

无服务器计算架构

Victor Meng 蒙维
亚马逊AWS 解决方案架构师
weimen@amazon.com

议程

什么是无服务器计算？

无服务器计算现状

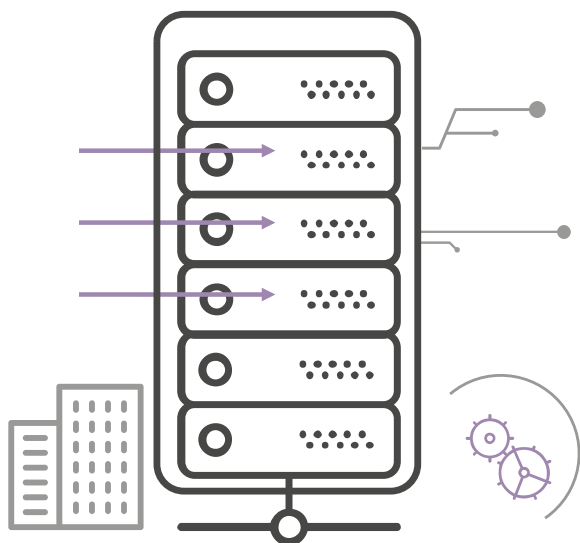
实现和典型案例

什么是 无服务器计算Serverless?

