# Activity 4: Serverless Architectures

You will write and deploy a calculator web application using Amazon's serverless architecture, AWS API Gateway and AWS Lambda. Your application will take a client's POST request that has a body like this:

```
{
  "a": "9",
  "b": "6",
  "op": "+"
}
```

and returns a result that looks like this:

```
{
  "result": "15"
}
```

The calculator should support these 4 operands: +, -, *, /. The logic should be implemented in Python 3.x. Use this code as a starting point.

```
import json

def lambda_handler(event, context):
    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps(json.loads(event["body"]))
    }
```

To test your application, you may use Postman or cURL from any EC2 instance (or from your notebook). Here is an example of using cURL:

```
curl -X POST -d "{\"key1\": \"value1\"}"  -H "Content-Type:
application/json" https://592tu1lmm8.execute-api.ap-northeast-
1.amazonaws.com/default/calculator
```

Next, install siege on an EC2 instance. We will use siege to benchmark our serverless application. We need to install siege a bit differently this time to make it support https (ssl). Follow these steps.

Get an **EC2 Amazon Linux 2** instance (micro instance free tier). Select an AZ in the same region as your deployed serverless app. Hmm..but which zone was your Lambda Function deployed to? Was it deployed in just one zone? :) Probably not...

SSH into your instance using your .pem file

Install gcc

```
$ sudo yum install gcc openssl openssl-devel
```

Download, build, and install siege like what you did for Activity 2 and 3

Run against a web application

```
$ siege -c5 -d1 -r1 --content-type "application/json" 'https://592tu1lmm8.execute-
api.ap-northeast-1.amazonaws.com/default/calculator POST {"key1" : "value2"}'
```

Look at the questions below and benchmark your serverless calculator application.

# Questions

Zip together your answers to these questions and your Python application source code, and upload them to mycourseville.

1. Run siege using the same number of clients (supposedly large) as Activity 2 and 3. Observe the response times, and the variations in response times across all requests. Do all requests see the same response times? Are there variations? Explain what you observe and why (same response time or variation)?
2. Wait for at least 20 minutes. Rerun siege. Observe the response times, and the variations in response times across all requests. Are the results the same as question 1? Do you observe any start up delays? Are they shorter or longer than before?
3. How much did this experiment cost you? How does this compare to if you were to use EC2 to run your application? How does this compare to if you were to use Elastic Beanstalk to run your application?
4. Apparently, with AWS Lambda, auto-scaling happens without user configuration. Compare the pro's and con's with configuring auto-scaling for IaaS and PaaS in Activity 3.
5. Compare the ease of development and ease of deployment between getting an app ready to work on AWS Lambda vs. if you were to get it to work on EC2.