

无服务器架构的设计模式和Demo演示

毛郸榕，AWS解决方案架构师

Mao Danrong, AWS Solution Architect

亚马逊 AWS

计算

-  **EC2**
云中的虚拟服务器
-  **EC2 Container Service**
运行和管理 Docker 容器
-  **Elastic Beanstalk**
运行和管理 Web 应用程序
-  **Lambda**
运行代码以响应事件

存储和内容分发

-  **S3**
可扩展的云存储
-  **Glacier**
云中归档存储
-  **Storage Gateway**
混合存储集成

联网

-  **VPC**
隔离的云资源
-  **Direct Connect**
AWS 专用网络连接

开发人员工具

-  **CodeDeploy**
自动代码部署

安全 & 身份

-  **Identity & Access Management**
管理用户访问和加密密钥

数据库

-  **DynamoDB**
托管 NoSQL 数据库
-  **ElastiCache**
内存缓存
-  **RDS**
管理型关系数据库服务
-  **Redshift**
快速、简单、经济高效的数据仓库

管理工具

-  **CloudFormation**
使用模板创建和管理资源
-  **Config**
跟踪资源库存和变更
-  **CloudTrail**
跟踪用户活动和 API 使用率
-  **Trusted Advisor**
优化性能和安全性
-  **CloudWatch**
监控资源和应用程序

物联网

-  **AWS IoT**
将设备连接到云

分析

-  **EMR**
Hadoop 托管框架
-  **Kinesis**
实时处理大数据流

移动服务

-  **SNS**
推送通知服务
-  **Cognito**
用户身份和应用程序数据同步

应用服务

-  **SQS**
消息队列服务
-  **SWF**
协调应用程序组件的工作流服务
-  **API Gateway**
构建、部署和管理 API

其他资源

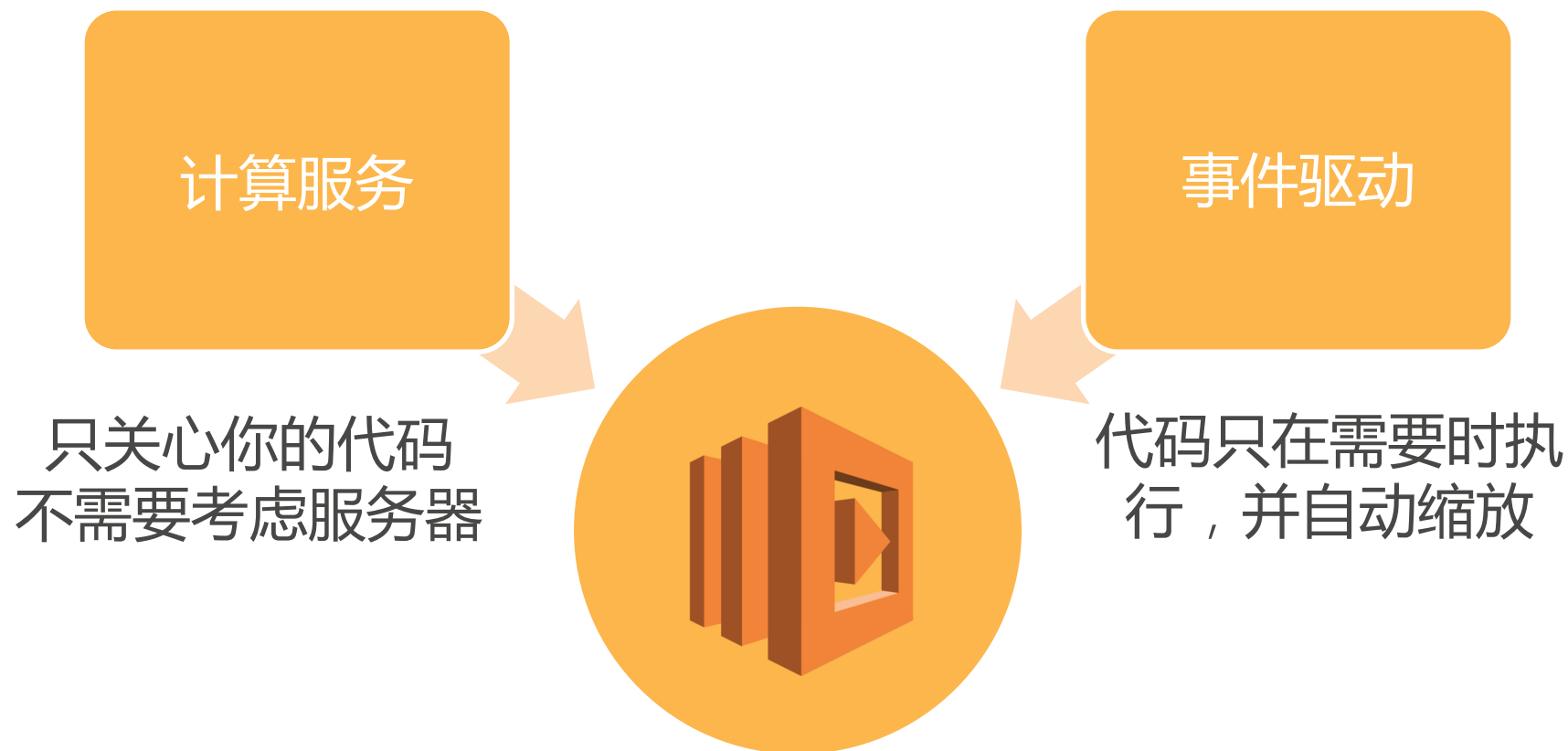
入门 [↗](#)

请参阅我们的[文档](#)或查看我们的[培训](#)，以了解更多有关 AWS 的信息。

**由光环新网运营，AWS中国（北京）区域
Lambda和API Gateway已经上线**

什么是AWS Lambda?

