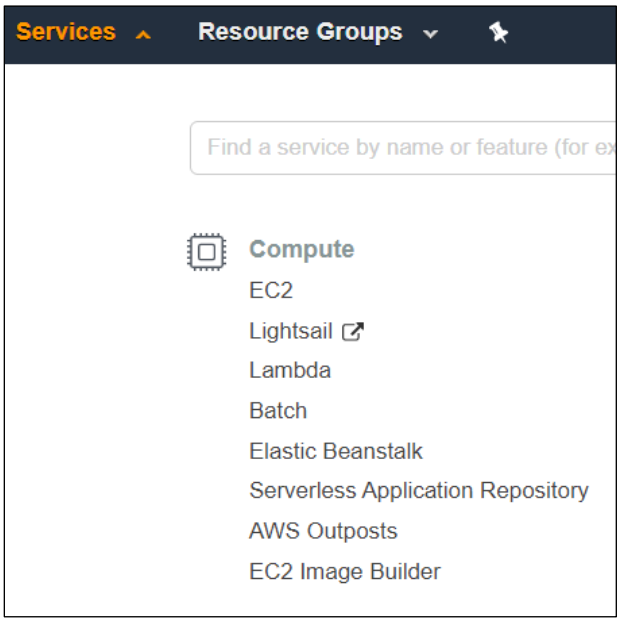
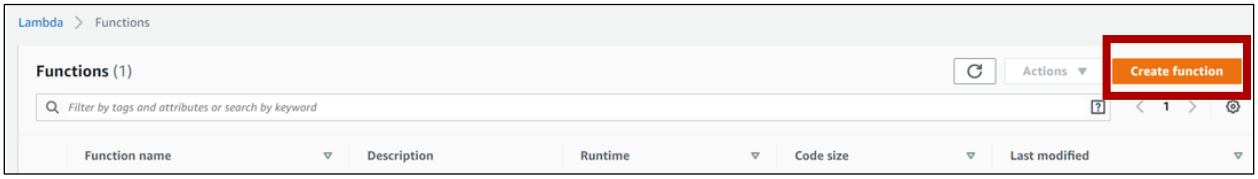
CS 470 Module Four Assignment One Guide

Part One – Creating and Testing a Lambda

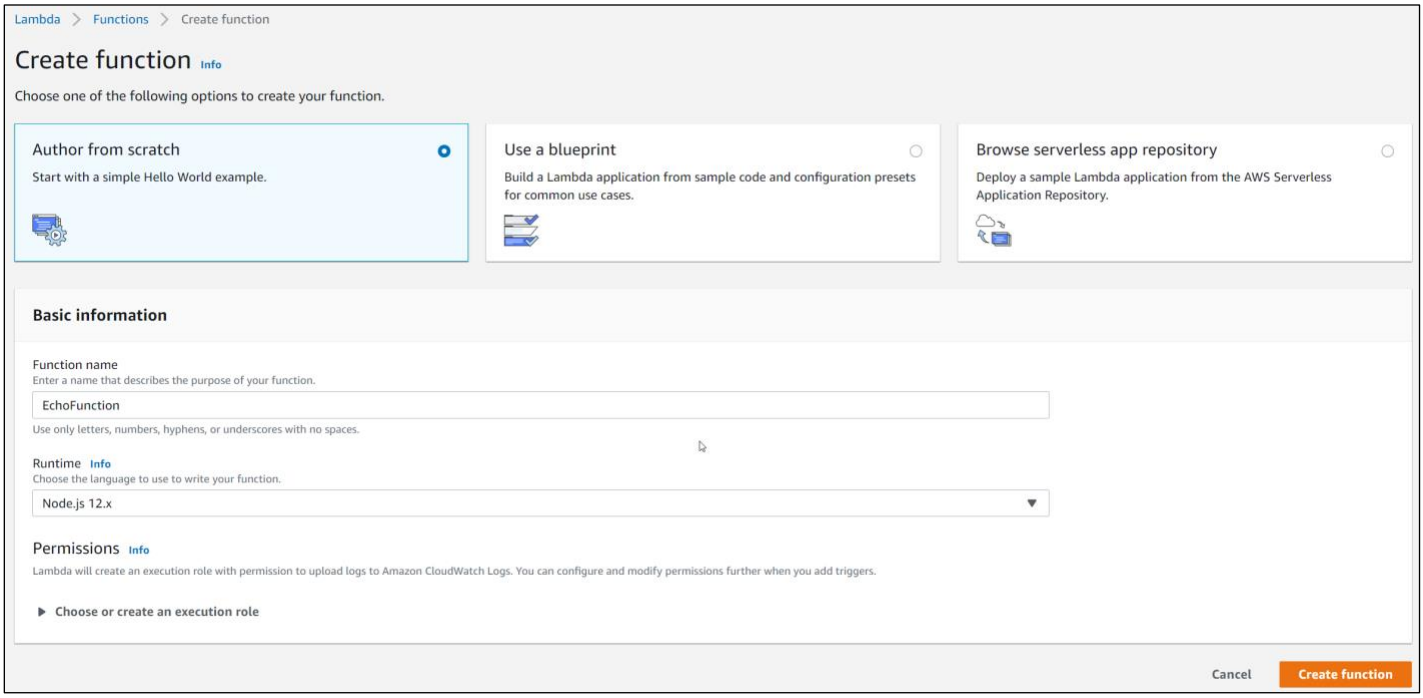
1. Navigate to **AWS Lambda** by typing “Lambda” into the console search bar or selecting **Lambda** under **Compute**.



