

AWSで画像リサイズAPIを作ろう

2023/12/14(金) AWS Gameday
石塚 峻斗

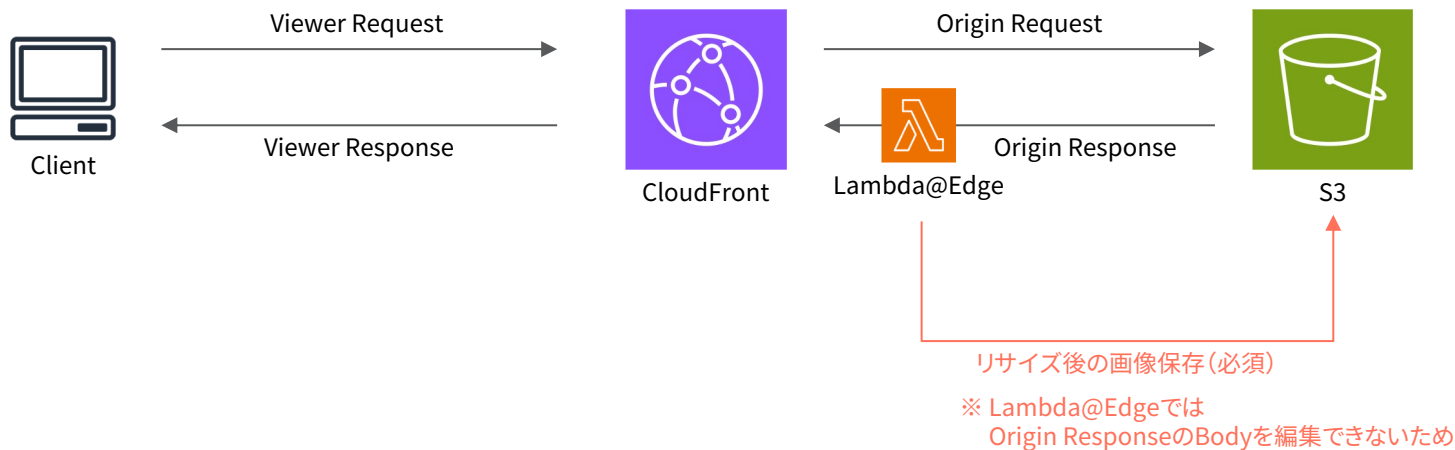
お題:

AWSで画像リサイズAPIを作成せよ

要件:

1. 画像はS3に保存する
2. CloudFrontでキャッシュする
3. リサイズ後の画像はS3に保存しない
※ 保守性の観点

1. Lambda@Edgeの利用 [[2018年3月7日の公式ブログ](#)]

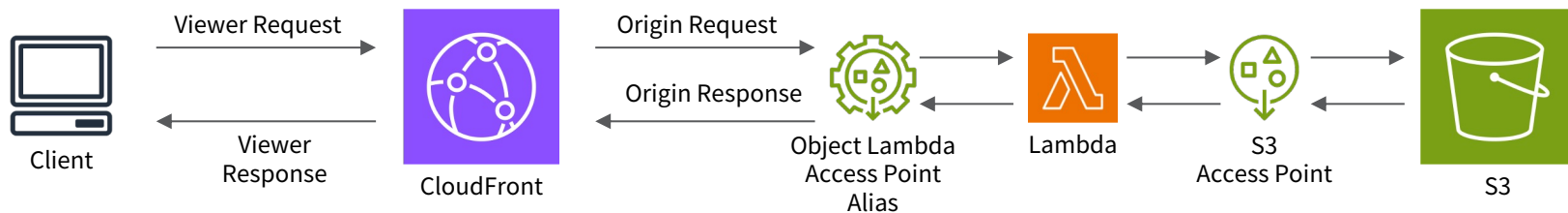


2. CloudFront Functionsの利用



CloudFront FunctionsはOrigin側に配置できない
画像を扱うヘビーな処理に向いていない

3. Object Lambdaの利用 [[2023年3月17日の公式ブログ](#)]



3案の比較検討

	キャッシュ	リサイズ後の 画像保存不要性	デプロイ容易性
Lambda@Edge	◎	×	× CloudFrontへのデプロイ (10分かかることも)
CloudFront Functions	×	-	-
S3 Object Lambda	◎	◎	◎ Lambdaのデプロイ (長くても数分)