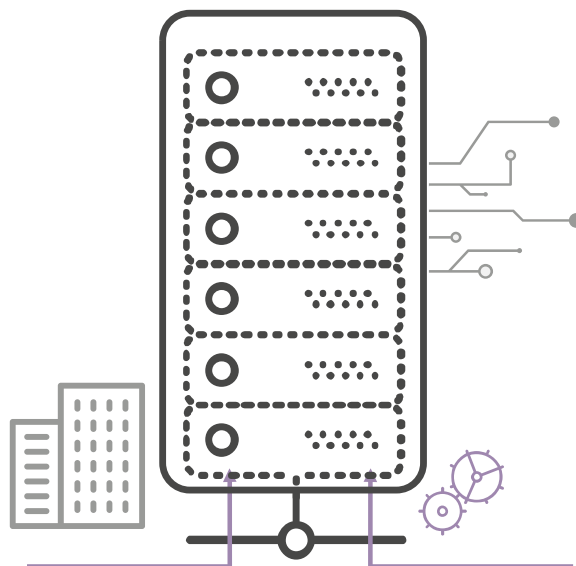
构建和运行应用程序的时候不用考虑底层的服务器资源分配和管理

让我们回顾一下计算服务的架构演进过程

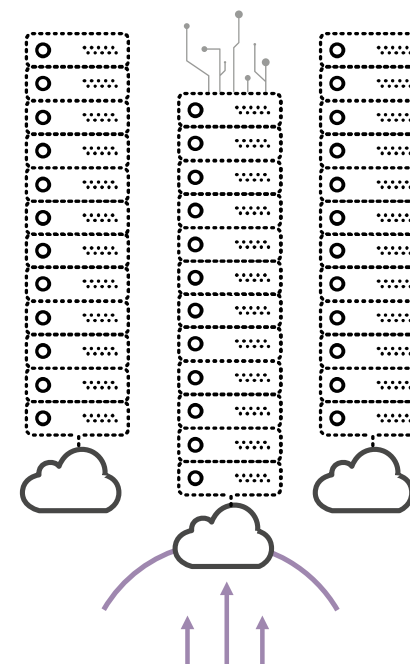
运行在本地数据中心中的物理服务器



运行在本地数据中心的虚拟服务器



运行在云端的虚拟服务器

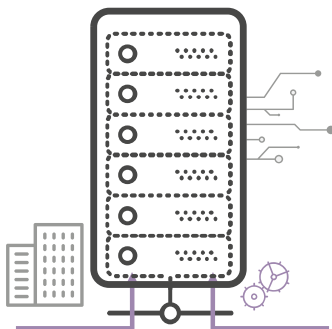


如今演进到 无服务器计算架构