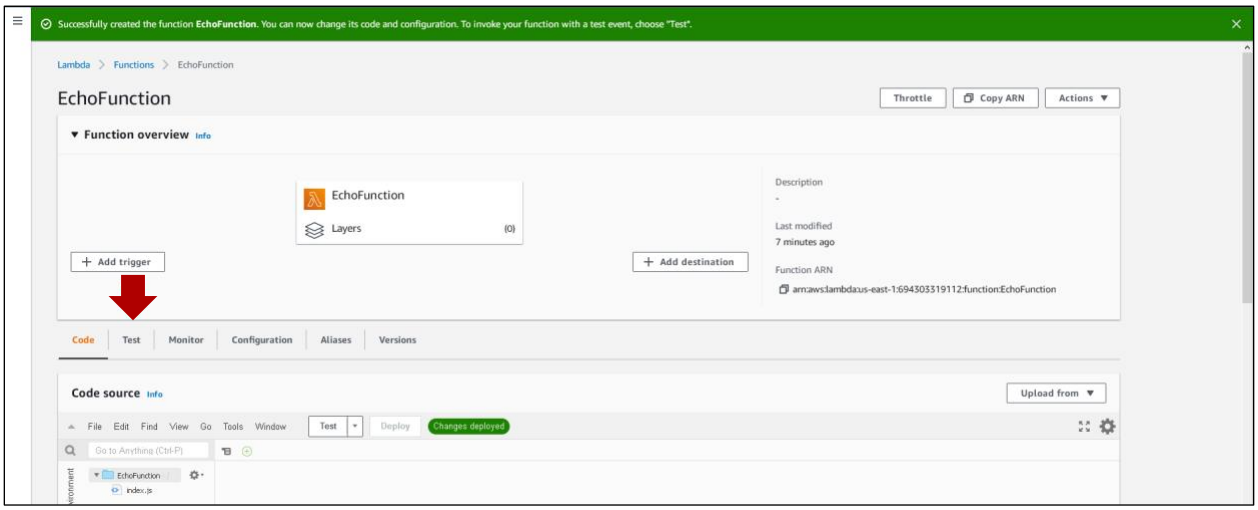
2. Click the orange **Create function** button in the upper-right corner.



