For the assignment, one of the first things that I did was to setup a free AWS account. After that I read through the Chalice documentation and understood how to set it up with Amazon AWS S3. I followed the instructions on the Chalice github page to create a simple HelloWorld application. Then I looked through the AWS account and read through the documentation for interacting with the S3 buckets.

My API implementation required a basic understanding of URL routing. Since I did not work with the Chalice code before, I decided to implement the basic ‘status’ page first. Then I worked on the more complex API post file to s3 bucket code. I set up a bucket on AWS and made sure it had the proper permissions for file access. I also setup a user role which was required for Chalice lambda and the boto3 package. The boto3 module provided the python library to connect to the AWS buckets. A bucket is how AWS stores files in folders. Although when the file was posted to the API it was not an simple task to convert it to an actual file.

The final step was utilizing an external API in a ‘fun way’. I chose to connect to the ‘recipe puppy’ API which provides public access to many recipes with a simple API call. I used the requests module to access the API url and get the json data. Then I parsed their result and converted it for display as an HTML page. I added bootstrap css for simple styling on the html.

In order to run the code, you would need to provide your AWS account credentials in the AWS json config file using the aws CLI command: aws configure. In app.py, be sure to update the BUCKET variable to the bucket listed on your AWS account. Check to be sure your AWS IAM is setup with the role to update the S3 bucket. IAM can be configured here: https://console.aws.amazon.com/iam. The code repository is located on my github account: https://github.com/vinny-santaiti/chalice-app.