什么是AWS Lambda



Lambda使用场景



S3 + Lambda

Dynamic data ingestion

图片压缩、视频转码、文件索引、日志处理、内容校验、筛选聚合



Kinesis + Lambda

Live stream processing

应用程序活动审计、订单事物处理、点击流分析、IoT设备响应、遥测和计量



SNS + Lambda

Smart IT, Custom message

自动IT警报响应、自定义行为、IT审计、定义广播消息、消息推送、邮件推送



Direct call + Lambda

Serverless backend

微服务架构、移动端后端架构、IoT后端架构



DynamoDB + Lambda

Database triggers

数据校验、数据筛选、Live 通知



Alexa + Lambda

Voice triggers

在云端构建自定义自动语音响应系统

由光环新网运营，AWS中国北京区域
Lambda和API Gateway已经在线上

AWS Lambda 编程模型、并发问题、版本控制

Lambda 编程模型

Lambda > Functions > GetHelloWithName

ARN - arn:aws:lambda:us-west-2:556776719183:function:GetHelloWithName

Qualifiers ▾

Save

Save and test

Actions ▾

Code

Configuration

Triggers

Monitoring



Code entry type

Edit code inline ▾

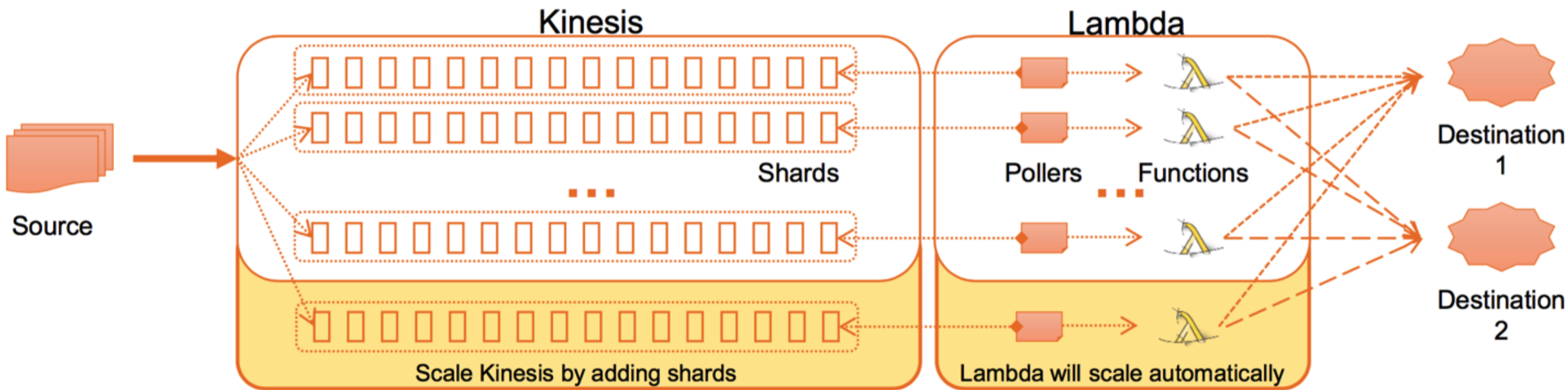
```
1 console.log('Loading function');
2
3 exports.handler = function(event, context) {
4     //console.log('Received event:', JSON.stringify(event, null, 2));
5     console.log('value1 =', event.key1);
6     console.log('remaining time =', context.getRemainingTimeInMillis());
7     console.log('functionName =', context.functionName);
8     console.log('AWSrequestID =', context.awsRequestId);
9     console.log('logGroupName =', context.logGroupName);
10    console.log('logStreamName =', context.logStreamName);
11    console.log('clientContext =', context.clientContext);
12    if (typeof context.identity !== 'undefined') {
13        console.log('Cognito identity ID =', context.identity.cognitoIdentityId);
14    }
15    context.succeed(event.key1); // Echo back the first key value
16    // context.fail('Something went wrong');
17 };
18
```

对此对

AWS Lambda 并发问题

对于Kinesis或者DynamoDB

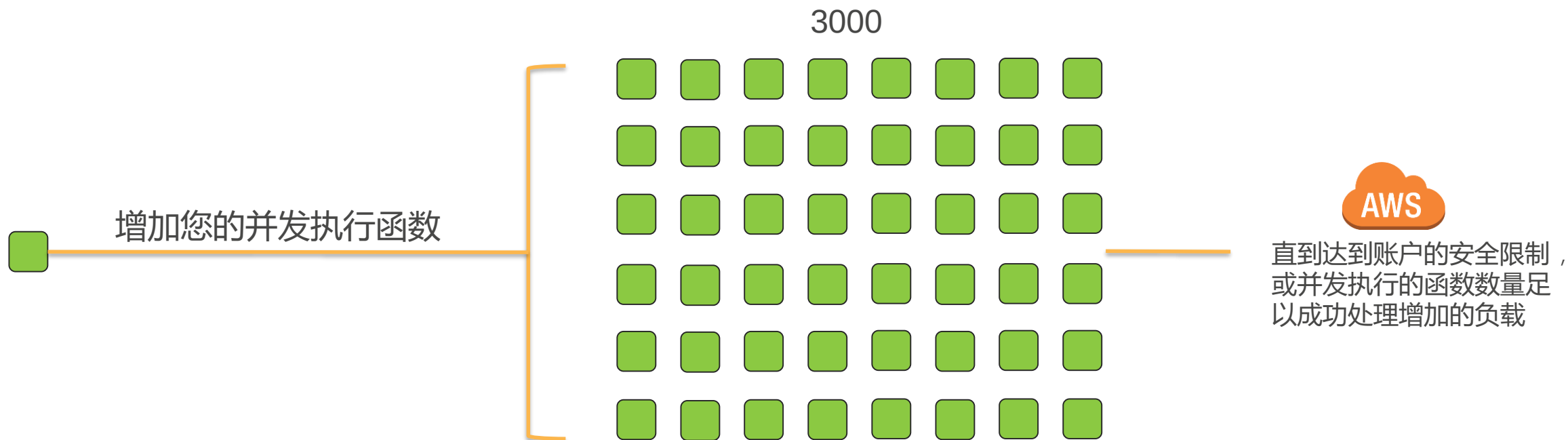
- **Lambda execute your function concurrently for each shard on the stream**
- 如果您的流有 100 个活动分区，则最多会有 100 个 Lambda 函数调用并发运行。