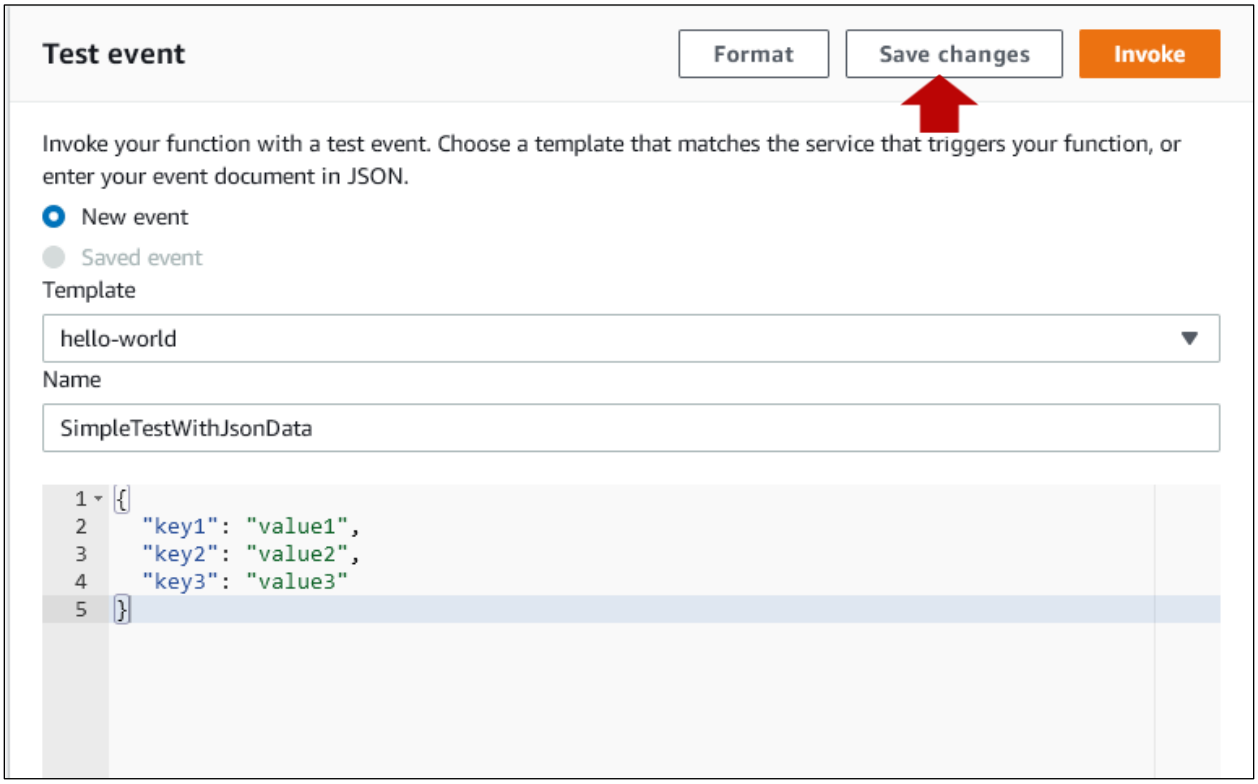
3. Make sure **Author From Scratch** is selected.
4. Enter the function name “EchoFunction”.
5. Set the runtime to “Node.js 12.x”.
6. Leave the permissions option set to **Create a new role with basic Lambda permissions.**
7. Click the orange **Create Function** button in the lower-right corner.



