Home Exercise: A Story of APIs and Cats

Goal

The following experiment is meant to
(a) be fun
(b) test your skills in rapidly utilizing and binding together a few state-of-the-art APIs to create new functionality, utilizing as off-the-shelf managed components as possible.

If you get stuck or find that the exercise takes too much of your time, you can wrap it up by documenting what you've accomplished vs. what was left, and note how you would tackle everything not yet implemented/working properly. Also, you can contact us for questions, if you're stuck and it becomes a choice of either a small hint or giving up...

In Summary

At Dynamic Yield, we have a very hungry cat. The cat is being fed by uploading images of appropriate food (fish, milk, or bread) into an S3 bucket. If the cat was not fed with a proper image for 15 minutes or more, an email should be automatically sent to an operator's address. If following a warning email, the cat has been fed again, a "back to normal" email should be sent. There should be only a single alert e-mail & back-to-normal e-mail, per each hunger period.

In more detail

1. Write a program that listens on a specific S3 bucket for new files. We strongly prefer that you use an AWS Lambda serverless function, which can be tied directly to S3 change notifications.

   If using a Lambda doesn't work for you, write a locally-running program that periodically checks for files it didn't yet process (or any other method you'd like).

2. When a new file is encountered, connect to Amazon Rekognition API to detect labels in the image.

3. If the labels returned from Rekognition include one of the “proper” food types
   (with 90% confidence or more), that means that suitable food has been given.
   In that case, we can update the last timestamp when valid food has been given.
   This value may be stored in a simple file, Redis, ElastiCache, or any super-simple storage.

4. A second program (or a scheduled AWS Lambda function) shall run periodically, and check the cat's feeding state:
   a. If the cat has not been fed for more than 15 minutes, send an email (but only once!)
   b. If, after a warning email was already sent, you detect that the cat was again fed, send a “back to normal” email - once per returning to a normal state.

      *** In order to make our life easier (and yours less) in testing your code, provisioning via terraform is a requirement**

How do | share my solution?

Create a new **private** GitHub repository, upload your code and grant **AlonMeron** read access.

Good Luck!