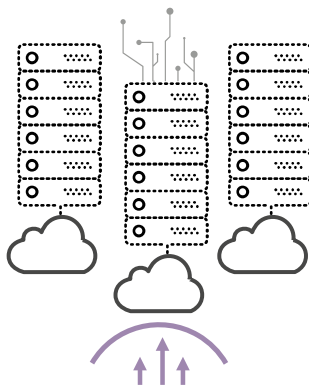
运行在本地数据中
心中的物理服务器



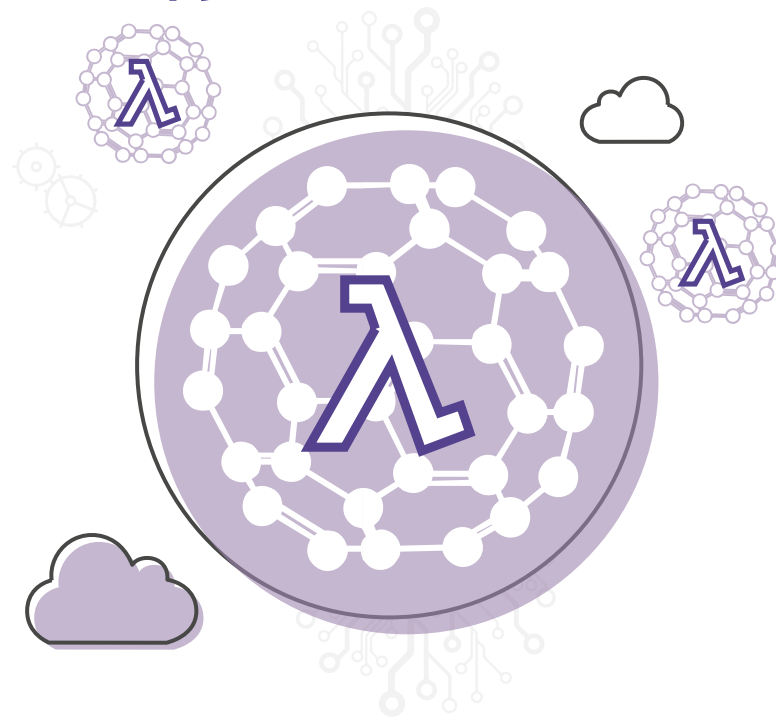
运行在本地数据中
心的虚拟服务器



运行在云端的虚拟
服务器



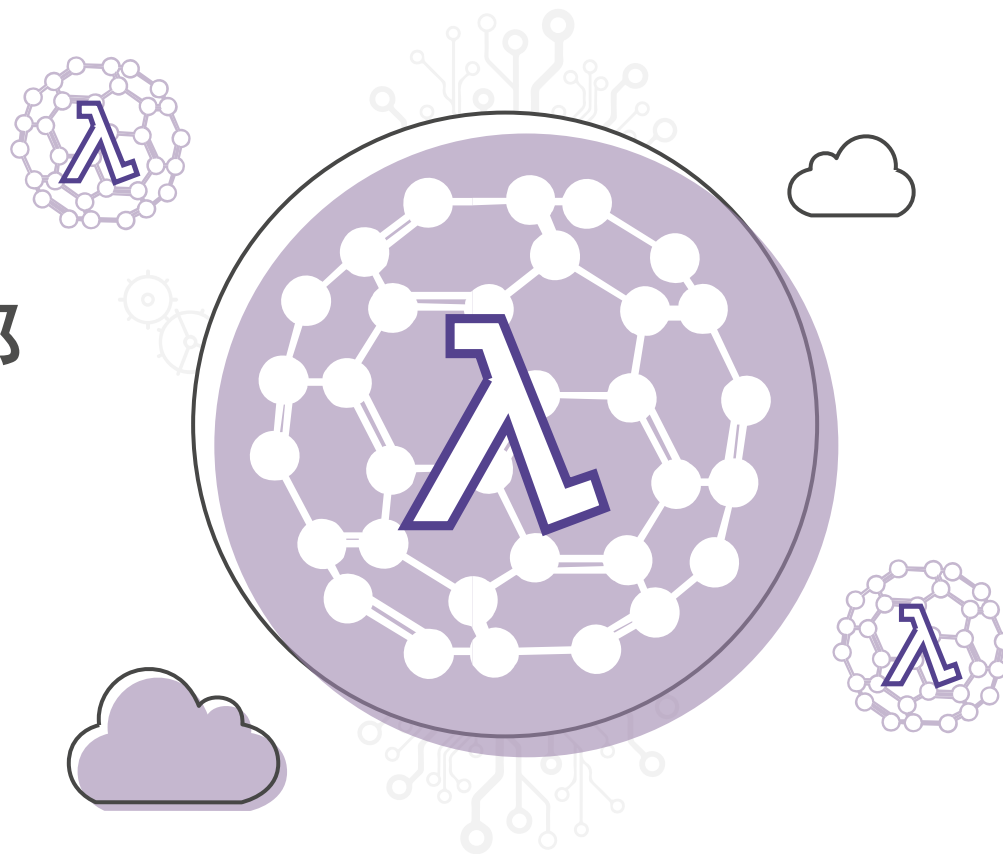
无服务器计算架 构SERVERLESS



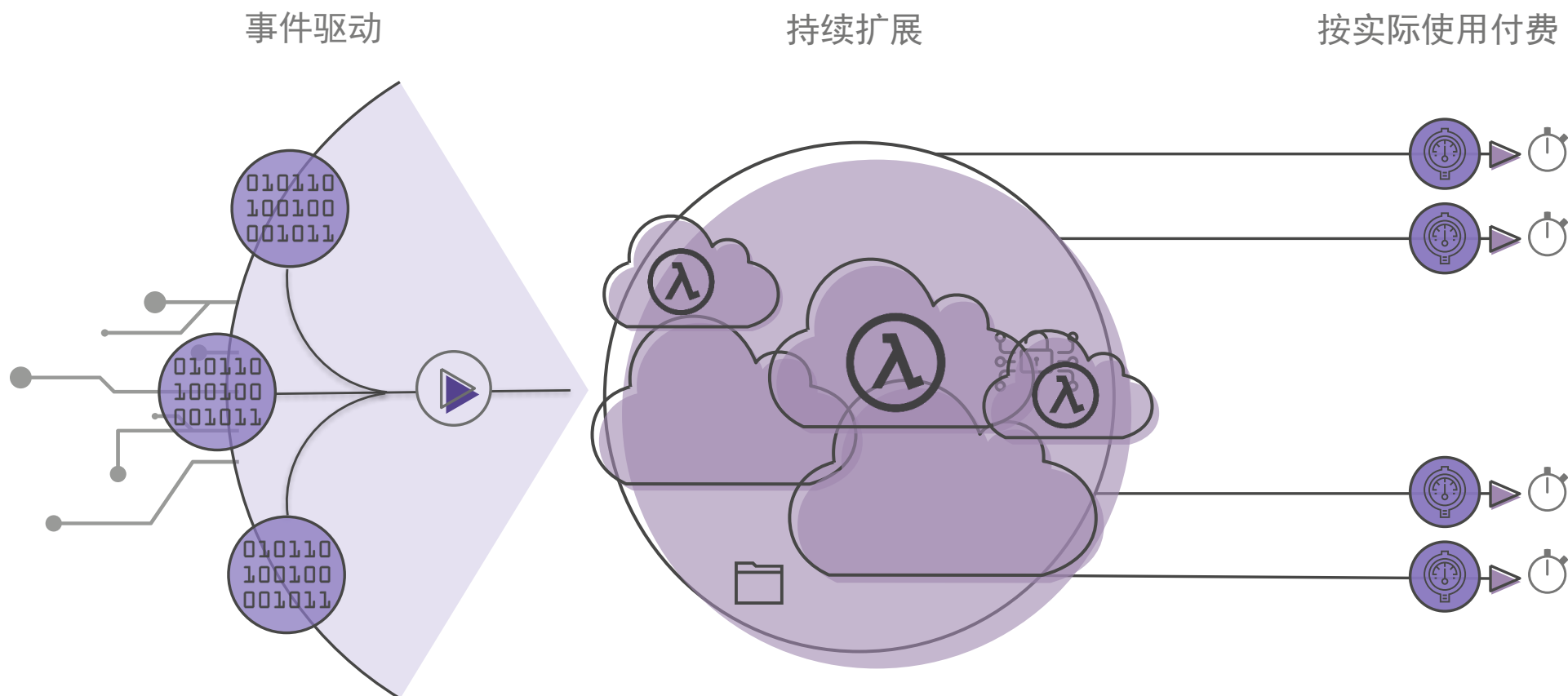
无服务器计算其实是指“没有服务器需要管理”而不是“没有运行服务器”

