**User Manual**

CIS 4911 - Senior Project (U01)

Event Driven Cloud Computing

Cory McAn

February 3, 2015

Jason Dettbarn

Introduction

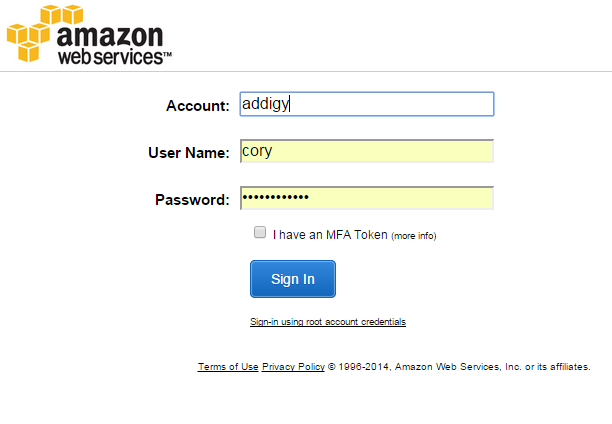
This manual is intended to assist users of the Event Driven Cloud Computing system developed by Cory McAn. This system is entirely back-end and lives on the Amazon Web Services S3 Bucket, so the steps listed here will be regarding the back-end functionalities of the Web Dashboard for Addigy.

Step 1

Navigate to <https://addigy.signin.aws.amazon.com/console>.

Ensure that the account is “addigy”.

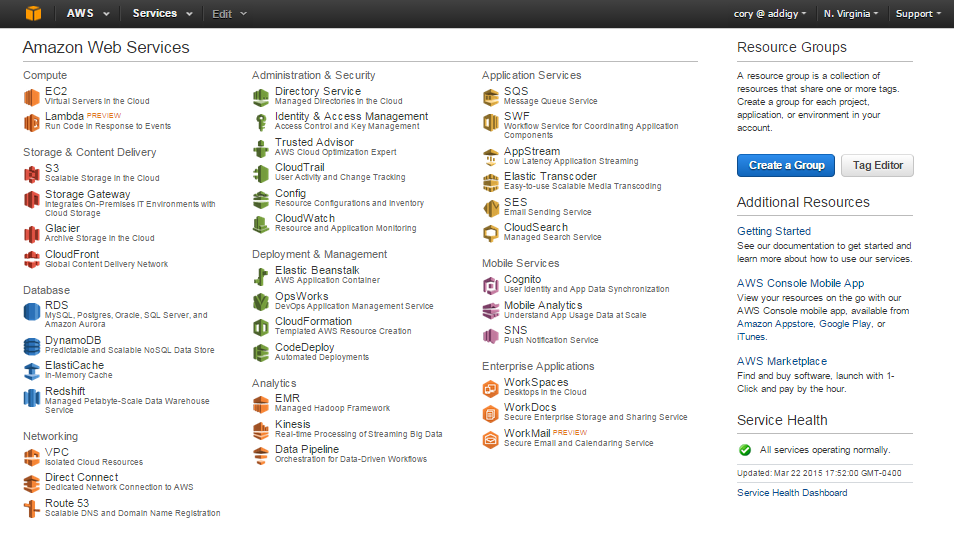
Log into Amazon Web Services with the credentials given to you by your AWS admin.