- **Shards (分片):** One Lambda function concurrently invoked per Kinesis shard
- Increasing shards will cause more Lambda function invoked concurrently
- Each individual shard follows ordered processing

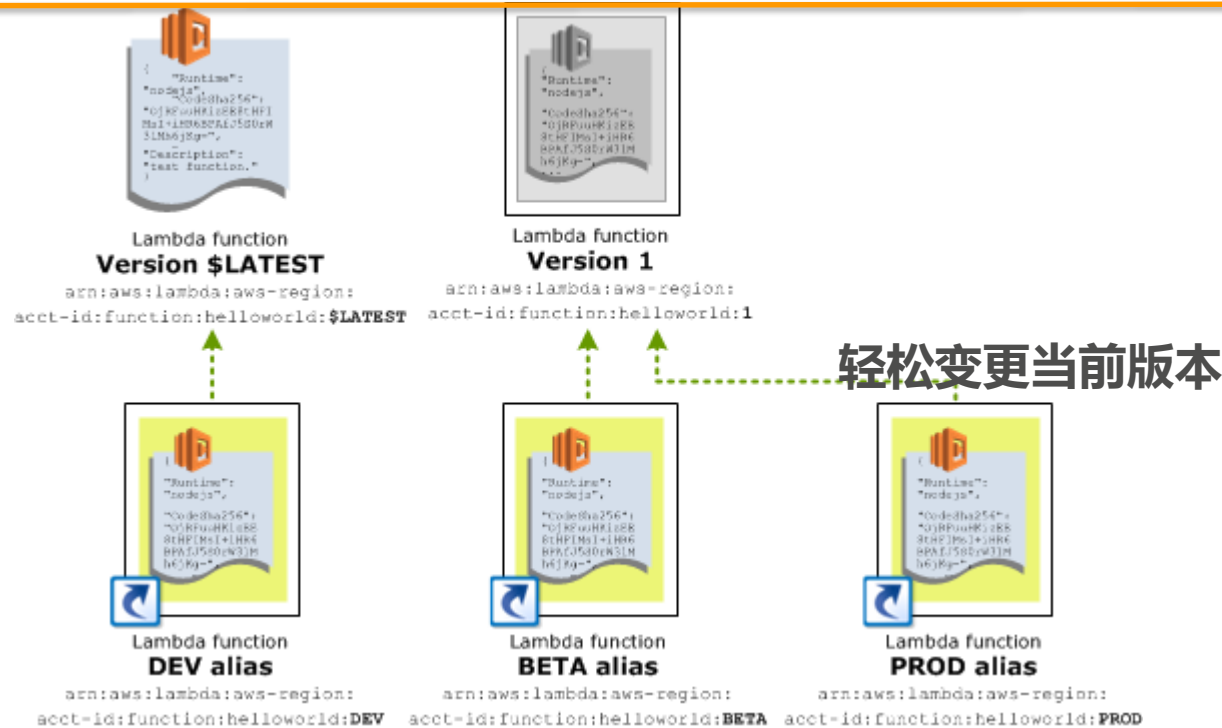
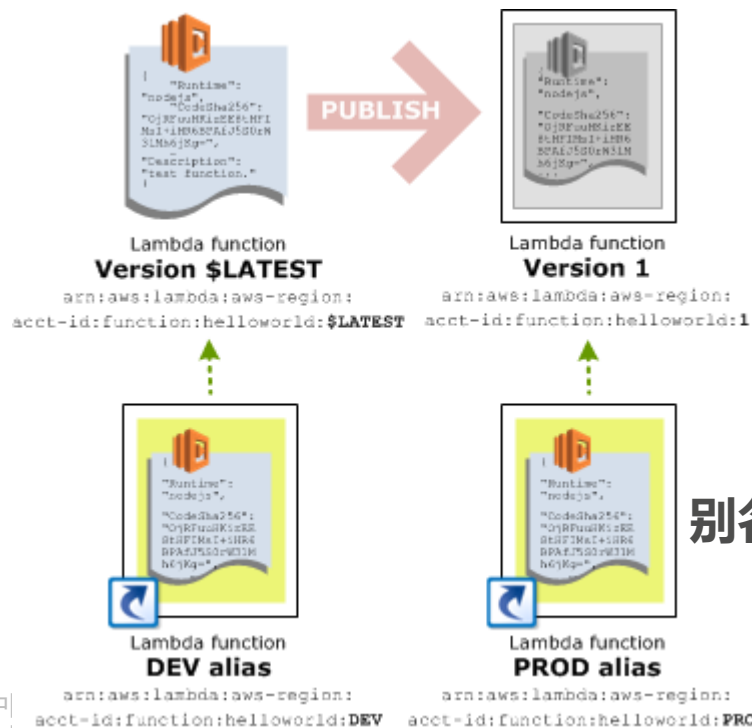
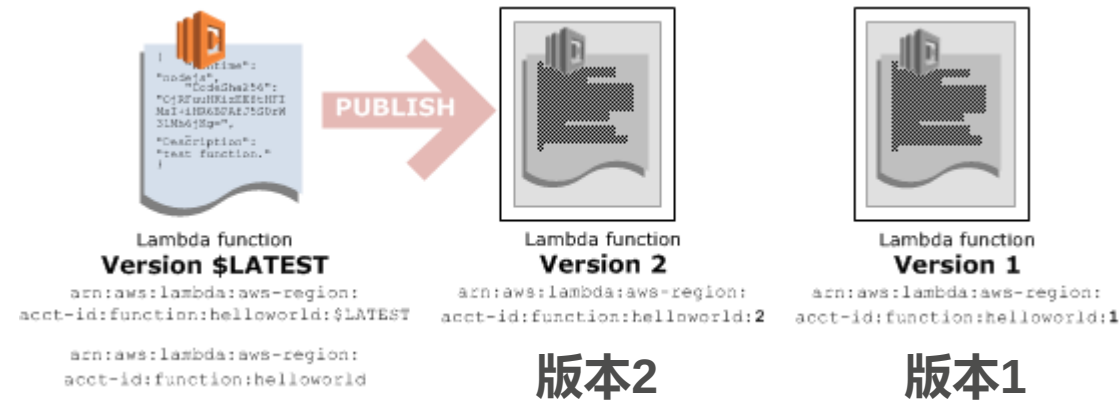
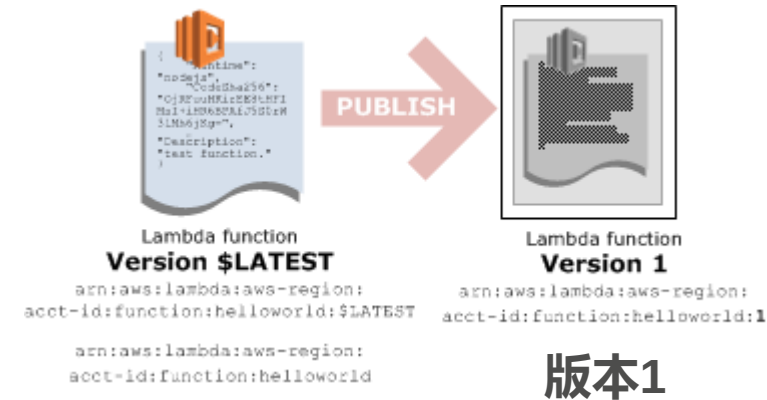
AWS Lambda 并发问题 - 流量激增问题

