS Lambda function in Node.js
$ serverless create --template aws-nodejs

# Deploy to live AWS account
$ serverless deploy

# Function deployed!
$ http://api.amazon.com/users/update

-> Read the docs or connect with the community
```

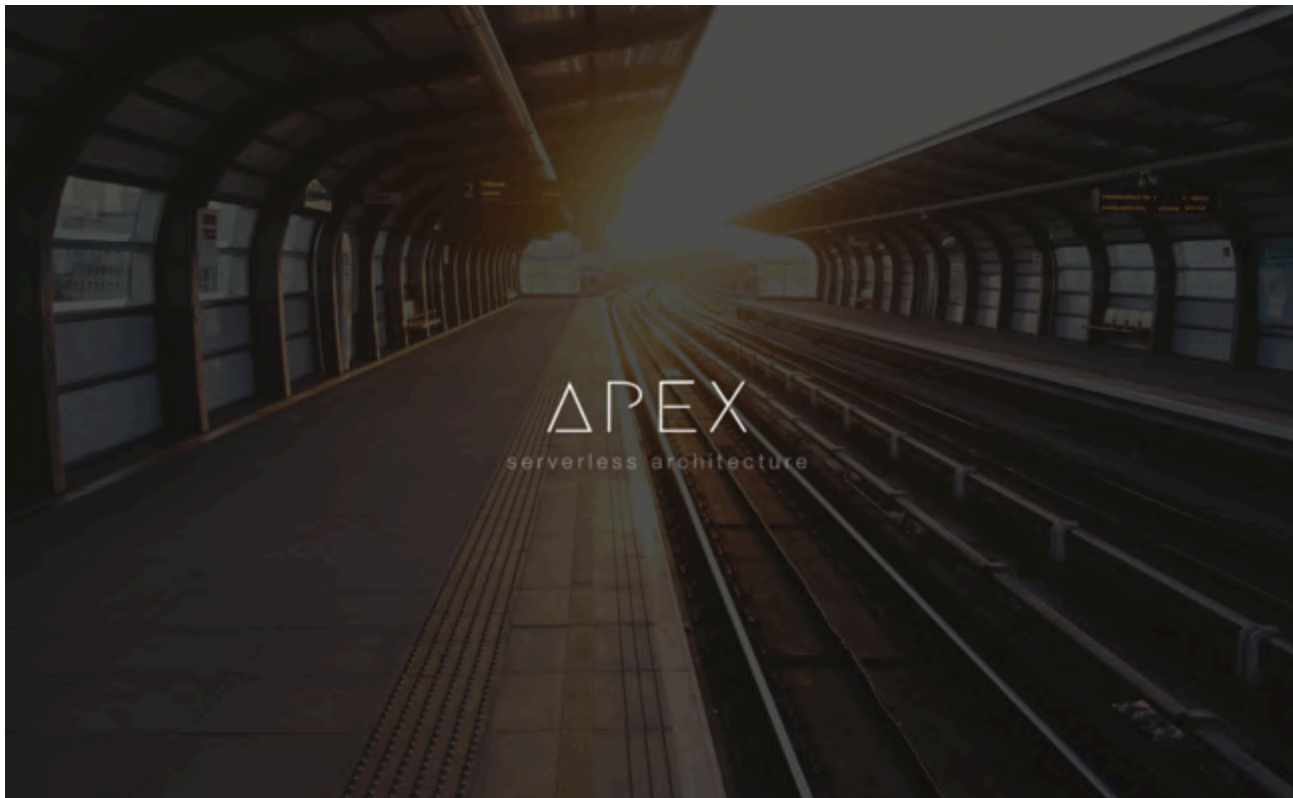
Powered by AWS Lambda



Claudia.js



Apex



测试



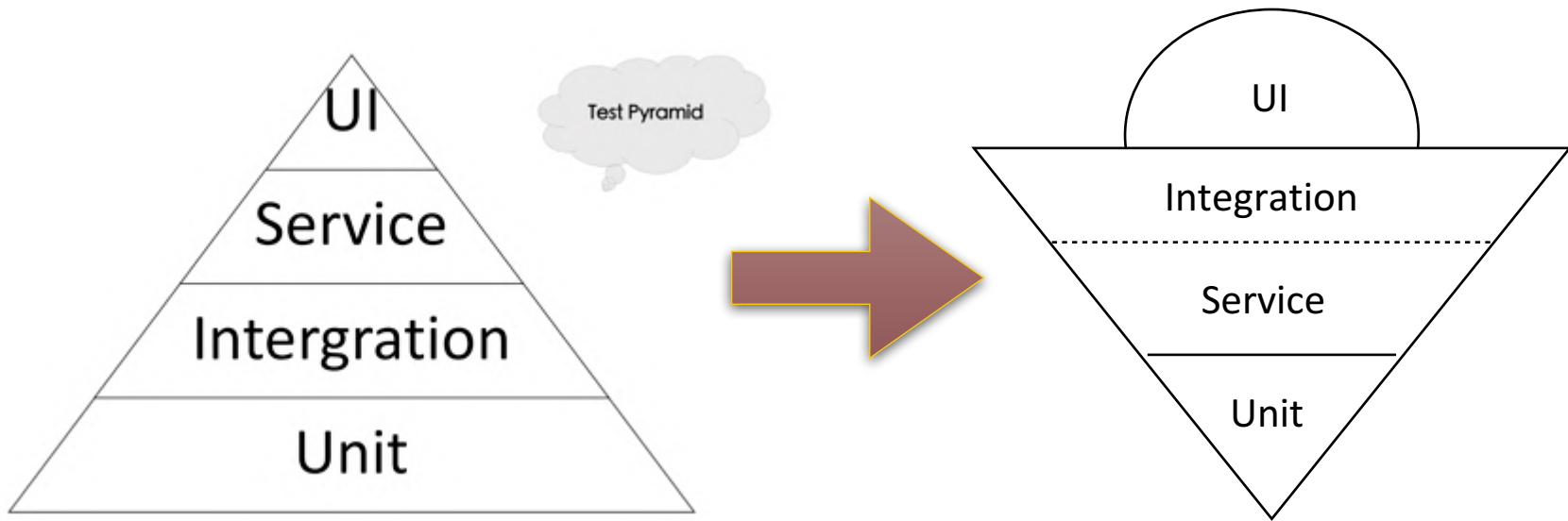
BDD/ATDD 更加重要。

单个函数虽然很好测试，但函数的集成不一定好测试。

所有的资源都在线上，你不需要 Mock 了。

从测试金字塔到测试冰激凌。

测试 —— 测试金字塔到测试甜筒



测试 —— 测试金字塔到测试甜筒



BDD/ATDD 更加重要，因为会带来更多的价值。

单个函数虽然很好测试，但函数的集成不一定好测试。

在线直接测试 不要尝试去 Mock

更多的精力在于和 AWS 的集成测试

配置管理

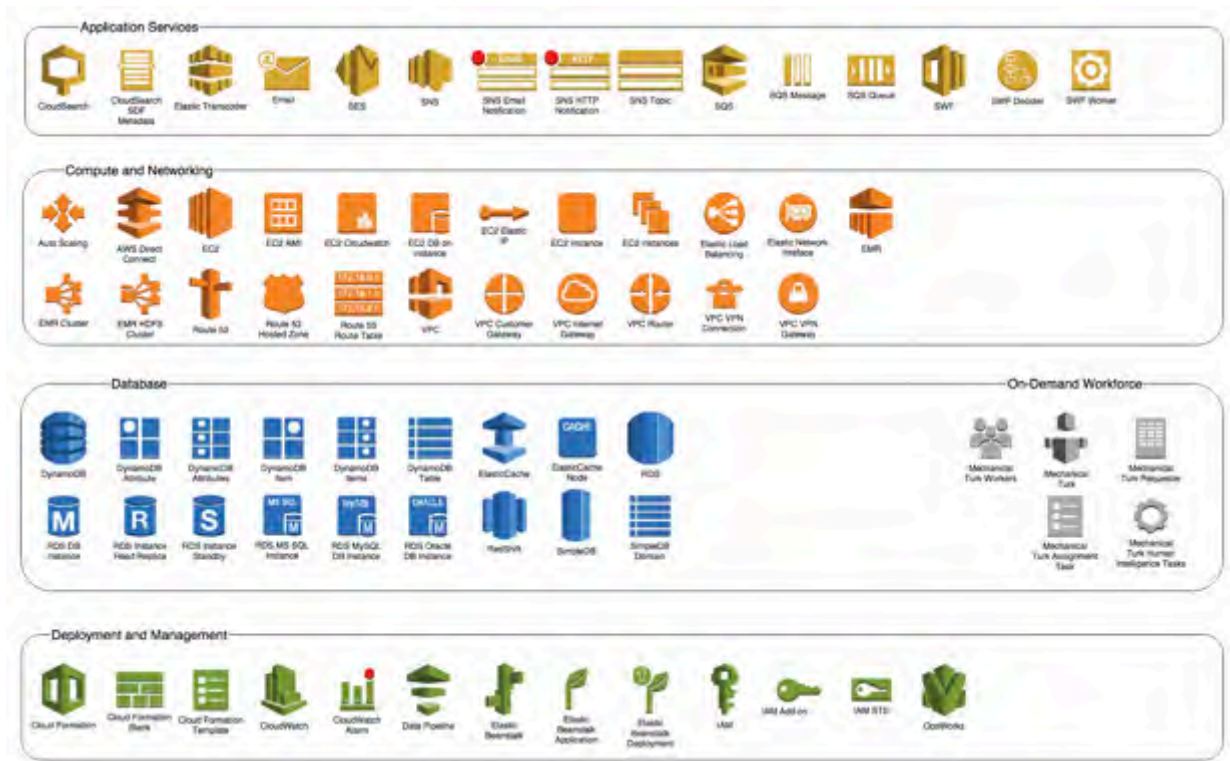


Serverless 的架构即函数 + 资源配置

微服务作为封装一组函数和资源的单元

由于粒度更小，对基础设施依赖更大，配置管理的难度更大。

云厂商的重度依赖



The image displays a collection of AWS service icons organized into five categories:

- Application Services:** CloudSearch, CloudSearch SF Messages, Elastic Transcoder, Email, SES, SNS, SNS Email Notification, SNS HTTP Notification, SNS Topic, SQS, RDS Message, RDS Queue, SWF, SWF Databse, SWF Worker.
- Compute and Networking:** Auto Scaling, AWS Direct Connect, EC2, EC2 AMI, EC2 Cloudwatch, EC2 DB on-instance, EC2 Elastic IP, EC2 Instance, EC2 Instance Profile, Elastic Load Balancing, Elastic Network Interface, EMR, EMR Cluster, EMR HDFS Cluster, Route 53, Route 53 Hosted Zone, Route 53 Route Table, VPC, VPC Customer Gateway, VPC Internet Gateway, VPC Router, VPC VPN Connection, VPC VPN Gateway.
- Database:** DynamoDB, DynamoDB Attribute, DynamoDB Attribute, DynamoDB Item, DynamoDB Item, DynamoDB Table, ElastiCache, ElastiCache Node, RDS, RDS DB Instance, RDS Instance Heat Replica, RDS Instance Standby, RDS MS SQL Instance, RDS MySQL DB Instance, RDS Oracle DB Instance, Redshift, S3, S3 Glacier, S3 Glacier Deep Archive.
- On-Demand Workforce:** Mechanical Turk Workers, Mechanical Turk, Mechanical Turk Requester, Mechanical Turk Assignment Tool, Mechanical Turk Human Intelligence Tasks.
- Deployment and Management:** CloudFormation, CloudFormation Stack, CloudFormation Template, CloudWatch, CloudWatch Alarm, Data Pipeline, Elastic Beanstalk, Elastic Beanstalk Application, Elastic Beanstalk Deployment, IAM, IAM Add-on, IAM STS, OpsWorks.

目录

1 什么是无服务器架构？

2 基于 AWS Lambda 的无服务器架构

3 打造无服务器化的微服务持续交付流水线

4 无服务器化的挑战

➔ 5 云计算竞争的下一个阶段

Serverless 架构的核心

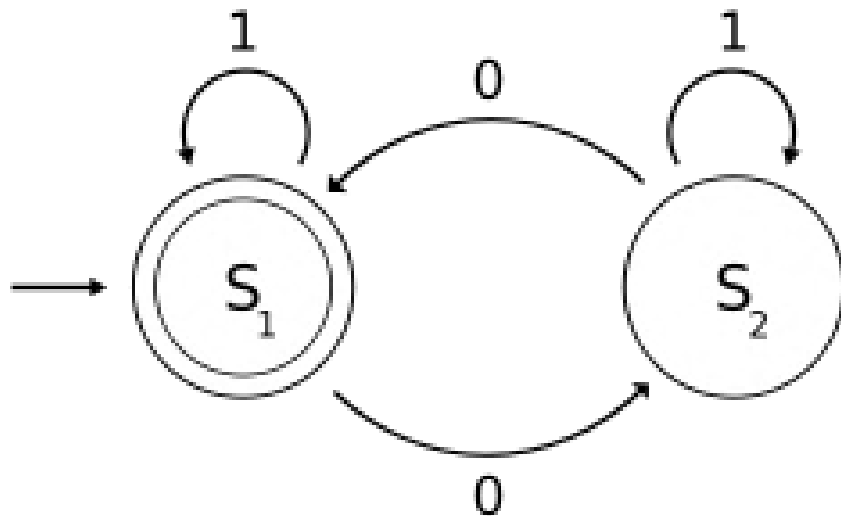


服务即状态机 (Service as a state machine)

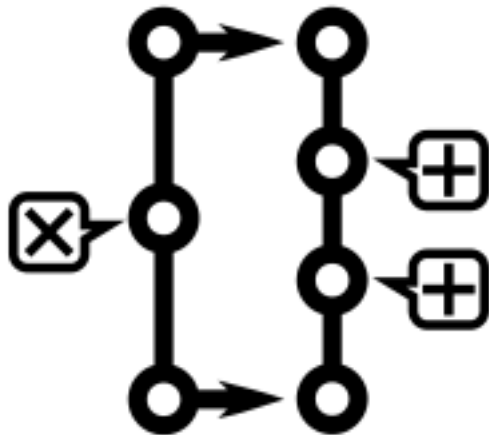