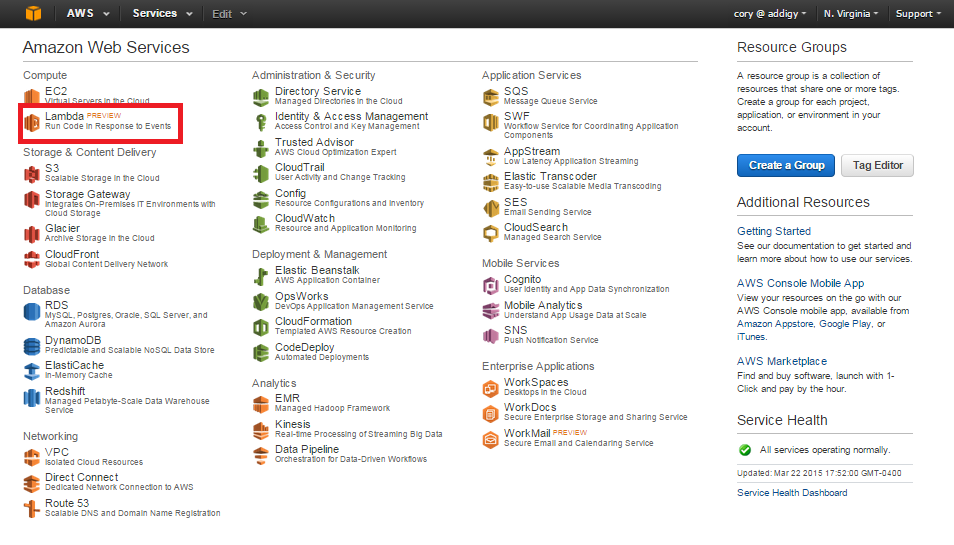
Step 2

You will now be at the Amazon Web Services home page.

The only necessary services here are Lambda and S3.



Let’s start by clicking on Lambda to understand how to create and invoke a Lambda Function.

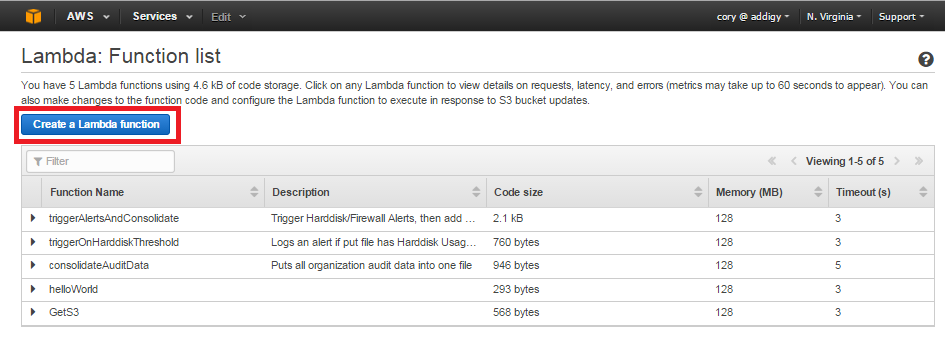


Step 3

By clicking Lambda, we arrive at the Lambda Function List.

Here you can manage your Lambda functions. There are many features such as creating a function, editing a function, configuring the event source, testing the function in the Lambda console, and checking the logs related to any Lambda function.

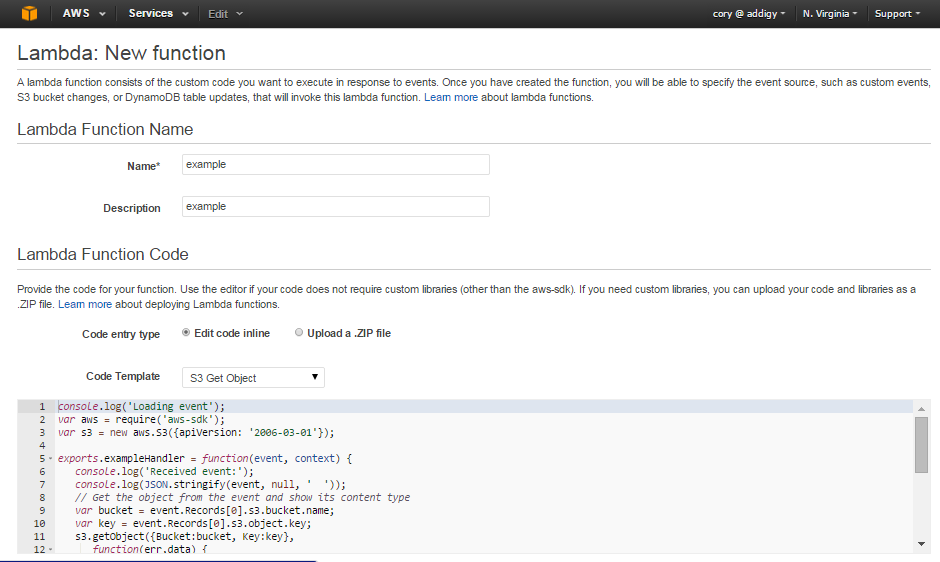
Let’s try creating a Lambda function by clicking “Create a Lambda function”.



Step 4

This is the page for creating a Lambda function. AWS provides you with function templates for testing and learning purposes, which would be helpful to look through.