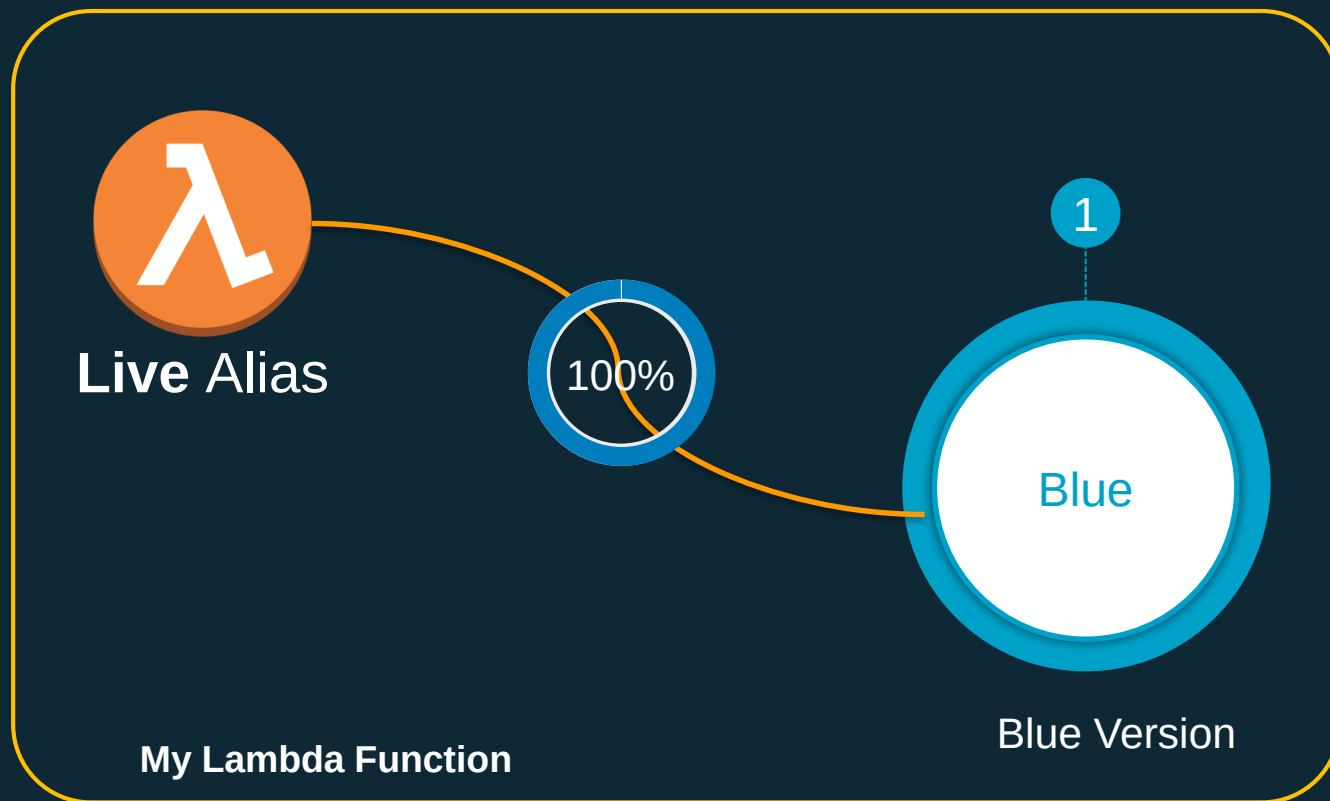
AWS Lambda 将根据增加的流量动态扩展容量，具体取决于您的AWS账户的并发执行限制。为了处理突增流量，Lambda 将立即根据预定量增加您的并发执行函数。