状态即版本化配置 (Status as versioning configs)

状态转换即事件消费 (Status transition as events consuming)

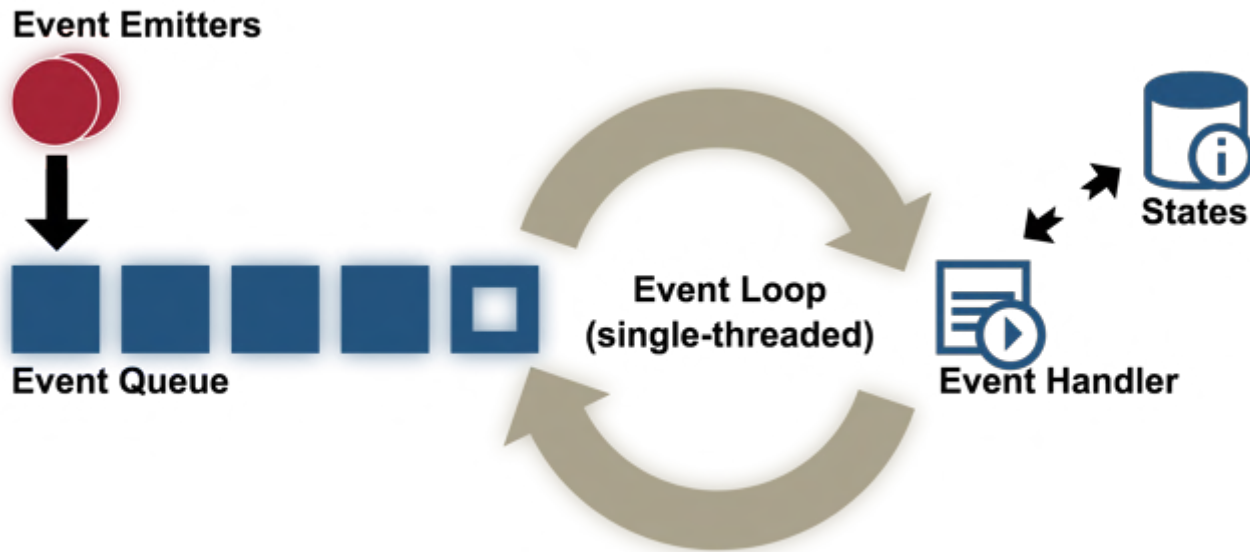
服务即状态机 (Service as a state machine)



状态即版本化配置 (Status as versioning configs)



状态转换即事件消费 (Status transition as events consuming)



云计算的未来是 配置 + 资源



代码本身就是一种配置

未来的应用是配置和资源的组合

高度自动化和抽象化的关键在于制定 CloudNative 的 DSL

平台即编程模型



现在的编程模型由编程语言平台决定，语言平台又和操作系统以及硬件相关联。

未来的编程模型由云平台决定，对于资源和资源的使用方式的管理会更加方便。

平台即编程模型举例：AWS IoT





高效运维社区
GreatOPS Community



会议

- 8月18日 DevOpsDays 上海
- 全年 DevOps China 巡回沙龙
- 11月17日 DevOps金融上海

培训

- EXIN DevOps Master 认证培训
- DevOps 企业内训
- DevOps 公开课
- 互联网运维培训

咨询

- 企业DevOps 实践咨询
- 企业运维咨询



商务经理：刘静女士
电话 / 微信：13021082989
邮箱：liujing@greatops.com



Thanks

荣誉出品

高效运维社区

国际最佳实践管理联盟



想第一时间看到
高效运维社区公众号
的好文章吗？

请打开高效运维社区公众号，点击右上角小人，如右侧所示设置就好