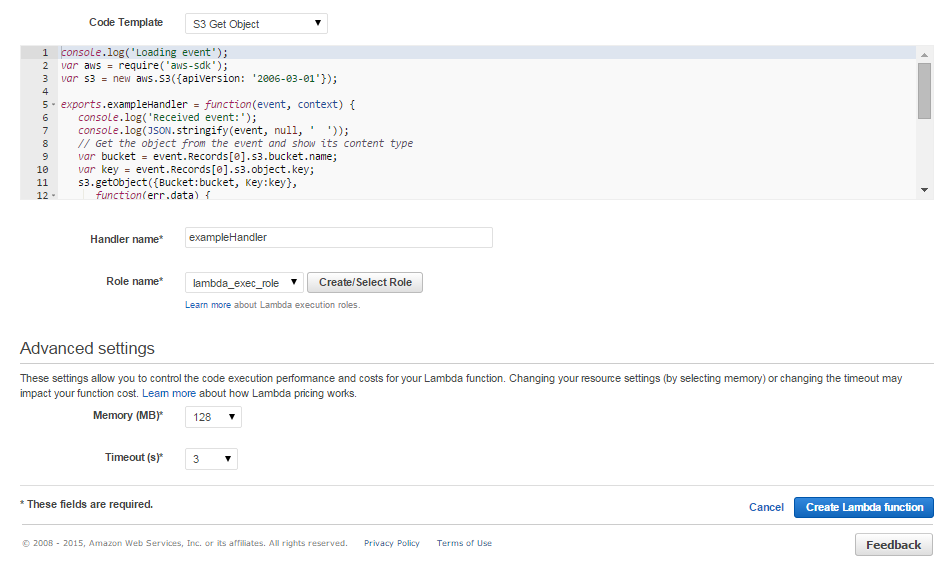
On this page we need to specify the name of our Lambda function and optionally describe the function. After that we can either upload a .zip file with our Node.js code (This is necessary if the code requires any packages not present in the core libraries of Node.js) or we can paste the Node.js code into the provided box if there are no additional libraries needed.



Next we need to name the handler, which is the same name that follows “exports.” In this case the handler is exampleHandler.

After we have a handler, we need to specify the execution role of the lambda function. Your Amazon Web Services admin must give you access to a lambda execution role as normal users cannot create them. Once your admin creates the role and gives you access you can select it from the drop down menu.

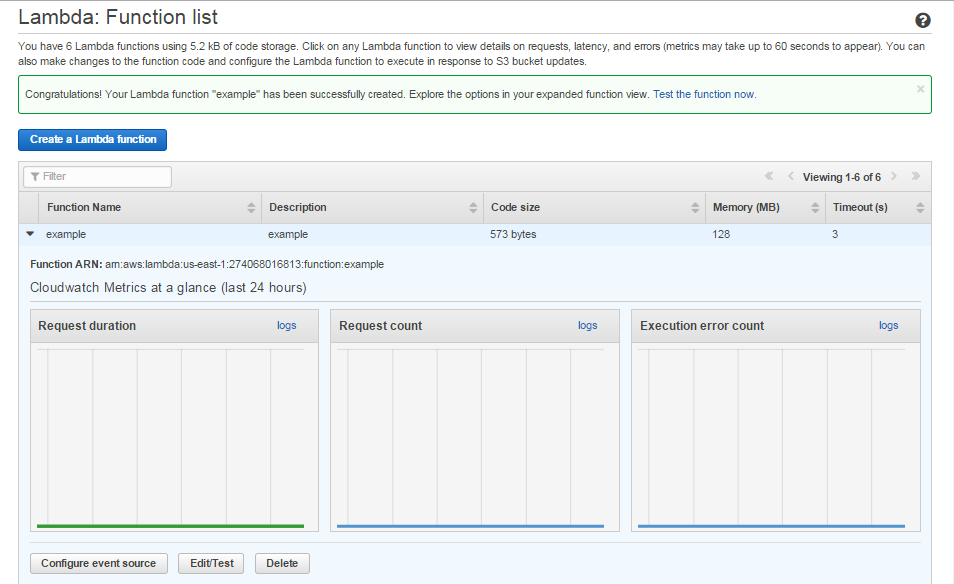
There are also advanced settings related to how long the Lambda function will run until it times out and the maximum amount of memory a Lambda function can use. As stated on the page, changing these values will have an impact on the cost of running the Lambda functions, so be sure about any changes you make to these values.



Now we simply need to click “Create Lambda function” and it will show up in our list!