http://docs.aws.amazon.com/zh_cn/lambda/latest/dg/concurrent-executions.html

AWS Lambda 版本控制和别名

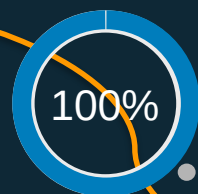




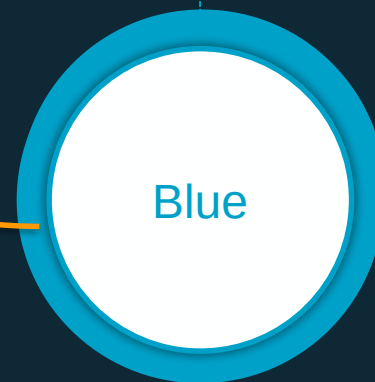
My Lambda Function



Live Alias



Green Version



Blue Version

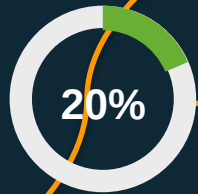


AWS Cloud9

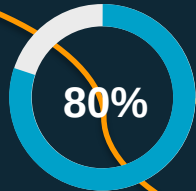
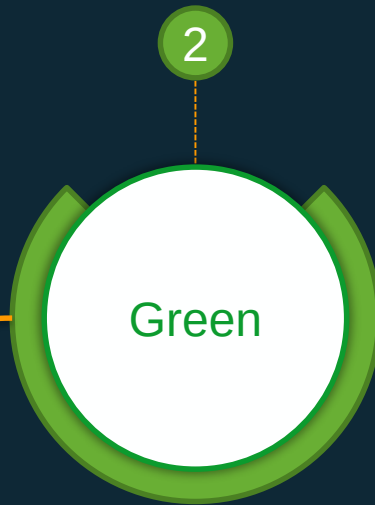
My Lambda Function



Live Alias



Green Version



1

Blue

Blue Version



Linear 20% Every 1 Minutes



AWS
CodeDeploy

Safe Lambda Deployments With SAM

NEW

Globals:

Function:

Runtime: `nodejs6.10`

Environment:

Variables:

MyName: `offsiteDemo`

MyLambdaFunction:

Type: `AWS::Serverless::Function`

Properties:

Handler: `index.handler`

AutoPublishAlias: `live`

DeploymentPreference:

Type: `Linear10PercentEvery10Minutes`

Alarms:

A list of alarms that you want to monitor

- `!Ref AliasErrorMetricGreaterThanZeroAlarm`
- `!Ref LatestVersionErrorMetricGreaterThanZeroAlarm`

Hooks:

Validation Lambda functions that are run before & after traffic shifting

PreTraffic: `!Ref PreTrafficLambdaFunction`

PostTraffic: `!Ref PostTrafficLambdaFunction`



Demo 演示

构建第一个Lambda函数（Node.js）

LambdaAPIGWGenerateRandomN...

Qualifiers ▼

Actions ▼

GeneratedNumber ▼

Test

Save



AWS Cloud9 File Edit Find View Goto Tools Window

Environment

LambdaAPIGWGenerateRan
index.js



index.js



```
1 'use strict';
2 console.log('Loading function');
3 exports.handler = (event, context, callback) => {
4     //var min = parseInt(event.min);
5     //var max = parseInt(event.max);
6     var min = 2;
7     var max = 10;
8     var generatedNumber = Math.floor(Math.random() * max) + min;
9     console.log(event);
10    callback(null, generatedNumber);
11 };
12
```

Demo 演示

由光环新网运营，AWS中国（北京）区域
Lambda和API Gateway已经上线

AWS Lambda 集成 API Gateway

Demo 演示

由光环新网运营，AWS中国（北京）区域
Lambda和API Gateway已经上线

通过API Gateway进行参数传递

SNSConfirmSubscription

Usage Plans

API Keys

Custom Domain Names

Client Certificates

Settings

Use Default Timeout ☒ 

▶ URL Path Parameters

▶ URL Query String Parameters

▶ HTTP Headers

▼ Body Mapping Templates 

Request body passthrough ☐ When no template matches the request Content-Type header 
☒ When there are no templates defined (recommended) 
☐ Never 

Content-Type

No mapping templates defined. The request body will be passed through to the integration endpoint

 [Add mapping template](#)

Demo 演示

由光环新网运营，AWS中国（北京）区域
Lambda和API Gateway已经上线

AWS Lambda配合DynamoDB 写入数据(Node.js)

LambdaDynamodbWrite

Qualifiers ▼

Actions ▼

LambdaDynamoDB ▼

Test

Save



AWS Cloud9 File Edit Find View Goto Tools Window

Environment

▼ LambdaDynamodbWrite

index.js



index.js



```
1 'use strict';
2 console.log('Loading function');
3 var AWS = require('aws-sdk');
4 var docClient = new AWS.DynamoDB.DocumentClient();
5 AWS.config.region = 'cn-north-1';
6 exports.handler = function (event, context, callback){
7     var params = {
8         Item: {
9             date: Date.now(),
10            message: event.message
11        },
12        TableName: 'Lambda-DynamoDB-Write-Read'
13    };
14    docClient.put(params, function(err, data){
15        if(err){
16            console.log("Fail to Write into AWS DynamoDB.");
17            callback(err, null);
18        }else{
19            console.log("Successfully write item into AWS DynamoDB.");
20            callback(null, data);
21        }
22    });
23 }
```

