

无服务器图像处理程序

部署手册

施 乔

版本：v4.2

最后更新时间: 2020 年 7 月



Copyright (c) 2020 by Amazon.com, Inc. or its affiliates.

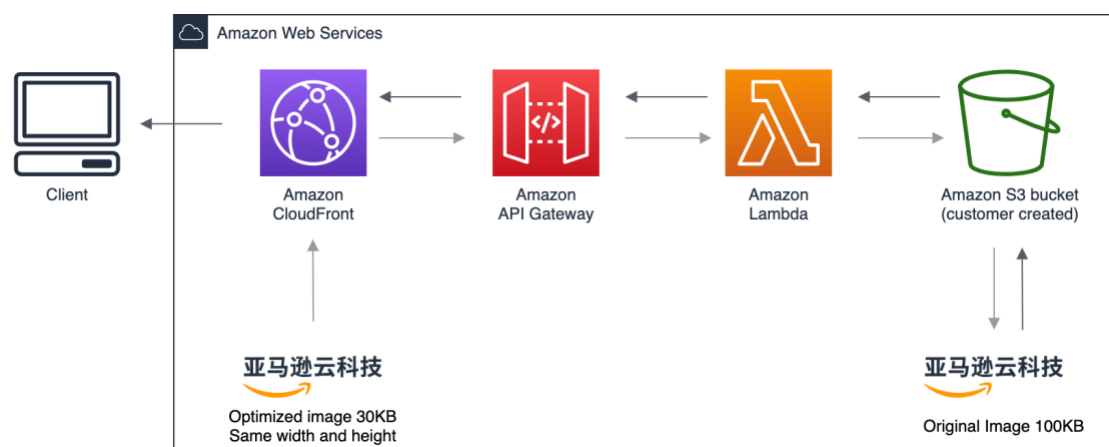
背景介绍

无服务器图像处理程序解决方案将高可用性，可信任的亚马逊云科技服务与开源图像处理套件 [Sharp](#) 结合在一起，在由西云数据运营的 亚马逊云科技（宁夏） 区域和由光环新网运营的亚马逊云科技（北京） 区域上实现快速且经济高效的图像处理。该解决方案自动部署和配置针对动态图像处理进行了优化的无服务器架构，并使用 Amazon CloudFront 进行内容的加速，使用 Amazon Simple Storage Service(Amazon S3) 以低成本提供可靠和持久的云存储。

该手册仅适用于在由西云数据运营的亚马逊云科技（宁夏） 区域和由光环新网运行的亚马逊云科技（北京） 区域部署无服务器图像处理程序。如您希望在 亚马逊云科技 其他区域部署该解决方案，请[访问此链接](#)。

架构与工作原理

架构图与资源



架构描述

- CloudFront 用于接收来自客户端的请求, 回源到 API Gateway。
- API Gateway 提供 API 服务, 用于接收来自 CloudFront 的 HTTP 请求, 并触发 Lambda, 将 Lambda 处理完的结果返回给 CloudFront。

- AMAZON Lambda 读取 Amazon Simple Storage Service (Amazon S3) 中存储的图像，并使用 [Sharp](#) 对图像进行处理。
- 对于已经处理过的图像请求，CloudFront 会缓存到 CDN 节点上，从而减少回源次数，降低费用并提高响应速度。

快速部署无服务器图像处理程序

前提条件

- **创建或指定一个或者多个用于图像存储的 Amazon S3 存储桶。**在启动解决方案的 AMAZON CloudFormation 模板之前，您必须在 **SourceBuckets** 模板参数中指定一个或者多个 Amazon S3 存储桶，用于存储要处理的图像。存储桶必须与即将启动的 AMAZON CloudFormation 堆栈位于**同一区域**。
- **准备已经 ICP 备案过的域名。**请准备一个或者两个经过 ICP 备案的域名，一个用于部署无服务器图像处理程序，另外一个用于部署演示程序(可选)。我们强烈建议您在首次部署解决方案以测试解决方案的功能时，部署演示用户界面。
- **(可选)上传 SSL 证书到 AMAZON Identity and Access Management (IAM)。**我们强烈建议您在生产环境中启用 SSL，如需启用，请提前使用 [AWS CLI](#) 的 [aws iam upload-server-certificate](#) 命令将 SSL 证书上传到 IAM 中。如果您选择启用 SSL，并且使用演示界面，请务必为两个域名皆配置证书。
- **配置 DNS 解析。**CloudFormation 堆栈部署完成后，您需要配置 CNAME 解析将域名指向 CloudFront，并且等待解析生效后方可使用该解决方案。您可以通过查看 CloudFormation 的输出来获取 CloudFront 的 DNS 地址。

步骤 1: 启动 CloudFormation 堆栈

此自动化 Amazon CloudFormation 模板在 亚马逊云科技 上部署无服务器图像处理程序。您负责运行此解决方案时使用的亚马逊云科技服务的成本。有关完整详细信息，请参阅此解决方案中所使用的每个亚马逊云科技服务的定价页面。

1. 登录到亚马逊云科技管理控制台，然后单击下面的按钮以启动无服务器图像处理程序 AMAZON CloudFormation 模板。



2. 默认情况下，该模板在由西云数据运营的亚马逊云科技（宁夏）区域启动。要在其他亚马逊云科技区域中启动无服务器图像处理程序，请使用导航栏中的区域选择器。
3. 在**创建堆栈**页面上，确认 **Amazon S3 URL** 文本框中显示正确的模板 URL，然后选择**下一步**。
4. 在**指定堆栈详细信息**页面上，为解决方案堆栈分配名称。
5. 在**参数**下，查看模板的参数并根据需要进行修改。此方案使用以下默认值：

	参数	默认	描述
1	ApiDomain	需要输入	访问该 API 服务的域名，必须是 ICP 备案过的域名
2	ApiCertificateId	需要输入（可选）	SSL 证书 ID, 需提前上传到 IAM, 如不输入，则不启用 HTTPS 访问
CORS Options			
4	CorsEnabled	Yes	选择是否启用跨域资源共享（CORS）。如果您希望部署演示界面，请选择 Yes。
5	CorsOrigin	*	该值将由 API 在 Access-Control-Allow-Origin 标头中返回。星号（*）可以支持任何源点。我们建议您指定一个特定来源（例如 `http://example.domain`），以限制跨站点访问您的 API。如果 CorsEnabled 参数设置为 No，则忽略此值。
Image Sources :			

11	SourceBuckets	需要输入	您帐户中的一个或多个 S3 存储桶，其中包含您将要操作的图像。如果提供多个存储桶，请用逗号分隔。请确保选择的 S3 存储桶和本解决方案位于同一个亚马逊云科技区域。
Demo UI :			
13	DeployDemoUI	Yes	将部署到 Demo S3 存储桶的演示界面
	DemoUIDomain	需要输入	演示界面所使用的域名，必须是 ICP 备案过的域名
	DemoUICertificateId	需要输入（可选）	SSL 证书 ID，需提前上传到 IAM，如不输入，则不启用 HTTPS 访问
Event Logging :			
	LogRetentionPeriod	1	将 Lambda 日志数据保留在 CloudWatch 日志中的天数
Other Parameters:			
	AutoWebP	No	您是否要基于接受 HTTP Header 启用自动 WebP？如果是，请选择“是”。

6. 选择**下一步**。
7. 在**配置堆栈选项**页面上，选择**下一步**。
8. 在**审核**页面上，查看并确认设置。确保选中确认模板将创建 AMAZON Identity and Access Management (IAM) 资源。
9. 选择**创建堆栈**以部署堆栈。

您可以在 AMAZON CloudFormation 控制台的状态列中查看堆栈的状态。您应该在大约 30 分钟内看到状态为 CREATE_COMPLETE。

步骤 2: 配置 DNS 解析

1. 点击创建完毕的 CloudFormation 堆栈，选择**输出**。

2. 在 CloudFormation 输出中可以看到为 **ApiEndpoint** 分配的 CloudFront CNAME, 即 **ApiEndpointCNAME**。以及为演示界面分配的 CloudFront CNAME, 即 **DemoUrlCNAME**。
3. 在您所使用的的 DNS 解析服务器配置 CNAME 解析, 分别将 API 和演示界面的域名指向 **ApiEndpointCNAME** 和 **DemoUrlCNAME**。

若您使用 Amazon Route53 作为您的 DNS 解析服务器, 请 [Route53 开发者手册](#)。

步骤 3：创建和使用图像请求

该解决方案生成一个 CloudFront 站点, 使您可以通过图像处理程序 API 访问原始图像和修改后的图像。您为 Image Handler Distribution 配置的 CNAME 域名即为该方案的访问地址, 我们称之为 **ApiEndpoint**。图像的位置和要进行的编辑等参数是在前端的 JSON 对象中指定的。

例如, 以下代码块将图像位置指定为 *myImageBucket*, 并指定灰度编辑: *true* 可以将图像更改为灰度。

```
const imageRequest = JSON.stringify({
  bucket: "myImageBucket",
  key: "myImage.jpg",
  edits: {
    grayscale: true
  }
});
const url = `${CloudFrontUrl}/${btoa(imageRequest)}`;

// Alternatively, you can call the url directly in an <img>
// element, similar to:
<img src=`${url}` />
```

1. 在代码沙箱或您的前端应用程序中, 创建一个新的 JSON 对象。该对象将包含成功检索图像并对其进行编辑所需的键值对。
2. 使用上面的代码示例和 Sharp 文档, 调整以下属性以满足您的图像编辑要求。

Bucket - 指定包含原始图像文件的 Amazon S3 存储桶。这是 **SourceBuckets**

模板参数中指定的名称。您可以通过将图像存储桶添加到图像处理程序

AMAZON Lambda 函数的 **SOURCE_BUCKETS** 环境变量中来更新图像位置。

Key - 指定原始图像的文件名。该名称应包括文件扩展名以及其位置与存储桶根之间的所有子文件夹。例如，**folder1 / folder2 / image.jpg**。

Edits - 将图像编辑方式指定为键值对。如果未指定图像编辑方式，则将返回原始图像，并且不做任何更改。

3. 对图像请求进行字符串化和编码。您可以使用 JavaScript 的 `JSON.stringify()` 属性，然后使用 `btoa()` 属性对结果进行编码。
4. 将编码结果附加到您的 **ApiEndpoint URL** 上，并将其用作 **HTML `img src`** 属性的值或 **GET 请求** 中的值。请参见以下示例。

```
const imageRequest = JSON.stringify({
  bucket: "myImageBucket",
  key: "myImage.jpg",
  edits: {
    grayscale: true
  }
});
const url = `${CloudFrontUrl}/${btoa(imageRequest)}`;

// Alternatively, you can call the url directly in an
<img> element, similar to:
<img src=`${url}` />
```

5. 以下是上述代码中产生编码图像请求的示例：

```
https://<ApiEndpoint>/<base64encodedRequest>
```

步骤 4: (可选) 使用演示用户界面

该解决方案提供了一个可选的演示用户界面，可以将其部署到您的亚马逊云科技账户中以显示基本功能。通过此用户界面，您可以使用帐户中指定的 S3 存储桶中的图像直

接与新处理程序进行交互。使用以下过程尝试所有受支持的图像编辑功能，预览结果并创建可在应用程序中使用的示例 URL：

The screenshot shows the 'Serverless Image Handler Demo' web application. It is divided into three main sections: 'Image Source', 'Editor', and 'Preview'.
1. **Image Source**: Contains two text input fields. The first is labeled 'Enter the name of an Amazon S3 bucket in your account that contains original image files.' and contains the text 'my-sample-bucket-10'. The second is labeled 'Enter the name of an original image file (with extension) stored in the above Amazon S3 bucket.' and contains the text 'sample-2.jpeg'. Below these is a blue 'Import' button.
2. **Editor**: Contains various controls for image editing. It has 'Width' and 'Height' input fields, both set to '200'. A 'Resize Mode' dropdown is set to 'Disabled'. There are 'Fill Color' and 'Background Color' input fields, both set to '#FF0000'. Below these are checkboxes for 'Grayscale', 'Negative', 'Flip', 'Flatten', 'Flop' (checked), and 'Normalize'. There are also 'Smart Cropping' and 'Crop Padding' input fields, both set to '0'. At the bottom of the editor are red 'Reset' and green 'Preview' buttons.
3. **Preview**: On the left, it shows the 'Original Image' as a night cityscape. On the right, it shows the 'Request Body' as a JSON object:

```
{  "bucket": "my-sample-bucket-10",  "key": "sample-2.jpeg",  "edits": {    "flip": true,    "tint": {      "r": 0,      "g": 0,      "b": 255    }  }  }
```

. Below the request body is the 'Encoded URL' field, which contains the URL: `https://d1f1r1zy8tv3q48.cloudfront.net/im`.

1. 在 AMAZON CloudFormation 堆栈的输出选项卡中，选择 **DemoUrl**。在新选项卡中打开无服务器图像处理程序 **DemoUrl**。
2. 在 **Image Source** 区域指定要用于演示的**存储桶名称**和**图像访问路径**。确保在密钥中包含文件扩展名。请注意，您指定的存储桶必须列在 AMAZON Lambda 函数的 `SOURCE_BUCKETS` 环境变量中。
3. 选择导入，原始图像出现在原始图像卡中。
4. 在**编辑器**部分中，调整图像设置，然后选择**预览**以生成修改后的图像。您可以选择重设将设置恢复为原始值。