Click the down arrow next to the function name and you will see the options below:

(next page)



Step 5

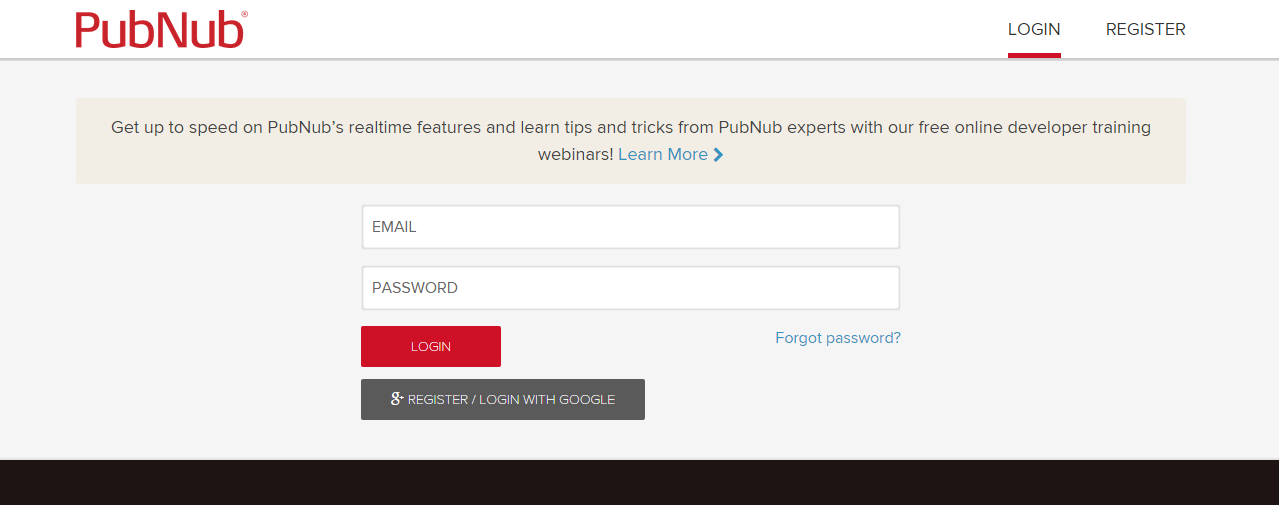
Now we have created a Lambda function. Each of your functions can be assigned to an S3 Bucket for the purpose of being triggered by a certain event on the Bucket.

Let’s open up the S3 Bucket List to see how to configure them.

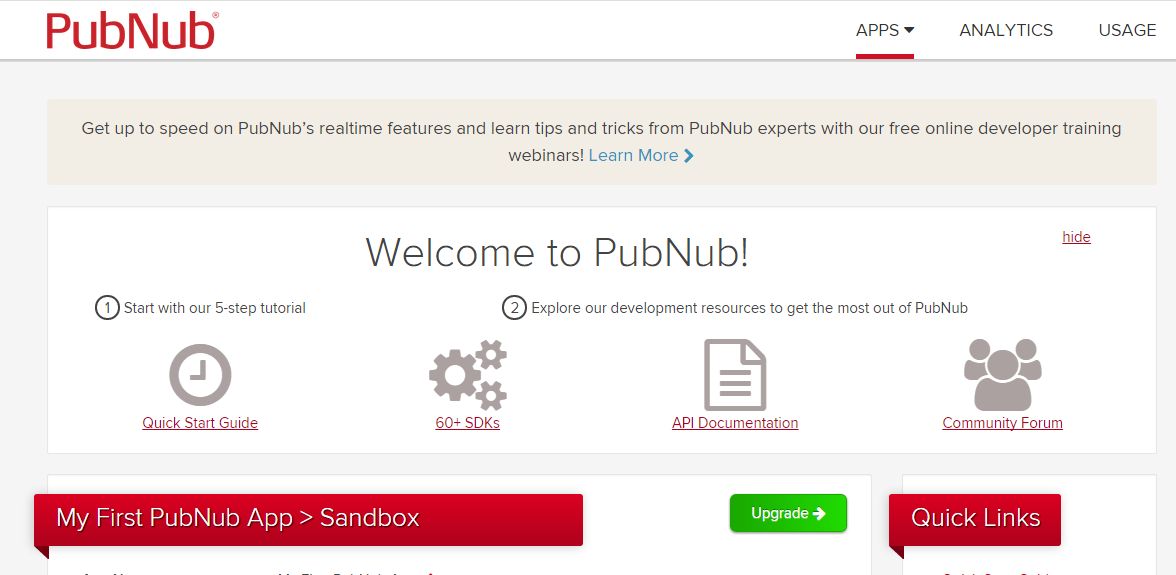
Step 6

Now that we have AWS and S3 configured, let’s get PubNub set up so that we can test our functions!