Demo 演示

由光环新网运营，AWS中国（北京）区域
Lambda和API Gateway已经上线

AWS Lambda配合DynamoDB 读取数据(Node.js)

LambdaDynamoDBRead

Qualifiers ▼

Actions ▼

LambdaDynamoDBRead ▼

Test

Save



AWS Cloud9 File Edit Find View Goto Tools Window

Environment

▼ LambdaDynamoDBRead

index.js



index.js



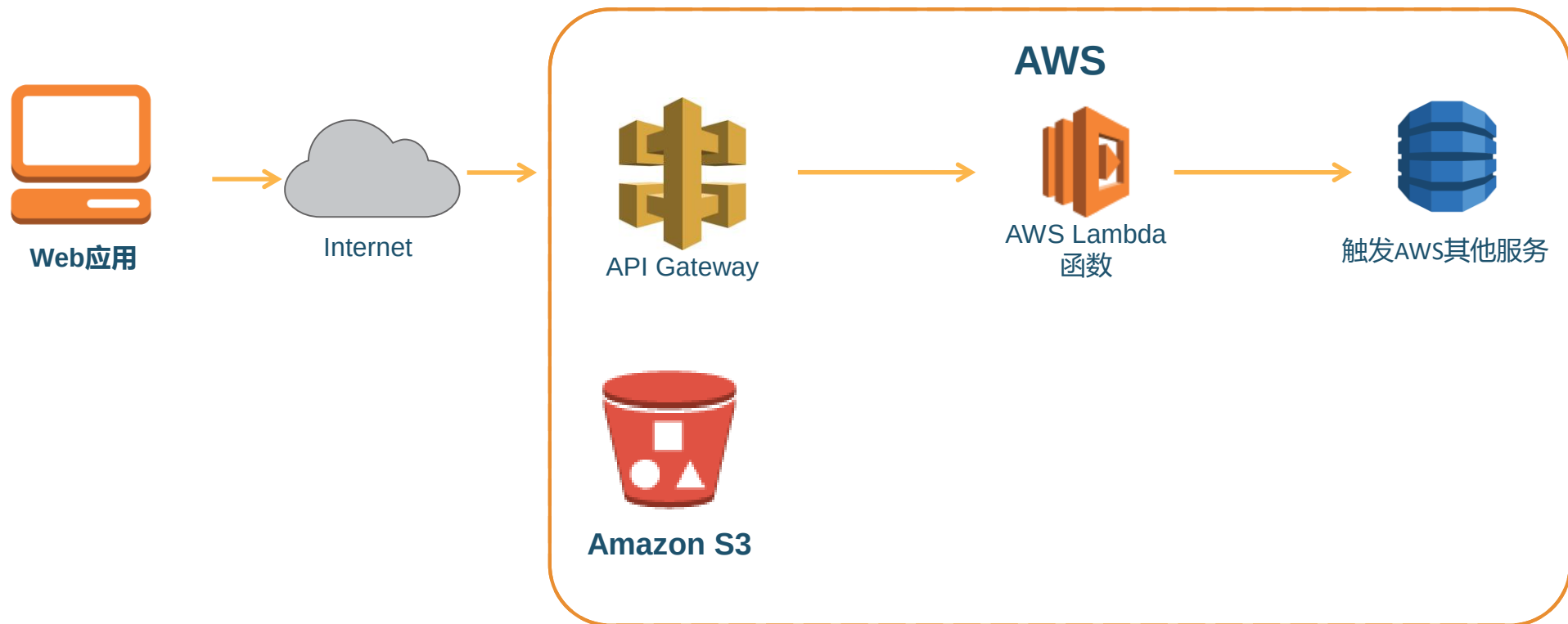
```
5 exports.handler = function(event, context, callback) {
6   var params = {
7     TableName: 'Lambda-DynamoDB-Write-Read',
8     Limit: 100
9   };
10  docClient.scan(params, function(err, data) {
11    if (err) {
12      console.log("Fail read from AWS DynamoDB Table.");
13      callback(err, null);
14    }
15    else {
16      console.log("Successfully read from AWS DynamoDB Table!!");
17      callback(null, data);
18    }
19  });
20 };
21
```

