AWS Introduction

Instructions:

1. Read the directions for each task very carefully before you begin working on it.
2. Every time the directions say something like "take a screenshot", do it and then copy it into the proper space in this document.

This worksheet is intended to be an introduction to AWS cloud infrastructure. In it, you will be creating your first Lambda. A Lambda is basically a short lived docker container that holds your code; AWS is in charge of spinning it up and tearing it down upon request.

Problem 0: Setup

I gave you AWS access to my AWS account (598791268315). Check your email for an email from me with the subject “AWS Creds” that contains your username and password. You should be able to log in by going to the console’s sign in page and putting in your username password.

Once you login:

1. Setup MFA by clicking on your user name at the top right and clicking on `Security Credentials`.

2. Navigate to the `United States: N. California` region (`us-west-1`) by selecting it from the top right of the console page.

3. Navigate to the Lambda Service by searching for it in the “service search bar” at the top left of the console.

Problem 1: Research

1. Watch as much of the following videos as you find useful:
   - AWS Lambda Introduction
   - AWS Lambda → Lambda Invocation

2. AWS is broken up into regions. I have assigned you the `us-west-1` region to avoid you colliding with anyone else who might be using these worksheets. In your 3 - 5 of your own sentences, explain what an AWS region is and why they are useful.

>

Problem 2: Action

Complete each of the following "steps" on your way to creating a simple AWS Lambda:

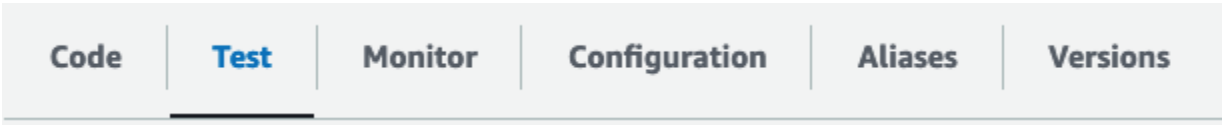
1. Double check that you're in the `United States: N. California` region (`us-west-1`).

2. Create your own `HelloWorldLambda` Python Lambda via the console that simply prints `"hi"` and returns the default JSON message. Copy the source code in the space below.

   ```

   ```

3. Deploy and test your changes of the Lambda by going to the Lambda's `Test` tab as seen in the picture below. Name your test `GoEvent` to trigger the lambda. Save and then Test it. Check the output to verify that you can see the `"hi"`



4. Alter the code such that it expects an event of the form: `{"name": "Bill", "age": 65}` and returns a json object of the form: `{"body": "Hello Bill, 65 is a great age", "statusCode": 200"}`.

5. Test `HelloWorldLambda` by going to the `Test` tab. Name your test `GreetingsEvent` to trigger the Lambda. Save and then test it. Check the output to verify it does what you expect. Paste a screenshot of the “correct output” in the space below:

   ```

   ```

6. Copy the source code in the space below.

   ```

   ```

7. Navigate to the Cloudwatch service and find the log group associated with the various runs of `HelloWorldLambda`. Take a screenshot of the logs of the various runs and paste it into the space below:

   ```

   ```

Problem 3: Reflection

Answer each of the following questions with a few sentences. Even if the answer seems trivial, take the time to write it down because it is most likely an important concept you will be building off of.

1. Two of the most important “services” you will be using for the rest of this project are “Lambda” and “Cloudwatch”. What parts of a normal program are these two services similar to?

   >

2. Having the correct “hook” is of the utmost importance when writing code to be used by AWS. Why is this the case?

   >

3. Status codes are an interesting part of the HTTP protocol that many people don't fully appreciate.

    a. What does a 200 mean?

        >

    b. When would you return a 400?

        >

    c. Why are status codes useful (rather than just some text explaining what happened)?

        >

4. What is a "lambda cold start" and a "lambda warm start"? What does this show about how your code is stored/run by AWS?

    >

5. Did you notice any obvious errors with this worksheet that should be fixed for future students? If so, what were they?

    >

6. Were there things in the worksheet that were overly confusing and that you think could be improved? If so, what were they?

    >