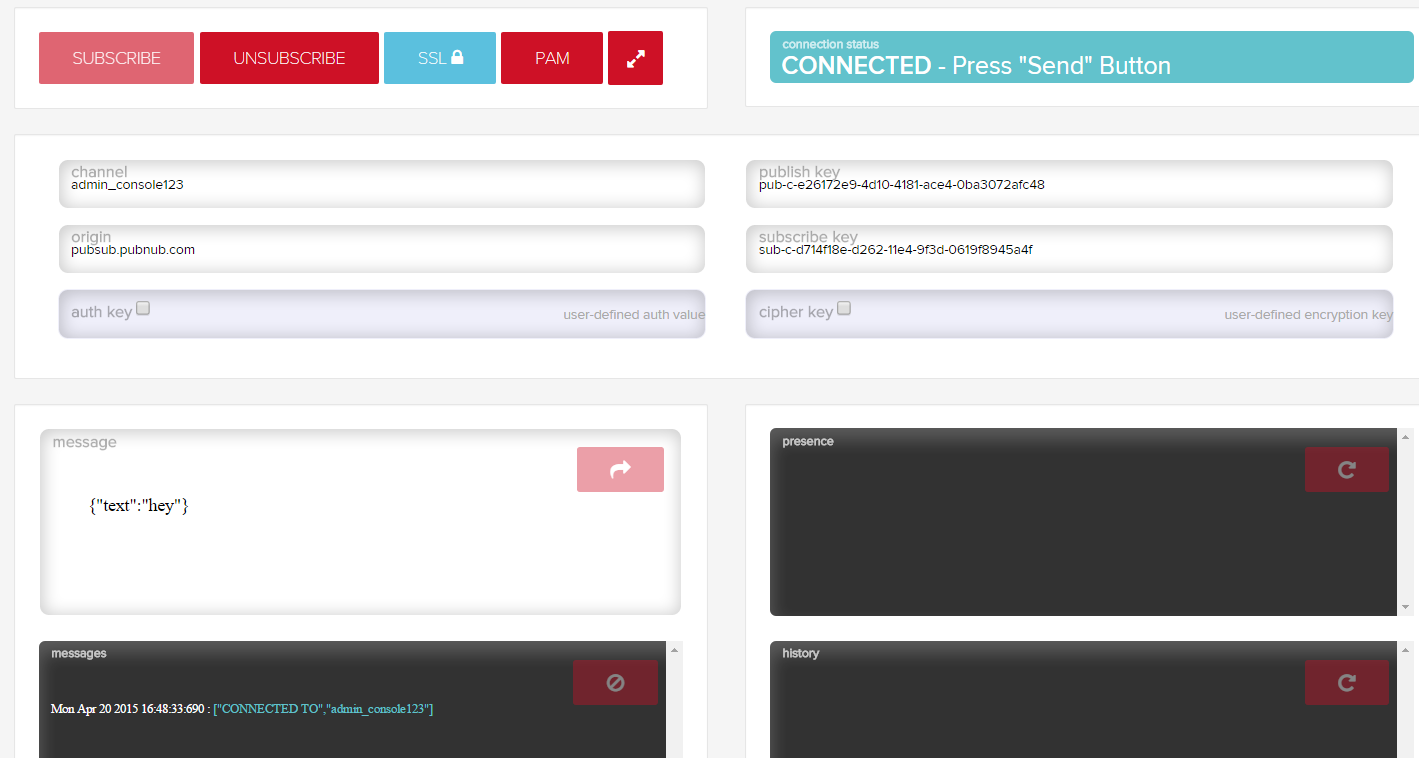
Navigate to <https://admin.pubnub.com/#/login>.



Input your login for PubNub and when you log in you’ll be at this screen:



Directly below “My First PubNub App” you will have your keys for publishing and subscribing to the PubNub channel. To the right under “Quick Links”, click the link titled “Debug Console”.



The debug console is all you need to test the lambda code in S3! Once you check the keys in the config file and make sure they match the publish and subscribe keys in the debug console, you can start testing a bucket by placing a file into the bucket and seeing the alert output in the console!