所有的运行环境管理任务都消失了

- ~~提供和使用~~
- ~~高可用和容错~~
- ~~扩展~~
- ~~操作管理~~



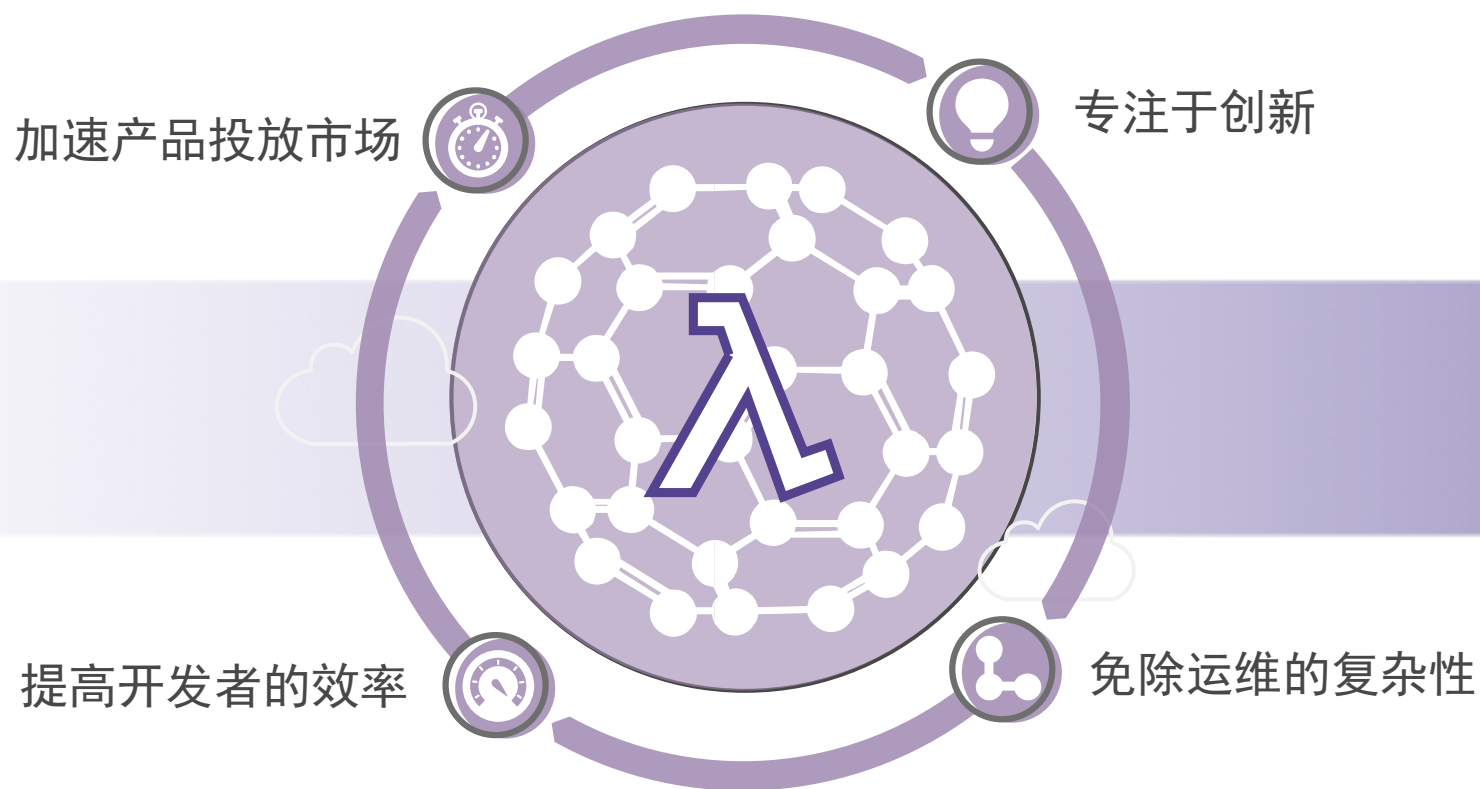
按实际需求提供计算能力, 不再为空闲资源付费



无服务器应用程序的构建工具集

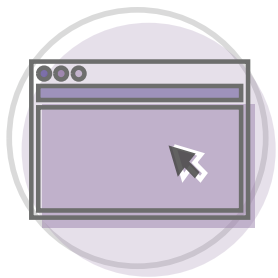
计算	存储	数据库
 AWS Lambda	 Amazon S3	 Amazon DynamoDB
API代理	消息和通知	数据分析
 Amazon API Gateway	 Amazon SQS  Amazon SNS	 Amazon Kinesis
任务编排和状态管理	监控和调试	
 AWS Step Functions	 AWS X-Ray	

无服务器计算架构会改变 您的应用交付方式



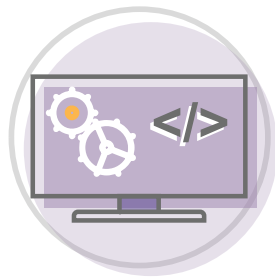
无服务器计算现状

典型的应用场景



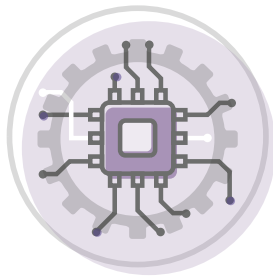
Web应用

- Static websites
- Dynamic web apps
- Packages for Flask and Express



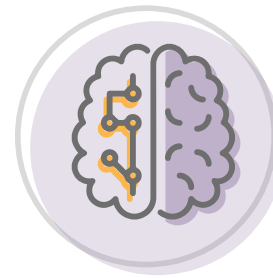
应用后台实现

- Apps & services
- Mobile
- IoT



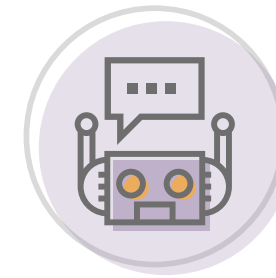
数据处理

- Real time
- MapReduce
- Batch



亚马逊Alexa

- Powering voice-enabled apps
- Alexa Skills Kit



聊天机器人

- Powering chatbot logic

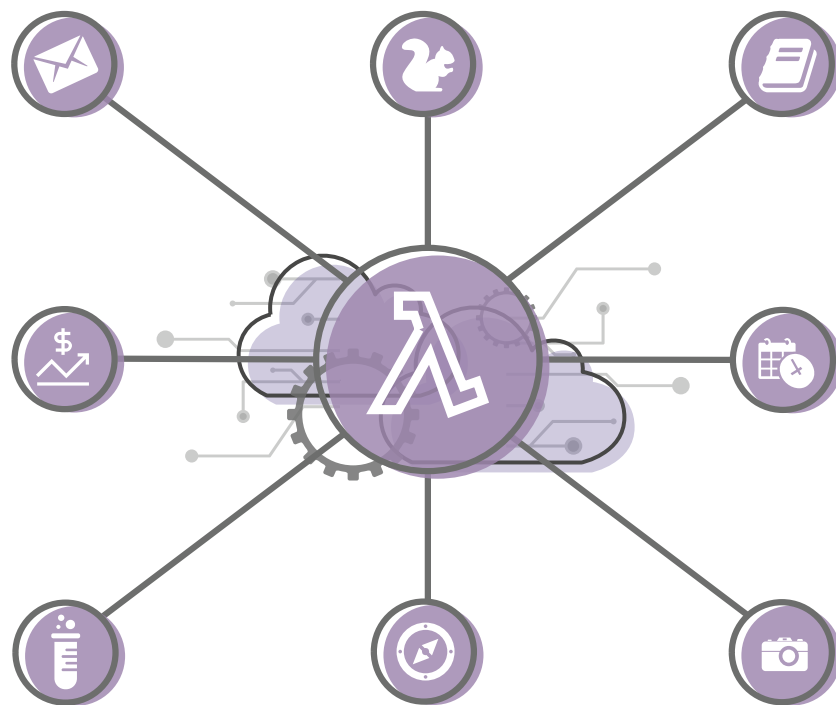
大量的用户在使用无服务器计算架构创新



大量企业利用无服务器计算架构实现了超大的扩展能力

- 汤森路透(Thomson Reuters) 每秒钟处理4,000条请求
- 美国金融业监管局 (FINRA) 每天检查数万亿笔股票交易
- 赫斯特集团 (Hearst) 将自己分析引擎中的数据采集和处理时间减少了90%
- Vevo 能够平滑处理超过平时80倍的峰值流量
- 亿客行 (Expedia) 每月触发12亿次请求

