

その接続先情報はどこに

株式会社Gunosy 浜地亮輔

おまえだれ

- @aibou (はまじ りょうすけ) 
- 「あいぼうさんって本名なんでしたっけ？」
- Gunosyサービスのインフラ全般担当(AWS歴1年ちょい)
- Java(Spring Boot), Ruby, Chef
- 趣味：スポーツ観戦    モータースポーツ
 - 昨日DAZN登録しました。興味ある人は懇親会でぜひ
- 炎上歴あり

AWS Lambda
使っている方に質問です

秘密な情報どこに書いてますか？
(DB接続情報等)

AWSコンソールのエディタに

コード直書き 

**Jenkins等の自前ビルドサーバ
でパラメータ化** 🙋

オンプレリポジトリ(GHE, svn)

だからハードコード👩🏻

GunosyではAWS KMSを使って 接続先情報を暗号化しています

Lambda使う時は、です。

普段はOpsworksのCustomJsonに書いてます

今日はそのお話をします

あっ・・・

Developers.IO

ホーム

AWS

iPhone

Android

Server

求人情報



KMSで認証情報を暗号化しLambda実行時に復号化する

Key Management Service

Lambda

2015年11月29日

irbbb (17)

37

24

f

10

はてぶ

View

t

G+1

Pocket

データベースやAPIサーバーなど外部システムと連携する際には、往々にして認証情報が必要になります。

今回は [AWS Key Management Service](#) (以下 KMS) の共通鍵暗号の仕組みを使い、暗号化した認証情報を AWS Lambda 関数のコードに埋め込み、関数呼び出し時に認証情報を復号化する方法を紹介します。

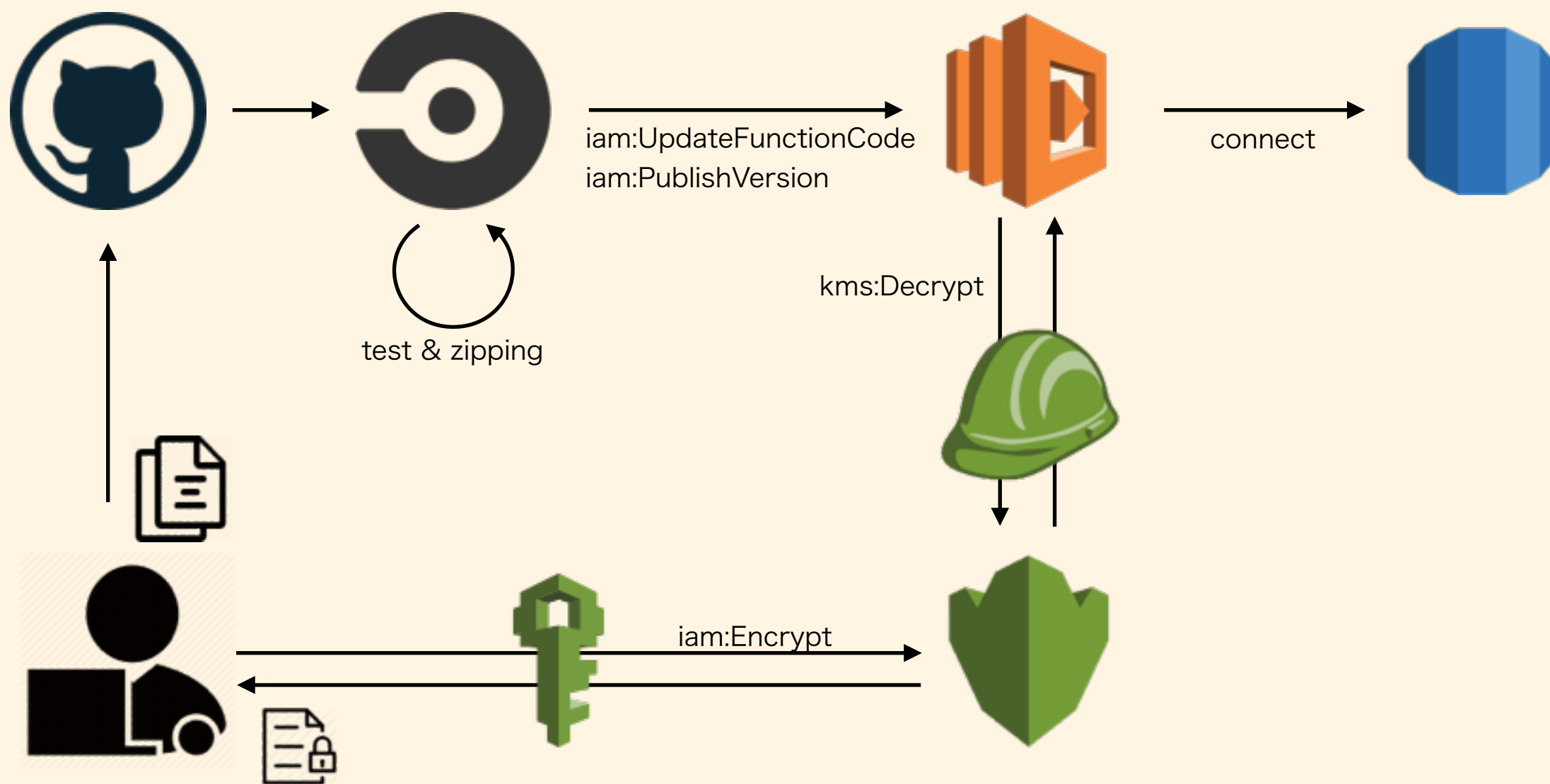
基本的なアイデアは次のブログで書かれており、KMS を使った暗号化処理だけを自分向けメモも兼ねて抜き出しました。

<http://ijin.github.io/blog/2015/08/06/github-to-lambda-to-slack/>

<http://dev.classmethod.jp/cloud/decrypt-sensitive-data-with-kms-on-lambda-invocation/>

Gunosyでは**AWS KMS**を使って
接続先情報が入ったファイルを
丸ごと暗号化しています

流れ的な



Configファイルのサンプル

```
{  
  "DATABASES": {  
    "master": {  
      "HOST": "",  
      "PORT": "",  
      "NAME": "",  
      "USER": "",  
      "PASSWORD": ""  
    }  
  }  
}
```

シンプル！

もちろんDB以外の設定も記載してたりします

Lambda上でのデコード (Python)

```
kms = boto3.client('kms')
with open(encrypted_json_path) as f:
    _settings = json.loads(
        kms.decrypt(CiphertextBlob=f.read())['Plaintext']
    )
```

```
DATABASES = _settings.get("DATABASES") or {}
```

```
# DATABASES['master']['HOST']
```

```
# DATABASES['master']['USER']
```

hoge/__init__.py に書いておけば
import hoge で使えるのも○

困ったこと

- 権限の払い出し
- 設定更新がめんどくさい
- 現状どういう設定になってるかわからない

この辺は  を使えば解決できそう

裏ワザ

lambda functionの

`description`に設定を埋める！

function description

	Function name ▼	Description ▼	Runtime ▼	Code size ▼	
☐	aibou-urawaza	{"name": "aibou"}	Python 2.7	314 bytes	3 m

Descriptionから取得 (Python)

```
import boto3
import json

def lambda_handler(event, context):
    configuration = boto3.client('lambda').get_function_configuration(
        FunctionName=context.function_name
    )

    json_conf = json.loads(configuration['Description'])
    return json_conf['name']
```

✓ Execution result: succeeded ([logs](#))

The area below shows the result returned by your function execution.

"aibou"

最後に

AWS Lambda **Ruby** オネシャス！