

# Function Compute on Alibaba Cloud

I've set up a function using Alibaba Cloud's Function Compute. My goal is to generate some normal-looking traffic to help obscure my proxy server's activity from the GFW. To do this, I've deployed a bandwidth server alongside my proxy. Now, I'm using Alibaba Cloud's Function Compute to make a request to this bandwidth API every minute, creating a mix of normal and proxy traffic.

```
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):
    # Log the request ID and other details
    rid = request.headers.get(REQUEST_ID_HEADER)
    print("FC Invoke Start RequestId: " + rid)
    data = request.stream.read()
    print("Path: " + path)
    print("Data: " + str(data))

    # Initialize counters
    start_time = time.time() # Record the start time
    duration = 60 # Run for 1 minute (60 seconds)
    total_calls = 0 # Track total API calls
    successful_calls = 0 # Track successful API calls

    # Loop for 1 minute
    while time.time() - start_time < duration:
        try:
            # Call the external /bandwidth API
            response = requests.get('https://www.lzwjava.xyz/bandwidth')
            response.raise_for_status() # Raise an exception for HTTP errors
            successful_calls += 1 # Increment successful calls counter
        except Exception as e:
            print("Error fetching bandwidth data:", e)
```

```

finally:
    total_calls += 1 # Increment total calls counter

    # Wait for 5 seconds before the next request
    time.sleep(5)

# Log the end of the request
print("FC Invoke End RequestId: " + rid)

# Return the number of calls and successful calls
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)

```