无服务器计算引擎是现代应用的核心组件



实现和案例

AWS Lambda- 无服务器计算架构的核心

AWS Lambda



Lambda = 事件驱动的无服务器的微服务

即一小段可运行的代码

AWS Lambda哲学

专注于您的业务逻辑

AWS Lambda – 如何使用

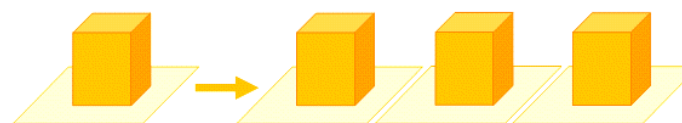
写代码



上传即部署



自动按需运行并扩展



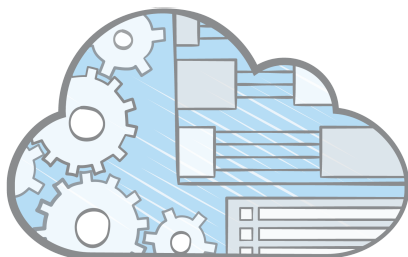
监控 & 日志



AWS Lambda的优势

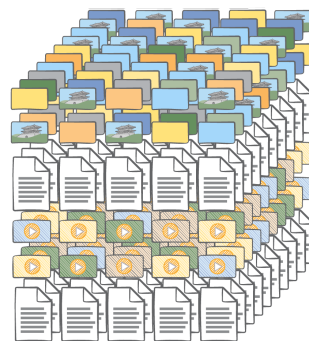
1

无需运维服务器



2

持续扩展



3

次秒级计量



AWS Lambda的卓越功能

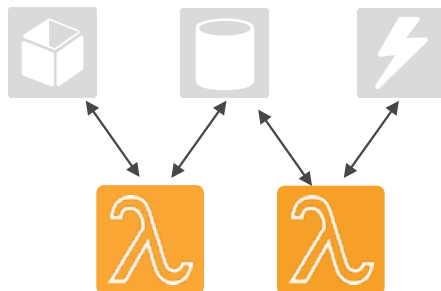
自带代码（BYOC）



简单的资源模型



灵活的调用方式



精细的权限控制



为什么那么容易？

这种架构模式，不再考虑：

- 服务器
- 按实际使用量自动匹配计算资源
- 默认的高可用
- 扩展能力、冗余问题
- 操作系统与语言的更新
- 记录状态和日志

但可轻松做到：

- 使用自己的代码（BYOC）
- 平行运行代码，低延时同步调用
- 创建后端，事件处理引擎，数据处理系统
- 不会有资源闲置浪费
- 启动成本很低
- 按需付费

还不够简单？ - Blueprint来帮忙

Step 1: Select blueprint

Select blueprint ?

Blueprints are sample configurations of event sources and Lambda functions. Choose a blueprint that best aligns with your desired scenario and customize as needed, or skip this step if you want to author a Lambda function and configure an event source separately. Except where otherwise noted, blueprints are licensed under [CC0](#).

All languages
« < Viewing 1-9 of 43 > »

s3-get-object-python An Amazon S3 trigger that retrieves metadata for the object that has been updated. python2.7 · s3	config-rule-change-triggered An AWS Config rule that is triggered by configuration changes to EC2 instances. Checks instance types. nodejs · config	dynamodb-process-stream An Amazon DynamoDB trigger that logs the updates made to a table. nodejs · dynamodb
microservice-http-endpoint A simple backend (read/write to DynamoDB) with a RESTful API endpoint using Amazon API Gateway. nodejs · api-gateway	node-exec Demonstrates running an external process using the Node.js child_process module. nodejs	slack-echo-command-python A function that handles a Slack slash command and echoes the details back to the user. python2.7 · api-gateway · slack
simple-mobile-backend A simple mobile backend (read/write to DynamoDB).	kinesis-process-record-python An Amazon Kinesis stream processor that logs the data being published.	splunk-kinesis-logging Demonstrates logging events streamed from AWS Kinesis to Splunk's HTTP Event Collector.

版本和别名

- 随时可以修改代码
- 每次的修改以版本的形式保留下来
- 最新的修改默认生效
- 可以增加alias

```
exports.handler =  
function(event, context)  
{context.succeed("hi");}
```

```
exports.handler =  
function(event, context)  
{context.succeed("bye");}
```

Versions

1

2 \$LATEST

Alias

Prod

Dev

AWS Lambda可以与多种AWS服务集成

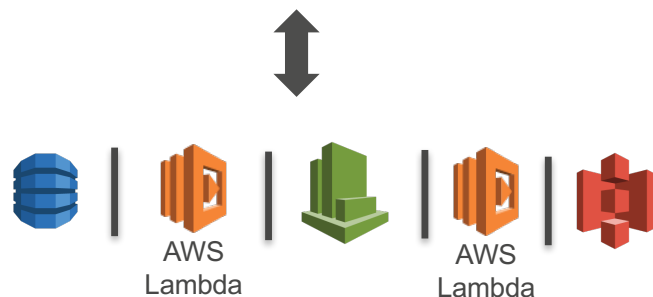


AWS Lambda和其他服务结合后的威力

服务粘合剂

- 各种服务能通过 AWS Lambda 串联起来
- 想象一下Unix/Linux的管道技术

```
cat poorly_formatted_report.txt | fmt | pr | lpr
```



让其他服务更智能



Serverless Framework – 服务治理

借助AWS Lambda和API Gateway来构建无服务架构应用的**框架**

- 本地运行/测试AWS Lambda
- 自动部署AWS Lambda和API Gateway
- 支持多region和多stage
- 支持project
- 支持插件扩展