Demo 演示

由光环新网运营，AWS中国（北京）区域
Lambda和API Gateway已经上线

Lambda配合DynamoDB和S3 构建交互式网站

Lambda配合DynamoDB和S3构建交互式网站

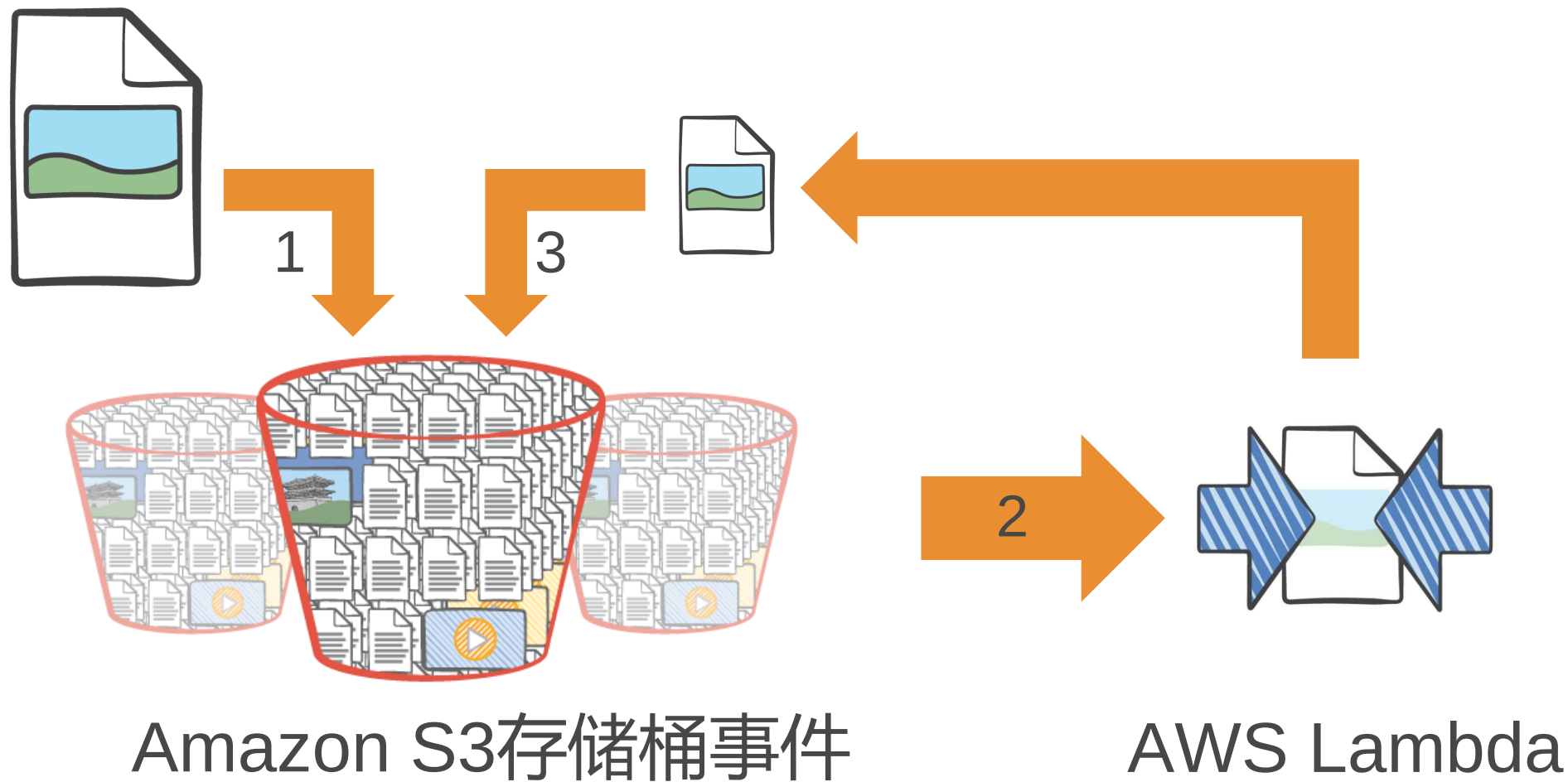


Demo 演示

由光环新网运营，AWS中国（北京）区域
Lambda和API Gateway已经上线

AWS Lambda配合S3 - 事件触发

扩展Amazon S3添加自动压缩功能



如何添加一个Amazon S3功能

1. 随便从网上找一段JavaScript压缩代码示例。
2. 从Amazon S3事件示例开始：
 1. 从S3上GET原对象
 2. 压缩
 3. 将压缩的版本PUT回S3
3. 选择一些S3存储桶应用这个事件。

1. 图片打水印 (Lambda + S3)

https://s3.cn-north-1.amazonaws.com.cn/bjsdemo/Lambda-watermark-demo/Lambda_watermark_demo.zip

2. 图片压缩/裁剪 (Lambda + S3)

<https://s3.cn-north-1.amazonaws.com.cn/bjsdemo/Lambda-CreateThumbnail-demo/CreateThumbnail.zip>

3. 根据输入参数线上裁剪图片 (API Gateway + Lambda + S3)

<https://s3.cn-north-1.amazonaws.com.cn/bjsdemo/Serverless-Image-Resizing-Demo/serverless-image-resizing-demo-Danrong.zip>

LambdaS3CreateThumbnail

Qualifiers ▼

Actions ▼

Select a test event.. ▼

Test

Save



AWS Cloud9 File Edit Find View Goto Tools Window



Environment

▼ LambdaS3CreateThumbnail
 ▼ node_modules
 CreateThumbnail.js

CreateThumbnail x +

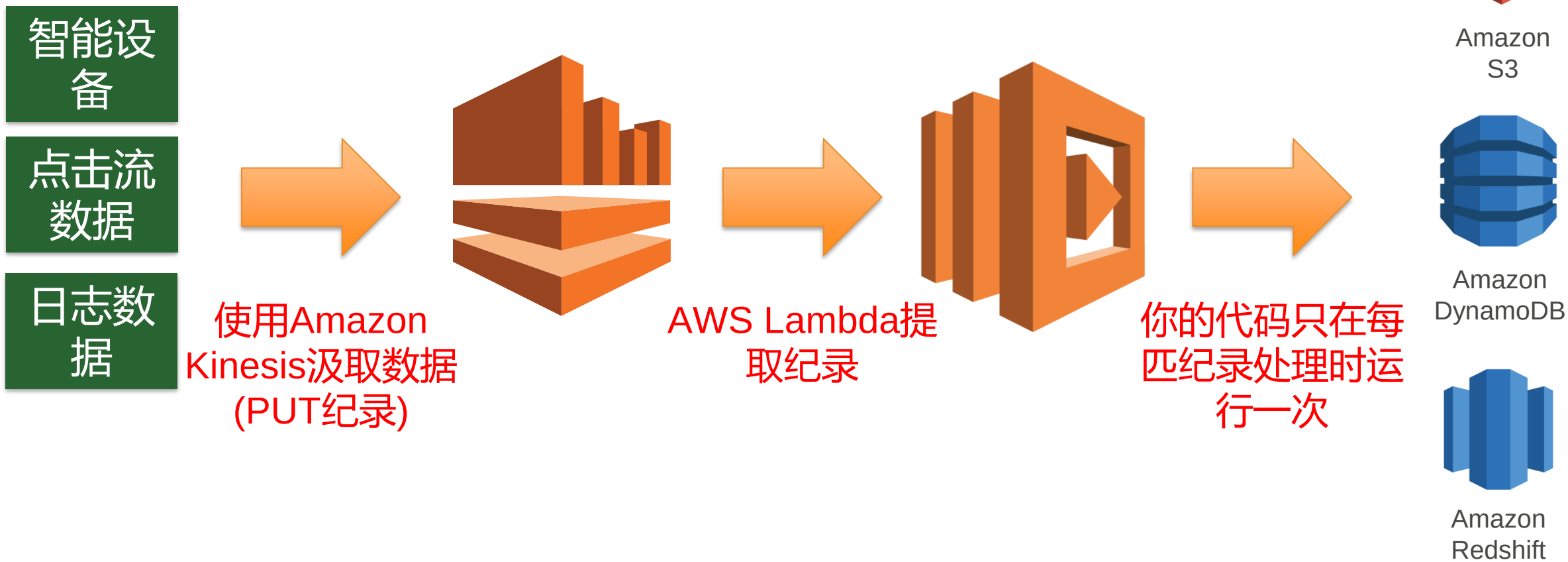
```
1 // dependencies
2 var async = require('async');
3 var AWS = require('aws-sdk');
4 var gm = require('gm').subClass({ imageMagick: true }); // Enable ImageMagick integration.
5 var util = require('util');
6
7 // constants
8 var MAX_WIDTH = 100;
9 var MAX_HEIGHT = 100;
10
11 // get reference to S3 client
12 var s3 = new AWS.S3();
13
14 exports.handler = function(event, context, callback) {
15     // Read options from the event.
16     console.log("Reading options from event:\n", util.inspect(event, {depth: 5}));
17     var srcBucket = event.Records[0].s3.bucket.name;
18     // Object key may have spaces or unicode non-ASCII characters.
19     var srcKey = decodeURIComponent(event.Records[0].s3.object.key.replace(/\+/g, " "));
20     //var dstBucket = srcBucket + "-resized";
21     var dstKey = "resized-" + srcKey;
22
23     // // Sanity check: validate that source and destination are different buckets.
24     // if (srcBucket == dstBucket) {
25     //     callback("Source and destination buckets are the same.");
26     //     return;
27     // }
28
29     // Infer the image type.
```

Demo 演示

由光环新网运营，AWS中国（北京）区域
Lambda和API Gateway已经在线上

AWS Lambda配合Kinesis - 实时流数据处理

实时数据流分析架构



LambdaKinesisDemo

Qualifiers ▼

Actions ▼

LambdaKinesis ▼

Test

Save



AWS Cloud9 File Edit Find View Goto Tools Window

Environment

▼ LambdaKinesisDemo

index.js



index.js



```
1 'use strict';
2 console.log('Loading function');
3 var AWS = require('aws-sdk');
4 var docClient = new AWS.DynamoDB.DocumentClient();
5 AWS.config.region = 'cn-north-1';
6 exports.handler = function(event, context, callback) {
7     event.Records.forEach(function(record) {
8         // Kinesis data is base64 encoded so decode here
9         var payload = new Buffer(record.kinesis.data, 'base64').toString('ascii');
10        console.log('Decoded payload:', payload);
11        var params = {
12            Item: {
13                date: Date.now(),
14                message: payload
15            },
16            TableName: 'Lambda-DynamoDB-Write-Read'
17        };
18        docClient.put(params, function(err, data){
19            if(err){
20                console.log("Fail to Write into AWS DynamoDB.");
21                callback(err, null);
22            }else{
23                console.log("Sucessfully write item into AWS DynamoDB.");
24                callback(null, data);
25            }
26        });
27    });
28    callback(null, "message");
29 };
```

[Privacy Policy](#)

[Terms of Use](#)

Demo 演示

由光环新网运营，AWS中国（北京）区域
Lambda和API Gateway已经在线上

AWS Lambda配合SNS - 消息通知事件触发

LambdaSNSNotification

Qualifiers ▼

Actions ▼

Select a test event.. ▼

Test

Save



AWS Cloud9 File Edit Find View Goto Tools Window

Environment

▼ LambdaSNSNotification

index.js



index.js



```
1 'use strict';
2 console.log('Loading function');
3 var AWS = require('aws-sdk');
4 var docClient = new AWS.DynamoDB.DocumentClient();
5 AWS.config.region = 'cn-north-1';
6 exports.handler = function (event, context, callback){
7     console.log("AWS China Lambda and SNS Notification");
8     const message = event.Records[0].Sns.Message;
9     console.log("From SNS:", message);
10    var params = {
11        Item: {
12            date: Date.now(),
13            message: message
14        },
15        TableName: 'Lambda-DynamoDB-Write-Read'
16    };
17    docClient.put(params, function(err, data){
18        if(err){
19            console.log("Fail to Write into AWS DynamoDB.");
20            callback(err, null);
21        }else{
22            console.log("Sucessfully write item into AWS DynamoDB.");
23            callback(null, data);
24        }
25    });
26 }
```

Privacy Policy

Terms of Use

Demo 演示

Thank You!