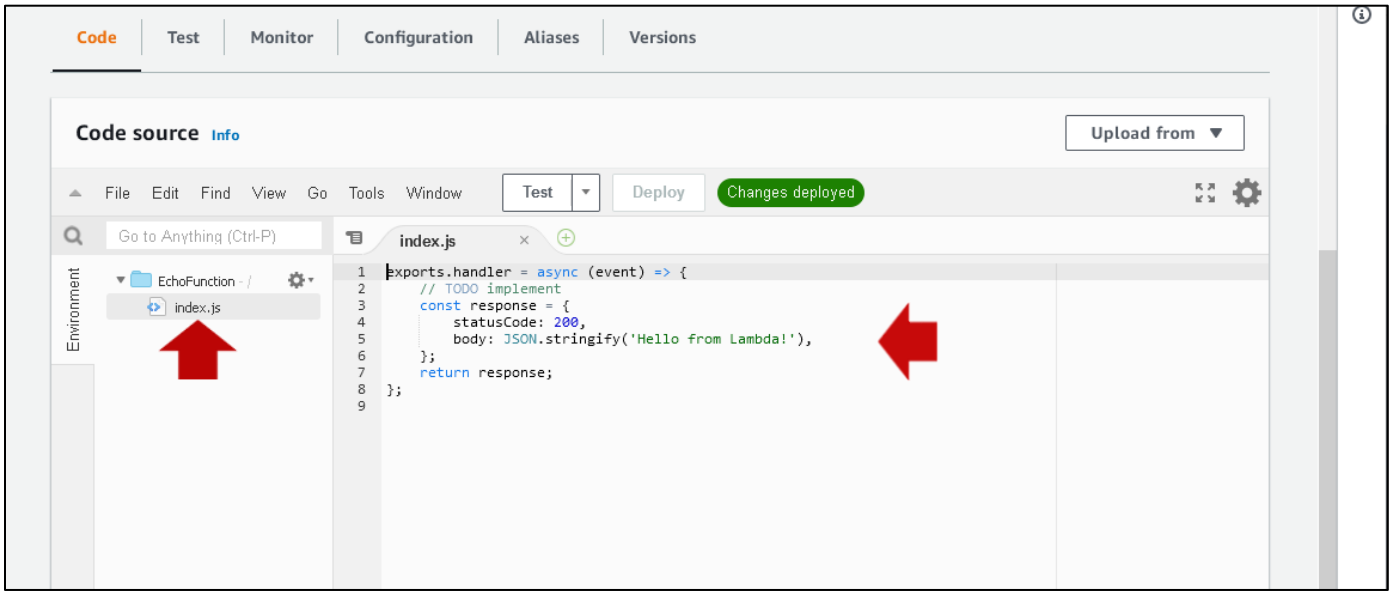
8. Congratulations! You have created your first serverless compute function – a bouncing baby Lambda! You should now see a screen like this:



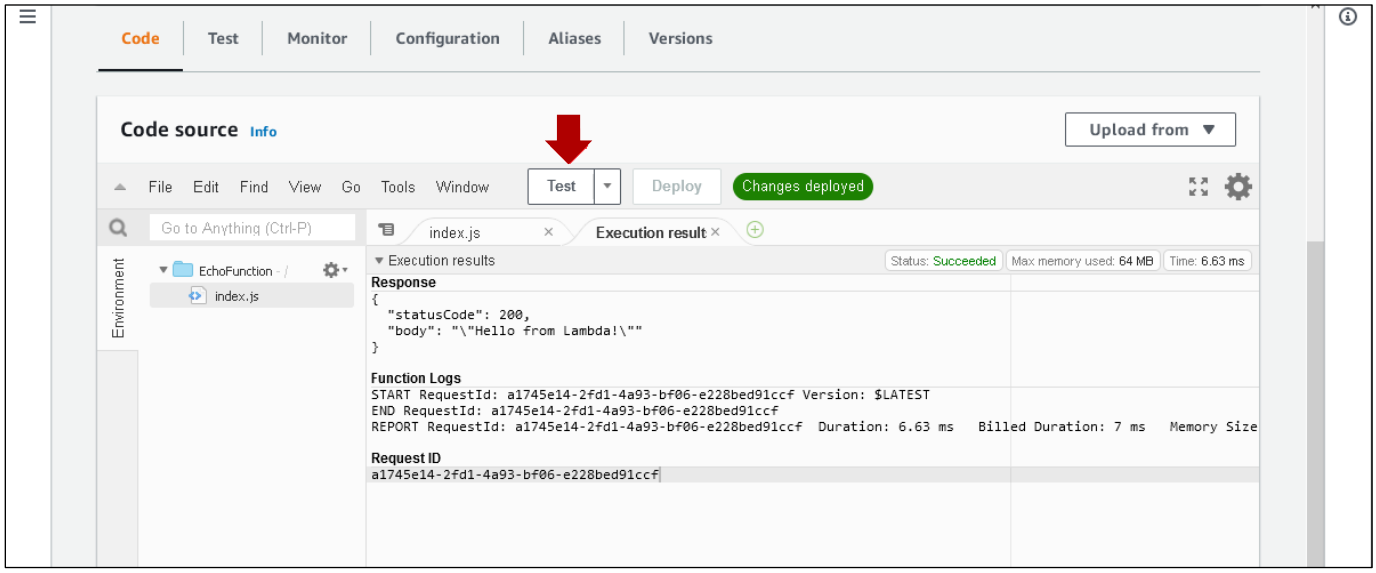
9. Click the **Test** tab in the left-hand side of the screen (see the red arrow).
10. In the **Test event** box, enter an event name of “SimpleTestWithJsonData”.
11. Leave everything else the same and click the **Save changes** Button in the upper-right part of the screen.