典型使用场景：移动应用后端

AWS新一代移动应用后端架构

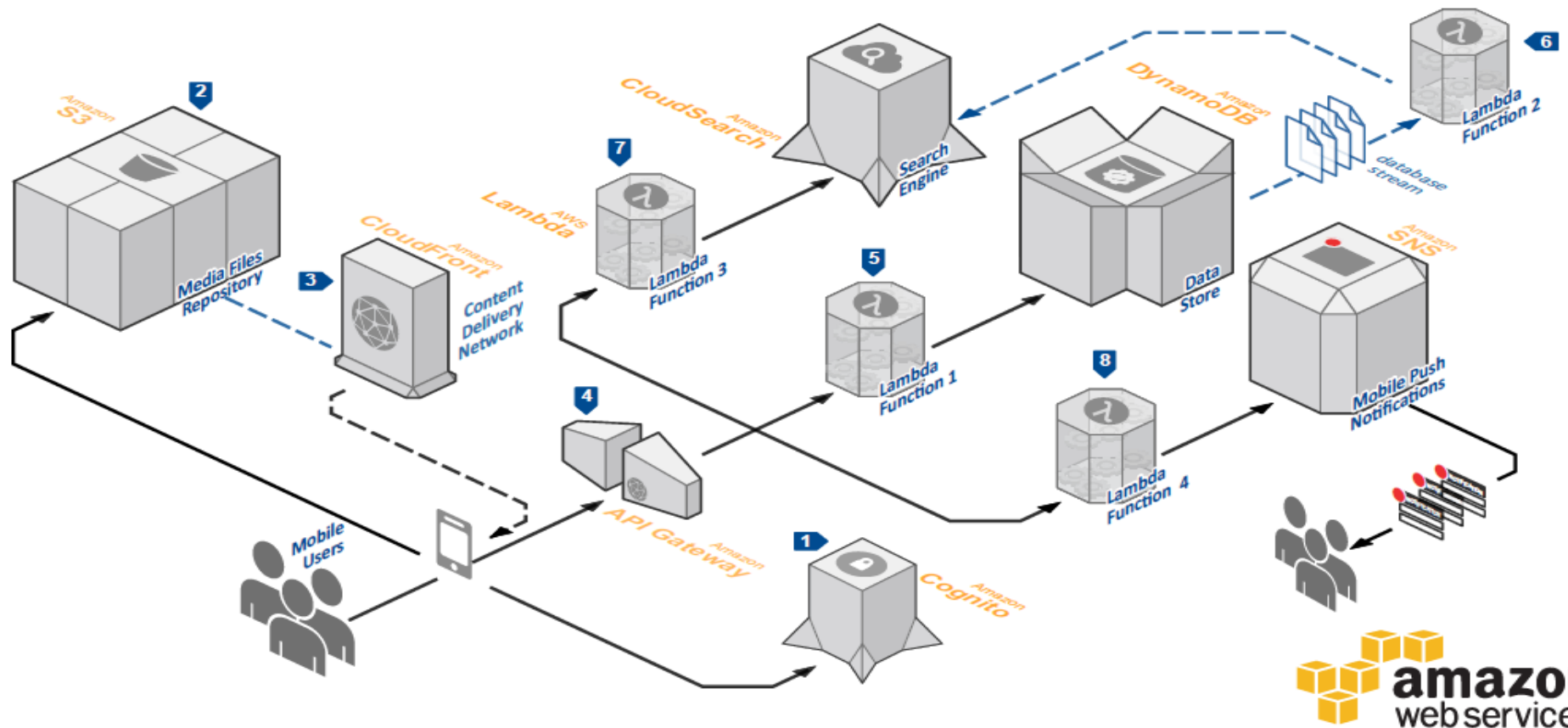
AWS LAMBDA: MOBILE BACKEND

Mobile application backend infrastructures require scalability, elasticity, and low latency to support a dynamic user base. Unpredictable peak usage and a global footprint of mobile users requires mobile backends to be fast and flexible.

The growing demand from mobile users means applications need a rich set of mobile services that work together seamlessly without sacrificing control and flexibility of the backend infrastructure.

With AWS Lambda, you can build applications that automatically scale without provisioning or managing servers. Since many mobile applications today have a limited budget for upfront infrastructure, a cost-effective, event-driven mobile architecture allows you to pay only for what you use.

This reference architecture shows an example of a highly available and scalable mobile backend featuring microservices, storage, and API endpoints requiring zero administration.



AWS新一代移动应用后端架构

AWS LAMBDA: MOBILE BACKEND

Mobile application backend infrastructures require scalability, elasticity, and low latency to support a dynamic user base. Unpredictable peak usage and a global footprint of mobile users requires mobile backends to be fast and flexible.

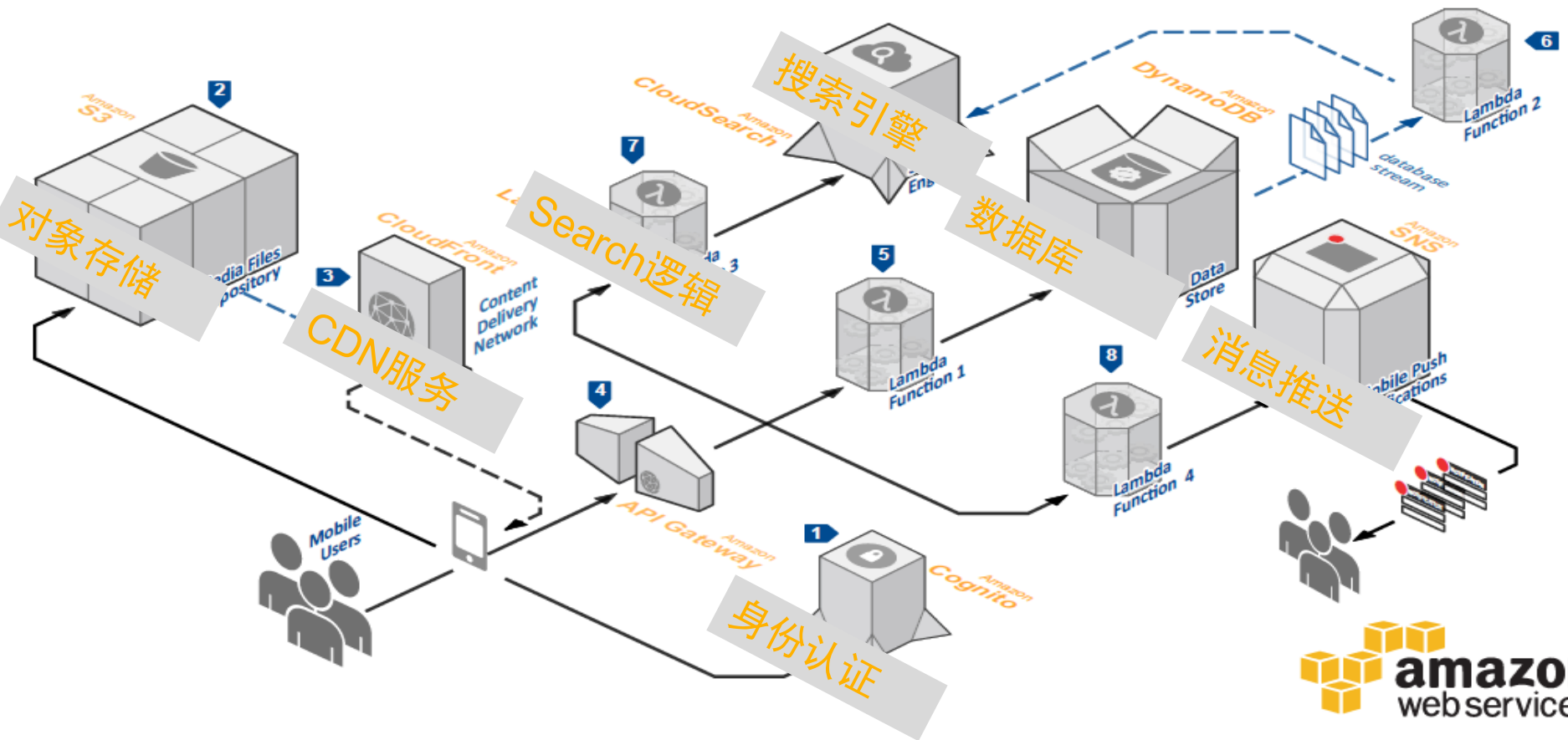
The growing demand from mobile users means applications need a rich set of mobile services that work together seamlessly without sacrificing control and flexibility of the backend infrastructure.

With AWS Lambda, you can build applications that automatically scale without provisioning or managing servers. Since many mobile applications today have a limited budget for upfront infrastructure, a cost-effective, event-driven mobile architecture allows you to pay only for what you use.

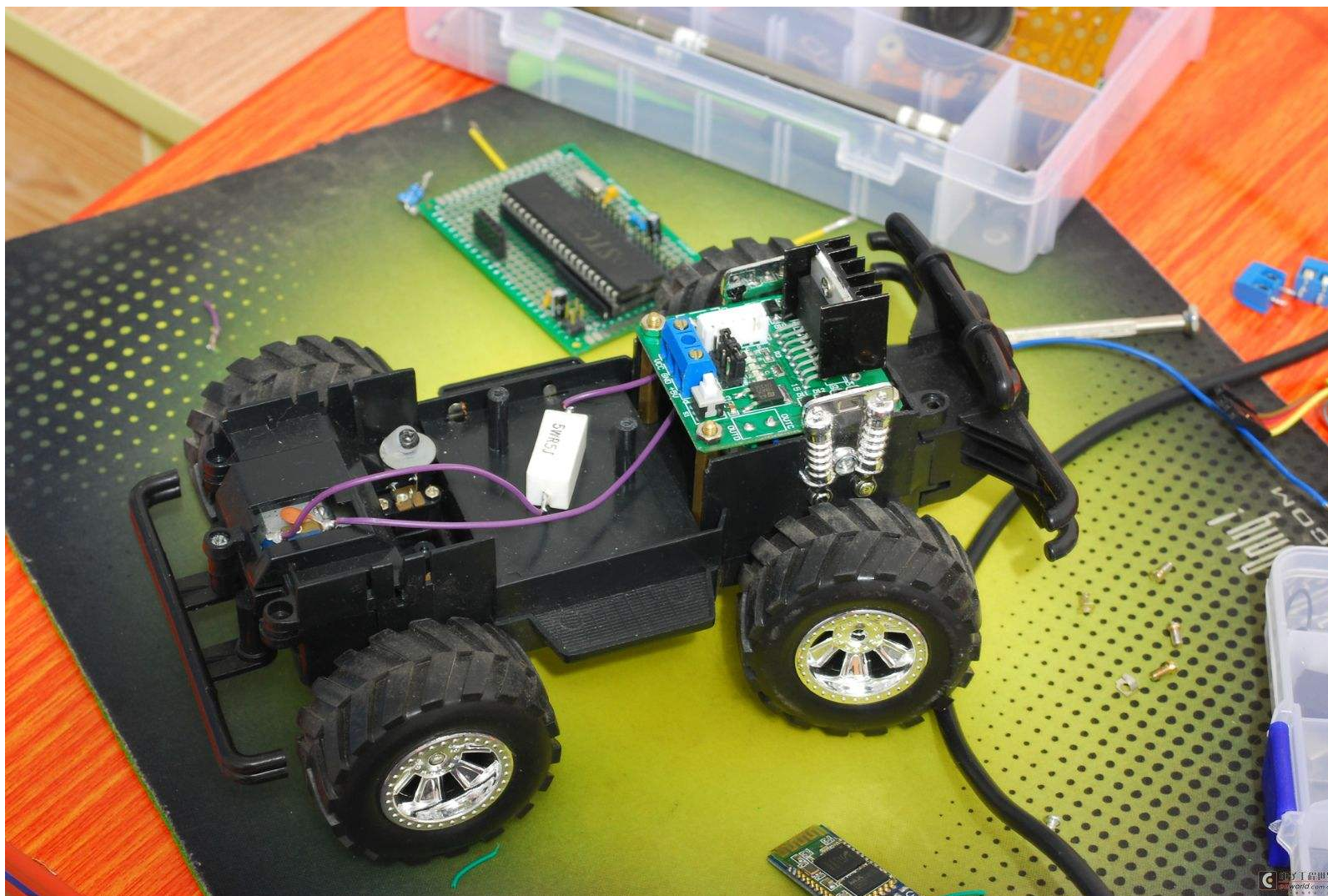
This reference architecture shows an example of a highly available and scalable mobile backend featuring microservices, storage, and API endpoints requiring zero administration.

AWS Reference Architectures

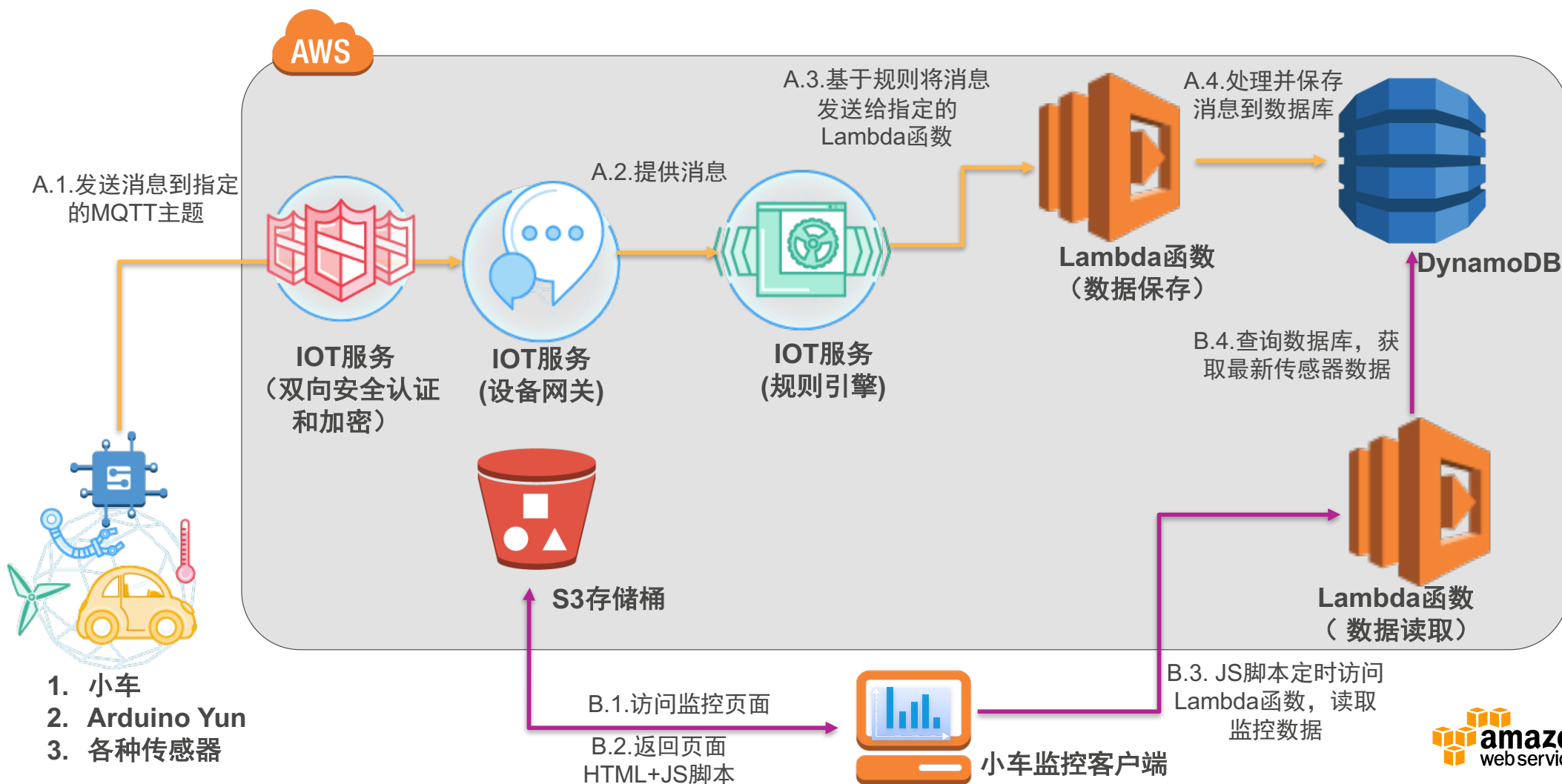
- Amazon Cognito
- Amazon S3
- Amazon CloudFront
- Amazon API Gateway
- AWS Lambda
- Amazon DynamoDB
- Amazon CloudSearch
- Amazon SNS



典型使用场景：车联网



互联网汽车工作原理架构图



小车运行状态监控



谢谢大家!