

Alibaba Cloud の Function Compute

Alibaba Cloud の Function Compute を使用して関数を設定しました。私の目標は、GFW (Great Firewall) からプロキシサーバーの活動を隠すために、通常に見えるトラフィックを生成することです。これを行うために、プロキシと並行して帯域幅サーバーをデプロイしました。そして、Alibaba Cloud の Function Compute を使用して、この帯域幅 API に毎分リクエストを送り、通常のトラフィックとプロキシのトラフィックを混在させています。

```
from flask import Flask, request, jsonify
import requests
import time

REQUEST_ID_HEADER = 'x-fc-request-id'

app = Flask(__name__)

@app.route('/', defaults={'path': ''})
@app.route('/<path:path>', methods=['GET', 'POST', 'PUT', 'DELETE'])
def hello_world(path):
    # リクエスト ID とその他の詳細をログに記録
    rid = request.headers.get(REQUEST_ID_HEADER)
    print("FC Invoke Start RequestId: " + rid)
    data = request.stream.read()
    print("Path: " + path)
    print("Data: " + str(data))

    # カウンターを初期化
    start_time = time.time() # 開始時間を記録
    duration = 60 # 1 分間 (60 秒) 実行
    total_calls = 0 # 総 API 呼び出し数を追跡
    successful_calls = 0 # 成功した API 呼び出し数を追跡

    # 1 分間ループ
    while time.time() - start_time < duration:
        try:
            # 外部の/bandwidth API を呼び出す
            response = requests.get('https://www.lzwjava.xyz/bandwidth')
            response.raise_for_status() # HTTP エラーが発生した場合に例外を発生させる
            successful_calls += 1 # 成功した呼び出しカウンターを増やす
        except Exception as e:
```

```
    print(" 帯域幅データの取得中にエラーが発生しました:", e)

finally:
    total_calls += 1 # 総呼び出しカウンターを増やす

# 次のリクエストまで 5 秒待機
time.sleep(5)

# リクエストの終了をログに記録
print("FC Invoke End RequestId: " + rid)

# 呼び出し数と成功した呼び出し数を返す
return jsonify({
    "message": "Hello, World!",
    "total_calls": total_calls,
    "successful_calls": successful_calls
})

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=9000)
```