12. You will still be looking at the Test tab in the Lambda console.
13. Now click on the Code tab and you will see the inline Lambda code editor in the box titled **Code source**. Double-left-click on the index.js symbol on the left side of the screen, and the code for the function will appear.



14. We will review the code in a moment, but for now, click the **Test** button on the editor.
15. Congratulations! You have now run your first Lambda! Your screen should look like this:



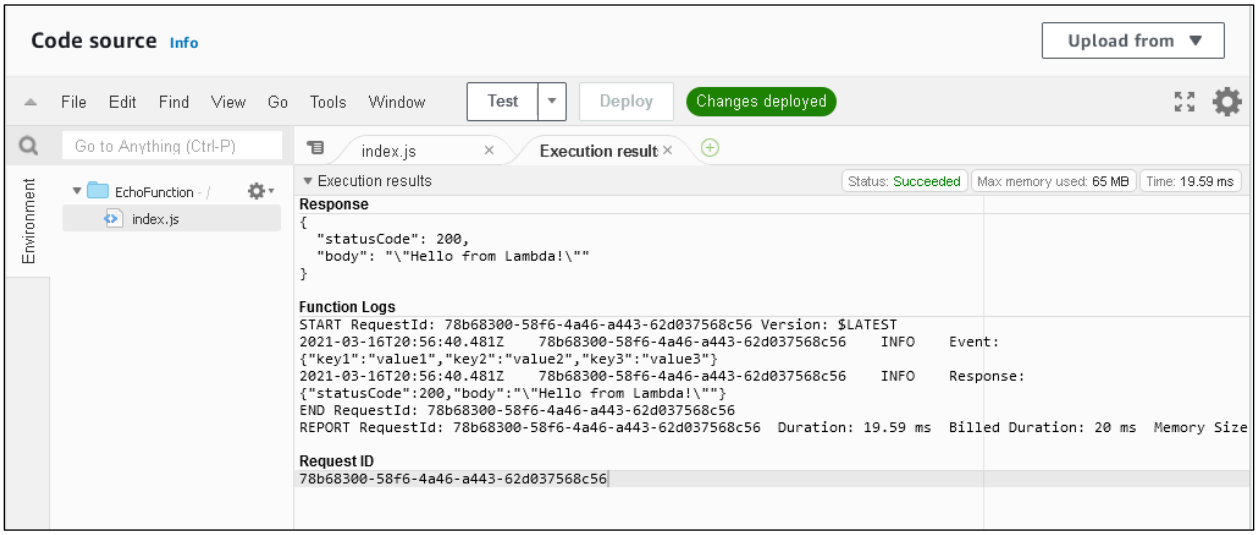
Part Two – Creating an Echo Function

1. You are going to modify the code provided by AWS to perform a simple echo function.
2. Add the following code to index.js after the creation of the response object and before the return:

```
// log the event to the logger
console.log("Event: \n" + JSON.stringify(event));
// log the response to the logger
console.log("Response: \n" + JSON.stringify(response));
```