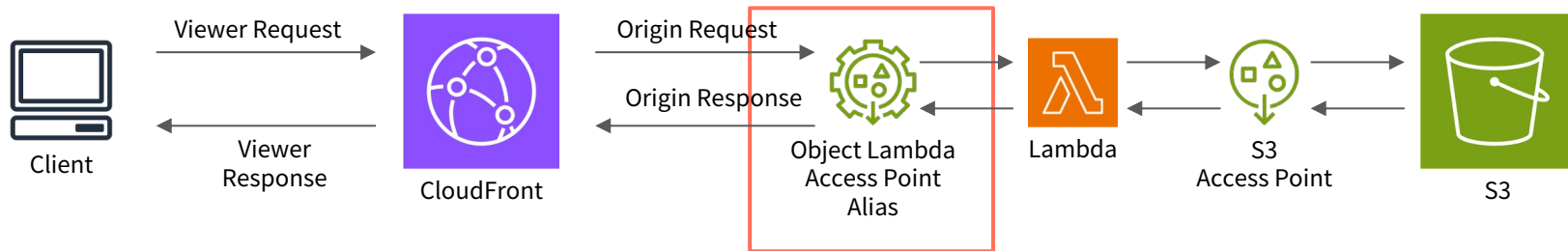
※ 実はLambda Function URLsを使う手もありますが、今回は割愛

実装してみよう

Terraformに落とし込むだけ！

…と思っていたのですが

Object Lambda Access Pointのalias attributeがない…

**Attribute Reference**

This resource exports the following attributes in addition to the arguments above:

- `arn` - Amazon Resource Name (ARN) of the Object Lambda Access Point.
- `id` - The AWS account ID and access point name separated by a colon (`:`).

※ v5.16.1時点

二段階でapplyするという
回避策はありますが

S3 Object Lambda Access Point aliases for Amazon CloudFront #30038

[New issue](#)

🔒 Closed

ewbankkit opened this issue on Mar 16 · 3 comments · Fixed by #33388



ewbankkit commented on Mar 16 · edited

Contributor · ...

You can now use an S3 Object Lambda Access Point alias as an origin for your Amazon CloudFront distribution...

- [Announcement](#)
- [Blog post](#)

Requires [AWS SDK for Go v1.44.222: #30040](#).

`aws_s3control_object_lambda_access_point` add `alias` attribute.

👍 11



github-actions bot commented on Mar 16

...

Community Note

Voting for Prioritization

- Please vote on this issue by adding a 🙌 [reaction](#) to the original post to help the community and maintainers prioritize this request.
- Please see our [prioritization guide](#) for information on how we prioritize.
- Please do not leave "+1" or other comments that do not add relevant new information or questions, they generate extra noise for issue followers and do not help prioritize the request.

Assignees

No one assigned

Labels

[enhancement](#) [service/s3control](#)

Projects

None yet

Milestone

v5.16.2

Development

Successfully merging a pull request may close this issue.

🔗 [t/aws_s3control_object_lambda_access_point: a...](#)
[beginaid/terraform-provider-aws](#)

2 participants



TerraformのAWS Providerに
Contributeしてみようと決心

1

そもそも何から
ハジメレバヨイノカ



公式ドキュメントが
懇切丁寧

2

Go言語
ナニモワカラン



A tour of Goを
1日で完走

3

終わりが
ナニモミエナイ



タイムボックスを切って
自分に鞭を打つ

 Terraform AWS Provider - Contributor Guide

Search

hashicorp/terraform-provi...
v5.30.0 ☆ 9.1k ▼ 8.5k

Terraform AWS Provider - Contributor Guide

[Welcome](#)

Contribute >

Developer Reference >

Submit an Issue

FAQ

Welcome

The Terraform AWS Provider is the work of thousands of contributors, and is maintained by a small team within HashiCorp. This site contains extensive instructions about how to contribute and how the AWS provider works.

 **Tip**

This documentation is intended for Terraform AWS Provider code developers. Typical operators writing and applying Terraform configurations do not need to read or understand this material.

Contribute

Please follow the following steps to ensure your contribution goes smoothly.

1. Configure Development Environment

Install Terraform and Go. Clone the repository, compile the provider, and set up testing. Refer to [Configure Development Environment](#).

2. Debug Code

If you are looking to *create or enhance code*, such as a new resource or adding an argument to an existing resource, skip to the next step.

Finding and fixing errors in the AWS Provider can be difficult. We have a [debugging guide](#) to help you get started.

3. Change Code

Table of contents

Contribute

1. Configure Development Environment

2. Debug Code

3. Change Code

4. Write Tests

5. Update the Changelog

6. Create a Pull Request

Submit an Issue

Join the Contributors Slack

学んだことはブログに 即時アウトプット



本稿では、[A Tour of Go](#)を読んで学んだことを整理してまとめています。

目次

- 1 | Basics
 - 1-1 | Packages
 - 1-2 | Exported names
 - 1-3 | Declaration
 - 1-4 | Functions
 - 1-5 | Named return values
 - 1-6 | Basic types
 - 1-7 | Zero values
- 2 | Flow control statements
 - 2-1 | For
 - 2-2 | If
 - 2-3 | Switch
 - 2-4 | Defer
- 3 | More types
 - 3-1 | Pointers
 - 3-2 | Struct
 - 3-3 | Arrays
 - 3-4 | Slices
 - 3-5 | Map
 - 3-6 | Closure
- 4 | Methods and Interface
 - 4-1 | Methods
 - 4-2 | Pointer receiver
 - 4-3 | Pointers and functions
 - 4-4 | Interfaces
 - 4-5 | Type assertions
 - 4-6 | Errors
 - 4-7 | Readers
- 5 | Concurrency
 - 5-1 | Goroutines
 - 5-2 | Channels
 - 5-3 | Buffer
 - 5-4 | Close
 - 5-5 | Select
 - 5-6 | sync.Mutex

折りたたむ



f/aws_s3control_object_lambda_access_point: add alias attribute to s3 object lambda #33388

New issue

Merged ewbankkit merged 7 commits into hashicorp:main from beginaid:f-add-alias-attribute-to-s3-object-lambda on Sep 11

Conversation 5 Commits 7 Checks 55 Files changed 4 +49 -5

beginaid commented on Sep 10 • edited

Description

As described in [this document](#), S3 object lambda access point can get an alias. Terraform provider does not support getting an alias, however. This PR modifies the existing `aws_s3control_object_lambda_access_point` resource to allow alias to be obtained.

Relations

Closes [#30038](#)

References

- API Reference
 - [GetAccessPointForObjectLambda](#)
 - [ObjectLambdaAccessPointAlias](#)
- Go Packages
 - [GetAccessPointForObjectLambdaWithContext](#)
 - [TestMatchResourceAttr](#)

Output from Acceptance Testing

```
$ make testacc TESTS=TestAccS3ControlObjectLambdaAccessPoint_basic PKG=s3control
==> Checking that code complies with gofmt requirements...
TF_ACC=1 go test ./internal/service/s3control/... -v -count 1 -parallel 20 -run='TestAccS3ControlObjectLambdaA
=== RUN   TestAccS3ControlObjectLambdaAccessPoint_basic
=== PAUSE TestAccS3ControlObjectLambdaAccessPoint_basic
=== CONT TestAccS3ControlObjectLambdaAccessPoint_basic
--- PASS: TestAccS3ControlObjectLambdaAccessPoint_basic (63.86s)
PASS
ok      github.com/hashicorp/terraform-provider-aws/internal/service/s3control  68.530s
...
```

Reviewers

ewbankkit ✓

Assignees

No one assigned

Labels

[documentation](#) [service/s3control](#) [size/S](#) [tests](#)

Projects

None yet

Milestone

v5.16.2

Development

Successfully merging this pull request may close these issues.

🔗 S3 Object Lambda Access Point aliases for Amaz...

3 participants

5.16.2 (September 11, 2023)

FEATURES:

- New Data Source: `aws_cognito_identity_pool` (#33053)
- New Resource: `aws_verifiedaccess_trust_provider` (#33195)

ENHANCEMENTS:

- resource/aws_autoscaling_group: Change the default values of `instance_refresh.preferences.scale_in_protected_instances` and `instance_refresh.preferences.standby_instances` from `Wait` to the [Amazon EC2 Auto Scaling console recommended value](#) of `Ignore` (#33382)
- resource/aws_s3control_object_lambda_access_point: Add `alias` attribute (#33388) 🥰

BUG FIXES:

- resource/aws_autoscaling_group: Fix `ValidationError` errors when starting Auto Scaling group instance refresh (#33382)
- resource/aws_iam_topic_rule: Fix `InvalidParameter` errors on Update with Kafka destinations (#33360)
- resource/aws_lightsail_certificate: Fix validation of `name` (#33405)
- resource/aws_lightsail_database: Fix validation of `name` (#33405)
- resource/aws_lightsail_disk: Fix validation of `name` (#33405)
- resource/aws_lightsail_instance: Fix validation of `name` (#33405)
- resource/aws_lightsail_lb: Fix validation of `lb_name` (#33405)
- resource/aws_lightsail_lb_attachment: Fix validation of `lb_name` (#33405)
- resource/aws_lightsail_lb_certificate: Fix validation of `lb_name` (#33405)
- resource/aws_lightsail_lb_certificate_attachment: Fix validation of `lb_name` (#33405)
- resource/aws_lightsail_lb_https_redirection_policy: Fix validation of `lb_name` (#33405)
- resource/aws_lightsail_lb_stickiness_policy: Fix validation of `lb_name` (#33405)

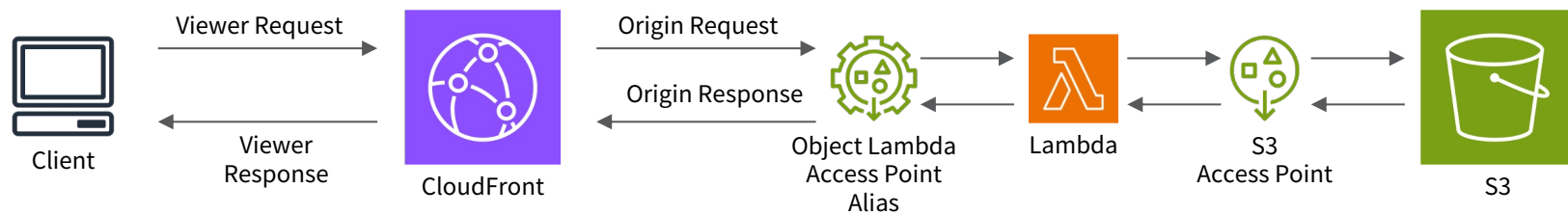
Attribute Reference

This resource exports the following attributes in addition to the arguments above:

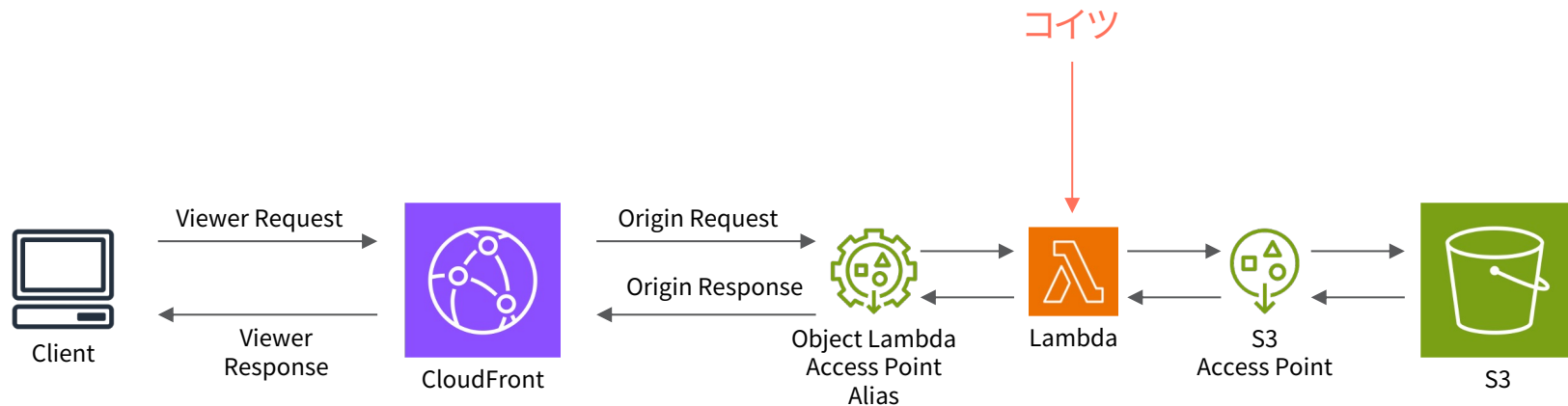
- `alias` - Alias for the S3 Object Lambda Access Point. 
- `arn` - Amazon Resource Name (ARN) of the Object Lambda Access Point.
- `id` - The AWS account ID and access point name separated by a colon (`:`).

※ v5.16.2以降

無事実装完了!



コイツ気になりますよね



Nodeで実装してみました (超省略コード)

仕様

Method	Query	Type	Example
GET	w	int	https://{domain}/{ファイル名}?w=500

エラーハンドリング後の世界

Status	Body	Example
200	バイナリ	キャッシュヒット/ミスに拘らず200を返す
400	空	Query Paramsが不正 指定されたobjectのcontent-typeが不正
404	空	指定されたパスに画像が存在しない
500	空	内部エラー

```
const { S3Client, WriteGetObjectResponseCommand } = require("@aws-sdk/client-s3");
const s3 = new S3Client({ region: "ap-northeast-1" });
const axios = require("axios").default;
const sharp = require("sharp");

exports.handler = async (event) => {
  // object lambdaのeventから必要なフィールドを抽出する
  const {
    getObjectContext: { inputS3Url, outputRoute, outputToken },
    userRequest: { url },
  } = event;
  // S3 originからレスポンスを受け取る
  const axiosResponse = await axios.get(inputS3Url, { responseType: "arraybuffer" });
  // レスポンスから画像データを抽出する
  let data = axiosResponse.data;
  // クエリパラメータからリサイズ後のwidthを取得
  const { searchParams } = new URL(url);
  const width = parseInt(searchParams.get("w"));
  // リサイズ実行
  data = await sharp(data).resize(width).toBuffer();
  // クライアントに画像を返す
  let response200 = {
    StatusCode: 200,
    RequestRoute: outputRoute,
    RequestToken: outputToken,
    ContentType: axiosResponse.headers["content-type"],
    Body: data,
  };
  // GetObjectResponseを書き換える
  await s3.send(new WriteGetObjectResponseCommand(response200));
  return { statusCode: 200 };
}
```

まとめ

1. 画像リサイズAPIにはS3 Object Lambdaが便利
2. Terraformは最新のAWSに追従しているとは限らない