3. Your code should look like this:

4. Click **Deploy**.
5. Click **Test** again.
6. Your execution results will now look like this:



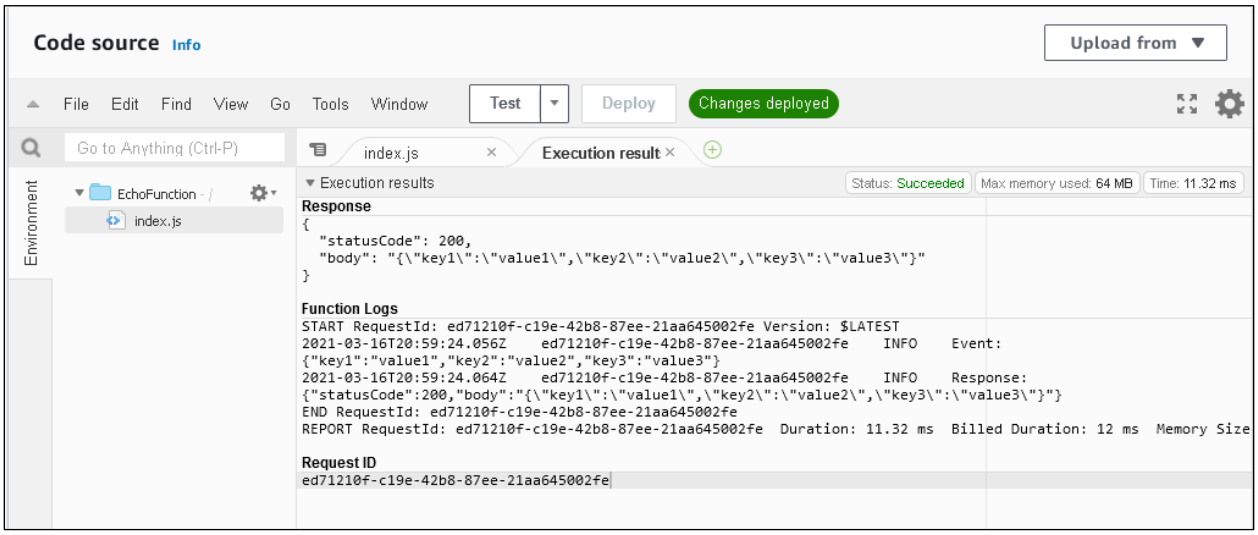
7. The values you passed in through the test event “SimpleTestWithJsonData” are now logged as the event object, and your response is the response object being created.
8. We want the response to echo what was passed in, so modify the code to do so. This will change the following code from:

```
// create some text to send back to the client
body: JSON.stringify('Hello from Lambda!'),
```

To:

```
// create some text to send back to the client
body: JSON.stringify(event),
```

9. Make sure to click **Save**.
10. Now click **Test** again and your execution results should be the following:



Part Three – Enhancing the Echo Function

1. You can now echo your results, at least for that JSON data that was passed in. Let's do something different.
2. First, create a new test class by clicking the down arrow next to **Test** and selecting **Configure test event.**
3. Select **Create New Test Event** and name it "EchoWithQuery", with the following JSON body:

```
{
  "queryStringParameters": {
    "name": "David"
  }
}
```

4. Click **Create**.
5. Now replace the code from the TODO to the log statements with the following:

```
var name = 'unknown';
if (event.queryStringParameters && event.queryStringParameters.name) {
    console.log("Received name: " + event.queryStringParameters.name);
    name = event.queryStringParameters.name;
}

const response = {
    statusCode: 200,
    body: "Hello " + name,
};
```

6. Click **Deploy**.
7. Run your EchoWithQuery test and you should see the following execution results: